Advances in Intelligent Systems

Research in Computing Science

Series Editorial Board

Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico
Gerhard X. Ritter, University of Florida, USA
Jean Serra, Ecole des Mines de Paris, France
Ulises Cortés, UPC, Barcelona, Spain

Associate Editors:

Jesús Angulo, Ecole des Mines de Paris, France
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel
Alexander Gelbukh, CIC-IPN, Mexico
Ioannis Kakadiaris, University of Houston, USA
Petros Maragos, Nat. Tech. Univ. of Athens, Greece
Julian Padget, University of Bath, UK
Mateo Valero, UPC, Barcelona, Spain
Rafael Guzmán, Univ. of Guanajuato, Mexico

Editorial Coordination:

Alejandra Ramos Porras
Carlos Vizcaino Sahagún

Research in Computing Science, Año 2019, Volumen 148, No. 8, agosto de 2019, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 30 de agosto de 2019.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

Research in Computing Science, year 2019, Volume 148, No. 8, August 2019, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

Advances in Intelligent Systems

Hiram Ponce (ed.)



Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2019

ISSN: 1870-4069

Copyright © Instituto Politécnico Nacional 2019

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Editorial

Este número de la revista “Research in Computing Science” contiene artículos relacionados con temas sobre sistemas inteligentes. Los trabajos aquí publicados fueron cuidadosamente seleccionados por el comité editorial y revisados por al menos dos revisores externos considerando su originalidad científica y la calidad técnica.

Este número contiene 39 artículos que abordan conceptos y aplicaciones relacionadas cuatro ejes temáticos: visión computacional y realidad virtual; robótica y sistemas de control; optimización y algoritmos evolutivos; y, aplicaciones de la inteligencia artificial. La intención de esta publicación es la recopilación de trabajos que aplican técnicas de inteligencia computacional, tales como lógica difusa, redes neuronales artificiales y cómputo evolutivo, para crear soluciones inteligentes a problemas de alta complejidad analítica. A manera general, se abordan soluciones en temas de medicina, navegación autónoma de sistemas robóticos, uso de tecnologías de realidad virtual y realidad aumentada como parte de las interacción humano-computadora, aplicaciones en lenguaje no verbal y soluciones a problemáticas del sector industrial.

En visión computacional y realidad virtual, se incluyen trabajos sobre: aplicaciones médicas mediante el análisis de imágenes con redes neuronales convolucionales; uso de la realidad virtual para aprender y para traducir el lenguaje de señas de la población mexicana; uso de realidad virtual y aumentada para guías virtuales en museos; implementación de herramientas para análisis 3D y profundidad en video juegos serios, así como el análisis del gasto energético en los mismos; diseño de interfaces de visión para la inspección de la calidad en la industria manufacturera; análisis de la incidencia criminal en regiones de México mediante procesamiento de imágenes; detección de rostros mediante técnicas de visión computacional; y, análisis de huecos en objetos de dos dimensiones.

Dentro de los temas de robótica y sistemas de control, se reportan trabajos de investigación dedicados al desarrollo de técnicas de planificación de movimiento basados en sensores para robots móviles, métodos de Bugs extendidos, y el uso de métodos de campos potenciales con visión para la navegación robótica automática; diseño de sistemas de control para exoesqueletos; uso de técnicas de aprendizaje por refuerzo para extremidades robóticas y propuestas de marcos de trabajo sobre este tipo de aprendizaje aplicado a robótica; técnicas para la manipulación de un brazo robótico mediante técnicas de evolución; sistemas de control inteligente para procesos de manufactura de precisión; así como, uso de inteligencia artificial para el análisis de la cinemática inversa en manipuladores.

Por otra parte, se incluyen trabajos sobre optimización implementado técnicas como: inteligencia colectiva de bacterias para el mejoramiento de un modelo de partículas contaminantes, evolución diferencial para procesos de ingeniería química, para la secuenciación de vehículos y para la sintonización de sistemas de control inteligente; optimización para la restructuración tridimensional de objetos; algoritmos genéticos uni-objetivo y multi-objetivo; optimización evolutiva aplicada al reconocimiento del

habla; además de trabajos metodológicos para la solución de problemas de bisección de vértices o aplicaciones puntuales en redes de distribución eléctricas.

Entre los trabajos también se presentan sistemas inteligentes tales como: el uso de sistemas distribuidos para actividades y monitoreo colaborativo, analítica de datos para mediciones en redes, modelado de sistemas multi-agentes para el monitoreo de residuos peligrosos en la industria, predicción de trayectorias, uso de redes neuronales y sistemas difusos para la simplificación de caminos, navegación de agentes en interiores, y uso de señales inteligentes mediante teléfonos celulares y software abierto.

También el volumen contiene un artículo regular sobre generación de grafos aleatorios.

Finalmente, cabe mencionar que el proceso de revisión y selección de artículos se llevó a cabo usando el sistema libremente disponible EasyChair (www.easychair.org).

Hiram Ponce
Universidad Panamericana, México
Editor Invitado

Julio 2019

ISSN 1870-4069

Table of Contents

Page

Propuesta de mejora de modelado de partículas contaminantes PM10 mediante algoritmos de inteligencia colectiva de bacteria BFOA 13
Cabrera-Hernández M.C., Ramírez-Montañez J.A., Aceves-Fernández M.A., Pedraza-Ortega J.C., Ramos-Arreguin J.M.

Evolución diferencial para la optimización global de procesos de ingeniería química 27
Maritza Yanet Martínez Zecua, Luis Angel Salamanca Vázquez, Leticia Flores Pulido, Edgar Alfredo Portilla Flores, Arturo Ortíz Arroyo

Clasificación de enfermedades del tórax usando aprendizaje profundo y aumento de datos de calidad en el conjunto de datos ChestX-ray8 41
Amhed A. Ortiz-Preciado, Juan Antonio Vega-Fernández, Gilberto Ochoa-Ruiz

Herramienta en realidad virtual para el aprendizaje del lenguaje de señas mexicano..... 55
Arturo Borja Araiza, David Araujo Díaz, Laura Méndez Segundo

Aspectos innovadores de realidad virtual y sensores cinéticos para una mejora significativa utilizando el algoritmo de Fireworks para un juego Wii de un deporte colaborativo para una sociedad en una Smart City 63
Alberto Ochoa, César Velázquez, Juan Hernández, Silvia Husted, Diego Oliva, Anahí Solis, Héctor Herrera, José Mejia, Saúl González, Edna Cruz-Flores

Traducción del lenguaje de señas usando visión por computadora 79
Eduardo Mancilla Morales, Oswaldo Vázquez Aparicio, Pedro Arguijo, Roberto Ángel Meléndez Armenta, Antonio Hiram Vázquez López

Reconstrucción tridimensional de objetos mediante el uso de luz estructurada..... 91
Ignacio Martínez Rojas, Mayra Pérez Robledo, Francisco Javier Cuevas de la Rosa, Pedro Arguijo

Actividades colaborativas en sistemas distribuidos construidos con Erlang 105
Gerardo Carreño-García, Manuel Hernández-Gutiérrez

Diseño de interfaces de inspección de calidad utilizando redes neuronales y visión artificial para la industria de manufactura 121
Luis Arturo Medina Muñoz, Samuel González-López, Jesús Antonio Mayorquín Robles, Gabriel Antonio López Valencia

Variante eficiente de red neuronal pulsante con auto-onda dinámica para resolver el problema del camino más corto 133
Oliver Espinosa, Manuel Mejía-Lavalle, José Ruiz, Gerardo Reyes, Alicia Martínez

Planificación de movimientos basada en sensores para robots móviles en ambientes desconocidos 147
Valerio Galiote T., Abraham Sánchez L., José F. Texcucano D., Alfredo Toriz P., Rogelio González V., Elberfeld Pérez G.

Navegación de robots móviles utilizando algoritmos Bugs extendidos 159
Mario Serna H., Abraham Sánchez L., Martín Estrada A., Ma. Beatríz Bernábe L., Elberfeld E. Pérez G.

Mapeo de incidencia criminal en el Estado de México mediante procesamiento digital de imágenes 173
Tania Lira-Baca, Miguel Borboya Melchor, Roberto Hernández Sánchez, David Tinoco Varela, Fernando Gudiño Peñaloza

Diseño e implementación de un algoritmo genético para la predicción de una variable 187
Hilda Avelar Uribe, Ludivina Gutiérrez Torres, Ismael Zúñiga Félix, Zindi Sánchez Hernández

Aplicación de un algoritmo genético multiobjetivo para la replaneación de liberaciones en proyectos ágiles de software 199
Victor Hugo Escandon Bailon, Humberto Cervantes Maceda, Abel García Nájera

Control de la distancia inter electrodos en un proceso de micro maquinado electroquímico por pulsos (PECMM) 215
Iván Gallegos Miranda, José Arturo Pérez Martínez, Everardo Efrén Granda Gutiérrez, Juan Fernando García Mejía

Movimientos en pierna robótica para el pateado de un balón a través de reinforcement learning 229
Gerardo Franco Delgado, M. De La Rosa, C. E. De Los Santos, J. D. Carrillo

Analítica de datos en simulador de redes para sistemas de medición inteligente 243
Juan C. Olivares-Rojas, Enrique Reyes-Archundia, José A. Gutiérrez-Gnecchi, Johan W. González-Murueta, Adriana Téllez-Anguiano, Jaime Cerda-Jacobo

Control basado en campos potenciales para un vehículo autónomo usando realimentación visual 253
Oscar González-Miranda, Santos M. Orozco-Soto, Juan M. Ibarra-Zannatha

Estimación automática del gasto energético de la actividad física de niños en videojuegos de ejercicio con el sensor Kinect 263
Benjamín Gutiérrez-Serafín, Humberto Pérez-Espinosa, Ismael Edrein Espinosa-Curiel, Pedro N. Figueroa-García, Edgar E. Pozas-Bogarín, Juan Martínez-Miranda

Algoritmo de evolución diferencial con reparador cromosómico aplicado a un problema de secuenciación de vehículos 279
Elvi Malintzin Sánchez Márquez, Héctor José Puga Soberanes, Luis Ernesto Mancilla Espinoza, Juan Martín Carpio Valadez, Manuel Ornelas Rodríguez, Javier Iván Manzanares Cuadro

Optimización evolutiva de contextos para la corrección fonética en sistemas de reconocimiento del habla 293
Rafael Viana Cámara, Diego Campos Sobrino, Mario Campos Soberanis

Predicción de trayectorias usando el filtro de Kalman 307
Astrid C. Estrada-Trejo, Isaac Ramos-Silva, Gabriel Sepúlveda-Cervantes, Edgar A. Portilla-Flores, Eduardo Vega-Alvarado

Modelo para la determinación de la ruta más corta con funciones experimentales para arcos difusos 317
Eduardo Chandomí-Castellanos, Elías N. Escobar-Gómez, Sabino Velázquez-Trujillo, Héctor R. Hernández De León, Madaín Pérez-Patricio, Carlos Venturino De Coss Pérez

Un nuevo algoritmo de ramificación y acotamiento para el problema de la bisección de vértices 331
Eduardo Del Ángel-Martínez, Héctor Fraire-Huacuja, Carlos Soto, Nelson Rangel, Laura Cruz-Reyes, Claudia Gómez-Santillán

Navegación en interiores utilizando la tecnología beacons 345
Alicia Martínez-Rebollar, Luis Antonio López-García, Hugo Estrada-Esquivel, Manuel Mejía Lavalle, Javier Ortiz Hernández

Adquisición de señales digitales utilizando teléfonos inteligentes y FIWARE 357
Alicia Martínez Rebollar, Leon Alberne Torres Restrepo, Fabio Alberto Vargas Agudelo, Javier Ortiz-Hernández

Detección efectiva de rostros en imágenes utilizando descriptores basados en HOG 371
Cinthia Viviana Alcántara-Montiel, Jesús Carlos Pedraza-Ortega, Juan Manuel Ramos-Arreguín, Efrén Gorrostieta-Hurtado, Saúl Tovar-Arriaga, José Emilio Vargas-Soto

Desarrollo de un guía virtual conversacional para museos en un entorno inmersivo de realidad virtual y aumentada 387
Isidro Hernán Martínez Juárez, María Eugenia Puga Nathal, Juan Martínez-Miranda

Topología de una RNA para calcular la cinemática inversa del Puma 560 para aplicaciones en tiempo real 397
David Bedolla-Martínez, Felipe de Jesús Trujillo-Romero, Esther Lugo-González

Aplicación para el monitoreo colaborativo de la calidad de cuerpos de agua usando bioindicadores 407
Irving Galindo-Márquez, Ángel López-Sánchez, Luis Hernández-Mercado, Carlos Sosa-Paz, Juan-Pablo Posadas-Durán

Sintonización óptima de un controlador PD utilizando evolución diferencial con un modelo dinámico virtualizado en Unity 3D 423
Rosaura A. Suárez-Santillán, Gabriel Sepúlveda-Cervantes, Edgar A. Portilla-Flores, María Bárbara Calva-Yáñez, Cuauhtémoc Morales-Cruz, Noemí Hernández-Oliva

Integración de aprendizaje por refuerzo a simulador físico y análisis de funciones de recompensa para robots humanoides 437
Alfonso Brown González, Fernando Cossío Ramírez, Edgar Omar López Caudana

Formulaciones para el cálculo eficiente del número de hoyos de un objeto binario 2-D 451
Juan H. Sossa, Elsa Rubio, Víctor H. Ponce, Luis E. Falcón

Un algoritmo genético con fertilización in-vitro aplicado a la sintonización de un regulador difuso 463
J. Fernando García-Mejía, Luis Antonio Gonzalez, Carlos E. Torres-Reyes, J. Arturo Pérez Martínez, J. Antonio Garcia-Mejía

Modelado de un sistema multi-agente para el monitoreo de residuos peligrosos en la industria manufacturera 477
Carlos A. Soto, Juan P. Soto, Adrián Vázquez, Elvira Rolón, Julio C. Rolón

Propuesta de red de distribución para reciclaje de desechos eléctricos y electrónicos 487
José-Raúl Morales-Contreras, Santiago-Omar Caballero-Morales, Patricia Cano-Olivos, Diana Sánchez-Partida

On Generating Random Graphs Based on Fuzzy Proximity Relationships between Vertices 499
Mohamed Amine Omrani, Wady Naanaa

ISSN 1870-4069

Propuesta de mejora de modelado de partículas contaminantes PM10 mediante algoritmos de inteligencia colectiva de bacteria BFOA

M.C. Cabrera-Hernández, J.A. Ramírez-Montañez, M.A. Aceves-Fernández, J.C. Pedraza-Ortega, J.M. Ramos-Arreguin

Universidad Autónoma de Querétaro, Facultad de Ingeniería, Querétaro, México
maria.delcarmen.cabrera@uaq.mx, julio.ramirezm27@gmail.com, marco.aceves@uaq.mx, caryoko@yahoo.com, jramos@mecamex.net

Resumen. La necesidad de monitorear la calidad del aire se ha vuelto fundamental para la humanidad ya que una calidad deficiente representa una amenaza grave en materia de salud, en especial para cierto tipo de partícula cuyo comportamiento no es lineal como lo es la partícula PM10. En esta contribución el objetivo es optimizar el modelo generado por ANFIS para partículas contaminantes utilizando algoritmos de inteligencia de enjambre. Específicamente, se utiliza BFOA para la optimización, demostrando que es posible disminuir el error del modelo ANFIS, para comprobar la efectividad del modelo propuesto, se llevaron a cabo varios experimentos con los datos y los parámetros propios del algoritmo BFOA.

Palabras clave: contaminación del aire, algoritmo de optimización de forrajeo por bacterias (BFOA), inteligencia de enjambre, ANFIS.

Proposal for Improving Modeling of PM10 Pollutant Particles through Collective Intelligence Algorithms of BFOA Bacteria

Abstract. The need to monitor air quality has become fundamental for humanity, since poor quality represents a serious threat in terms of health, especially for a certain type of particle whose behavior is not linear, such as the PM10 particle. In this contribution the objective is to optimize the model generated by ANFIS for pollutant particles using swarm intelligence algorithms. Specifically, BFOA is used for optimization, demonstrating that it is possible to reduce the error of the ANFIS model, to verify the effectiveness of the proposed model, several experiments were carried out with the data and parameters of the BFO algorithm.

Keywords: air pollution, bacterial foraging optimization algorithm (BFOA), swarm intelligence, ANFIS.

1. Introducción

1.1. Material particulado

El material particulado (MP) o partículas atmosféricas son piezas pequeñas de materia suspendidas en la atmosfera que bien pueden encontrarse en forma de partículas sólidas o gotas. El origen de esta materia bien pude ser debido a procesos naturales (polen, erosión de la tierra), o actividades humanas (construcciones, actividad industrial, etc.) [1]. Así mismo el MP es liberado directamente de la fuente, o puede ser producto de reacciones químicas que ocurren directamente en la atmósfera.

Las partículas atmosféricas se pueden clasificar según su tamaño, hay partículas con un diámetro de 10 μm (micras de metro) llamadas PM10 y otras más finas que son de 2,5 μm (PM2.5), lo que las hace adecuadas para ser inhaladas fácilmente por los seres humanos y causar graves problemas de salud en las vías respiratorias. y sistemas cardiovasculares [2]. Es importante destacar que, aunque se monitorean gases como el ozono (O3), dióxido de nitrógeno (NO2), dióxido de azufre (SO2) y PM10 como parte de los contaminantes atmosféricos, PM10 ha demostrado en ser uno de los contaminantes que causan mayores problemas de salud, incluso siendo un factor de mortalidad en un sector de la población [3], de ahí la importancia del desarrollo de herramientas más precisas para modelar su concentración en el medio ambiente.

Contaminantes atmosféricos y PM10. Para el del modelado comportamiento de las partículas PM10 ya se han utilizado varias técnicas como redes neuronales para predecir el comportamiento del contaminante con 24 horas de anticipación [4], en general las redes neuronales han sido de los métodos más utilizados para el modelado de contaminantes atmosféricos [5, 6], incluyendo algunas variantes de dicho método como las redes neuronales autorregresivas (ARNN, autoregressive neural network) [7]. Algunos otros métodos utilizados para el modelado de contaminantes ambientales incluyen, lógica difusa de intervalo tipo 2 [8]; así como el uso de cuantificación de recurrencia para investigar los cambios en el comportamiento no lineal de los contaminantes atmosféricos [9] Uno de los métodos innovadores en esta área es el uso de algoritmos de inteligencia de enjambre, por ejemplo, el algoritmo de Colonia de Hormigas [10] que mostró resultados alentadores para los algoritmos basados en la naturaleza aplicados a este problema.

1.2. Inteligencia de enjambre

Los algoritmos de inteligencia de enjambre existen desde finales de los años 80 del siglo pasado, Beni y Wang [11]introdujeron el término. La inteligencia de enjambre está basada en un conjunto de agentes que interactúan entre sí, así como con su ambiente, con el objetivo de alcanzar un comportamiento integral y cooperativo que da como resultado una interacción con cierto grado de complejidad que de otra manera no podría ser completado por los agentes de manera individual, de esa manera encontrar mejores soluciones como una comunidad. Estos algoritmos están inspirados principalmente en fenómenos naturales como las colonias de Hormigas [12], de abejas [13] o de bacterias, gotas de agua [14], el comportamiento de los murciélagos [15], termitas [16], etc.

Algoritmo de optimización de forrajeo bacteriano. El algoritmo de optimización de forrajeo bacteriano (BFOA, Bacterial Foraging Optimization Algorithm) [17] está inspirado en el proceso de forrajeo de la bacteria E. Coli. La forma en que esta bacteria realiza los procesos de forrajeo y reproducción, maximiza la energía que obtiene del ambiente, haciendo de este un método de optimización.

BFOA ha sido aceptado como un algoritmo de optimización global, su eficiencia se ha demostrado en varias áreas. Por ejemplo, su aplicación en el campo de la ingeniería eléctrica y el control [18], reconocimiento de patrones [19], diseño PID [20] programación de horarios de trabajo utilizando sólo BFOA [21], así como el desarrollo de aplicaciones para resolver ecuaciones simultáneas [22], incluso se ha utilizado para la selección de características en problemas de reducción de dimensionalidad de conjuntos de datos que así lo requieran [23]. El algoritmo de BFO se ha utilizado también de manera híbrida junto con otros métodos por ejemplo junto con el algoritmo de Colonia de Hormigas para el mismo problema de programación de horarios [24] y se ha utilizado en conjunto con el Algoritmo de Optimización de Enjambre de Partículas PSO (Particle Swarm Optimization) para diseño óptimo de estabilizadores del sistema de potencia [25] y en problemas de generación de energía [26].

2. Metodología y materiales

2.1. Materiales

Los datos usados para construir el modelo provienen del Sistema de Monitoreo Atmosférico (SIMAT) que es responsable de la medición permanente de los niveles de contaminantes en la atmósfera en la Ciudad de México y el área Metropolitana [27]. El SIMAT a su vez tiene un subsistema, la Red Automática para el Monitoreo Atmosférico (RAMA), este sistema usa equipo de medición continua de contaminantes del aire, tales como dióxido de azufre, monóxido de carbono, dióxido de nitrógeno, ozono, PM10 y PM2.5. La ubicación de las estaciones de monitoreo están divididas en 5 áreas, Este, Centro, Norte, Sur y Oeste, la organización de las estaciones se puede observar en la Fig. 1.

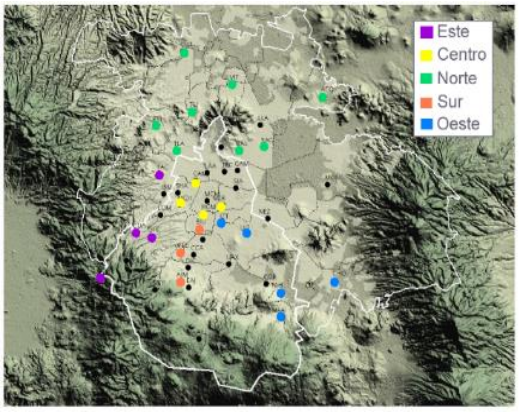


Fig. 1. Mapa de las estaciones de RAMA.

Las estaciones de monitoreo recolectan información de las concentraciones de los contaminantes cada hora, estas concentraciones son medidas en µg/m3.

2.2. Construcción del modelo

Sistema adaptativo de inferencia neurodifusa (ANFIS). El enfoque propuesto en esta contribución es optimizar un modelo existente (modelo base), utilizando BFOA como un optimizador. El modelo base propuesto es un sistema adaptativo de inferencia neurodifusa, ANFIS (Adaptive neuro fuzzy inference system) que es un tipo de red neuronal artificial que incluye un sistema de inferencia difusa Takagi-Sugeno (Fig. 2).

ANFIS construye un Sistema de inferencia difusa (FIS) dado un conjunto de datos de tipo entradas/salida, los parámetros de las funciones de membresía se ajustan utilizando un algoritmo de retropropagación o en combinación con un método de mínimos cuadrados.

Un FIS se puede definir como un conjunto de reglas difusas de tipo SI-ENTONCES (if-then), que son expresiones de la forma SI A ENTONCES B, donde A y B son etiquetas de conjuntos difusos [28]

Para simplificar el ejemplo supongamos que tenemos dos entradas *(x,y)* y una salida f, y se utilizan cinco capas para construir el modelo. Cada una de las capas contienen varios nodos que pueden ser adaptativos (nodos cuadrados) o fijos (nodos circulares) [29].

Capa 1. Es la capa difusa y convierte las entradas del modelo en conjuntos difusos por medio de funciones de membresía (MF) y las funciones del nodo se describen así:

$$O_{1,i} = \mu_{Ai}(X_1) \text{para i} = 1,2, \quad (1)$$

$$O_{1,i} = \mu_{B_{1-2}}(Y_1) \text{para i} = 3,4, \quad (2)$$

donde X_1 y Y_1 son los nodos de entrada , A y B corresponden a las etiquetas lingüísticas asociadas a estos nodos , $\mu(X_1)$ y $\mu(Y_1)$ son las funciones de membresía (MF), los parámetros en esta capa se llaman parámetros de premisa.

Capa 2. Los nodos en esta capa son fijos, la función de cada nodo se multiplica por las señales de entrada, lo cual sirve como señal de salida y se etiquetan con Π.

$$O_{2,i} = w_i = \mu_{Ai}(X_1).\mu_{B_{1-2}}(Y_1) \text{ para i} = 1,2, \quad (3)$$

donde $O_{2,i}$ es la salida de esta capa y la señal de salida w_i representa la fuerza de disparo de la regla.

Capa 3. Los nodos de esta capa también son fijos y están etiquetados con N, y su función es normalizar la fuerza del disparo calculando la proporción de la fuerza de disparo del nodo i-ésimo para sumar la fuerza de disparo de todas las reglas.

$$O_{3,i} = \overline{w} = \frac{w_i}{w_1 + w_2} \; para \; i = 1,2, \quad (4)$$

donde $O_{3,i}$ es la salida de la capa 3 y $\overline{w}$ es la fuerza de disparo pero ya normalizada.

Capa 4. Los nodos de esta capa se ajustan y se definen como:

$$O_{4,i} = \overline{w_i}.\, f_i \; para \; i = 1,2, \tag{5}$$

donde $f_1 \; y \;\; f_2$ son las reglas difusas SI- ENTONCES definidas de la siguiente manera:

– Regla 1. SI X_1 es A_1 y Y_1 es B_1, ENTONCES $f_1 = p_1 X_1 + q_1 Y_1 + r_1$,
– Regla 2. SI X_1 es A_2 y Y_1 es B_2, ENTONCES $f_2 = p_2 X_1 + q_2 Y_1 + r_2$,

donde p_i, q_i y r_i son parámetros establecidos y referidos como parámetros consecuentes.

Capa 5. En esta capa los nodos son fijos etiquetados con ∑ y su función es calcular la salida total y que se define por:

$$O_{5,i} = \sum_i \overline{w_i}.\, f_i = \frac{\sum_i w_i f_i}{w_i} = f_{out} = salida\ total. \tag{6}$$

ANFIS tiene una regla de aprendizaje muy sencilla, que es la "propagación hacia atrás", esta regla calcula las señales de error de forma recurrente, comenzando desde la capa de salida (Capa 5) hacia los nodos de entrada (Capa 1).

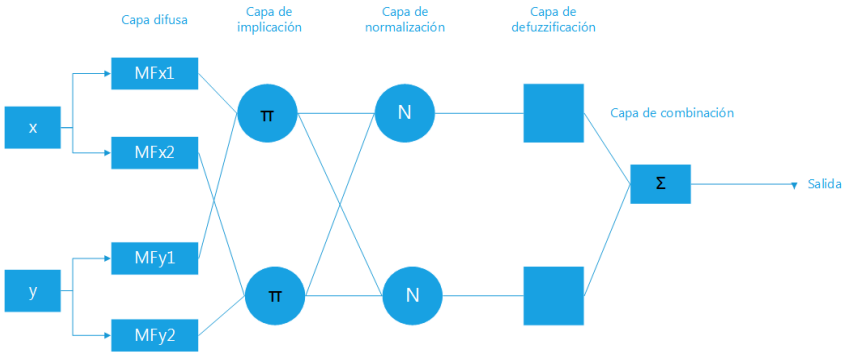


Fig. 2. Estructura ANFIS (adaptado de [25]).

2.3. Modelo de optimización

El algoritmo de BFO imita el proceso de forrajeo de una bacteria real cuya locomoción es lograda a través del movimiento de sus flagelos que ayudan a la bacteria a nadar o girar, estas operaciones son básicas en el proceso de forrajeo. Si los flagelos giran en el sentido de las manecillas del reloj genera un movimiento de giro, en un ambiente nocivo la bacteria va a dar más giros para encontrar nutrientes y cuando los flagelos giran en el sentido contrario a las manecillas del reloj la bacteria realiza un nado, en un ambiente adecuado para la bacteria el movimiento de nado recorre mayores distancias [30]. Los movimientos mencionados anteriormente forman parte del proceso de quimiotaxis en donde la bacteria buscará moverse en un ambiente con nutrientes y evitará las zonas dañinas. En la Fig. 3 se puede observar los movimientos de giro y nado.

Cuando la bacteria se encuentre con nutrientes suficientes y la temperatura del ambiente lo permita la bacteria se reproducirá dividiéndose en dos y creando una réplica de sí misma creando así una colonia de bacterias. Igualmente, si ocurre un ataque o el ambiente cambia repentinamente un grupo de bacterias es dispersado hacia otras áreas del ambiente o son eliminadas, este evento se llama eliminación-dispersión.

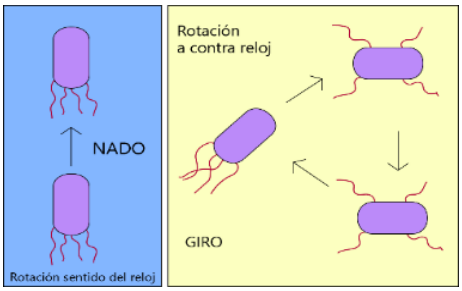


Fig. 3. Quimiotaxis de una bacteria (traducido de [17]).

Si suponemos que buscamos encontrar el mínimo de J (θ) donde θ ∈ ℜp (en donde θ es un vector p-dimensional) y se ignora la naturaleza del gradiente ∇J (θ) ya que no contamos con una descripción analítica o incluso mediciones de ∇J (θ). BFOA implementa una imitación de los mecanismos principales presentes en una colonia de bacterias E, Coli real: quimiotaxis, formación de la colonia o enjambre, reproducción y eliminación-dispersión con el cual se puede resolver el problema de optimización sin gradiente. La forma de explicar en qué consiste una bacteria virtual es que es realidad una solución de prueba que se mueve en la superficie funcional para ubicar el óptimo global [23].

Para poder implementar BFOA es necesario definir el paso quimiotáctico como un giro seguido de un nado o un giro seguido de un giro. Entonces j es el índice el paso quimiotáctico, k el índice del paso de reproducción, y por último l es el índice es del evento de eliminación-dispersión.

El algoritmo cuenta con ciertos parámetros que deben inicializarse y de los cuales depende el desempeño del algoritmo dependiendo del problema. Sean:

- p: Dimensión del espacio de búsqueda
- S: Bacterias que forman la población
- Nc : Pasos quimiotácticos
- Ns: Longitud de nado
- Nr : pasos de reproducción
- Ne : Eventos de eliminación dispersión
- Ped: Probabilidad de que una bacteria sea eliminada o dispersada
- C (i): Tamaño del paso tomado en una dirección aleatoria especificada por el giro.
- Sea entonces $P(j,k,l) = \theta^i(j,k,l) donde\ i = 1,2,\dots,S$ la posición de cada miembro en la población de S bacterias en el paso quimiotáctico j-ésimo, el pas k-ésimo de repoducción yu el evento l-ésimo de eliminación-dispersión, entonces podemos asociar un costo $J(j,k,l)$ a esa posición $\theta^i(j,k,l)$.

A continuación, se describe cada una de las etapas de BFOA.

Quimiotaxis. Supongamos que $\theta^i(j,k,l) donde\ i = 1,2,\dots,S$ la posición de cada miembro en la población de S bacterias en el paso quimiotáctico j-ésimo, el paso k-ésimo de reproducción y el evento l-ésimo de eliminación-dispersión y C(i) es el paso tomado en una dirección aleatoria especificada por el giro, entonces el movimiento de quimiotaxis artificial se representa con:

$$\theta^i(j,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}\ , \tag{7}$$

donde Δ es un vector que contiene una dirección aleatoria cuyos elementos se encuentran entre [-1,1].

Enjambre. Las celdas reales responden a estímulos químicos para formar grupos de células y así viajar en el ambiente. Las señales de célula a célula se representan de la siguiente manera:

$$\begin{aligned} J_{cc}\big(\theta, P(j,k,l)\big) = &\sum_{i=l}^{S}[-d_{attractant}\exp(-w_{attractant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \\ &+ \sum_{i=l}^{S}\left[-h_{repellant}\exp\left(-w_{repellant}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2\right)\right], \end{aligned} \tag{8}$$

donde J_{cc} es el valor que se agregará a la función objetivo que será minimizada $w_{attractant}$, $d_{attractant}$, $h_{repellant}$, $w_{repellant}$ son coeficientes que deben escogerse de manera adecuada.

Reproducción. Las bacterias menos saludables deben morir mientras las bacterias más saludables que son las que tienen un valor más bajo en la función objetivo se van a reproducir dividiéndose en dos, manteniendo el tamaño de la población constante.

Eliminación/dispersión. Para simular los eventos de dispersión y eliminación, se eliminan un grupo de bacterias de manera aleatoria con una pequeña probabilidad, y se inicializan los reemplazos de manera aleatoria sobre el espacio de búsqueda.

2.4. Metodología propuesta

A continuación, se presentan los pasos que conforman la metodología para obtener el modelo optimizado (Fig. 4).

1. Adquisición de datos: Los datos puros para crear el modelo son obtenidos de una base de datos.
2. Filtrado de datos: La base de datos contiene algunos datos no válidos o que pudieran ser parciales y deben de ser filtrados para no tener un modelo sesgado.
3. Entrada de datos: una vez que los datos han sido revisados y son válidos, entonces pueden ser alimentados al modelo.

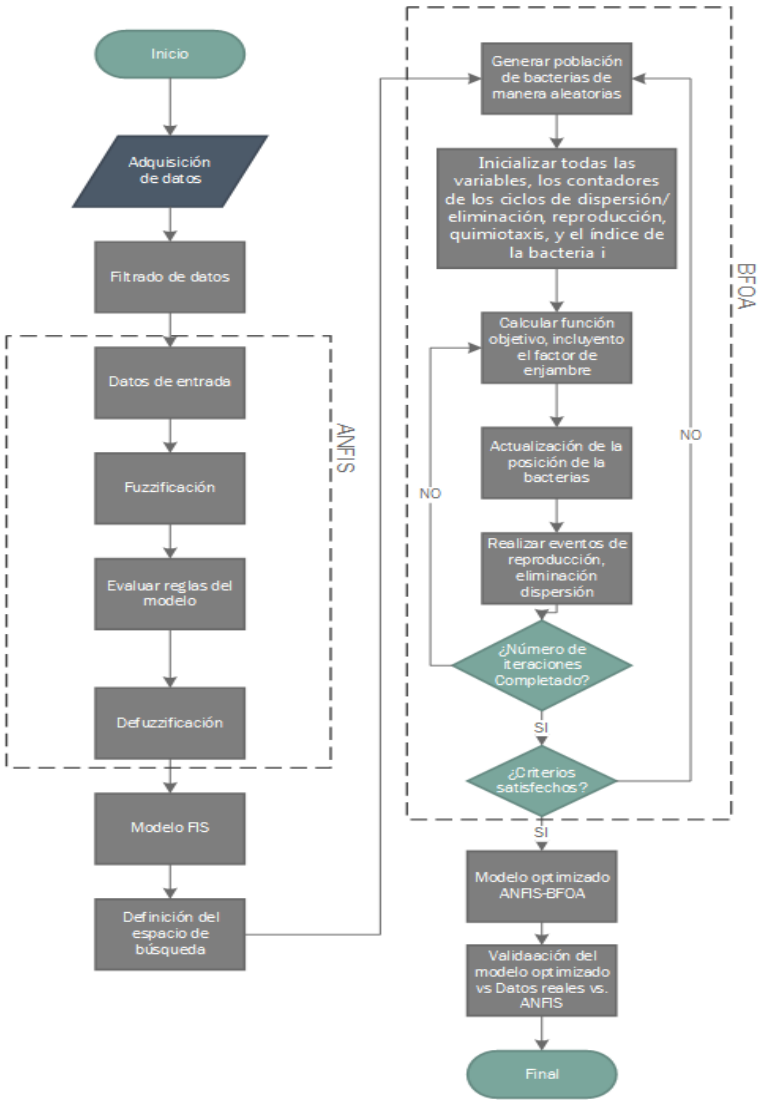


Fig. 4. Modelo optimizado.

4. Fuzzificación: Es el proceso de convertir en los datos de entrada en un valor lingüístico, esto depende de las funciones de membresía (ecuaciones 1 y 2).
5. Evaluación de las reglas: las reglas del modelo son evaluadas con respecto a las reglas difusas y los valores de las funciones de membresía (ecuaciones 3 y 4).
6. Defuzzificación: El método de defuzzificación utilizado fue el de promedio ponderado de todas las salidas de reglas (ecuación 5).
7. Modelo Sistema de inferencia Difuso: Una vez que se completaron los pasos de evaluación y defuzzificación, el modelo se construye con las ecuaciones propias de ANFIS expresadas en la sección de construcción del modelo (ecuación 6).

8. Definición del espacio de búsqueda: Un espacio de búsqueda se define como todas las soluciones factibles dentro de nuestro problema, es por esto que nuestro espacio de búsqueda debe de encontrarse primero, entre los datos válidos para los factores ambientales Humedad relativa, temperatura, dirección y velocidad del viento así como también dentro de los valores factibles de PM10, ya que dada la naturaleza del problema, otra forma de definir el espacio de búsqueda es muy difícil ya que las posibilidades son muy amplias
9. Generación de la población: se deben de generar la población con S bacterias, en posiciones iniciales aleatorias dentro de un rango de posibles valores que puedan tener los datos reales.
10. Inicialización de parámetros: Los parámetros de BFOA se deben inicializar, estos parámetros incluyen inicializar los contadores de los ciclos de eliminación/dispersión j, reproducción k, quimiotaxis l, y el índice s de la bacteria i, así como los parámetros de atracción y repelente ($w_{attractant}$, $d_{attractant}$, $h_{repellant}$, $w_{repellant}$), que son los que generan el efecto de enjambre, en este caso se utilizaron los valores de $d_{attractant} = 0.05, w_{attractant} = 0.15$, $h_{repellant}$=$d_{attractant}$, $w_{repellant} = 10$

    Estos valores fueron seleccionados en un principio de acuerdo a estudios realizados por algunos autores[17] mismos que recomiendan se seleccionen siguiendo ciertos criterios, por ejemplo la altura o magnitud del repelente debe ser igual al ancho de la señal de atracción para evitar tener dos bacterias en el mismo punto y esto se modela haciendo que $h_{repellant} = d_{attractant}$,

    Por otro lado, también recomienda que la señal de atracción sea muy pequeña en comparación a los valores de concentración de nutrientes en nuestro espacio de búsqueda y por eso el repelente debe ser lo suficientemente grande para evitar que las bacterias estén muy cerca, sin embargo, el experimentar con diferentes variantes de estos valores es parte de trabajos futuros para localizar el valor óptimo de este parámetro.

11. Cálculo de la función objetivo: Se debe de calcular el criterio que nos dirá que tan saludable está nuestra bacteria, en este algoritmo se agrega el cálculo del factor de enjambre J_{cc} con la ecuación 8 y agregarse al costo J de la bacteria en la posición actual. Actualizar posición de las bacterias: Se hace una actualización de las bacterias de acuerdo al costo de la bacteria en la posición actual y comparando la posición siguiente, para lograr esto se calcula el giro de la bacteria (ecuación 7) y se realiza un nado en esa dirección, se calcula el costo de la nueva posición, si esta tiene menor costo entonces se convierte en la mejor posición de la bacteria y se sigue moviendo en esa dirección, de lo contrario si termina el ciclo de nado y no encontró una mejor posición significa que no está ubicada en un ambiente adecuado y continua con la siguiente bacteria.
12. Se realizan los eventos de reproducción, dispersión y eliminación en donde la mitad mejor de la población se reproduce, es decir se hace una copia exacta de las bacterias con menor costo total y la otra mitad es reemplazada por bacterias generadas aleatoriamente, al igual un grupo con probabilidad baja son dispersadas en el espacio de búsqueda de manera aleatoria.

13. Modelo optimizado: si los ciclos del algoritmo BFO han sido completados y se cumplen los criterios de optimización se dice que se ha creado un Modelo Optimizado.

3. Resultados

En la Fig. 5 se pueden observar los datos reales de concentración de PM10 y su comportamiento no lineal.

El objetivo de los experimentos realizados con el método de optimización BFOA es la reducción del error que existe al aplicar el modelo creado con ANFIS. Este error se puede observar en la Fig. 6, la cuantificación del error entre los datos reales y los calculados por el modelo ANFIS se llevó a cabo utilizando la raíz del error cuadrático medio (RMSE por sus siglas en inglés) que es un método históricamente utilizado para medir la precisión de los pronósticos de datos [31].

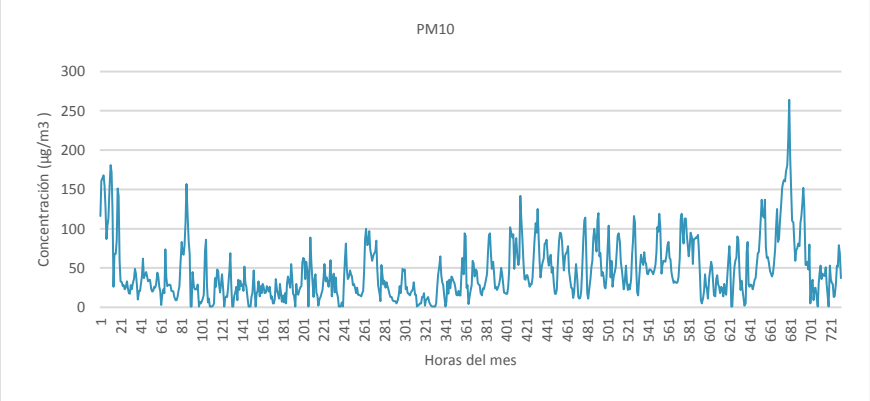


Fig. 5. Concentración de PM10 enero 2016 estación FAC.

En el caso del modelo generado con ANFIS se obtiene un RMSE= 24.147, el cual debe de ser minimizado con BFOA.

Por otro lado, si se busca generar un modelo más preciso, podemos realizar más pruebas variando los parámetros del algoritmo de optimización de Forrajeo de Bacterias, como lo son el número de bacterias que se moverán en el espacio de búsqueda, la cantidad de pasos de quimiotaxis y de pasos de reproducción para determinar cuáles son los valores óptimos de tales parámetros.

En la Fig. 7 y la tabla 1 observamos cómo la cantidad de bacterias, afecta en el rendimiento del modelo optimizado. Como puede observarse, al tener más agentes que busquen en el espacio de búsqueda, es más probable que encuentren una solución óptima a nuestro caso de estudio.

Así mismo, se puede observar que por la naturaleza de BFOA, ya que es un algoritmo que cuenta con ciclos anidados, el tiempo de ejecución aumenta de manera considerable, y aunque el error se ve reducido al aumentar el tamaño de la población, esta diferencia en la optimización no parece importante, tomando en cuenta los recursos computacionales que requiere su cálculo. Como contexto para las pruebas de tiempo de

ejecución, mencionamos que se utilizó una PC con Windows 7, de 64 bits, Procesador Intel Core i3-2100 3.10 GHz y memoria RAM de 12Gb.

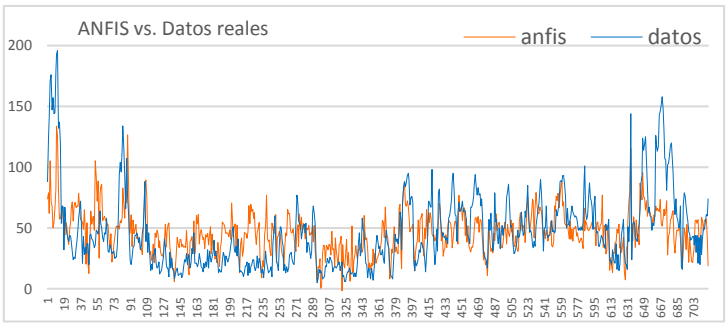


Fig. 6. Comparación datos reales vs ANFIS.

Tabla 1. Relación de cantidad de bacterias, RMSE y tiempo de ejecución.

Bacterias	RMSE	tiempo/seg
10	19.8851195	419.84
50	10.1072	2268.786
100	7.07398003	4185.117
200	5.32684396	8062.675
500	3.75769854	20625.8073

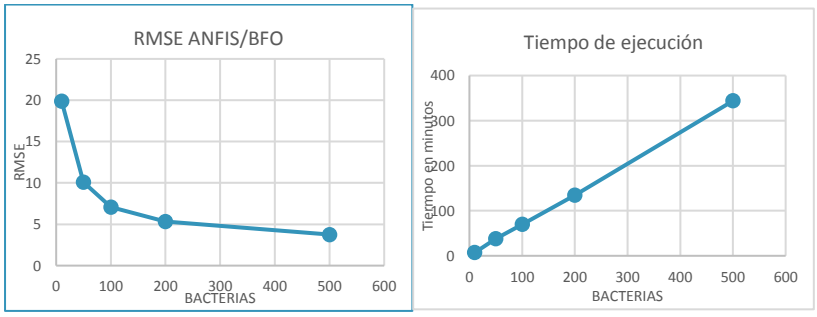


Fig. 7. Aumento de tiempo de ejecución.

Otro parámetro con el que se experimentó es el número de pasos de quimiotaxis, Nc, cuando este parámetro tiene un valor grande significa que tendremos numerosos pasos quimiotácticos lo cual podría significar un mayor avance de optimización, en la Fig. 8 podemos observar una disminución en el error cuando Nc=10 y en especial cuando la población S es mayor (S=500), pero, esto supone también una mayor complejidad computacional. Si elegimos que el tamaño de Nc va a ser muy pequeño, el algoritmo va a depender de más de la suerte al generarse la población aleatoria y la reproducción de las mejores bacterias, incluso en algunos casos, podría quedar atorado en un mínimo local (convergencia prematura).

En la Fig. 9 se observa el error correspondiente al método ANFIS con respecto a los datos reales (línea azul), en comparación con la tasa de error del modelo optimizado

ANFIS/ BFOA respecto a los datos reales (línea naranja), esta comparación muestra que al ser menor su tasa de error, el modelo optimizado con BFOA se ajusta mejor a los datos reales que el método ANFIS, por lo tanto, sí puede ser optimizado el modelo usando éste método.

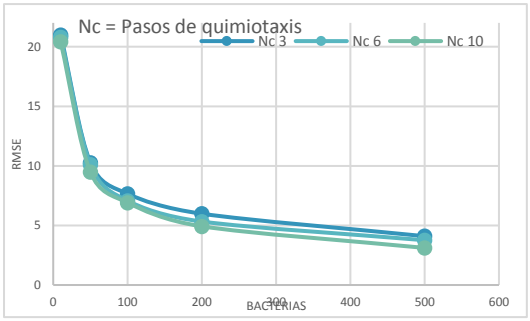


Fig. 8. Diferentes valores para Nc (pasos de quimiotaxis).

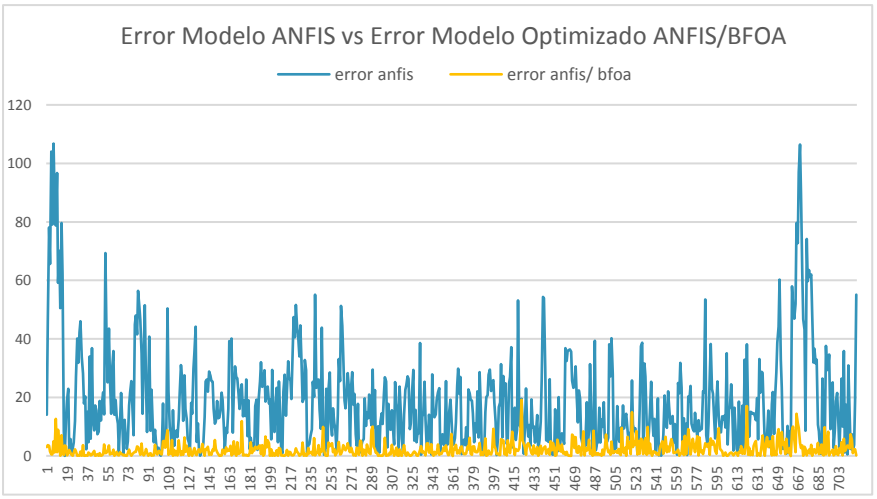


Fig. 9. Comparación de errores del Modelo ANFIS y Modelo optimizado BFOA/ANFIS.

4. Conclusiones

Este trabajo muestra que se puede optimizar un modelo previo construido en ANFIS utilizando el algoritmo de optimización de forrajeo bacteriano (BFOA). El incremento de ajuste del modelo optimizado depende en su mayoría de los parámetros de tamaño de la población S y número de pasos quimiotácticos, pasos de nado, por mencionar algunos.

Y aunque se demostró que el método sí presentó una mejora en el modelo en cuanto a error, dependerá del problema, su dimensionalidad y los requerimientos de optimización donde sea aplicado ya que también demostró aumentar la complejidad del

tiempo de cómputo lo cual podría ser un problema si se cuenta con recursos limitados para construir el modelo.

En futuros trabajos se puede utilizar el algoritmo de forma híbrida con otros métodos para buscar una forma de reducir el tiempo de ejecución e investigar si su rendimiento mejora.

Referencias

1. Kelly, F.J., Fussell, J.C.: Size, source and chemical composition as determinants of toxicity attributable to ambient particulate matter. Atmospheric environment, 60, pp. 504–526 (2012)
2. Brook, R.D., Rajagopalan, S., Pope III, C.A., Brook, J.R., Bhatnagar, A., Diez-Roux, A.V., Peters, A.: Particulate matter air pollution and cardiovascular disease: an update to the scientific statement from the American Heart Association. Circulation 121(21), 2331–2378 (2010)
3. Tellez-Rojo, M.M., Romieu, I., Ruiz-Velasco, S., Lezana, M.A., Hernandez-Avila, M.M.: Daily respiratory mortality and PM10 pollution in Mexico City: importance of considering place of death. European Respiratory Journal, 16(3), 391–396 (2000)
4. Cortina–Januchs, M.G., Quintanilla–Dominguez, J., Vega–Corona, A., Andina, D.: Development of a model for forecasting of PM10 concentrations in Salamanca, Mexico. Atmospheric Pollution Research, 6(4), 626–634 (2015)
5. Moustris, K.P., Ziomas, I.C., Paliatsos, A.G.: 3-Day-ahead forecasting of regional pollution index for the pollutants NO 2, CO, SO 2, and O 3 using artificial neural networks in Athens, Greece. Water, Air, & Soil Pollution, 209(1-4), 29–43 (2010)
6. Hooyberghs, J., Mensink, C., Dumont, G., Fierens, F., Brasseur, O.: A neural network forecast for daily average PM10 concentrations in Belgium. Atmospheric Environment, 39(18), 3279–3289 (2005)
7. Alkasassbeh, M., Sheta, A.F., Faris, H., Turabieh, H.: Prediction of PM10 and TSP air pollution parameters using artificial neural network autoregressive, external input models: a case study in salt, Jordan. Middle-East Journal of Scientific Research, 14(7), 999–1009 (2013)
8. Zarandi, M.F., Faraji, M.R., Karbasian, M.: Interval type-2 fuzzy expert system for prediction of carbon monoxide concentration in mega-cities. Applied soft computing, 12(1), 291–301 (2012)
9. Aceves-Fernandez, M.A., Pedraza-Ortega, J.C., Sotomayor-Olmedo, A., Ramos-Arreguín, J.M., Vargas-Soto, J.E., Tovar-Arriaga, S.: Analysis of key features of non-linear behavior using recurrence quantification. Case study: urban airborne pollution at Mexico City. Environmental Modeling & Assessment, 19(2), 139–152 (2014)
10. Aceves-Fernandez, M.A., Estrada, A.L., Pedraza-Ortega, J.C., Gorrostieta-Hurtado, E., Tovar-Arriaga, S.: Design and Implementation of Ant Colony Algorithms to Enhance Airborne Pollution Models. Int. J. Environ. Sci. Toxic.. vol. 3(1), (2015)
11. Beni, G., Wang, J.: Swarm intelligence in cellular robotic systems. In: Robots and biological systems: towards a new bionics? Springer, Berlin, Heidelberg.w, pp. 703–712 (1993)
12. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics, 26(1), 29–41 (1996)
13. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department (2005)
14. Hosseini, H.S.: Problem solving by intelligent water drops. In: 2007 IEEE congress on evolutionary computation, pp. 3226–3231 (2007)

15. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, Berlin, Heidelberg, pp. 65–74 (2010)
16. Hedayatzadeh, R., Salmassi, F.A., Keshtgari, M., Akbari, R., Ziarati, K.: Termite colony optimization: A novel approach for optimizing continuous problems. In: 2010 18th Iranian Conference on Electrical Engineering. IEEE, pp. 553–558 (2010)
17. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE control systems magazine, 22(3), 52–67 (2002)
18. Tripathy, M., Mishra, S.: Bacteria foraging-based solution to optimize both real power loss and voltage stability limit. IEEE Transactions on Power Systems, 22(1), 240–248 (2007)
19. Das, S., Biswas, A., Dasgupta, S., Abraham, A.: Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: Foundations of Computational Intelligence vol. 3, Springer, Berlin, Heidelberg, pp. 23–55 (2009)
20. Ali, E.S., Abd-Elazim, S.M.: BFOA based design of PID controller for two area Load Frequency Control with nonlinearities. International Journal of Electrical Power & Energy Systems, 51, pp. 224–231 (2013)
21. Wu, C., Zhang, N., Jiang, J., Yang, J., Liang, Y.: Improved bacterial foraging algorithms and their applications to job shop scheduling problems. In: International Conference on Adaptive and Natural Computing Algorithms. Springer, Berlin, Heidelberg, pp. 562–569 (2007)
22. Mahapatra, G., Banerjee, S.: A study of bacterial foraging optimization algorithm and its applications to solve simultaneous equations. International Journal of Computer Applications, 72(5), 1–6 (2013)
23. Chen, Y.P., Li, Y., Wang, G., Zheng, Y.F., Xu, Q., Fan, J.H., Cui, X.T.: A novel bacterial foraging optimization algorithm for feature selection. Expert Systems with Applications 83, pp. 1–17 (2017)
24. Narendhar, S., Amudha, T.: A hybrid bacterial foraging algorithm for solving job shop scheduling problems. arXiv preprint arXiv:1211.4971 (2012)
25. Abd-Elazim, S.M., Ali, E.S.: A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design. International Journal of Electrical Power & Energy Systems, 46, pp. 334–341 (2013)
26. Azizipanah-Abarghooee, R.: A new hybrid bacterial foraging and simplified swarm optimization algorithm for practical optimal dynamic load dispatch. International Journal of Electrical Power & Energy Systems 49, pp. 414–429 (2013)
27. Sistema de Monitoreo Atmosférico (SIMAT). El monitoreo de la calidad del aire, http://www.aire.cdmx.gob.mx/default.php?opc=%27ZaBhnmI=%27. Último acceso: 2019/05/01
28. Jang, J.S.: ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics, 23(3), 665–685 (1993)
29. Abdulshahed, A.M., Longstaff, A.P., Fletcher, S.: The application of ANFIS prediction models for thermal error compensation on CNC machine tools. Applied Soft Computing 27, pp.158–168 (2015)
30. Hansen, C.H., Endres, R.G., Wingreen, N.S.: Chemotaxis in Escherichia coli: a molecular model for robust precise adaptation. PLoS computational biology, 4(1), e1 (2008)
31. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. International journal of forecasting, 22(4), 679–688 (2006)

ISSN 1870-4069

Evolución diferencial para la optimización global de procesos de ingeniería química

Maritza Yanet Martínez Zecua, Luis Angel Salamanca Vázquez, Leticia Flores Pulido, Edgar Alfredo Portilla Flores, Arturo Ortíz Arroyo

[1] Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas, Ingeniería y Tecnología, Apizaco, Tlaxcala, México
[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Ciudad de México, México
yanetzecua.13@gmail.com, angel.salamanca.vazquez@gmail.com, leticia.flores.p@uatx.mx, arturo.ortiz@uatx.mx, aportilla@ipn.mx

Resumen. El diseño de reactores es un aspecto fundamental para la producción de casi todas las sustancias químicas industriales, ya que la selección de un sistema de reacción que opere de la forma más segura y eficiente posible puede ser la clave del éxito o del fracaso productivo de una planta de productos quṕicos. Existen dos grandes procesos dentro de una planta química: los reactores y los procesos en redes de calentadores de sustancias químicas. En este artículo se definen dos problemas de esta clase por medio de el enfoque de optimización. Una vez que se hace esta clase de definiciones, se procede a optimizar sus funciones con la intencion de mejorar los minimos propuestos por planteamientos previos en el estado del arte por medio de tecnicas no tradicionales, es decir, hacendo uso de técnicas heurísticas. Se implementaron entonces seis variantes del algoritmo de evolución diferencial. Cada una de las variantes optimiza las funciones de ambos problemas dentro del contexto de ingeniería química, alcanzando minimos comparablemente mas rapido que enfoques previos. Se realiza una comparativa entre las seis variantes del algoritmo y se menciona cual de ellos alcanza una mejor factibilidad y convergencia.

Palabras clave: evolución diferencial, diseño de intercambiador de calor, diseño de red de reactores, pptimización, metaheurística.

Evolution for the Global Optimization of Chemical Engineering Processes

Abstract. The design of reactors is a fundamental aspect for the production of almost all industrial chemical substances, since the selection of a reaction system that operates in the safest and most efficient way possible can be the key to success or the productive failure of a chemical products plant. There are two major processes within a chemical plant: reactors and processes in networks of chemical substance heaters. In this

article two problems of this kind are defined by means of the optimization approach. Once this kind of definition is made, we proceed to optimize its functions with the intention of improving the minimums proposed by previous approaches in the state of the art by means of non-traditional techniques, that is, making use of technical heur Isticas. Six variables were then implemented in the differential evolution algorithm. Each of the variants optimizes the functions of both problems within the context of chemical engineering, reaching minimums comparably faster than previous approaches. A comparison is made between the six variants of the algorithm and it is mentioned which of them achieves a better feasibility and convergence.

Keywords: evolution on differential, design or heat exchanger, design or reactor network, optimization, metaheuristic.

1. Introducción

Los algoritmos evolutivos (AE) basan su funcionamiento en el proceso de selección natural propuesto por Charles Darwin. Estos algoritmos clasificados como metaheurísticos no aseguran obtener el óptimo global, sin embargo producen una muy buena solución en un tiempo de cómputo razonable, además que no se requiere del conocimiento específico del problema a resolver. Debido a que se pretende que el método de solución no tenga una alta sensitividad a las condiciones iniciales para la búsqueda de soluciones, se eligió que las estrategias del algoritmos evolutivos utilizadas en el presente trabajo fueran las siguientes: generación aleatoria de una población inicial y creación de nuevas direcciones de búsqueda basadas en la distribución de las soluciones (individuos) de la población actual. Se busca entonces, que el proceso de optimización avance, y que la selección de las direcciones de búsqueda (las cuales en un principio son altamente aleatorias) estén basadas en las mejores soluciones encontradas a partir de la población actual. Un algoritmo que satisface lo establecido anteriormente es el denominado Evolución Diferencial (ED) [2]. Uno de los trabajos realizados en aplicaciones de ingeniería qímica es el trabajo realizado en [3] donde Evolución Diferencial es aplicada en dos problemas no-lineales de ingeniería química, además de que se realiza la comparación de su rendimiento con el del algoritmo ramificación y acotación, estos problemas nosotros también los aplicamos y así mismo comparamos los resultados que ellos obtuvieron con los nuestros.

Más adelante se presenta Evolución Diferencial Modificado (EDM) para la optimización de procesos químicos no-lineales[4], en el cual se elabora una modificación a lo que es Evolución Diferencial (ED), mediante el argumento de que a pesar de que ED ha demostrado ser un eficiente, efectivo y robusto método de optimización evolutiva, aun toma largo tiempo computacional para optimizar costosas funciones objetivas, por lo tanto se considero necesario mejorar la tasa de convergencia sin comprometer la calidad de la solución. En dicha investigación el algoritmo de EDM es implementado en cinco problemas de ingeniería química no lineales, comparando los resultados obtenidos contra Evolución Diferencial

(ED) considerando el historial de convergencia (tiempo de CPU y el número de ejecuciones convergidas a óptimo global) así como las técnicas estadísticas, tomando en cuenta la variabilidad de los resultados. Otro trabajo similar es el realizado en [5], donde se resuelven problemas que pertenecen a la amplia clase de NLP restringidas dos veces diferenciables. El algoritmo se basa en un enfoque de ramificación y unión, donde se obtiene un límite inferior en la solución óptima en cada nodo mediante la generación automática de un problema válido de subestimación convexo.

Finalmente en [6] por medio de búsqueda y ramificación se resuelven múltiples problemas no lineales aplicandose en diseño de procesos, de igual manera se muestran los problemas de la red de calentadores y de reactores químicos con planteamientos de optimiación desde otros enfoques. Este artículo entonces describe la propuesta de el uso de evolución diferencial para resolver un problema de optimización de aplicaciones de ingeniería química, se analiza el algoritmo de ED y finalmente se muestran resultados de factibilidad y convergencia, comparables con los planteamientos dentro del estado del arte. Se logra identificar la variante mas eficiente de ED para cada problema y se mejoran los resultados obtenidos en planteamientos variados para las mismas aplicaciones.

2. Propuesta de investigación

El reactor químico realiza un proceso de concentración de soluciones y concentraciones químicas, a partir de ciertas condiciones y funciones ya definidas que permiten modelar un proceso de optimización para generar concentraciones químicas como nafta, gasolina y otro tipo de sustancias derivadas del petróleo. Posteriormente, se dan dos problemas de optimización como entrada al algoritmo de evolución diferencial en seis variantes y se procede a realizar experimentos para analizar cual de ellos arroja un mejor desempeño para la función objetivo a los casos de estudio presentados por medio de comparaciones, después con los resultados obtenidos en la optimización se procede a mostrarlo mediante una interfaz demostrativa. Este proceso se presenta en la Figura 1.

3. Evolución diferencial

La optimización es un proceso de búsqueda, de la mejor solución o conjunto de soluciones en un problema específico, bajo circunstancias determinadas, como método para alcanzarla se utiliza Evolución Diferencial (ED) donde la idea principal es un nuevo esquema para generar vectores. ED es un algoritmo de optimización estocástico propuesto por Storn y Price en 1995 para resolver problemas de optimización numérica [1]. Se trata de un algoritmo evolutivo, altamente competitivo en problemas de optimización global, que basa su estrategia en la búsqueda poblacional. De este modo, ED parte de una población de NP individuos, también llamados vectores, que representan las soluciones candidatas al problema en cuestión, como se muestra en la ecuación (1):

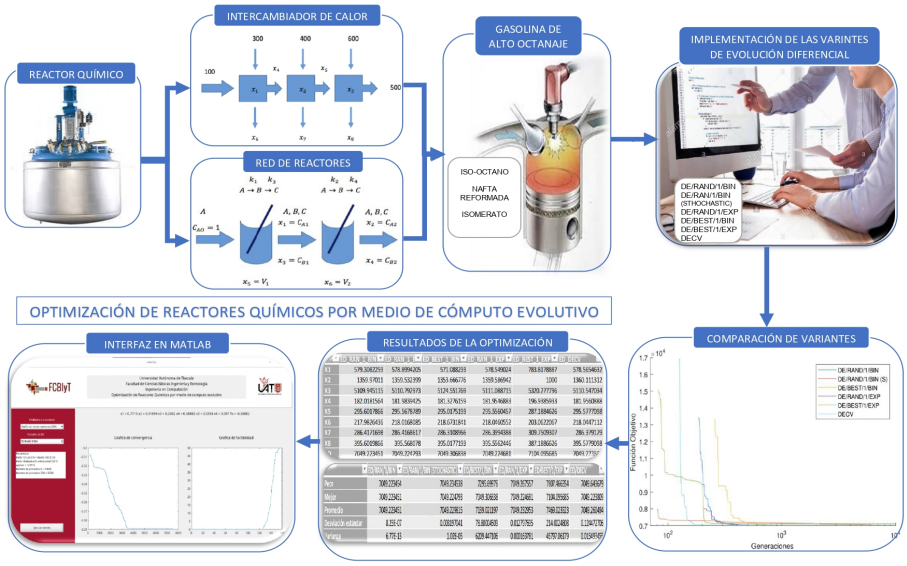


Fig. 1. Propuesta de Evolucion Diferencial para la Optimización de Procesos en Ingenieria Quimica.

$$x_{g,i} = (x_{g,i,j}), j = 1, \ldots, D, i = 1, \ldots, NP, \tag{1}$$

donde $x_{g,i,j}$ es una variable de decisión del problema, los parámetros de cada vector son indexados con la variable j, que se extiende desde 1 hasta D. Los vectores son identificados a través de un índice de población, i. La g corresponde al número de la generación actual, $g = 1, \ldots, gmax$. A continuación se describen cada una de las etapas del algoritmo de ED.

En la *Etapa de Inicialización*, los individuos de la población son inicializados aleatoriamente dentro de sus límites inferiores (L_j) y superiores (U_j), es decir: $L_j \leq x_j \leq U_j$. Dichos individuos serán sometidos a operaciones de mutación, recombinación y selección para producir nuevos descendientes que participarán en la siguiente generación reemplazando a la anterior. En la *Etapa de Mutación*, se enfatiza la dirección de búsqueda puesto que se controla la magnitud con la que los vectores se desplazan en el espacio de búsqueda y la velocidad de convergencia hacia la solución óptima. Por consiguiente, la mutación diferencial añade una diferencia de dos vectores ponderada por un factor escala, a un tercer vector, todos ellos seleccionados aleatoriamente, como se muestra en la ecuacion (2):

$$v_{g,i} = x_{g,r1} + F(x_{g,r1} - x_{g,r1}), r_1 \neq r_2 \neq r_3 \neq i \tag{2}$$

El parámetro $F \in [0, 1)$ controla el rango en que la población evoluciona. En la *Etapa de Cruza (CR)* se toma como un proceso complementario a la mutación

con la cual se logra el intercambio de información entre dos o más vectores para crear uno o más vectores hijo. Valores altos implican menor influencia del padre. Existen dos tipos de recombinación: discreta (el vector hijo obtiene los parámetros del vector padre o del vector mutante) y la continua o aritmética (el vector hijo hereda sus parámetros por medio de una combinación lineal de vectores. En la *Etapa de Selección* se evalúa si el valor de la función objetivo del vector trial u_g, i es menor (asumiendo minimización) que el vector target x_g, i , de ser así, reemplazará su posición en la siguiente generación, de otro modo, el mejor se mantiene, como se expresa en la ecuación (3):

$$x_{g+1,i} = \begin{cases} \mathrm{u}_{g,i} \; si \; (u_{g,i}) \leq f(x_{g,i}) \\ \mathrm{x}_{g,i} \; de \, otro \, modo \end{cases} \tag{3}$$

A continuación, en la Figura 2 se muestra el tradicional algoritmo de Evolución Diferencial implementado en los experimentos.

```
1:   Inicio
2:   Crear una población inicial aleatoriamente
                  x_{0,i} i = 1, ..., NP
3:   Evaluar población f(x_{0,i}) i = 1, ..., NP
4:   for g=0 to MAX_GEN do
5:     for i=1 to NP do
6:     Seleccionar aleatoriamente r1 ≠ r2 ≠ r3 ≠ i
7:     j_rand = randint[1, D]
8:     for j=1 to D do
9:       if (rand_j[0,1) < CR or j = j_rand) then
10:        u_{g,i,j} = x_{g,r1,j} + F(x_{g,r2,j} − x_{g,r3,j})
11:      else
12:        u_{g,i,j} = x_{g,i,j}
13:      end if
14:    end for
15:    if f(u_{g,i}) ≤ f(x_{g,i}) then
16:      x_{g+1,i} = u_{g,i}
17:    else
18:      x_{g+1,i} = x_{g,i}
19:    end if
20:  end for
21:  end for
22:  Fin
```

Fig. 2. Algoritmo de Evolución Diferencial, variante DE/rand/1/bin.

Existe un esquema de nomenclatura desarrollado para hacer referencia a las variantes del algoritmo de Evolución Diferencial, utilizando la siguiente notación: **DE/x/y/z**. Donde: **DE** se refiere al algoritmo Evolución Diferencial (Differential Evolution), **x** especifica la forma en la que el vector base (r_0) es seleccionado, puede ser de forma aleatoria (rand), el mejor individuo hasta el momento (best), entre otros. **y** indica el número de diferencias de vectores que contribuyen al

diferencial. **z** representa el tipo de recombinación que se utiliza, en las versiones del algoritmo DE se utiliza recombinación binomial **bin** o exponencial **exp**.

4. Implementación del diseño de red de intercambiador de calor

Los intercambiadores de calor son equipos principalmente para transferir calor entre corrientes calientes y frías. Tienen caminos separados para las dos corrientes y operan continuamente. El diseño de la red del intercambiador de calor se realiza para hacer frente a una situación en la que una variedad de flujos de procesos deben sufrir cambios de estado simultáneos que se pueden lograr mediante la adición y / o eliminación de calor. Las ventajas económicas concurrentes con el uso eficiente de flujos de proceso, como medios de calentamiento o enfriamiento, sirven como incentivo para administrar cuidadosamente las especificaciones de los equipos auxiliares e instalaciones tales como enfriadores de agua y vapor de los calentadores encendidos. Bajo las circunstancias habituales, sin embargo, no está claro de antemano cómo hacer frente a las demandas de calor y los suministros de manera que se optimice el rendimiento del sistema requerido. Este problema aborda el diseño de una red de intercambiador de calor como se muestra en la Figura 3. Un flujo frio debe ser calentado de 100 °F hasta 500 °F utilizándo tres flujos calientes con diferentes temperaturas de entrada. El objetivo es minimizar el tamaño del área total de intercambio de calor. Este problema se ha tomado como referencia de Floudas & Pardalos, con un planteamiento realizado en 1990 [8]. Incluso este caso de estudio ha sido resuelto por Adjiman en 1998 [5] utilizándo el algorimo $\alpha\beta$, también conocido como el algoritmo de búsqueda y ramificación.

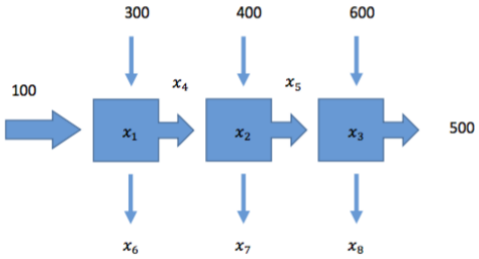


Fig. 3. Caso de Estudio 1: Diseño de la Red de Intercambio de Calor.

La función a minimizar está dada por la ecuacion (4):

$$minf(x) = x_1 + x_2 + x_3. \tag{4}$$

Sujeto a las restricciones descritas en las ecuaciones de la (5) a la (10):

$$g_1(x) = 0.0025(x_4 + x_6) - 1 = 0, \tag{5}$$

$$g_2(x) = 0.0025(-x_4 + x_5 + x_7) - 1 = 0, \tag{6}$$

$$g_3(x) = 0.01(-x_5 + x_8) - 1 = 0, \tag{7}$$

$$g_4(x) = 100x_1 - x_1x_6 + 833.33252x_4 - 83333.333 \leq 0, \tag{8}$$

$$g_5(x) = x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0, \tag{9}$$

$$g_6(x) = x_3x_5 - x_3x_8 - 2500x_5 + 1250000 \leq 0, \tag{10}$$

donde los rangos de las variables son sujetos a las expresiones (11), (12) y (13):

$$100 \leq x_1 \leq 10000, \tag{11}$$

$$1000 \leq x_2, x_3 \leq 10000, \tag{12}$$

$$10 \leq x_4, x_5, x_6, x_7, x_8 \leq 1000, \tag{13}$$

donde x_1, x_2 y x_3 son áreas de intercambio de calor y x_4, x_5, x_6, x_7 y x_8 son temperaturas de flujos, como se muestra en la Figura 4. La mejor solución está localizada en los puntos mostrados en la expresión (14):

$$x^* = \langle 579.19, 1360.13, 5109.92, 182.01, 295.60, 217.9, 286.40, 395.60 \rangle. \tag{14}$$

Se sabe además que $f(x)^* = 7049.25$.

5. Implementación del diseño de red de reactores

Este es un ejemplo propuesto por Ryoo & Sahinidis en 1995 [6], es un problema de diseño de una red de reactores, descrito en el sistema mostrado en la Figura 4. Involucra el diseño de una secuencia de dos reactores de *tanque agitado continúo* (CSTR), donde la reacción consecutiva $A \rightarrow B \rightarrow C$ toma lugar. El objetivo es maximizar la concentración del producto B en la salida del flujo. Este problema es conocido por causar dificultades para algunos métodos de optimizacion global [8].

La función a minimizar está dada por la ecuación (15):

$$minf(x) = -x_4. \tag{15}$$

Sujeto a las siguientes restricciones de las ecuaciones de la (16) a la (20):

$$g_1(x) = x_1 + k_1x_2x_5 = 1, \tag{16}$$

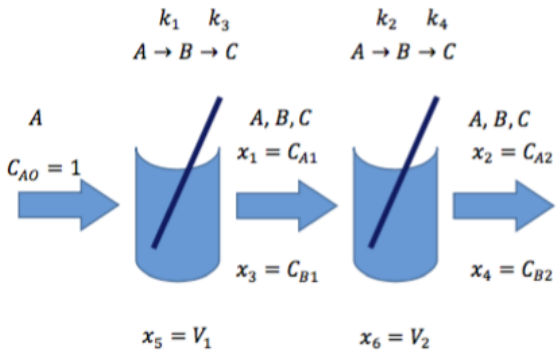


Fig. 4. Caso de Estudio 2: Diseño de la Red de Reactores.

$$g_2(x) = x_2 - x_1 + k_2 x_2 x_6 = 0, \tag{17}$$

$$g_3(x) = x_3 + x_1 + k_3 x_3 x_5 = 1, \tag{18}$$

$$g_4(x) = x_4 - x_3 + x_2 - x_1 + k_4 x_4 x_6 = 0, \tag{19}$$

$$g_5(x) = x_5^{0.5} + x_6^{0.5} \leq 4, \tag{20}$$

donde los rangos de las variables son se definen en las expresiones (21) a la (26):

$$0 \leq x_1 \leq 1, \tag{21}$$

$$0 \leq x_2 \leq 1, \tag{22}$$

$$0 \leq x_3 \leq 1, \tag{23}$$

$$0 \leq x_4 \leq 1, \tag{24}$$

$$10^{-5} \leq x_5 \leq 16, \tag{25}$$

$$10^{-5} \leq x_6 \leq 16. \tag{26}$$

Y las constantes tienen los valores descritos en las expresiones (27) a la (30):

$$k_1 = 0.09755988, \tag{27}$$

$$k_2 = 0.99k_1, \tag{28}$$

$$k_3 = 0.0391908, \tag{29}$$

$$k_4 = 0.9k_3. \tag{30}$$

El óptimo global está descrito en la expresion (31):

$$x^* = \langle 0.771462, 0.516997, 0.204234, 0.388812, 3.036504, 5.096052 \rangle. \tag{31}$$

Se sabe además que $f(x)^* = -0.388812$. Este caso constituye un caso de prueba muy complejo puesto que se puede observar un minimo local con una función objetivo que es muy cercana a la solución global. Las soluciones locales son $f = -0.375$ y $f = -0.3881$. Por si no fuera suficientemente interesante, las dos soluciones locales utilizan solamente uno de los dos reactores mientras la solución global hace uso de ambos reactores.

6. Parámetros de sintonización

Al tratar con Algoritmos Evolutivos (AE), no se puede dejar a un lado la fuerte dependencia de sus parámetros de entrada con respecto al comportamiento del algoritmo, ya que éstos juegan un papel importante en la búsqueda de mejores soluciones; de ahí que realizar una configuración adecuada de sus parámetros sea crucial para obtener un correcto desempeño del algoritmo. La configuración de parámetros aplicada en nuestro trabajo, es aquella que suele hacerse sin una clara justificación del valor asignado, siendo a prueba y error, entre un rango de valores aleatorios con una calibración de parámetros, los valores adecuados para ellos son obtenidos mediante un proceso previo donde se ejecuta el algoritmo muchas veces con diferentes valores de parámetros, y mediante validación estadística o búsqueda, se proveen aquellos valores que proporcionen buenos resultados.

En esta última sección se muestra el análisis de los parámetros de las diferentes versiones de Evolución Diferencial donde se determinaron los parámetros relevantes del algoritmo y sus correspondientes rangos de aplicación para cada problema de optimización. Para ello las seis variantes del algoritmo se trabajaron bajo los mismos parámetros a excepción del número de generaciones respecto a cada problema, como se muestra en la Tabla 1 y en la Tabla 2.

7. Gráficas de factibilidad y convergencia

A continuación se muestran las gráficas de factibilidad para ambos casos de estudio. También se muestran las tablas de convergencia para las 6 variantes de ED.

Tabla 1. Tabla de parámetros para el Diseño de Intercambiador de Calor.

Parámetro	Valor
Factor de Cruza	Aleatorio[0.8,1.0]
Factor de Mutación	Aleatorio[0.3,0.9]
Número de población	50
Generacion máxima	3000
Epsilón	$1X10^{-6}$

Tabla 2. Tabla de parámetros para el Diseño de Red de Reactores.

Parámetro	Valor
Factor de Cruza	aleatorio[0.8,1.0]
Factor de Mutación	aleatorio[0.3,0.9]
Número de población	50
Generacion máxima	40000
Epsilón	$1X10^{-6}$

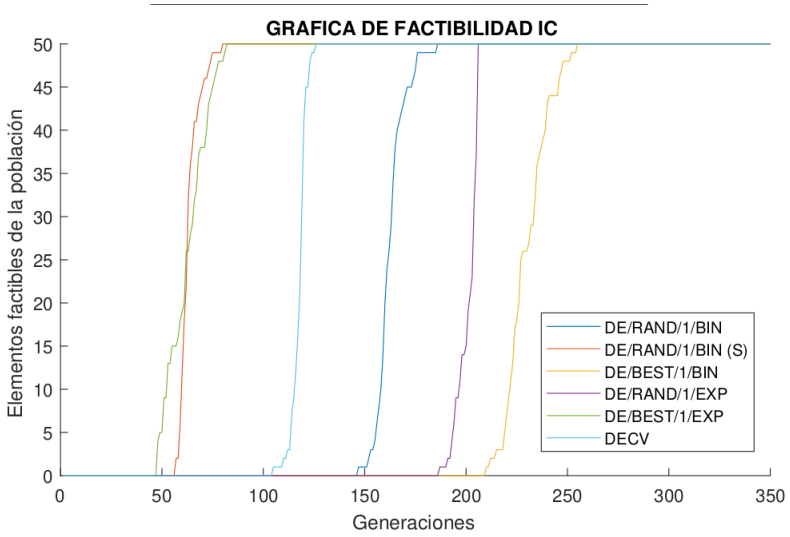


Fig. 5. Gráfica de Factibilidad del diseño de intercambiador de calor.

En la Figura 5, se visualiza que para el Caso 1: Red de intercambiador de calor, tanto el algoritmo de DE/RAND/1/BIN y DE/BEST/1/EXP son los que encuentran mas soluciones factibles que el resto de las variantes de ED. Esto indica que el espacio de búsqueda es explorado de manera mas eficiente, encontrando el óptimo en la mayoría de los casos. Dicha gráfica de factibilidad puede observarse en la Figura 5.

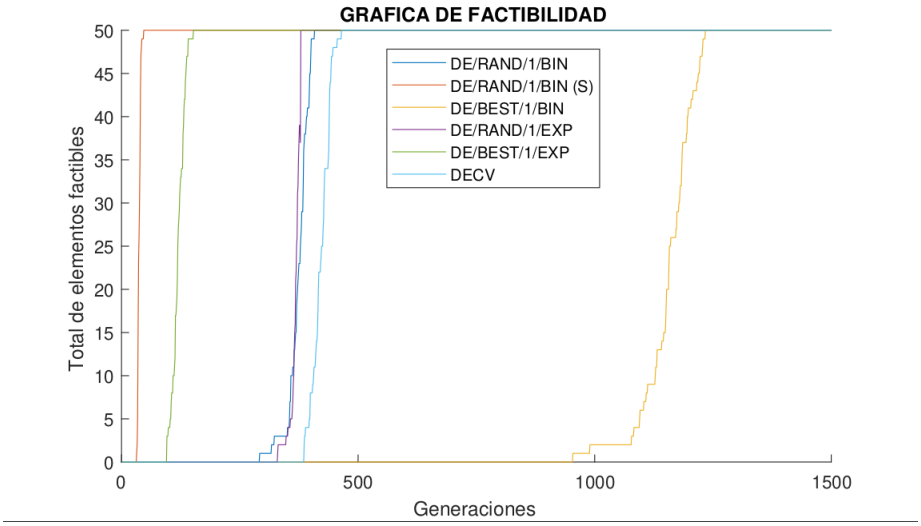


Fig. 6. Gráfica de Factibilidad del diseño de red de reactores.

Respecto a la factibilidad para el Caso 2: Red de reactores, puede mencionarse que DE/RAND/1/BIN, logró nuevamente ser la variante mas eficiente en cuanto a la exploración del espaciio de búsqueda en zonas factibles (ver Figura 6). Por otro lado, en cuanto a las convergencias de los algoritmos, podemos observar que para el Caso 1: Red de intercambiador de calor, se tiene una convergencia casi inmediata con DE/RAND/1/BIN, como se observa en la Figura 7. Para la convergencia del Caso 2: Red de reactores, se muestra una convergencia inmediata con DE/RAND/1/EXP (ver Figura 8).

8. Conclusiones

Una vez analizadas las factibilidades y convergencias, podemos mostrar quecen los casos de estudio de procesos de ingeniería química los resultados obtenidos indican que respecto al primer caso de estudio con el *intercambiador de calor*, definido previamente en tres trabajos planteados en [5] y [6], se logran los mismos valores que ellos reportan en su literatura, sin embargo con el planteamiento de [3], se logra alcanzar un mejor valor que él que se muestra en los artículos.

En lo que corresponde al segundo caso de estudio, es decir, a la *red de reactores*, los resultados de [7] y [8] solo logran encontrar óptimos locales. Pero en el planteamiento realizado en [6] y debido a su correcta exposición del problema, el resultado es el óptimo global planteado en su artículo, no obstante es importante destacar que con ED logramos un mejor resultado respecto al óptimo global; teniendo en cuenta que no se reporta un épsilon en ninguno de los artículos. dicho lo anterior, se utilizó un épsilon de $1x10_6$ puesto que los autores presentan

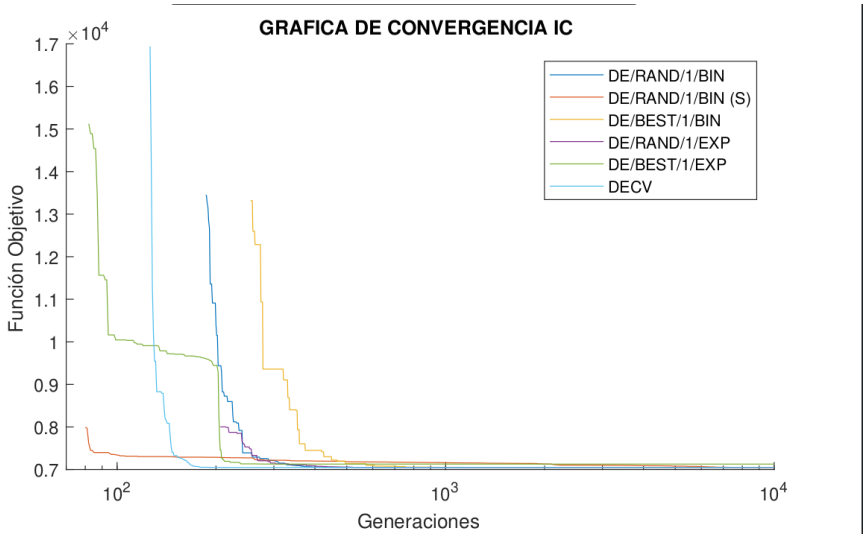


Fig. 7. Gráfica de Convergencia del diseño de intercambiador de calor.

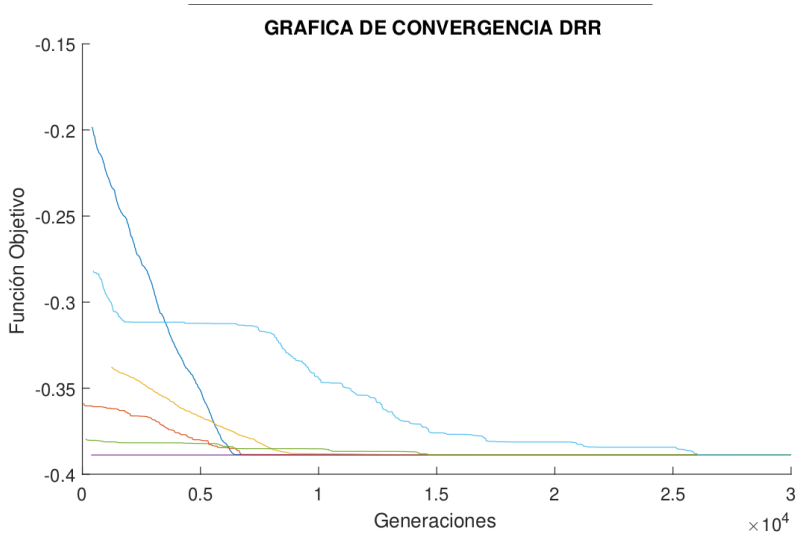


Fig. 8. Gráfica de Convergencia del diseño de red de reactores.

sus resultados con un máximo de cinco cifras significativas, estimando que con el valor de épsilon antes mencionado se cubre este rango de precisión.

Finalmente se enfatiza que la adecuada implementación de los métodos metaheurísticos brindan un buen rendimiento, eficiencia y por lo tanto un mejor enfoque en problemas de optimización, especialmente la variante de DE/RAND /1/BIN que resultó ser la variante mas eficiente en cuanto a exploración del

espacio de búsqueda del caso de estudio. También es importante mencionar que en cuanto a explotación, existe una particularidad de los algoritmos, es decir, DE/RAND/1/BIN es el mas eficiente para el Caso del intercambiador de calor, pero para el caso de la red de reactores, DE/RAND/1/EXP resulta ser el que mejor explota el espacio de soluciones factibles, puesto que encuentra un óptimo global mas rápido que las demas variantes de evolución diferencial.

Referencias

1. Storn, R., Price, K.: Differential Evolution, A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report, International Computer Science Institute, Berkeley (1995)
2. Vela, C. R., Puente, J., Alonso, C. L., Varela, R.: Computación Evolutiva para Resolución de CSPs. Revista Iberoamericana de Inteligencia Artificial, 7, 20 (2003)
3. Babu, B. V., Angira, R.: Evolutionary computation for global optimization of Nonlinear chemical engineering processes. Birla Institute of Technology and Science, 31, 5 (2003)
4. Babu, B. V., Angira, R.: Modified differential evolution (MDE) for optimization of non-linear chemical processes. Computers and Chemical Engineering, vol. 30, pp. 989–1002 (2006)
5. Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A Global Optimization Method, BB, for general twice differentiable constrained NLPs – II. Implementation and computational results, Computers and Chemical Engineering, vol. 22(9), pp. 1159–1179 (1998)
6. Ryoo, H.S., Sahinidis, B.P.: Global Optimization of Nonconvex NLPs and MINLPs with Application in Process Design. Computers and Chemical Engineering, 19, 551 (1995)
7. Floudas, C.A., Aggarwal, A., Ciric A.R. Global optimum search for nonconvex NLP and MINLP problems. Computers & Chemical Engineering, vol. 13, no. 10, pp. 1117–1132, Editor Pergamon (1998)
8. Floudas, C.A., Christodoulos, A., Pardalos P.M.: State of the Art in Global Optimization. Computational Methods and Applications. 654 p., Kluwer Academic Publishers (1996)

ISSN 1870-4069

Clasificación de enfermedades del tórax usando aprendizaje profundo y aumento de datos de calidad en el conjunto de datos ChestX-ray8

Amhed A. Ortiz-Preciado, Juan Antonio Vega-Fernández, Gilberto Ochoa-Ruiz

Universidad Autónoma de Guadalajara,
Departamento de Ciencias Computacionales, Guadalajara, México
antonio.vega@edu.uag.mx, gilberto.ochoa@edu.uag.mx

Resumen. Presentamos un método novedoso para la clasificación de enfermedades del tórax en el conjunto de datos estándar y ampliamente utilizado ChestX-ray8, el cual contiene más de 100,000 imágenes de vista frontal con 8 enfermedades. Las principales contribuciones de este artículo son tres: 1. La introducción de una evaluación de la calidad de la imagen para el pre-filtrado del conjunto de datos, ya que ha sido demostrado que los ejemplos de entrenamiento de baja calidad afectan el rendimiento de las soluciones basadas en aprendizaje profundo. 2. Un enfoque de aumento de datos basados en la calidad de la imagen, y 3. Una arquitectura simplificada de red convolucional basada en AlexNet con solamente 200,000 parámetros. Con estas modificaciones estructurales a la arquitectura de la red y el aumento de datos basados en calidad, fuimos capaces de obtener resultados comparables con otras propuestas del estado del arte que usan el mismo conjunto de datos ChestX-ray8 en 4 de 8 categorías y logramos superar los resultados en las otras 4 categorías o enfermedades, tal como lo demuestran los resultados experimentales, todas usando una red convolucional que puede ejecutarse en dispositivos pequeños.

Palabras clave: aprendizaje profundo, CNN, aprendizaje débilmente-supervisado, chestX-Ray8.

Thoracic Disease Classification Using Deep Learning and Quality-Aware Augmentation on the ChestX-Ray8 Dataset

Abstract. We present a novel method for thoracic disease classification on the now standard and widely used ChestX-ray8 dataset, which contains over 100,000 frontal view images with 8 diseases. The main contributions of this paper are threefold: 1. The introduction of a quality image assessment for prefiltering the dataset, as it has been demonstrated that training examples with low quality undermine the performances on deep learning-based solutions. 2. A quality aware data augmentation approach and 3. A simplified ConvNet Architecture based on AlexNet with only 200,000 parameters. With these architectural modifications and via the use of quality-aware data augmentation, we were able to attain comparable results to other state of the art proposals that use the ChestX-ray8

dataset on 4 of the 8 categories and surpass them in the other 4, as validated by thorough experimental results, all using a ConvNet that can run on constrained devices.

Keywords: deep learning, CNN, weakly-supervised learning, chestX-Ray8.

1. Introducción

El campo de la visión computacional ha visto un progreso exponencial en una amplia gama de aplicaciones debido al uso de aprendizaje profundo y especialmente a la existencia de grandes conjuntos de datos de imágenes anotadas [1]. Se han mostrado mejoras significativas en el rendimiento de problemas previamente considerados difíciles, tales como el reconocimiento de objetos, detección y segmentación sobre los enfoques basados en obtener las características de la imagen a mano [2].

En años recientes, se ha demostrado que el aprendizaje profundo puede producir incrementos en rendimiento similares en el dominio del análisis de imágenes médicas para tareas de detección y segmentación de objetos [3]. Trabajos recientes notables incluyen aplicaciones médicas importantes, por ejemplo, en el dominio de la neumología (clasificación de enfermedades de los pulmones [4] y detección de nódulos pulmonares en imágenes de TC [5]). Comparado con otras aplicaciones en visión computacional, la principal limitante en las aplicaciones médicas de aprendizaje automático es que la mayoría de los métodos propuestos son evaluados utilizando un conjunto de datos mas bien pequeño con cientos de pacientes cuando mucho, pero se han hecho avances en los últimos años con la introducción de conjuntos de datos disponibles públicamente. En este artículo, enfocamos nuestros esfuerzos en la identificación y clasificación de enfermedades del tórax, un problema que presentamos en las siguientes secciones.

1.1. Uso de aprendizaje profundo en la identificación de enfermedades del tórax

Más de un millón de adultos son hospitalizados con neumonía y cerca de 50,000 mueren de esta enfermedad solamente en los EE. UU. cada año [6]. Las radiografías o placas del tórax actualmente son el mayor método para diagnosticar la neumonía [7], y juegan un papel crucial en los cuidados clínicos y los estudios epidemiológicos. Sin embargo, detectar la neumonía en las radiografías es una tarea difícil que depende de la disponibilidad de especialistas radiólogos, por lo que existe una amplia oportunidad de ayudar a los especialistas con nuevas tecnologías de Inteligencia Artificial para facilitar su trabajo y mejorar el sistema de salud.

Como se mencionó anteriormente, la placa del tórax se utiliza en el cribado y diagnóstico de muchas enfermedades de los pulmones. Un gran número de estudios de radiografías y reportes radiológicos típicamente son almacenados en los sistemas PACS (Sistema de Comunicación y Archivo de Imágenes) de los hospitales. Se han realizado una buena cantidad de investigaciones para aprovechar el conocimiento contenido en estas bases de datos, pero existe un reto muy específico: los registros médicos frecuentemente están débilmente etiquetados y no contienen anotaciones. Esto significa que la radiografía tiene asociado el diagnóstico del médico especialista -como negativo

(paciente sano) o positivo (identificando la enfermedad). Pero la radiografía no contiene delimitada la región de la imagen que lo llevo a tal conclusión o diagnóstico. Así que la investigación reciente se ha enfocado en explotar estos conjuntos de datos con los paradigmas de aprendizaje profundo, los cuales requieren de grandes cantidades de datos o imágenes, construyendo sistemas CAD de gran escala para propósitos médicos.

Detectar las enfermedades del tórax y especialmente la neumonía en las radiografías puede ser difícil para el especialista. La apariencia de varias enfermedades en la placa es frecuentemente vaga, se puede sobreponer con otros diagnósticos, y puede imitar otras anormalidades benignas u otras enfermedades.

En la figura 1, se muestran algunas de las enfermedades más comunes y el enfoque de este y otros enfoques del estado del arte. Como se puede observar, la calidad de muchas de estas imágenes es subóptima y no es adecuada para los algoritmos de aprendizaje automático.

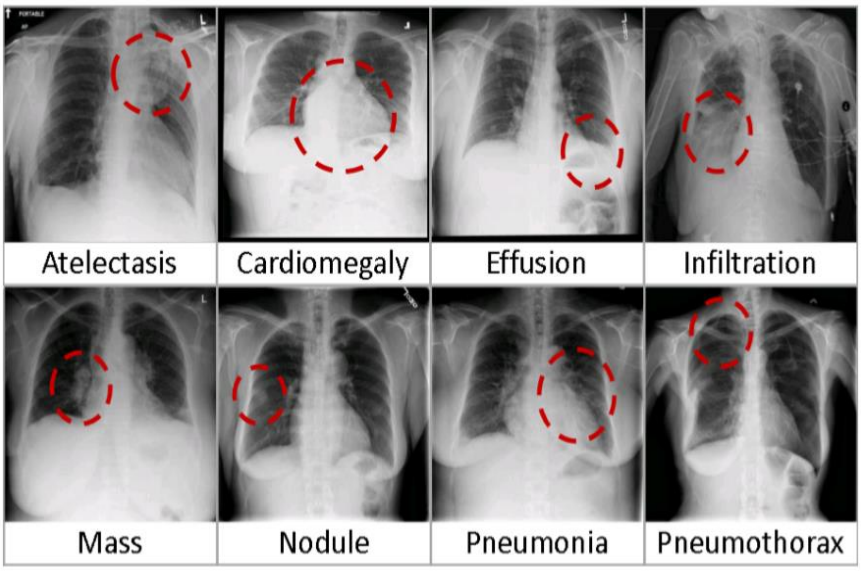


Fig. 1. Ocho enfermedades comunes del tórax observadas en radiografías que validan la dificultad del reto de automatizar el diagnóstico, tomado de [11].

Además, el etiquetado anatómico y patológico genérico y abierto, no se puede obtener por medio del *crowdsourcing* o subcontratación pública, dado que es implausible para anotadores no entrenados médicamente. También, la anotación completa al nivel de la región en cajas delimitantes para localizar la patología específica, lo cual sería necesario en conjuntos de datos para visión computacional [8] es actualmente inviable.

Consecuentemente, la mayoría de la investigación en la identificación de enfermedades del tórax es formulada como un problema de clasificación multi etiquetas débilmente supervisado, para superar las dificultades antes mencionadas y utilizando la información de la imagen completa en tándem con anotaciones débiles de la imagen y explorando diferentes avenidas de investigación.

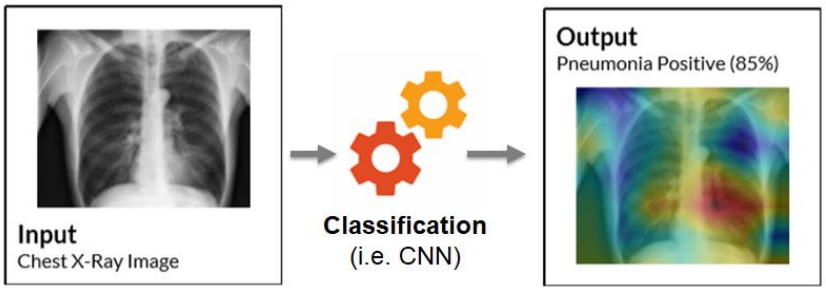


Fig. 2. El Modelo propuesto de CNN toma una radiografía como entrada y obtiene la probabilidad de la patología y la región en la imagen más probable donde se localiza esta patología como salida [9].

Con el fin de aliviar la escasez de conjuntos de datos disponibles públicamente, se han llevado a cabo varios esfuerzos recientes para crearlos [9], con números de pacientes que varían entre unos pocos cientos y hasta dos mil. Particularmente para las imágenes de placas del tórax, el conjunto de datos público más grande era Open-I [10], el cual contenía 3,955 reportes radiológicos de la Indiana Network for Patient Care y 7,470 placas del tórax asociadas de los sistemas PACS de hospitales.

Más recientemente, Wang et al. (2017) presentaron una nueva base de datos muy grande de radiografías, "ChestX-ray8" [11]. Esta basa de datos se compone de 108,948 imágenes de rayos-X de vista frontal de 32,717 pacientes únicos con etiquetas de 8 enfermedades obtenidas usando minería de textos sobre el reporte radiológico y en donde cada imagen podía tener múltiples enfermedades. Este conjunto de datos se ha utilizado desde entonces en varios estudios utilizando enfoques débilmente supervisados para el entrenamiento de modelos de aprendizaje profundo.

Debido a la cuidadosa curación y a su gran volumen, ChestX-ray8 representa un estudio clínico más realista y completo que Open-I, y consecuentemente más adecuado para el desarrollo y comparación de modelos. Además, se ha convertido en el estándar de facto sobre el cuál, otros métodos han entrenado y validado sus modelos. Varios trabajos han tratado de resolver el problema de la anotación de la imagen en las placas del tórax [12], incluyendo el estudio inicial presentado junto con el conjunto de datos.

La mayoría de estos estudios, incluyendo el presentado en [13] hacen uso de modelos ConvNet RNN con el objetivo de modelar las dependencias entre etiquetas. las cuales son muy fuertes cuando se evalúan enfermedades torácicas. Adicionalmente, la mayoría de estos trabajos detectan neumonía y localizan las áreas en la imagen más indicativas de esta patología, creando un mapa de calor en el proceso, como se ilustra en la figura 2 arriba.

Comparado con trabajos previos, nuestro enfoque difiere en tres maneras principales: 1. Hacemos uso de aumento de datos para incrementar el tamaño del conjunto de datos de ChestX-ray8. 2. Implementamos un esquema de filtrado de datos para eliminar imágenes que pudieran impactar negativamente el proceso de entrenamiento, y 3. Diseñamos una red convolucional mucho más pequeña y reducida que logra mejores resultados que los presentado por Wang et al [2017]. En la siguiente sección, explicaremos en detalle cada una de estas contribuciones y en las secciones 3 y 4, discutiremos a fondo los resultados obtenidos por nuestra metodología. Finalmente, discutiremos las limitaciones de nuestro enfoque y las líneas de investigación futura.

2. Formulación del problema

El enfoque presentado aquí es un problema de clasificación múltiple, ya que identificamos y clasificamos la imagen de la radiografía en 8 posibles clases o enfermedades. Tomamos los datos del conjunto de ChestX-ray8 y lo entrenamos con una red convolucional para hacer la clasificación, y obtener un modelo después de algunos ajustes finos como el *dropout* o eliminación de pesos y la regularización L2, entre otras estrategias. Sin embargo, uno de los mayores problemas del aprendizaje profundo, cuando se trabaja con conjuntos de datos como el utilizado en este estudio es el problema del sobre entrenamiento o *overfitting*.

Una solución ampliamente usada es el aumento de datos (A.D.), esto es, generar más datos de entrenamiento a partir de las muestras existentes. aplicando a estos ejemplos una variedad de transformaciones aleatorias que produzcan imágenes creíbles. El proceso de aumento de datos puede reducir significativamente la pérdida de validación, como se ilustra en la figura 3, en donde se muestra la gráfica con datos de entrenamiento y validación para dos redes convolucionales idénticas, una sin usar aumento de datos y otra usando esta estrategia. Utilizando aumento de datos claramente combate el problema de sobre entrenamiento que se puede observar en el lado izquierdo de la figura, en donde se observa que el valor de la función de pérdida con los datos de validación inicialmente decrementa y luego se incrementa de nuevo. mientras que continúa decrementando con los datos de entrenamiento. El lado derecho de la figura muestra como la pérdida de validación se puede regularizar usando el aumento de datos, entre otras estrategias.

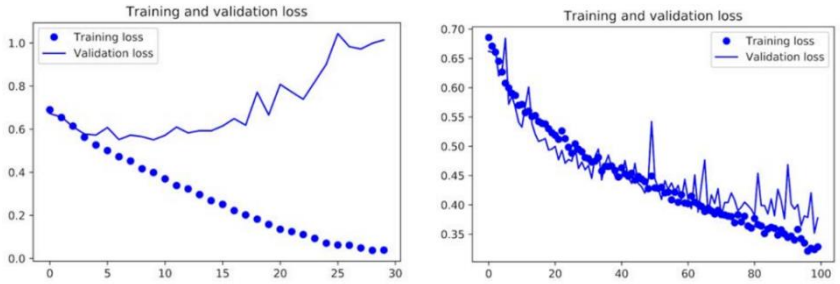


Fig. 3. Función de pérdida por entrenamiento y validación para dos redes convolucionales idénticas, a) sin usar aumento de datos y b) con aumento de datos. La función de pérdida claramente muestra como el A.D. previene el sobre entrenamiento.

Sin embargo, como veremos en la siguiente sección, si las estrategias de aumento de datos se implementan sin cuidado de la calidad de las imágenes, algunos ejemplos pobres en el entrenamiento nos pueden llevar a un modelo subóptimo de la red convolucional [14,15].

2.1. El conjunto de datos ChestX-ray8 y sus limitaciones

El enfoque de aumento de datos ha probado ser un método exitoso en diferentes aplicaciones de visión computacional, pero como veremos en la siguiente sección, hay algunas consideraciones importantes cuando se aplica este método a imágenes médicas. El

problema surge de la pobre calidad de algunas de las imágenes en el conjunto de datos, el cual puede afectar el entrenamiento y producir un modelo subóptimo. Estos modelos pueden introducir un gran número de falsos positivos o aún peor, clasificar equivocadamente un nuevo ejemplo de prueba, como lo documentan exhaustivamente los autores en [14]. Los autores de este estudio aplicaron una batería de pruebas en las cuales evaluaron el rendimiento de algunas redes convolucionales bien conocidas como VGG16 y GoogLeNet utilizando imágenes modificadas con diferentes tipos de distorsiones de la calidad de la imagen: compresión, bajo contraste, desenfoque y ruido gaussiano. Los autores encontraron que las redes son resilientes a distorsiones de compresión pero que se ven grandemente afectadas por las otras tres. Esto ha motivado una gran cantidad de investigación en esta área, con algunos trabajos concentrándose en los efectos que producen estas distorsiones en la calidad de la imagen para distintas tareas de visión computacional basadas en redes de aprendizaje profundo [15].

Esto es especialmente importante en el dominio de las imágenes médicas, particularmente con el conjunto de datos ChestX-ray8, el cual contiene 108,948 radiografías frontales de 32,717 pacientes únicos recolectados entre 1992 y 2015, con etiquetas de 8 enfermedades comunes obtenidas por mineo de texto de los reportes radiológicos utilizando técnicas de procesamiento de lenguaje natural. Hasta donde sabemos, no se ha reportado ninguna evaluación de la calidad de las imágenes en los trabajos previos con ese conjunto de datos, el cual incluye un gran número de imágenes con calidad pobre, como se observa en la figura 4, probablemente introducidos por la manera en la que los autores crearon el conjunto de datos. Algunas de las imágenes tienen un contraste muy bajo, mientras que otras están desenfocadas o carecen de definición, y otras sufren de saturación extrema.

Además de la estrategia de aumento de datos descrita anteriormente, la segunda contribución de este artículo es la implementación de una evaluación de la calidad de la imagen para determinar si el ejemplo de entrenamiento puede considerarse un buen candidato para la fase de aumento de datos y posterior procesamiento en la fase de entrenamiento. Algunas de las consideraciones para evaluar la calidad de la imagen, así como los algoritmos que se utilizaron para determinarla, serán presentados en la siguiente sección.



Fig. 4. Ejemplos de imágenes de baja calidad consideradas no útiles para clasificación, las cuales fueron automáticamente filtradas en nuestra fase de preprocesamiento.

2.2. Pre-proceso y filtrado de imágenes

Como se mencionó con anterioridad, el rendimiento de los modelos de redes convolucionales puede ser afectado severamente por la calidad de las imágenes de entrada; los autores de [14], realizaron pruebas individuales para ruido gaussiano, desenfoque, compresión y bajo contraste. Individualmente, aún para cantidades

moderadas de ruido o desenfoque, la exactitud de la red decrementa significativamente, y uno solo puede asumir que la combinación de varias de estas distorsiones en las imágenes puede producir resultados más pobres. Así, en este artículo implementamos un "filtro" o proceso de selección basado en varias métricas[1] tradicionales del procesamiento de imágenes para estimar si una imagen puede ser considerada para el procesamiento posterior o no, tales como, la taza de señal a ruido (SNR), el índice de desenfoque, y la varianza de la imagen para determinar el contraste. Típicamente, estos parámetros están correlacionados: una imagen con altos niveles de ruido presenta un bajo contraste y un alto desenfoque.

Las imágenes en los datos de ChestX-ray8 fueron primero pre-procesados usando técnicas de procesamiento de imágenes tradicionales: 1. Calculamos el índice de desenfoque utilizando un operador de estimación de desenfoque espectral. 2. Para estimar el contraste hacemos uso de la varianza como medida de la calidad de la imagen, usando la taza de contraste de Michelson[1], como se aprecia en la figura 5, dejando fuera las imágenes con una varianza alta. 3. Adicionalmente, realizamos una prueba SNR utilizando los métodos propuestos por Lagendijk y Biemond, para determinar si la imagen es adecuada para las fases de aumento de datos y entrenamiento.

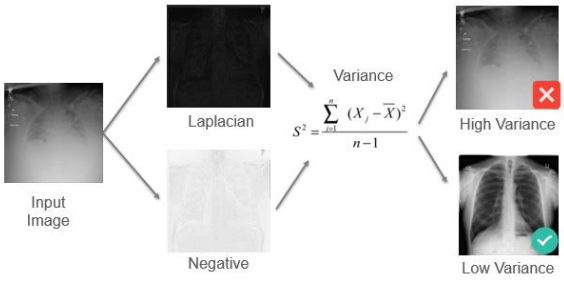


Fig. 5. Proceso para descartar imágenes con bajo contraste usando el cálculo de la varianza.

Una vez que estas pruebas se han llevado a cabo, realizamos los procedimientos de restauración de la imagen removiendo las distorsiones como el desenfoque (usando la convolución de Wiener), y bajo contraste (usando CLAHE, Contrast Limited Adaptive Histogram Equalization) y repitiendo la prueba SNR para determinar si la imagen corregida puede ser utilizada en el procesamiento posterior. Como veremos en la siguiente sección, este proceso produjo un conjunto de datos aumentado en donde las métricas de calidad de la imagen fueron tomadas en cuenta.

2.3. Resultados del proceso de aumento de datos en ChestX-ray8

En esta sección, discutiremos brevemente los resultados obtenidos con aumento de datos vía la evaluación de calidad de la imagen. Como se mencionó anteriormente, ChestX-ray8 contiene 108,948 radiografías frontales de 32,717 pacientes únicos. Debido a consideraciones de espacio, no incluimos la tabla completa aquí, y nos concentramos solamente en las patologías que hemos decidido enfrentar en este

[1] Lagendijk and Biemond, "Basic methods for image restoration and identification"

artículo. Para una revisión más completa del conjunto de datos, el lector es dirigido a [9, 11] pero presentamos aquí una breve introducción.

Primero, el tamaño del conjunto de datos se incrementó a 136,526 imágenes, aumentando el número de ejemplos de entrenamiento para algunas de las clases que previamente estaban pobremente representadas, (por ejemplo, cardiomegalia y neumonía, por citar dos casos). Además, la mayoría de los ejemplos corresponden a pacientes sanos (63,832) y el resto fueron divididos en 8 clases (como se mostró en la figura 1). El número de imágenes para cada una de estas clases se muestra en la primera fila de la Tabla 1 (nombrada "original") y en la segunda fila mostramos el número de imágenes que se obtuvieron después del proceso de filtrado descrito en la sección 2.2 y el aumento de datos (nombrado "modificado").

Tabla 1. Comparación de los datos de ChestX después del aumento de datos y filtrado.

	Cardiomegalia	Efusión	Infiltrado	Masa	Nódulo	Neumonía
Original	9,986	10,235	18,018	3,681	4,965	1,513
Nuestro	10,977	11,217	19,722	4,086	5,541	1,656

Nótese que después de aplicar el aumento de datos, obtuvimos 221,760 imágenes, de las cuales 70,908 fueron descartadas después del filtrado que evaluaba la calidad de la imagen. Por lo tanto, nuestro conjunto de datos modificado, creció a un tamaño de 150,852 imágenes, las cuales fueron utilizadas para el entrenamiento de la red convolucional que se presenta en la siguiente sección.

3. Solución propuesta

Comparado a otros conjuntos de datos para clasificación como COCO e ImageNet, la extensión espacial de las enfermedades en ChestX-ray8 ocupan una región más bien pequeña de la imagen, lo cual puede afectar el rendimiento de la red convolucional cuando se cuenta con poder de cómputo limitado, como es el caso aquí. Los autores de [11] extrajeron las imágenes de rayos-X del archivo DICOM y las ajustaron a un tamaño de 1024 x 1024 pixeles (de su tamaño original de 3,000 x 2,000) sin perder detalles significativos. En nuestro caso, las imágenes del conjunto de datos modificados fueron reducidos todavía más a un tamaño de 224 x 224 pixeles, debido a limitantes del hardware disponible, pero fuimos capaces de obtener resultados satisfactorios, como veremos a continuación.

En las siguientes secciones veremos que tuvimos que hacer compromisos respecto a la profundidad de la red convolucional, para acomodarnos al hardware utilizado. Sin embargo, en la sección 4 veremos que fuimos capaces de obtener y en algunos casos superar los resultados reportados por Wang et al. [11].

3.1. Arquitectura propuesta

La red convolucional implementada en nuestro trabajo se puede ver en la parte superior de la figura 6. Es una arquitectura más bien mínima, ya que está conformada

por 5 capas convolucionales y 3 capas completamente conectadas, con una cuenta de parámetros baja de solamente 254,138.

La arquitectura está basada en AlexNet que tiene un total de 62.4 millones de parámetros y fue simplificada gradualmente hasta que el modelo se ajustó al hardware disponible, un GPU Nvidia GeForce 1050 GTX. Compare con la arquitectura usada en [11] que realizó pruebas en varios modelos pero que obtuvo los mejores resultados con ResNet-50, mostrada en la parte inferior de la figura 6, y la cual tiene 25.6 millones de parámetros.

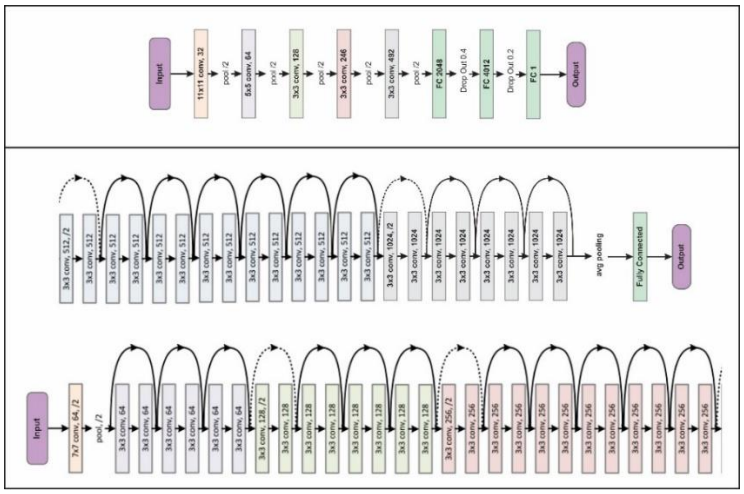


Fig. 6. Arquitecturas para a) la red convolucional propuesta, y b) el sistema en [11] basada en RestNet50.

3.2. Proceso de entrenamiento

Debido al tamaño de la imagen y al límite de memoria del GPU, fue necesario reducir el tamaño de la imagen y del batch para poder cargar la red completa en el GPU e incrementamos el número de iteraciones para acumular los gradientes. La combinación de ambos puede variar en diferentes redes convolucionales, pero en nuestro caso mantuvimos el tamaño del batch constante a 32.

La red fue entrenada de principio a fin usando el optimizador de Adam con los parámetros estándar ($\beta1 = 0.9$ y $\beta2 = 0.999$). Usamos un valor inicial de 0.001 para la taza de aprendizaje que disminuíamos por un factor de 10 cada vez que la pérdida de validación se estancaba después de una época y tomábamos el modelo con la pérdida de validación más baja.

En total, 150,852 imágenes de rayos-X de vista frontal fueron incluidas en la base de datos, de las cuales 87,020 imágenes contenían una o más patologías. Las restantes 63,832 imágenes eran casos normales. Para la tarea de clasificación de la patología, dividimos de manera aleatoria el conjunto de datos completo en tres grupos: entrenamiento (70%), validación (10%), y pruebas (20%). En entrenamiento y validación el modelo fue ajustado finamente con *Stochastic Gradient Descent*. En nuestros experimentos, solo reportamos los resultados obtenidos en la clasificación de las 8 enfermedades sobre los datos de prueba.

4. Resultados y discusión

A continuación, presentamos los resultados obtenidos de nuestros experimentos y los comparamos con enfoques similares, específicamente, los modelos presentados por Wang et al. [11].

4.1. Resultados de entrenamiento y validación

Entrenamos a nuestra CNN para clasificar las 8 enfermedades del tórax presentadas en el conjunto de datos de ChestX-ray8. En la mayoría de los casos, los modelos convergieron rápidamente, debido a la baja complejidad del modelo de nuestra arquitectura CNN, requiriendo menos de 3,000 iteraciones o épocas y obteniendo mejores resultados que la investigación base reportada en [11]. Las gráficas de la figura 7 muestran la exactitud obtenida con los datos de validación en 4 de las 8 enfermedades, la línea roja representa la prueba de nuestro modelo usando los datos originales, no aumentados, mientras que la línea azul muestra los resultados en el conjunto de datos modificado, con aumento de datos.

La exactitud obtenida con los datos de validación para las 8 clases se muestra en la Tabla 2, para ambos casos, los datos originales no aumentados, y con los datos aumentados con imágenes de calidad en el conjunto de datos ChestX-ray8. Como se mencionó anteriormente, podemos observar claramente una mejora significativa en todas las 8 clases, lo cual es especialmente sorprendente dado que nuestra red convolucional es más simple.

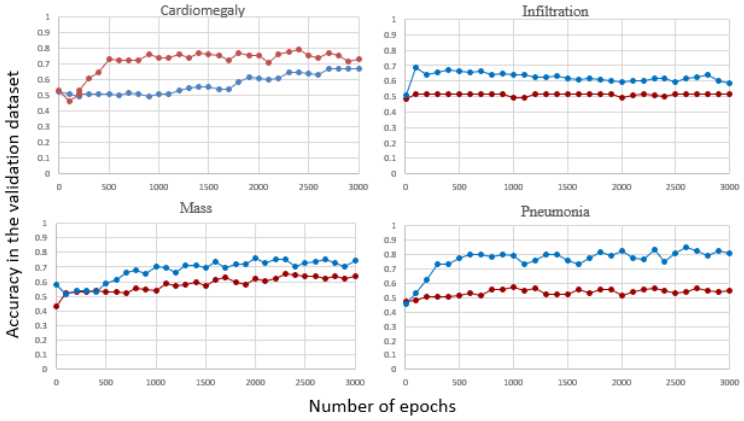


Fig. 7. Resultados de las pruebas de exactitud usando los datos de validación de ChestX-Ray, sin usar aumento de datos (línea en rojo) y con aumento de datos y filtrado (en azul), usando la red propuesta.

Tabla 2. Resultados para exactitude en el major de los casos para clasificación múltiple en diferentes modelos de DCNN model settings.

	Cardiomelagia	Efusión	Infiltrate	Mass	Nodule	Pneumonia
No DA	0.7065	0.5966	0.5598	0.6875	0.6449	0.6051
W/ DA	0.8071	0.6471	0.6957	0.7898	0.6989	0.8636

4.2. Comparación con el modelo original de ChestX-ray8

Obtuvimos muy buenos resultados usando una red convolucional relativamente simple inspirada en AlexNet, como se muestra en la Tabla 3. Los autores de [11] probaron 4 redes diferentes con ChestX-ray8: AlexNet, GoogLeNet, VGG-Net-16 y RestNet50. Los mejores resultados se obtuvieron con ResNet50, excepto para "Masa" donde AlexNet obtuvo los mejores resultados. Cuando comparamos nuestro modelo con los resultados reportados para AlexNet y ResNet50, vemos una mejora significativa para todos los casos del primero, excepto "Efusión" y mejores resultados en "Cardiomegalia", "Infiltración", "Masa" (una clase particularmente difícil) y "Neumonía" con el segundo. Creemos que el aumento de datos con calidad jugó un papel significativo para alcanzar estos resultados y para un futuro, deseamos investigar si las clases que no obtuvieron los mejores resultados se pueden mejorar con otros modelos de redes o modificando el proceso de filtrado.

Tabla 3. Comparación en términos de exactitud de nuestros modelos con otros modelos del estado del arte.

Red-Clase	Cardiomegalia	Efusión	Infiltrado	Masa	Nódulo	Neumonía
AlexNet	0.6458	0.6925	0.6041	0.5644	0.6487	0.5493
ResNet	0.8071	0.7362	0.6090	0.5609	0.7164	0.6333
Nuestra	**0.8141**	0.6471	**0.6957**	**0.7898**	0.6989	**0.8636**

4.3. Discusión sobre limitantes y extensiones del enfoque propuesto

Como hemos discutido a fondo en este artículo, el enfoque presentado de aumento de datos basado en la calidad de la imagen ha producido resultados muy interesantes con los datos de ChestX-ray8. Hasta donde sabemos, este enfoque no se ha reportado en la literatura y consideramos que esta es la principal contribución de este trabajo. Como se mencionó en la sección 3.1, hemos hecho uso de una arquitectura relativamente simple de red convolucional, debido principalmente a la disponibilidad de hardware con la que contábamos. Esta limitante, también nos llevó a utilizar imágenes más pequeñas que las reportadas en la literatura (224 x 224, en contraste a imágenes de 1024 x 1024 como se menciona en [11] y otros trabajos).

Sin embargo, los resultados reportados en la subsección anterior son muy prometedores y creemos que nuestro enfoque de aumento de datos se puede poner a prueba usando otras redes convolucionales más complejas, como ResNet-50, una línea que podemos explorar en el futuro utilizando herramientas como Google Colab o los Kernels de Kaggle.

5. Conclusiones

El diagnóstico médico se ha convertido en un dominio más interesante y viable para el aprendizaje automático. En particular, las placas del tórax son el tipo de estudio radiológico más común en el mundo y un ejemplo particularmente retador de clasificación múltiple para el diagnóstico médico. Acumulando cerca del 45% de todos

los estudios radiológicos, la placa del tórax ha alcanzado una ubicuidad global como una herramienta de cribado de bajo costo para una gran cantidad de patologías incluyendo el cáncer de pulmón, la tuberculosis y la neumonía.

Sin embargo, la escasez general de conjuntos de datos médicos, disponibles públicamente, han inhibido el desarrollo de soluciones basadas en aprendizaje profundo en este campo. Además, el preprocesamiento sobre conjuntos de datos como ImageNet o COCO pueden introducir sesgos no intencionados los cuales pueden ser inaceptables en ambientes clínicos, haciendo la transferencia de aprendizaje más difícil de alcanzar en este contexto.

Igualmente, la mayoría de los ambientes clínicos demandan modelos que puedan predecir exactamente múltiples diagnósticos. Esto transforma muchos problemas médicos en problemas de clasificación múltiple, en donde una sola imagen pueda tener un gran número de resultados, los cuales pueden ser ambiguos o pobremente definidos y es probable que la imagen sea etiquetada de manera inconsistente. Estos problemas son expuestos debido a la introducción del conjunto de datos ChestX-ray8 en 2017, ya que los conjuntos de datos disponibles previamente, tales como Open-I eran considerados demasiado pequeños.

Desde entonces, varios trabajos han mejorado los resultados de manera progresiva al diagnosticar enfermedades torácicas basados en este conjunto de datos. Sin embargo, estos trabajos no toman en cuenta la calidad de la imagen de los ejemplos de entrenamiento, lo cual ha producido resultados pobres en trabajos previos, tal como lo demostramos experimentalmente aquí utilizando el enfoque de aumento de datos con imágenes de calidad. Los resultados de nuestros experimentos fueron capaces de igualar los alcanzados en [11] en 4 enfermedades y superarlos en las otras 4, utilizando una red más pequeña. Estamos planeando un estudio más completo donde se evalúen otras redes más complejas tales como ResNet-50 para validar todavía mejor nuestros resultados.

Referencias

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, pp. 1097–1105 (2012)
2. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Presa Reyes, M., Shyu, M.-L., Chen, S.-C., Iyengar, S.S.: A Survey on Deep Learning: Algorithms, Techniques, and Applications. ACM Comput. Surv 51(5), Article 92, 36 pages (2018)
3. Litjens, G., Kooi, T., Ehteshami Bejnordi, B., Adiyoso Setio, A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical Image Analysis, vol. 42, pp. 60–88 (2017)
4. Shin, H., Roth, H., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.: Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learnings. IEEE Trans. Medical Imaging 35(5), 1285–1298 (2016)
5. Setio, A., Ciompi, F., Litjens, G., Gerke, P., Jacobs, C., van Riel, S., Wille, M., Naqibullah, M., Snchez, C., van Ginneken. B.: Pulmonary nodule detection in ct images: False positive reduction using multi-view convolutional networks. IEEE Trans. Medical Imaging 35(5), 1160–1169 (2016)
6. CDC, 2017, https://www.cdc.gov/features/ pneumonia/index.html. Último acceso: 2019/05/11

7. WHO. Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children (2001)
8. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D. A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations (2016)
9. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Ball, R.L., Langlotz, C., Shpanskaya, K., Lungren, M. P., Ng, A.Y.: CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. https://arxiv.org/abs/1711.05225
10. Demner-Fushman, D., Kohli, M.D., Rosenman, M., Shooshan, B., Sonya, E., Rodriguez, L., Antani, S., Thoma, G.R., McDonald, C.J.: Preparing a collection of radiology examinations for distribution and retrieval. Journal of the American Medical Informatics Association 23 (2), 304–310 (2015)
11. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. arXiv preprint arXiv:1705.02315 (2017)
12. Chen, S.-F., Chen, Y.-C., Yeh, C.-K., Wang, F., Yu-C.: Order-free rnn with visual attention for multi-label classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
13. Yao, L., Poblenz, E., Dagunts, D., Covington, B., Bernard, D., Lyman, K.: Learning to diagnose from scratch by exploiting dependencies among labels. arXiv preprint arXiv:1710.10501 (2017)
14. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: Quality of Multimedia Experience (QoMEX), Eighth International Conference on, IEEE, arXiv:1604.04004v2, pp. 1–6 (2016)
15. Diamond, S., Sitzmann, V., Boyd, S., Wetzstein, G., Heide, F.: Dirty Pixels: Optimizing Image Classification Architectures for Raw Sensor Data. arXiv:1701.06487

ISSN 1870-4069

Herramienta en realidad virtual para el aprendizaje del lenguaje de señas mexicano

Arturo Borja Araiza, David Araujo Díaz, Laura Méndez Segundo

Instituto Politécnico Nacional, Escuela Superior de Cómputo,
Ciudad de México, México
aborja.isc@gmail.com, daraujo@ipn.mx
lmendezs.escom@gmail.com

Resumen. Este trabajo brinda una herramienta visual que permite el aprendizaje de múltiples palabras que son parte del lenguaje de señas mexicano (LSM) a través de un modelo antropomórfico completamente manipulable y programable, haciendo uso de los recursos brindados por la tecnología de realidad virtual utilizando el lenguaje VRML (Virtual Reality Modeling Language) y utilizando una base de datos creada a partir de los movimientos y posiciones bien establecidos por el LSM, representando distintos gestos y expresiones manuales, desde distintos ángulos, profundidades y a diversas velocidades, según lo desee y requiera el usuario, para optimizar el aprendizaje de dicho lenguaje utilizando una interfaz gráfica amigable.

Palabras clave: realidad virtual, lenguaje de señas, aprendizaje, aplicación, interacción humano-computadora.

Virtual Reality Tool for Learning Mexican Sign Language

Abstract. This work provides a visual tool for learning multiple words that are part of Mexican Sign Language (LSM) through a fully manipulable and programmable anthropomorphic model, using resources provided by the technology of virtual reality using the language VRML (Virtual Reality Modeling Language) and using a database created from the movements and positions well established by LSM, representing different gestures and manual expressions, from different angles, depths at various speeds, as desired and required by the user , to optimize the learning of said language using a friendly graphical interface.

Keywords: virtual reality, language, learning, application, human-computer Interaction.

1. Introducción

El lenguaje es el medio de acceso humano al mundo. Los lenguajes tienen la virtud de abrirnos formas alternativas de pensar y entender el lugar que habitamos, relacionarnos con él, ampliarlo y modificarlo. Como posibilidad de comunicación, los lenguajes nos abren oportunidades para relacionarnos con otras personas, acercarnos a ellas y desarrollar una comprensión más amplia de ellas, de lo social y lo humano [1].

Aprender un lenguaje es acceder de un modo distinto al mundo que todas y todos conformamos, forma parte de la diversidad, los ideales, los conceptos mediante los cuales nos reconocemos, nos relacionamos y en último de los casos nos reinventamos.

El lenguaje tiene la virtud de permitirnos transformar las concepciones que tenemos de nosotros mismos y de las y los demás. Las personas nos conocemos fundamentalmente por medio de procesos comunicativos; por ello, entre más amplios sean éstos más posibilidades tenemos de diversificar nuestro mundo.

Sin embargo, hay muchas personas con problemas para comunicarse, siendo la sordera una de las más comunes. Las personas con este problema hacen uso del Lenguaje de Señas (LS) para poder comunicarse e incluso se han desarrollado sistemas de traducción (Voz/Texto-a-LS) para asistir en esta tarea. Sin embargo, debido a que los LS son dependientes de países y culturas, hay diferencias entre gramáticas, vocabularios y señas, incluso si estos provienen de lugares con lenguajes hablados similares.

1.1. Estado del arte

En 2003, la lengua de señas mexicana se declaró oficialmente una "lengua nacional", junto con las lenguas indígenas y el español, que se utiliza en el sistema nacional de educación para sordos. Antes de eso la principal filosofía educativa en el país se centró en el oralismo (voz y lectura de labios) y con pocas escuelas donde las clases se impartieran en el lenguaje de señas mexicano (LSM) [2].

En el 2018 se estimó que alrededor de 694,451 habitantes del país padecen deficiencia auditiva, de acuerdo con cifras del Censo de Población y Vivienda más reciente, realizado por el Instituto Nacional de Estadística y Geografía (INEGI) [3].

Debido a la falta de interés por parte de los especialistas en sistemas en abordar el LSM no se encontraron programas o sistemas similares que cubran la necesidad del aprendizaje del LSM, pues existen algunos sistemas similares que buscan el aprendizaje por medio de un videojuego[6] o reconocimiento visual de gestos humanos[7], pero están centrados en su mayoría para el lenguaje de señas americano, por lo que se propone este trabajo para cubrir las necesidades de este sector, y se propone el uso de la realidad virtual para dar soporte a las personas interesadas en aprender el LSM.

Recientemente se ha incrementado el número de personas que, sin padecer deficiencias auditivas, busca aprender el LSM para poder comunicarse con este grupo.

1.2. Realidad virtual

La Realidad Virtual (VR) es una representación de la realidad a través de medios electrónicos, que nos da la sensación de experimentar una situación real, en la que podemos interactuar con lo que nos rodea [4].

En este caso se trabajó con la VR no inmersiva, también conocida como realidad virtual de escritorio, que se limita al uso de una computadora y no requiere el uso de otros dispositivos como cascos o visores de VR; Debido a su bajo costo este tipo de realidad virtual ha sido ampliamente difundida y aceptada.

1.3. Metodología

Modelo de Desarrollo Rápido de Aplicaciones. El desarrollo rápido de aplicaciones o RAD por sus siglas en ingles Rapid Application Development., es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE. Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso permite al equipo de desarrollo crear un "sistema completamente funcional" dentro de periodos cortos de tiempo. Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque comprende las siguientes fases:

Modelado de gestión. El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso?
Modelado de datos. El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.

Modelado de proceso. Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
Generación de aplicaciones. El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.

Pruebas de entrega. Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

2. Diseño

El diseño de esta herramienta consiste en un modelo de VR para imitar los movimientos hechos con el cuerpo hechos por las personas con discapacidad auditiva para comunicarse.

Entre las acciones necesarias para llevar a cabo este proyecto fue necesario conocer y realizar los movimientos en VR de algunas palabras del LSM para probar la funcionalidad del modelo, así como programar los interpoladores de tiempo y posición, para posteriormente relacionarlos con los sensores de toque, para las funciones de reproducción, avance, retroceso, retardo y pausa de la animación del modelo (Fig. 1).

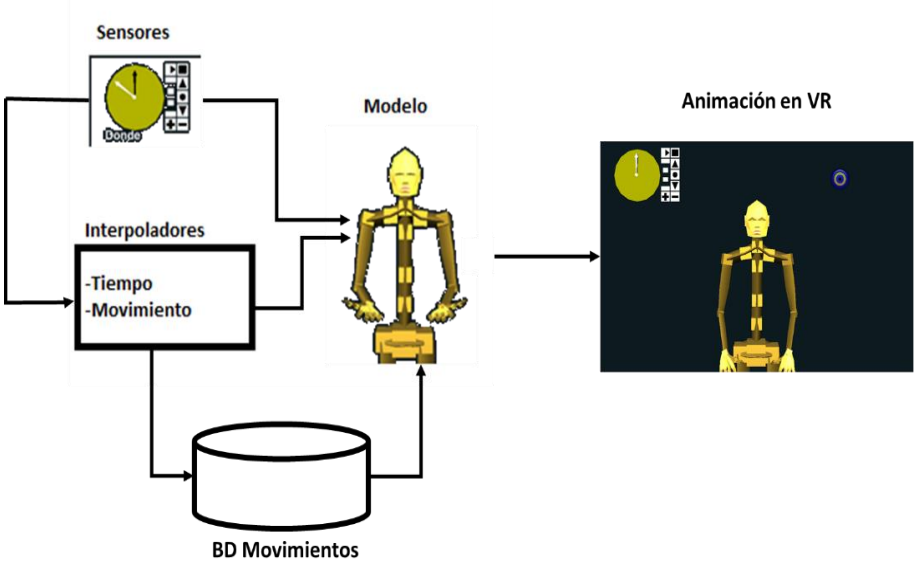


Fig.1. Diseño de la herramienta.

3. Desarrollo

La presente herramienta en Realidad Virtual (VR) para el aprendizaje del Lenguaje de Señas Mexicano (LSM) tiene como objetivo permitir a cualquier persona interesada, aprender diversas palabras del LSM, a través de un medio visual y permitiendo la manipulación de diferentes aspectos de la animación, como son, la velocidad de los gestos, su sentido, las palabras que se representarán, así como el poder pausar y reanudar la animación, entre otras.

Por ejemplo, para aprender correctamente los movimientos necesarios para representar la palabra "Rojo" podemos observar que se debe llevar el índice a la barbilla y realizar un movimiento vertical sobre esta (Fig.2), y para representar un vaso se hace el ademán de tomar un vaso o copa con una mano sobre la otra (Fig. 3).

La animación en paralelo, nos permite ver conjuntamente la palabra que se representa y la animación del modelo humano como se observa en la Fig. 4, asi como manipular la animación a través de los controles superiores (Fig. 5) con los cuales podremos manipular la velocidad a la que el modelo hace los gestos, que gesto es el que queremos ver, pausar la animación e invertir el sentido de las palabras; esto debido a que muchas veces difiere la estructura de una oración en contraste con su representación en LSM.



Fig. 2. Modelo diciendo ROJO. Elaboración Propia.



Fig. 3. Representación de un vaso en el LSM. Elaboración Propia.

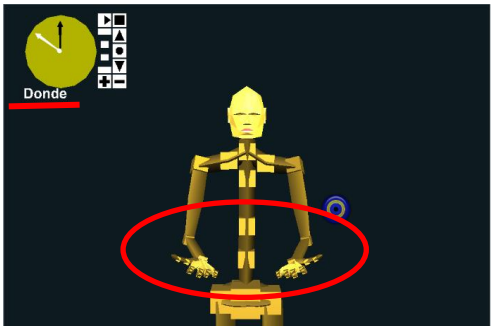


Fig. 4. Funcionamiento paralelo. Elaboración Propia.

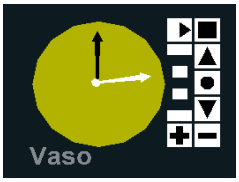


Fig. 5. Controles y timer de la aplicación. Elaboración Propia.

El desarrollo de esta herramienta se realizó utilizando el lenguaje VRML (Virtual Reality Modeling Language) [4], debido a la sencillez y la facilidad que este lenguaje ofrece para representar mundos virtuales con recursos computacionales reducidos.

También se utilizó una base de datos propia, creada a partir de los movimientos y posiciones bien establecidos por el LSM, se basó principalmente en el diccionario del Consejo Nacional para Prevenir la Discriminación publicado en 2011 [2] y que se puede consultar en la red.

Para quien tenga el deseo de aprender, practicar y comprender el LSM, esta herramienta resulta una alternativa didáctica, interactiva, dinámica y amigable al usuario, ya que permite aprender y trabajar a un ritmo propio [5].

4. Trabajo a futuro

Para ampliar el presente trabajo se propone implementar una base de datos más extensa que permita crear un compendio completo de las palabras y expresiones que posee el LSM, y posteriormente aplicarlo a otros LS de alguna región o cultura diferente, así como evaluar si utilizando este método se presenta alguna mejora significativa en el aprendizaje en comparación con el método estándar de enseñanza de este lenguaje.

Cuando el sistema contenga una base de datos más completa, se evaluará su usabilidad que es un criterio importante en el rendimiento y calidad del sistema de RV, y aunque su evaluación puede tener un componente subjetivo, el estudio de normas, guías, estándares y diferentes propuestas permitirá unificar los aspectos relevantes en este proceso y servir como base para estructurar una guía que permita mejorar el sistema de LSM.

Existen múltiples campos del cómputo actual que podrían aprovechar herramientas como esta para desarrollar programas o traductores para gente con discapacidad auditiva.

5. Conclusiones

El prototipo desarrollado es un prototipo que conjuga la tecnología y herramientas existentes en la actualidad, para ayudar al aprendizaje de un lenguaje nuevo, mediante el uso de modelos virtuales. Este tipo de herramientas podrían servir para cerrar la brecha que existe en cuanto a la comunicación se refiere, permitiendo que cualquiera pueda acceder o utilizar las nuevas tecnologías y herramientas, así como ser aprovechado para implementarse con sistemas de inteligencia artificial [8] o *deep learning* [9] que puedan ser operados por medio de gestos del LSM.

Se considera que el uso de la realidad virtual permite mejorar el aprendizaje de un nuevo lenguaje, pues a diferencia de un video o un conjunto de imágenes, la interacción con el ambiente virtual es mayor, debido a que en el Sistema de RV propuesto se tienen controles de velocidad, lo que hace posible ver los gestos la cantidad de veces necesarias y controlando el tiempo de que estos se reproducen.

Referencias

1. García Mejía, J.F., Tinoco Monroy, L.A. (et. al.): Modelado e implementación de un sistema de enseñanza de Lenguaje de Señas Mexicano. CIENCIA ergo-sum 14(2), 185–190 (2007)
2. Serafín de Fleischmann, M.E., González Pérez, R.: Manos con voz: Diccionario de lenguaje de señas mexicano. Consejo Nacional para Prevenir la Discriminación, México (2011)
3. Portal INEGI, https://www.inegi.org.mx/temas/discapacidad/default.html. Último acceso: 02/04/2019
4. Araujo Diaz, D.: El Placer de Programar con VRML 2.0, Instituto Politécnico Nacional, México (2006)
5. Báez, M.: Las TIC: oportunidades para la alfabetización de jóvenes y adultos sordos. Centro de Estudios Interdisciplinarios de la Universidad Nacional de Rosario (CEI/UNR) Rosario, Argentina (2012)
6. Gameiro, J., Cardoso, T., Rybarczyk, Y. (eds): Innovative and Creative Developments in Multimodal Interaction Systems. Kinect-Sign: Teaching Sign Language to "Listeners" through a Game (2014)
7. Cooper, H., Holt, B., Bowden, R.: Sign Language Recognition. Visual Analysis of Humans, pp. 539–562. Springer (2011)
8. Stoll, S., Camgöz, N.C., Hadfield, S., Bowden, R.: Sign Language Production using Neural Machine Translation and Generative Adversarial Networks. In: 29th British Machine Vision, Northumbria University, Newcastle Upon Tyne, UK (2018)
9. Chung, S., Lim, J.Y., Ju Noh, K., Kim, G., Jeong, H.: Sensor Data Acquisition and Multimodal Sensor Fusion for Human Activity Recognition Using Deep Learning (2019)

ISSN 1870-4069

Aspectos innovadores de realidad virtual y sensores cinéticos para una mejora significativa utilizando el algoritmo de Fireworks para un juego Wii de un deporte colaborativo para una sociedad en una Smart City

Alberto Ochoa[1], César Velázquez[2], Juan Hernández[1], Silvia Husted[1], Diego Oliva[3], Anahí Solis[1], Héctor Herrera[1], José Mejia[1], Saúl González[1], Edna Cruz-Flores[4]

[1] Universidad Autónoma de Ciudad Juárez, Ciudad Juárez, Chihuahua, México
[2] Universidad Autónoma de Aguascalientes, Aguascalientes, Ags., México
[3] CUCEI-UDG, Guadalajara, Jalisco, México
[4] Universidad Autónoma del Estado de Morelos, FCAeI, Cuernavaca, Morelos, México
alberto.ochoa@uacj.mx

Resumen. Cada cierto tiempo se publica un nuevo informe sobre la obesidad infantil. Los malos hábitos alimentarios y la vida cada vez más sedentaria de los niños en una sociedad fronteriza, ha provocado un aumento alarmante en los casos de niños con sobrepeso u obesos. Anteriormente, parecía un problema de países con hábitos alimentarios poco saludables, como los Estados Unidos o México en América Latina, donde la comida chatarra es parte de la dieta en la infancia. Sin embargo, la obesidad es un problema que ya tenemos a la vuelta de la esquina y que no es tan difícil combatir en los niños. En la presente investigación, el desarrollo de una aplicación que reduce el problema de la falta de movimiento en la infancia de una ciudad inteligente se considera un problema futuro, que es la principal contribución, junto con el logro de una forma innovadora de buscar un deporte olímpico sin la complejidad de trasladarse físicamente a un espacio con altos costos de mantenimiento y considerando las condiciones climáticas adversas.

Palabras clave: juegos basados en ejercicio virtual, cinética, sistema Wii, aplicaciones inteligentes en una smart city.

Innovative Aspects of Virtual Reality and Kinetic Sensors for Significant Improvement using Fireworks Algorithm to a Wii Game of a Collaborative Sport to a Society in a Smart City

Abstract. A new report on childhood obesity is published every so often. The bad habits of food and the increasingly sedentary life of children in a border society, has caused an alarming increase in the cases of children who are overweight or obese. Formerly it seemed a problem of countries with unhealthy eating habits, such as the United States or Mexico in Latin-America, where junk food is part of the diet in childhood. However, obesity is a problem that we already have around the corner and that is not so difficult to fight in children. In the present research the development of an application

that reduces the problem of the lack of movement in the childhood of a smart city is considered a future problem which It is the main contribution, coupled with achieving an innovative way of looking for an Olympic sport without the complexity of physically moving to a space with high maintenance costs and considering the adverse weather conditions.

Keywords: exergaming, kinetic, Wii system, smart applications in a smart city.

1. Introducción

El aumento de la obesidad infantil, un problema de gran importancia en una ciudad inteligente, determina los desafíos que se deben construir con respecto a las aplicaciones que involucran la Inteligencia Artificial. Los juegos de computadora para combatir la obesidad infantil son muy importantes para reducir un problema futuro en nuestra sociedad. Exergaming, juegos de computadora para ejercitar, los niños juegan cada vez menos en la calle y pasan más tiempo con los videojuegos y los juegos de computadora, por lo que llevan una vida más sedentaria. Esto, junto con los malos hábitos alimenticios, aumenta los casos de niños obesos cada año. ¿Qué pueden hacer los padres para evitar el sobrepeso en la infancia? Una apuesta que nos viene de la Universidad de Australia Occidental, la Universidad John Mores de Liverpool y la Universidad de Swansea en el Reino Unido es el exergaming, un anglicismo que viene de unir la palabra "exerdizze" en turco (ejercicio en inglés) con juegos (juego). Estos son juegos que ofrecen consolas como Xbox, Kinect o Wii en las que interactúas a través de la actividad física en las pruebas en las que tienes que correr, andar en bicicleta, jugar bolos o saltar vallas.

Los investigadores evaluaron a los niños que realizaron exergaming de alta y baja intensidad y midieron su gasto de energía. La conclusión a la que se llegó fue que el exergaming generó un gasto de energía comparable al ejercicio de intensidad moderada o baja, dependiendo de la dificultad del juego. Además, el juego fue satisfactorio para los niños, quienes disfrutaron de las actividades que realizaron.

Es una ventaja que los padres pueden aprovechar para evitar que los niños pasen tantas horas sentados frente a la consola, ya que se ha demostrado que puede obtener beneficios de salud a largo plazo. En cualquier caso, siempre debe ser uno de los medios que podemos utilizar para alentar a los niños a hacer alguna actividad física, pero no el único. Salir a la calle para jugar, correr, saltar, siempre debe estar en la agenda de los niños, como se muestra en la figura 1.

A. Emociones en los niños desde que surgió la psicología infantil basada en la ciencia en la segunda mitad del siglo XIX, se prometió proporcionar una base racional para la educación para el desarrollo general del niño. El desarrollo de toda esta área ofrece nuevas propuestas interesantes, como el uso del estudio de los niños, no solo para la psicología infantil, sino que también propone la creación de una ciencia enfocada al niño que va más allá de la psicología infantil [2]. Esto se lo agradece a William Preyer, quien es considerado el padre de la psicología infantil y el ojo no es el primero, pero si lo hizo con una precisión rigurosa, lo mismo se hace en sus observaciones de forma muy sistemática [1].



Fig. 1. Aplicación inteligente utilizando Kinect.

B. Trastornos emocionales en niños. Los trastornos mentales en los niños afectan a muchos niños y sus familias. Los niños de todas las edades, orígenes étnicos o raciales, y las regiones de los Estados Unidos tienen trastornos mentales. Según el informe del Consejo Nacional de Investigación y el Instituto de Medicina de la división (Prevención de trastornos mentales, emocionales y del comportamiento en jóvenes: progreso y posibilidades, 2009) que recopilaron los resultados de estudios anteriores, se estima que de 13 a 20% de los niños que viven en los Estados Unidos (hasta 1 de cada 5) tiene un trastorno mental en un año determinado, y se gastan aproximadamente 247,000 millones de dólares al año en trastornos mentales infantiles [3]. En la literatura encontramos varias definiciones para referirse a problemas emocionales, mentales o de comportamiento. En estos días encontramos que se les conoce como "trastornos emocionales". El Acta de Educación para Personas con Discapacidades de la Ley de Educación ("IDEA" para abreviar) define el desempeño de los trastornos emocionales como "esa condición que exhibe una característica en mucho tiempo en un grado de nitidez que afecta negativamente al niño". Inmediatamente enumeramos las clasificaciones de los tipos de comportamientos como trastorno emocional [4]. a) La falta de habilidad para aprender que es inexplicable por razones intelectuales, sensoriales o de salud. b) La falta de capacidad para mantener relaciones personales en buenos términos con compañeros o con sus maestros. c) Comportamiento o sentimientos inconsistentes en circunstancias normales. d) Suele ser tristeza o depresión. e) Desarrollar síntomas físicos o miedos asociados con problemas personales o escolares. Hiperactividad: en este tipo de comportamiento se manifiesta al niño desatento, fácilmente distraído y con un grado de impulsividad. Asaltos: cuando el resultado del comportamiento termina en heridas, ya sea ellos mismos o a personas que los rodean. Retirada: la vida social muestra signos de retraso o la incapacidad de los individuos para relacionarse con su entorno. Incluye temores excesivos o ansiedad. Inmadurez: cuando experimenta episodios de llanto donde no está justificado; es incapaz de adaptarse a los cambios. Dificultades de aprendizaje: el aprendizaje no se desarrolla al mismo ritmo que el promedio en su entorno. Se mantiene por debajo del nivel por debajo de sus compañeros. Hay aún más niños pequeños con trastornos emocionales graves en los que el pensamiento se distorsiona, ansiedad severa, no son actos motores comunes y comportamientos demasiado irritables. A veces se diagnostica psicosis grave o esquizofrenia [4]

C. Terapia de juego: la terapia de juego se define como un modelo terapéutico de reconocimiento formal y también ha demostrado efectividad en niños con problemas de estrés emocional que contribuyen y se manifiestan en el niño durante el desarrollo

normal. La terapia de juego se basa en el juego del niño como un medio natural de autoexpresión, experimentación y comunicación. Mientras el niño juega aprende sobre el mundo y sus relaciones, la realidad probada, explora las emociones y los roles. También le da al niño la posibilidad de externalizar su historia personal, liberando así sentimientos negativos y frustraciones, mitigando los efectos de experiencias dolorosas y que se estremezca, aliviando los sentimientos de ansiedad y estrés [5]. El terapeuta de juego es un especialista especializado y entrenado en el juego y adecuado para las diversas etapas de desarrollo profesional infantil, técnicas y métodos terapéuticos. Esto debe ser comprendido y entendido, así como participar en el juego del niño para poder crear una relación de confianza y confianza para la expresión y el manejo de los conflictos internos del niño. Además, la descarga y la comprensión de sus emociones están más atrapadas para que pueda reconocerlas y comprenderlas. Explorar temas que afectan sus vidas [6]. El juego serio propuesto no reemplaza la terapia lúdica de su terapeuta, sino que busca otorgar una herramienta de apoyo auxiliar y también es eficaz como apoyo para aquellos que están en contacto con niños en algún momento y muestran signos de un estado de ánimo bajo. Inclusive, los profesionales de la salud también podrían utilizarlo como parte de sus herramientas terapéuticas para controlar las emociones de sus pacientes.

D. Juegos serios: una antigua definición de los 70's que ha permanecido hasta ahora es que los juegos serios son aquellos que tienen un propósito expreso educativo o terapéutico y no solo divertidos. Dentro de esta definición es donde está el verdadero desafío para los desarrolladores porque mantener este equilibrio entre la diversión y el propósito del aprendizaje cuidadosamente planificado. Una vez que la construcción del juego se enfoca en el aprendizaje o el propósito terapéutico, sucede que la parte divertida de un zumo descuidado tan importante y necesario socava la calidad y el impacto del juego que debería tener en los usuarios. Las aplicaciones son variadas y van desde las más comunes a la educación, gestión, políticas, promoción, planificación, entre otras [7]. ¿Por qué juegos como soporte? La respuesta es simple e intuitiva, ya que son parte de la capacitación de las personas. Abarcan desde juegos individuales hasta juegos en computadoras, ya sea dos o tres o incluso miembros de muchos equipos. También un aspecto importante de la socialización y construcción de la identidad de los individuos. Cada cultura tiene juegos que se practican de generación en generación. Aunque siente que sus primeros juegos no fueron diseñados con propósitos de aprendizaje o terapéuticos, se puede ver que, además de la diversión y el entretenimiento para el individuo, ayuda en su desarrollo físico y mental porque no logran estimularlo. Y para reforzar lo anterior, el tiempo de la revolución es la tecnología que vivimos, donde prácticamente el uso de la tecnología se ha vuelto indispensable [8].

E. Lo importante a la hora de construir elementos de juegos serios. Dentro de una lista importante de características que incluyen un juego serio, debemos comenzar refiriéndonos a la memoria del Juego. Esta virtud es que el juego debe memorizar los logros o lo que ha aprendido el jugador, es que después del juego no solo se reinicia la dificultad del juego en el mismo punto, ya que esto puede hacer que el jugador no apela al juego porque lo hará ser el mismo para jugar de nuevo. Pero debe incluir una forma en que el juego recuerde lo que el jugador aprendió, debe ser educativo o apostar, saber qué parte del jugador de terapia se mantuvo así si el juego se utilizó

como terapia de recuperación de algún tipo [9]. El flujo del juego es muy importante para sembrar la lealtad del jugador. Para ello debemos intentar modular la dificultad que no frustra ni aburre en el otro extremo para el usuario. Por esta razón, la dificultad debe ser modulada cuidadosamente [9]. Otra cosa muy importante es la historia o la narrativa. La combinación de historia, paisaje y contexto debe ser consistente con la cultura y el país. La narrativa es un elemento básico que debe indicar que el entorno gráfico debe ser inmersivo (sumergido, sumergido, escondido) para que el jugador sienta que está en el juego. Como la interactividad anterior debe estar presente para que el jugador interactúe tanto como sea posible jugando. No debemos olvidar también que deben proporcionar elementos coherentes con la realidad, ya que un objetivo del juego es que el jugador desarrolle y / o mejore sus habilidades. Otra de las cosas más importantes de un juego serio es proporcionar cierto grado de adaptabilidad, este es el juego que se adapta al estilo del jugador para que responda correctamente y, dependiendo del estilo de un juego serio, piense en el público quién jugará, las historias que puedan interesar, los entornos y los personajes con los que el jugador se sentirá atraído para jugar son una parte esencial del juego. Las reglas deben estar claramente definidas. El jugador. Y, sobre todo, no debemos dejar pasar un juego serio, por obligación debe ser divertido [8].

2. Metodología de la propuesta

Algoritmo de fuegos artificiales. Para cumplir con el tiempo, en la ausencia de un modelo de integración social, y políticas públicas, se ha incluido considerar el modelo de juego de depredador-presa para analizar las relaciones entre estas minorías y el resto del grupo mayoritario [3, 4]. En el marco de algoritmo de fuegos artificiales, cuando se encuentra un fuego artificial, una lluvia de chispas llenará el espacio local alrededor del fuego artificial [1, 2].

En nuestra opinión, el proceso de explosión de un fuego artificial se puede ver como una búsqueda en el espacio local alrededor de un punto específico donde se muestra el fuego artificial a través de las chispas generadas en la explosión. Cuando se nos pide que encontremos un punto *xj* que satisfaga, ecuación (1):

$$f(xj)=y, \tag{1}$$

podemos activar continuamente "fuegos artificiales" en el espacio potencial hasta que una "chispa" apunte o esté bastante cerca del punto *xj*. Al igual que el proceso de activación de los fuegos artificiales, en la figura 2 se muestra un marco aproximado del FA. En el FA, para cada generación de explosión, primero se seleccionan y se publican, donde se disparan *n* fuegos artificiales.

Luego, después de la explosión, se obtienen y se evalúan las fuentes de las chispas. Cuando se encuentra la ubicación óptima, el algoritmo se detiene. De lo contrario, en otras ocasiones, se seleccionan las cines y fuegos artificiales actuales para la próxima generación de explosiones. En la figura 2, se puede ver que el éxito del FA radica en un buen diseño, un proceso de explosión y un método adecuado para seleccionar las preferencias, que se detallan respectivamente en las siguientes subsecciones.

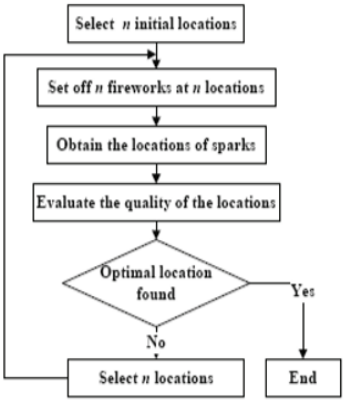
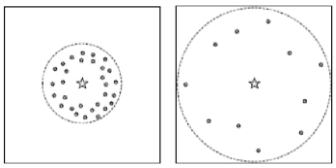


Fig. 2. Marco del algoritmo de fuegos artificiales.

Diseño de la explosión de fuegos artificiales. Al observar la exhibición de fuegos artificiales, hemos encontrado dos comportamientos específicos de explosión de fuegos artificiales. Cuando los fuegos artificiales están bien fabricados, se generan numerosas chispas, y las chispas centralizan el centro de la explosión. En este caso, disfrutamos de la espectacular exhibición de los fuegos artificiales. Sin embargo, para una mala explosión de fuegos artificiales, se generan muy pocas chispas, y las chispas se dispersan en el espacio. Las dos formas se muestran en la Figura 3. Desde el punto de vista de un algoritmo de búsqueda, un buen fuego artificial indica que el fuego artificial se ubica en un área prometedora que puede estar cerca de la ubicación óptima. Por lo tanto, es apropiado utilizar más chispas para buscar en el área local alrededor del fuego artificial. En contraste, un fuego artificial malo significa que la ubicación óptima puede estar lejos de donde se ubica el fuego artificial. Entonces, el radio de búsqueda debe ser mayor. En el FA, se generan más chispas y la amplitud de la explosión es menor para un buen fuego artificial, en comparación con uno malo [11].



(a) Buena explosión / (b) Mala explosión

Fig. 3. Dos tipos de explosión de fuegos artificiales.

Número de chispas. Supongamos que el FA está diseñado para el problema general de optimización, ecuación (2):

$$\text{Minimize } f(\boldsymbol{x}) \in \mathbb{R},\ \boldsymbol{x}_{\min} \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}_{\max}\ , \tag{2}$$

donde *x = x1, x2, . . ., xd* denota una ubicación en el espacio potencial, *f (x)* es una función objetivo, y *xmin* y *xmax* denotan los límites del espacio potencial. Luego, el número de chispas generadas por cada *xi* de fuego artificial, ecuación (3), se define de la siguiente manera:

$$s_i = m \cdot \frac{y_{\max} - f(x_i) + \xi}{\sum_{i=1}^{n} (y_{\max} - f(x_i)) + \xi} \ , \tag{3}$$

donde *m* es un parámetro que controla el número total de chispas generadas por los *n* fuegos artificiales, *ymax = max (f (xi)) (i = 1, 2, ..., n)* es el valor máximo (peor) de la función objetivo entre los *n* fuegos artificiales, y ξ, que denota la constante más pequeña en la computadora, se utiliza para evitar el error de división cero. Para evitar los efectos abrumadores de los espléndidos fuegos artificiales, los límites se definen para sí, como se muestran en la ecuación (4).

$$\hat{s}_i = \begin{cases} round(a \cdot m) & \text{if } s_i < am \\ round(b \cdot m) & \text{if } s_i > bm \ , \ a < b < 1 \\ round(s_i) & otherwise \end{cases} \ , \tag{4}$$

donde *a* y *b* son parámetros *const*. Amplitud de explosión. En contraste con el diseño del número de chispas, la amplitud de una buena explosión de fuegos artificiales es menor que la de una mala. La amplitud de explosión para cada fuego artificial se define de la siguiente manera, ecuación (5):

$$A_i = \hat{A} \cdot \frac{f(x_i) - y_{\min} + \xi}{\sum_{i=1}^{n} (f(x_i) - y_{min}) + \xi} \ , \tag{5}$$

donde *Aˆ* denota la amplitud máxima de explosión, y *ymin = min (f (xi)) (i = 1, 2, ..., n)* es el valor mínimo (mejor) de la función objetivo entre los *n* fuegos artificiales. Generando chispas. En una explosión, las chispas pueden sufrir los efectos de una explosión desde direcciones *z* aleatorias (dimensiones). En el FA, obtenemos el número de direcciones afectadas al azar de la siguiente manera, ecuación (6):

$$z = round(d \cdot rand(0, 1)) \ , \tag{6}$$

donde *d* es la dimensionalidad de la ubicación *x*, y *rand (0, 1)* es una distribución uniforme sobre *[0,1]*. La ubicación de una chispa del fuego artificial *xi* se obtiene utilizando el algoritmo 1. Al simular el proceso de explosión, se genera primero la ubicación de una chispa *xx*. Luego, si la ubicación obtenida se encuentra fuera del espacio potencial, se asigna al espacio potencial de acuerdo con el algoritmo.

Algorithm 1. Obtain the location of a spark

Initialize the location of the spark: $\tilde{x_j} = x_i$;
z=$round(d \cdot rand(0,1))$;
Randomly select z dimensions of $\tilde{x_j}$;
Calculate the displacement: $h = A_i \cdot rand(-1,1)$;
for each dimension $\tilde{x}_k^j \in$ {pre-selected z dimensions of $\tilde{x_j}$} **do**
  $\tilde{x}_k^j = \tilde{x}_k^j + h$;
  if $\tilde{x}_k^j < x_k^{\min}$ or $\tilde{x}_k^j > x_k^{\max}$ **then**
    map $\tilde{x}_k^j$ to the potential space: $\tilde{x}_k^j = x_k^{\min} + \mid \tilde{x}_k^j \mid \%(x_k^{\max} - x_k^{\min})$;
  end if
end for

Para mantener la diversidad de chispas, diseñamos otra forma de generar chispas: la explosión gaussiana, que se muestra en el algoritmo 2. Se utiliza una función

gaussiana *(1, 1)*, que denota una distribución gaussiana con media 1 y desviación estándar 1, para Definir el coeficiente de la explosión. En nuestros experimentos, se generan *ˆm* chispas de este tipo en cada generación de explosión.

Algorithm 2. Obtain the location of a specific spark

Initialize the location of the spark: $\hat{x}_j = x_i$;
z=$round(d \cdot rand(0,1))$;
Randomly select z dimensions of $\hat{x}_j$;
Calculate the coefficient of Gaussian explosion: $g = Gaussian(1,1)$;
for each dimension $\hat{x}_k^j \in$ {pre-selected z dimensions of $\hat{x}_j$} do
  $\hat{x}_k^j = \hat{x}_k^j \cdot g$;
  if $\hat{x}_k^j < x_k^{\min}$ or $\hat{x}_k^j > x_k^{\max}$ then
    map $\hat{x}_k^j$ to the potential space: $\hat{x}_k^j = x_k^{\min} + | \hat{x}_k^j | \%(x_k^{\max} - x_k^{\min})$;
  end if
end for

Selección de ubicaciones. Al comienzo de cada generación de explosión, se deben seleccionar *n* ubicaciones para la explosión de fuegos artificiales. En el FA, la mejor ubicación actual *x*, sobre la cual la función objetivo *f (x ∗)* es óptima entre las ubicaciones actuales, siempre se mantiene para la siguiente generación de explosión. Después de eso, se seleccionan *n - 1* ubicaciones en función de su distancia a otras ubicaciones para mantener la diversidad de chispas. La distancia general entre una ubicación *xi* y otras ubicaciones se define de la siguiente manera:

$$R(x_i) = \sum_{j \in K} d(x_i, x_j) = \sum_{j \in K} \| x_i - x_j \| , \tag{7}$$

donde *K* es el conjunto de todas las ubicaciones actuales de fuegos artificiales y chispas. Entonces, la probabilidad de selección de una ubicación *xi*. (ecuación (8)). Se define de la siguiente manera:

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_j)} . \tag{8}$$

Al calcular la distancia, se puede utilizar cualquier medida de distancia, incluida la distancia de Manhattan, la distancia euclidiana, la distancia en ángulo, etc. [9]. Cuando *d (xi, xj)* se define como | *f (xi) - f (xj)* |, la probabilidad es equivalente a la definición de la probabilidad basada en la densidad inmune en [10].

3. Prototipo propuesto

Debido al clima existente en Ciudad Juárez - Una sociedad fronteriza - donde la temperatura durante más de tres meses no permite actividades al aire libre, y se practican pocos deportes además del béisbol y el fútbol, para nuestra investigación decidimos elegir el waterpolo. Los estudiantes de los niveles escolares básicos aceptan colaborar en equipo, respetar las reglas y que el juego pueda interconectarse con otros usuarios en un modelo llamado "Modelo cooperativo para la resolución de problemas", una idea innovadora propuesta en el rol masivo de jugadores múltiples en línea. Los juegos (MMORPG). La cooperación entre jugadores para realizar tareas

difíciles es a menudo una mecánica integral del juego, y surgen grupos organizados de jugadores, a menudo llamados gremios, clanes o facciones. A veces las relaciones con los jugadores dentro del juego se extienden a amistades o relaciones románticas en el mundo material. En otros casos, los socios románticos y los grupos de amigos del mundo material encuentran que jugar juntos fortalece sus vínculos. Utilizamos componentes comprados con nuestro Proyecto para desarrollar nuestro modelo innovador como se muestra en la figura 4.



Fig. 4. Componentes de la aplicación Wii desarrollados.

La idea principal del Juego serio para una plataforma Wii es desarrollar las habilidades asociadas con la empatía y las habilidades psicomotoras para mejorar el rendimiento de los niños, es por eso que, para construir el prototipo del juego serio, inicialmente utilizamos tres escenarios asociados con la práctica del Waterpolo: una piscina cubierta, una piscina olímpica o un área delimitada en un parque acuático. Al comenzar nuestro juego serio, cada jugador debe registrar su nombre y la posición asignada para que el hardware refleje los movimientos necesarios para jugar en equipo. Y determine la modalidad del juego, en este caso, el jugador seleccionará un nivel que va de simple a complejo. En el nivel más simple, el jugador tendrá la posibilidad de verificar si los movimientos realizados son apropiados de acuerdo con el regalo de FINA, es por eso que la importancia de nuestra propuesta radica en que, junto con jugar simultáneamente con otros jugadores, puede aprender de una manera más apropiada. El juego irá demostrando que puede completar los desafíos propuestos y, a su vez, desarrollar sus habilidades y conocimientos en el Waterpolo, es por eso que Serious Game propone al usuario demostrar sus habilidades y conocimientos, como el juego serio progresa cada vez que el jugador logra los objetivos deseados, se crea un modelo de sonido para recompensar el buen desempeño en el juego serio, como se muestra en la figura 5.



Fig. 5. Nuestro Wii Serious Game Model utilizando una tarea de colaboración para jugar Waterpolo.

Como se menciona en este documento, este proyecto se encuentra actualmente en la fase de construcción. Este prototipo Serious Play tiene la firme intención de tomar la siguiente fase de construcción completa y funcional. Una fundación patrocinada por uno de los hospitales de niños más importantes de la ciudad de Paso Texas en los Estados Unidos. Ha mostrado un interés genuino en el proyecto, por lo que mencionó que proporciona el apoyo necesario para su realización. Esta fundación cuenta con programas de cuidado infantil y cuenta con estrategias interesantes y muy eficientes. Es por eso que le gustaría tener este tipo de herramientas integradas en sus programas.

Para esto, pronto será formalizado por ambas partes comprometidas a apoyar el proyecto que se muestra en este documento. En futuras investigaciones, tratamos de modificar un juego basado en el trabajo colaborativo en un grupo (estamos eligiendo rugby seven) con una alta intensidad de presión para cada niño y modificamos la importancia relacionada con el soporte de este tipo de presión relacionada con la responsabilidad de una actividad colectiva, una aproximación se relacionará con lo que implica Waterpolo, como se muestra en la figura 5.

Un aspecto muy relevante, es considerar que si alguien pregunta por qué le gusta usar nuestro Juego serio, este usuario podrá responder: porque ha tenido un ámbito lúdico y una selección adecuada con el avatar, por lo que podría tener empatía por nuestra propuesta.

Al analizar con más detalle el grupo de personas que usaron nuestros Serious Games, determinamos que, como el juego de roles, es un pasatiempo que los une y les brinda oportunidades para ayudarse mutuamente y a su comunidad de videojuegos. Es un entorno seguro en el que puede experimentar interacciones sociales, algo fundamental cuando el clima no lo permite

Este grupo de usuarios de nuestro Juego serio dice que han sido testigos del crecimiento personal de los individuos en términos de su autoestima y la expansión de sus interacciones sociales como resultado del juego. Este es solo uno de los beneficios del juego.

Nuestra investigación mostró que se descubrió que todos pueden encontrar algunas horas a la semana para "salvar el universo, atrapar a los villanos o resolver misterios", incluido aprender a practicar waterpolo, y jugar con la computadora es tan divertido como cualquier otra actividad en nuestra investigación. Jugar nuestro juego serio puede fortalecer una variedad de habilidades como matemáticas y leer recomendaciones en línea.

Aumente la capacidad de pensar y hablar de manera clara y concisa, al formular e implementar sus planes, cooperar y comunicarse con otros, así como aumentar la capacidad de análisis. Información escrita y verbal. Colocado en el mercado, nuestro juego serio determinará que los jugadores sean miembros cohesivos del grupo en los juegos multijugador, pueden ayudar a las personas a desarrollar habilidades de liderazgo y promover la cooperación, el trabajo en equipo, la amistad y la comunicación abierta.

En otros estudios relacionados con este tipo de Wii Serious Games, tratamos de compararlos con nuestros colegas de Montenegro que proponen y desarrollan un juego serio innovador que involucra un modelo para los practicantes de esgrima. Este deporte está alcanzando una gran popularidad en esta sociedad. En la siguiente figura 6 se puede mostrar un modelo representativo:



Fig. 6. Uso del software de Wii para mejorar el rendimiento en el Pentatlón moderno.

¿Qué esperaban los usuarios de nuestra propuesta de Juego Serio en este modelo de Wii para aprender y practicar el Waterpolo? Mejore el estado de ánimo del jugador según el juego transcurrido a través de las herramientas que se utilizarán como música de fondo, utilizando técnicas de musicoterapia para animar a los espíritus Jugador [11]. Otro elemento de la estrategia son los colores del escenario. A través del efecto del color en el estado de ánimo es tener al jugador en un estado emocional positivo [10]. El tercer elemento es a través de los sonidos en el juego. Con cada éxito y cada falla, el entorno ambiental suena el escenario en el que estás jugando. Y el cuarto elemento es el reconocimiento de los logros. A través de insignias, medallas, trofeos, puntuaciones y citas, usted desea que el jugador se sienta satisfecho con el reconocimiento de cada logro [12].

4. Resultados

Las escenas estructuradas asociadas con los agentes no se pueden reproducir en general, ya que solo representan un momento dado en el espacio y el tiempo de las diferentes sociedades. Estos representan una forma única e innovadora de comportamiento adaptativo que resuelve un problema computacional que no trata de agrupar las sociedades solo con un factor asociado con su apariencia externa (fenotipo), tratando de resolver un problema computacional que involucra un cambio complejo entre las relaciones existentes con la población que practican estos deportes. Las configuraciones generadas pueden relacionarse metafóricamente con el conocimiento del comportamiento de la comunidad con respecto a un problema de optimización (para ajustarse al clúster social y culturalmente con otras personas similares, sin ser del mismo deporte [4]). En la tabla 1 se muestra una muestra de siete deportes que describen las características analizadas en cada deporte diferente y su práctica requerida. Intentamos mejorar la siguiente ecuación 4, en esta descripción, K es el desempeño de la práctica de cada deporte en la ciudad y sus respectivas promociones en diversos lugares. (ecuación 9).

$$K = [C + CS + GAL] \pm CBS, \tag{9}$$

en donde: C = representa si el deporte se puede practicar en cualquier lugar, por ejemplo, el Ajedrez. CS = representa el capital simbólico asociado con la perspectiva de una sociedad, la práctica de esgrima se considera sofisticada. G = es el género del rango de la población, en la Ciudad de Juárez la pirámide de la población es diferente al resto de América Latina porque la violencia provoca muertes y el éxodo de la

ciudad. *AL* = Aspecto lúdico relacionado con la edad de los niños que practican un deporte, por ejemplo, en el Trampolín, el aspecto lúdico se asocia con la alta promoción de este deporte. *CBS* = Beneficios sociales: los costos relacionados con cada deporte, en el caso de la gimnasia rítmica, se asocian con la mejora de la salud en los niños sedentarios menos peso porque requieren muchas actividades deportivas relacionadas con los cuatro eventos: pelota, garrote, aro y cinta.

Tabla 1. Análisis multivariable con la información del número de espacios a recuperar para practicar estos deportes virtuales y sus representaciones por género (problema) –valores a masculino y femenino respectivamente–, el estado social por su práctica, divertido de usar la aplicación Wii, el clima, mejora el aumento de la salud para cambiar los paradigmas al sedentarismo en los niños y, finalmente, la relación entre el costo / beneficio social asociado con su práctica.

Sport Virtual	Space	Gender	Social Status	Fun	Complexity	Kinetic Adaptation	Climate	Increase Health	Cost-Benefit
Aquatic Sky	8	m-.50,f-.45	0.498	0.914	0.928	0.815	0.774	0.715	0.387
Tae Kwon Do	6	m- 40,f- 30	0.857	0.714	0.628	0.914	0.851	0.879	0.568
Baseball	5	m-50, f-.40	0.617	0.658	0.574	0.971	0.385	0.712	0.514
Syncronized Swimming	3	f-47	0.275	0.637	0.416	0.989	0.814	0.748	0.885
Waterpolo	14	m-.45,f- .40	0.578	0.784	0.925	0.997	0.627	0.914	0.879
Bowling	4	m-.30,f- .30	0.620	0.631	0.715	0.884	0.856	0.802	0.744
BMX Bike	3	m-.40,f- .40	0.562	0.748	0.611	0.996	0.303	0.897	0. 448
Fencing	7	m-.57, f-.37	0.912	0.815	0.478	0.914	0.487	0.914	0.872
Dance Rolling	9	m-.27, f-.49	0.884	0.924	0.788	0.956	0.589	0.895	0.916
Figure Sky	5	m-.19, f-.50	0.948	0.912	0.916	0.978	0.398	0.917	0.887

Hasta la fecha, se está desarrollando una Herramienta para la toma de decisiones para identificar en forma gráfica la forma en que la violencia afecta un índice compuesto al IDH en diferentes partes de la ciudad y con diferentes escenarios según la perspectiva de aumento o disminución del mismo. Para estos tres escenarios se están construyendo (progreso optimista, pesimista y gradual) para la identificación de cambios con respecto a la temporalidad y la ubicuidad de la información. El análisis e interpretación de estos escenarios con respecto al tiempo debe tratarse con sumo cuidado, buscando hacer un balance entre la predicción numérica realizada y justificada por el uso de ecuaciones, pudiendo seleccionar un índice, un índice compuesto, un factor o un rango asociado. La figura 7 muestra la perspectiva de la Herramienta de toma de decisiones propuesta y sus componentes principales, incluido un generador de escenarios asociado con el año de predicción numérica, que puede mostrarse mediante el uso de gráficos de radar y tacómetros de variabilidad estadística, los cambios en los escenarios mostrados, el ajuste de parámetros con respecto a los aspectos sociales y su relación con el año a proyectar, la visualización de un mapa que muestra en color los cambios en el IDH para cada AGEP de la ciudad asociada con el aumento en la violencia y la visualización de un guion narrativo que detalla los parámetros finales más importantes de la simulación realizada, como se puede ver en la figura 7.

Uno de los aspectos más relevantes de la investigación realizada es la migración de la herramienta de toma de decisiones al sitio del Centro de Investigación Social de la UACJ, de modo que cualquier usuario pueda consultar con respecto a los indicadores asociados a HDI y su efecto de la Violencia existente en la ciudad. Uno de los aspectos metodológicos que justifican los escenarios para su simulación, se asocia con las encuestas de percepción de violencia que se han llevado a cabo en el Centro de Investigación Social y que muestran las variables asociadas con diversos elementos del estado que la sociedad mantiene con respecto a los indicadores de violencia más

prevalentes. Otro aspecto que se desarrollará será la implementación de dicha Herramienta de toma de decisiones en dispositivos móviles como se propone en [11], que tienen la capacidad de modificar valores en situaciones de alta incertidumbre, como el tipo de cambio en el dólar mexicano. -prelación de la relación con la economía y en la parte social, el aumento de la violencia en un mes específico -en esa categoría, la disminución de las muertes durante 2011 puede estar asociada con el aumento de la extorsión, el secuestro y el robo con violencia.

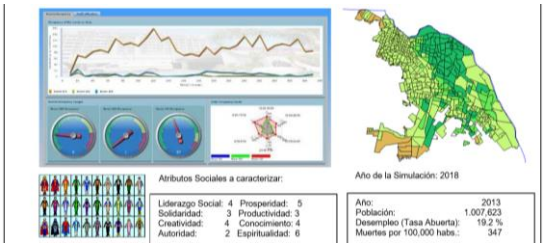


Fig. 7. Simulador de la Herramienta de Toma de Decisiones.

5. Discusiones e investigación futura

El aspecto más relevante de esta investigación es consolidar la práctica de deportes en niños y jóvenes que, debido a las condiciones climáticas, no pueden ponerse de acuerdo para unirse para practicar un deporte junto con la confianza en sí mismos que generan los jugadores, lo que permite mejorar su desempeño en otras áreas de su vida diaria a través de un modelo de apoyo emocional en los niños, lo que conlleva un compromiso y una complejidad intrínseca en su desarrollo motor. Teniendo en cuenta que la infancia es una etapa vulnerable en la que también se encuentran en pleno desarrollo y cualquier evento o suceso puede causar efectos negativos y puede dejar al niño marcado permanentemente por el resto de su vida [13]. Por lo tanto, es muy importante centrarse más en cómo obtener resultados asociados con el desempeño del grupo, patrocinado por el individuo. Es por eso que el futuro del Juego serio requerirá una investigación más profunda que permita tener un gran impacto al tener la oportunidad de ayudar a los niños y jóvenes que no tienen acceso a los deportes por varias razones. En el futuro, los Juegos serios pueden presentar una oportunidad natural debido en gran parte a la aceptación de que los videojuegos tienen en esta era y más aún con la ventaja de que estas generaciones tienen fácil acceso a la tecnología. Como se mencionó en la presente investigación, este prototipo desea continuar la consolidación para establecer diversos equipos asociados principalmente con el avatar. Este prototipo en lo que respecta al Juego Serio para lograr la fase de construcción completa y funcional. En este momento y debido a un proyecto con financiamiento de la Unión Europea y la colaboración de FINA, queremos hacer una aplicación Wii que permita ser inclusivo a través de estrategias interesantes y muy eficientes asociadas con la movilidad. FINA está muy interesada en este tipo de aplicaciones que podrían diversificar la práctica del waterpolo en los países en desarrollo. En futuras investigaciones, tratamos de mejorar la práctica de un deporte asociado con un juego y en base al trabajo colaborativo en un grupo con alta

intensidad de presión para cada niño, como lo es el tenis profesional, y modificamos la importancia relacionada con el apoyo de este tipo de presión relacionado con la responsabilidad de una actividad colectiva [14], como se muestra en la figura 8.



Fig. 8. Un juego serio basado en actividades colectivas y relacionado con el aumento de habilidades sociales.

El experimento principal consistió en detallar cada uno de los 47 deportes, con 500 agentes y una condición de desempleo de 50 generaciones, lo que nos permitió generar diferentes escenarios relacionados con los horizontes temporales, que se obtuvieron después de comparar diferentes similitudes culturales y sociales en cada comunidad, y para determinar las relaciones existentes entre cada uno en relación con la Distancia de Mahalanobis (el número de puntos que indica cada deporte y el tamaño de las personas representa el número de personas que determinan la magnitud relacionada con la sociedad). Diseño propuesto de experimentos.

Por otra parte, los objetivos específicos se determinarán a través de un estudio correlacional, en el que experimentará con diferentes variables. Esto con el propósito de conocer la relación existente entre dos o más variables para comprender qué variables son las que más afectan el IDH, como; el diseño de políticas públicas que mejoren la calidad de vida de las personas con problemas como la inseguridad, la violencia y el secuestro. Asimismo, se pretende analizar el grado de asociación entre estas variables. Estos resultados serán arrojados a través de un diseño factorial que intentará analizar mediante un diseño de experimentos con arreglo ortogonal, el efecto que las diferentes variables independientes tienen como violencia, inseguridad y secuestro en diferentes niveles como edad, género y nivel socioeconómico, aproximadamente la variable dependiente que, en nuestra investigación, es un efecto asociado con la calidad de vida denominada HDI, o que, en su caso, no está presente. En la presente investigación, se propone un diseño de experimentos en el que se llevarán a cabo diferentes interacciones de las variables a estudiar: violencia, inseguridad, extorsión y secuestro, que son las variables de control (Figura 9). Asimismo, estas variables se estudiarán en cuatro niveles: área geográfica, nivel de estudios, género y edad, que son las variables de ruido.

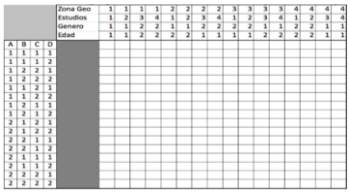


Fig. 9. Diseño de experimentos propuestos.

Es importante determinar cambios paradigmáticos en nuestro modelo y justificar que se necesita estimar un costo/beneficio social.

Referencias

1. Jaeger, S.: Origins of Child Psychology: William Preyer, en W.R. Woodward. Woodward: s.n., pp. 300–321 (1982)
2. Chrisman, O.: Paidologie; Entwurf zu einer Wissenschaft des Kindes. s.l.: Jena, Bernhard Vopelius (1896)
3. HHS of USA. Center for Disease Control and Prevention. Especiales de los CDC: Salud mental de los niños: Apoyo de los servicios de salud conductual para quienes los necesitan http://www.cdc.gov/spanish/especialesCDC/ SaludMentalNinos/index.html
4. U.S. Department of Education of USA. Center for Parents Information and Resources. Sobre la Ley IDEA. http://www.parentcenterhub.org/repository/ sobreidea/
5. Tómas, U.: Elpsicoasesor. http://elpsicoasesor.com/25-poderes-terapeuticos-del-juego/
6. Schaefer, C.E.: Terapia de Juego. México: Editorial El Manual Moderno,S.A. de C.V. (2012)
7. Braga, P.H.C., Silveira, I.F.: A Pattern Language for semi-automatic generation of Digital Animation through hand-drawn Storyboards. Programa de Pós Graduação em Engenharia Eléctrica e Computação (2015)
8. Silveira, I.F.: Juegos Serios. Brazil: UACJ, Juegos Serios: teoría y práctica (2016)
9. Csíkszentmihályi, M.: Flow – The Psychology of optimal experience. New York: Harper & Row (1990)
10. Heller, E.: Psicología del color: Como actúan los colores sobre los sentimientos y la razón. s.l.: Rústica (2016)
11. Lopez, J.J.B.: Musicoterapia como alternativa terapéutica en la depresión. Guatemala: Universidad de San Carlos de Guatemala (2012)
12. Leary, M.E.: Interpersonal Rejection. New York: Oxford University Press (2001)
13. The Hospital for Sick Children. About Kids Health. Efectos psicológicos del desastre sobre los niños. http://www.aboutkidshealth.ca/En/HealthAZ/ Multilingual/ES/Pages/psychological-effects-of-disaster-on-children.aspx
14. Garcia-Mundo, L., Vargas-Enriquez, J., Genero, M., Piattini, M.: ¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática? Actas de las XX JENUI, pp. 303–310 (2014)

ISSN 1870-4069

Traducción del lenguaje de señas usando visión por computadora

Eduardo Mancilla Morales, Oswaldo Vázquez Aparicio, Pedro Arguijo, Roberto Ángel Meléndez Armenta, Antonio Hiram Vázquez López

Tecnológico Nacional de México/Instituto Tecnológico Superior de Misantla, México
eduardomancillamorales@gmail.com, 192T0035@itsm.edu.mx, pedro_arguijo@excite.com, ramelendeza@itsm.edu.mx, jahvazquezl@itsm.edu.mx

Resumen. Se realiza el análisis del desempeño de redes neuronales multicapa y SVM como clasificador para el reconocimiento de señas del LSM considerando tres conjuntos de características: Momentos de Hu, Momentos de Zernike e Histogramas de orientación del gradiente (HOG). Se creó un dataset de 21 señas y se le aplicó un preprocesamiento a las imágenes antes de extraer el conjunto de características, las técnicas aplicadas son: reducción de ruido, escalamiento de la imagen y eliminación del fondo de la imagen. Así mismo, se analizó el efecto de la reducción de dimensionalidad del conjunto de características con los algoritmos PCA y LDA. Los resultados indican que los momentos de Zernike junto con LDA ofrece la mejor exactitud en los resultados.

Palabras clave: LSM, reconocimiento de imágenes, segmentación de piel, aprendizaje automático, momento de Hu, momento de Zernike, HOG.

Sign Language Translation Using Computer Vision

Abstract. We analyzed the perfomance of multilayer neural networks and SVM as classifier for the Mexican Sign Language (LSM) considering three diferent sets of features: Hu moments, Zernike moments and Histogram of oriented gradients (HOG). A 21-sign dataset was created and in order to extrac the features the images were pre-processed, the applied techniques were: noise reduction, image scaling, and image background removal. Likewise, we analyzed the dimensionality reduction of the set of features with PCA and LDA. The obtained results show that the moments of Zernike together with LDA offers the best accuracy in the classification using the features set.

Keywords: LSM, image recognition, skin segmentation, automatic learning, Hu moments, Zernike moments, HOG.

1. Introducción

La comunicación es fundamental para el desarrollo social del ser humano. Entre las diferentes formas de comunicación, la oral es la más común. Cuando el habla se ve impedida por algún motivo, se disminuye la capacidad de interacción social del individuo, lo que limita sus oportunidades de inclusión. Las personas con problemas del habla y auditivos se comunican utilizando señas que incluyen movimientos de las manos o cuerpo y expresiones faciales. Con este fin, en México se ha desarrollado el lenguaje de señas mexicano (LSM) cuyo alfabeto consta de 27 señas, de las cuales 21 son estáticas y las restantes dinámicas. Rautaray y Agrawal [9] mencionan que las señas estáticas se definen por la orientación y posición de la mano, durante algún tiempo sin mover ésta. Si la mano varía durante ese tiempo, se denomina gesto dinámico. El reconocimiento de gestos dinámicos requiere de secuencias de imágenes en el tiempo, debe detectarse que las secuencias tengan un patrón de movimiento, lo cual implica un conocimiento de las secuencias anteriores.

Se han tenido grandes avances en el desarrollo de sistemas de traducción automatizada de señas a texto utilizando técnicas de visión computacional. Sin embargo, dicho avance se ha dado principalmente para el lenguaje de señas americano, el japonés y el de la India. Con este fin se propuso un sistema un sistema que utiliza un sensor de movimiento para capturar imágenes de profundidad y el esqueleto de la mano humana para rastrear movimientos considerando la altura y profundidad de la mano [1]. También se ha presentado un sistema basado en visión artificial [11] que realiza el reconocimiento de veintiuna señas estáticas cuya descripción está dada por momentos centrales normalizados invariantes a transformaciones de escala y rotación. Obtienen un 93 % de exactitud con un perceptrón multicapa para la etapa de reconocimiento. El sensor Kinect se utilizó para la interpretación de palabras [12], consideran los momentos geométricos como extractor de características y modelos ocultos de Markov como clasificador para encontrar las relaciones entre la secuencia de imágenes, obteniendo tanto una sensibilidad como especificidad del 80 %. Montaño y Rodríguez-Aguilar [5] proponen un sistema para la traducción automática de lenguaje de señas mexicano a texto reportando un porcentaje de exactitud del 80 % comparando el vector de características obtenido con vectores previamente almacenados.

El presente trabajo se centra en el análisis del desempeño de redes neuronales y SVM como clasificador para el reconocimiento de señas del LSM. Se considera el enfoque basado en apariencia con una cámara como medio de captura. En la siguiente sección se describe el dataset utilizado. La solución propuesta se aborda en la sección 3. La sección 4 describe la implementación y en las secciones 5 y 6 se aborda el análisis de los resultados y las conclusiones, respectivamente.

2. Dataset de señas estáticas del LSM

En la búsqueda de un dataset para el lenguaje de señas mexicano, solamente se encontró uno pero a las imágenes de éste les faltaba variación en traslación

y escala [3], por lo cual se creó un dataset con variaciones de traslación, escalamiento y rotación. En la Fig. 1, se muestran variaciones en la posición de la mano en traslación y rotación. La traslación se logra moviendo la mano de un lado hacia otro y la rotación se obtiene al girar la mano, en la Fig. 1 se muestra con una flecha el movimiento que se realizó al capturar las imágenes. El dataset consta de 300 imágenes por cada una de las 21 señas estáticas, con las variaciones mencionadas, dando un total de 6300 imágenes. La Fig. 2, exhibe un ejemplo de las 21 señas estáticas capturadas. Todas las señas se realizaron por una sola persona. El dataset utilizado en este artículo se encuentra disponible, para las personas interesadas en su posterior análisis, en la dirección electronica: `https://1drv.ms/f/s!Ag_D60aA-1xrggXOp-G9eJrs-u3l`



Fig. 1. Imágenes de la letra A con variaciones en traslación y rotación.

3. Solución propuesta

El método propuesto en el presente trabajo incluye una etapa de preprocesamiento, una etapa de extracción de caracteristicas y una etapa de clasificación. El proceso es el siguiente:

- Creación del dataset de las 21 señas.
- Eliminar el ruido de la imagen.
- Escalar la imagen a un tamaño predeterminado.
- Segmentar la imagen por color de piel.

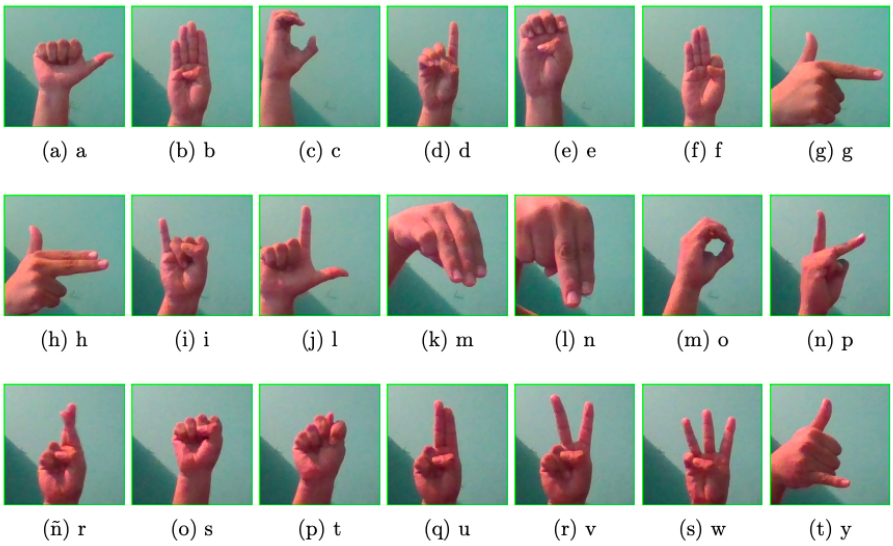


Fig. 2. Ejemplos de las imágenes de todas las señas del dataset creado.. Sólo se incluyen las letras estáticas.

- Transformar la imagen a un vector de características.
- Reducir la dimensionalidad del vector.
- Entrenar el clasificador.

3.1. Reducción de ruido

Se realizó un comparación entre los filtros promediador, gaussiano y bilateral [7] concluyendo que este último permite conservar los bordes sin una pérdida significativa de la mano.

3.2. Escalamiento de la imagen

Con la finalidad de aumentar la velocidad de procesamiento se escalan la imágenes a un tamaño de 200×200 pixeles, con lo cual se conserva suficiente información. Este tamaño de imagen da un total de 40,000 píxeles en comparación con los 777600 de la imagen original (1080×700).

3.3. Eliminación del fondo de la imagen

Se necesita eliminar el fondo de la imagen, para evitar tener información innecesaria, que puede interferir en los resultados de la detección de la mano. Consecuentemente, se utiliza la segmentación por color de piel. La cual consiste en eliminar las partes de la imagen que no tengan un color similar al color de la

piel, eliminando el fondo, en la Fig. 3 se muestra el resultado de la segmentación. Se utiliza el espacio de color YCbCr debido a que puede ser aplicado a imágenes con colores complejos y con iluminación irregular. Los valores que se consideran son [10]:

$$150 < Cr < 200 \tag{1}$$

$$100 < Cb < 150 \tag{2}$$

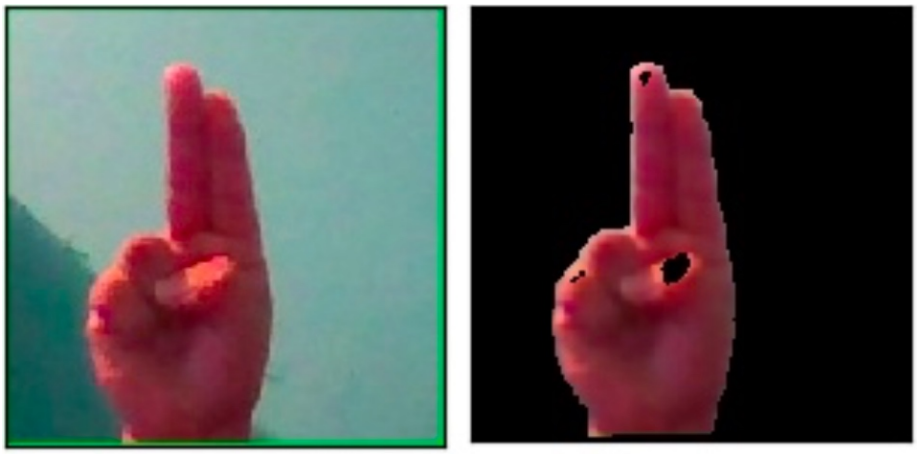


Fig. 3. Segmentación basada en el color de la piel.

3.4. Transformar la imagen a un vector de características

Se utilizaron tres algoritmos de extracción de características: momentos de Hu [6], momentos de Zernike [6] e histograma de orientación del gradiente [4].

El algoritmo de Hu da como resultado un vector de longitud igual a 7, la longitud del vector de características que produce el algoritmo de Zernike depende del grado del polinomio usado, y la longitud del vector de características para el algoritmo de Histograma de Orientación del Gradiente depende del número de píxeles por celda y el número de orientaciones seleccionadas. El algoritmo que da como salida el mayor número de características, es el Histograma de Orientación del Gradiente.

Los parámetros considerados para la obtención del vector de características correspondiente a los momentos de Zernike fueron: grado 5 y radio 80, con lo cual se obtuvieron 12 características. Para HOG se consideraron 8 orientaciones, con celdas de 16×16 píxeles, obteniendo un total de 1153 características.

Debido a la gran dimensionalidad del vector de características de HOG se emplearon técnicas de reducción a fin de lograr una mejor eficiencia del clasificador, los métodos utilizados fueron: PCA y LDA.

3.5. Entrenamiento del clasificador

Se consideraron dos clasificadores para el reconocimiento de las señas, una red neuronal multicapa [11] y SVM [8]. En ambos caso es necesario reajustar los valores mediante el escalamiento de los valores. En el presente trabajo se utiliza la estandarización para reescalar las características. El proceso de estandarización se expresa como:

$$x(i)_{std} = \frac{x(i) - \mu_x}{\sigma_x}, \tag{3}$$

donde:

- $x(i)$ es un ejemplo en particular.
- μ_x es el promedio de una columna de características.
- σ_x es la desviación estándar de la columna de características.

El presente trabajo se evalúa usando validación cruzada, porque es uno de los métodos que dan una mejor evaluación del clasificador, usando $k = 4$, debido al tamaño del dataset. Por lo tanto primero se dividen los datos en 70 % para entrenamiento y 30 % para prueba, este 70 % de entrenamiento posteriormente es dividido en k particiones y evaluado con validación cruzada [2].

Los mejores parámetros para la configuración de los clasificadores se encontraron empleando grid search, que consisten en buscar las combinaciones de los diferentes parámetros, y mediante la técnica de validación cruzada, seleccionar la mejor combinación de parámetros.

La entrada de la red neuronal depende del algoritmo de extracción de características, en el caso de los momentos de Hu, la red neuronal tiene como entrada 7 características, y para los otros algoritmos dependerá de la configuración que tengan, porque la salida depende de los parámetros de configuración del algoritmo de extracción de características. La capa de neuronas de salida de la red neuronal para todos los algoritmos será de 21 neuronas, ya que son 21 tipos de señas, que corresponden a las diferentes letras del alfabeto. El número de neuronas en las capas internas depende de los algoritmos y cambia para cada combinación de técnicas.

Para evitar el sobre-entrenamiento de la red se utilizó Dropout, que consiste en bloquear algunas neuronas, para permitir que las otras neuronas aprendan patrones independientes. La función de activación de la última capa de las neuronas tiene la función de activación softmax, debido a la clasificación del tipo multiclase. La función de activación de las demás neuronas es RELU. La función de optimización que se utiliza es Adam, utilizando decay = learning rate/número de épocas, permitiendo a la red converger más rápido, si se ajusta de la manera correcta. La métrica que se utilizan para evaluar el clasificador es exactitud.

Cabe mencionar que se debe configurar de acuerdo al problema el número de capas de neuronas ocultas, así mismo se debe establecer la cantidad de neuronas en cada capa oculta, existen varias formas de configurar la red, pero las que dan mejores resultados es la configuración escalonada, es decir poner 5 neuronas en

la primera capa, 4 en la segunda y así sucesivamente, por lo tanto se utiliza esta configuración en la búsqueda de los parámetros.

Para la máquina se soporte vectorial (SVM) se utilizó un kernel de función de base radial (RBF). Se empleó el enfoque uno contra el resto para realizar la clasificación multiclase, debido a que este clasificador está diseñado para la clasificación binaria. Los modelos SVM tienen un parámetro de costo C que controla la compensación entre permitir errores de entrenamiento y forzar márgenes estrictos. El parámetro gamma define hasta dónde llega la influencia de un sólo ejemplo de entrenamiento, con valores bajos que significan lejos y valores altos que significan cercanos. Los parámetros a buscar mediante grid search para el clasificador SVM son C y gamma.

4. Implementación

Para la implementación de las configuraciones mencionadas junto con la etapa de preprocesamiento se utilizó Python 2.7, con distribución Anaconda. Las librerías empleadas para la implementación fueron:

- Pandas: Para analisis de datos, guardar y leer archivos CSV.
- Numpy: Para operaciones con arreglos.
- OpenCV: Procesamiento de imagenes.
- Scikit-learn: Para preprocesamiento y clasificación de los datos con SVM.
- Mahotas: Para extraer características con el algoritmo de momentos de Zernike.
- Scikit-image: Para extraer características con el algoritmo de momentos de HOG.
- Keras: Para preprocesamiento y clasificación de los datos con la red neuronal.
- Matplotlib: Para visualizar las imagenes.
- Pickle: Para guardar los clasificadores despues del entrenamiento.

El equipo de cómputo utilizado fue un ordenador portátil con un procesador Intel core i5, 8gb de memoria RAM con sistema operativo macOs Mojave 10.14.

5. Análisis de resultados

Los resultados de todas las configuraciones se muestran en la Tabla 1 y Tabla 2. Como era de esperarse, el tamaño del vector de características influye en el resultado obtenido por los clasificadores. El algoritmo de Hu obtiene la menor exactitud, aún así el entrenamiento es más rápido, debido a que el vector de características es menor. El algoritmo de HOG obtiene mejores resultados que el algoritmo de Hu, sin embargo, el entrenamiento y el cálculo de las características son procedimientos muy lentos, esto se debe a que el número de características es mayor que los otros algoritmos de extracción de características. El algoritmo de Zernike obtiene una mejor exactitud en algunos casos, comparado a HOG y Hu, también tiene menor número de características, y tarda menos tiempo

de entrenamiento que HOG. Los algoritmos de reducción de dimensionalidad ayudan al algoritmo en algunos casos, pero en otros disminuyen la exactitud, en algunas situaciones se obtiene mejores resultados con PCA y en otras situaciones se obtiene mejores resultados con LDA, por este motivo es mejor evaluar la combinación completa y no sólo al clasificador. El algoritmo de clasificación SVM y la red neuronal obtienen resultados parecidos en exactitud de la clasificación, sin embargo, la red neuronal requiere un configuración mayor de sus parámetros y mayor tiempo de entrenamiento, y sólo para obtener una exactitud parecida.

Para validación cruzada se utilizó el número de particiones es 4. Primero se dividió el conjunto en una parte de entrenamiento y una de test. Para los resultados de la Tabla 1 se usó el 30 % de entrenamiento y 70 % de prueba, dando buenos resultados incluso usando más datos de prueba que de entrenamiento.

Se realizó otra prueba con en 70 % de entrenamiento y 30 % de prueba, como se esperaba al aumentar el numero de imágenes de entrenamiento, se obtuvo mejores resultados, como se muestra en la Tabla 2.

Tabla 1. Resultados de la exactitud con 30 % de entrenamiento y 70 % de prueba.

Extracción de características	Preprocesamiento de datos	Reducción de dimensionalidad	Clasificación	Exactitud
Zernike	std scaler		SVM	95.07 %
Zernike	std scaler	PCA	SVM	95.39 %
Zernike	std scaler	LDA	SVM	96.50 %
HU	std scaler		SVM	89.88 %
HU	std scaler	PCA	SVM	85.70 %
HU	std scaler	LDA	SVM	87.87 %
HOG	std scaler		SVM	89.30 %
HOG	std scaler	PCA	SVM	92.90 %
HOG	std scaler	LDA	SVM	67.97 %
Zernike	std scaler		NN	94.37 %
Zernike	std scaler	PCA	NN	94.69 %
Zernike	std scaler	LDA	NN	95.89 %
HU	std scaler		NN	91.15 %
HU	std scaler	PCA	NN	90.0 %
HU	std scaler	LDA	NN	90.3 %
HOG	std scaler		NN	88.5 %
HOG	std scaler	PCA	NN	40.5 %
HOG	std scaler	LDA	NN	71.0 %

El tiempo de procesamiento para la extracción de características fue el indicado:

- Momentos de Hu: 68.64 segundos en crear dataset.
- Histograma de orientación del gradiente: 1319.21 segundos en crear dataset.
- Momentos de Zernike: 134.29 segundos en crear dataset.

Tabla 2. Resultados de la exactitud con 70 % de entrenamiento y 30 % de prueba.

Extracción de características	Preprocesamiento de datos	Reducción de dimensionalidad	Clasificación	Exactitud
Zernike	std scaler		SVM	98.35 %
Zernike	std scaler	PCA	SVM	98.5 %
Zernike	std scaler	LDA	SVM	98.7 %
HU	std scaler		SVM	93.43 %
HU	std scaler	PCA	SVM	92.0 %
HU	std scaler	LDA	SVM	93.06 %
HOG	std scaler		SVM	95.44 %
HOG	std scaler	PCA	SVM	99.1 %
HOG	std scaler	LDA	SVM	93.3 %
Zernike	std scaler		NN	98.04 %
Zernike	std scaler	PCA	NN	97.35 %
Zernike	std scaler	LDA	NN	97.77 %
HU	std scaler		NN	92.75 %
HU	std scaler	PCA	NN	93.59 %
HU	std scaler	LDA	NN	93.17 %
HOG	std scaler		NN	94.97 %
HOG	std scaler	PCA	NN	81.37 %
HOG	std scaler	LDA	NN	91.48 %

Analizando los resultados del clasificador en conjunto con los diferentes algoritmos de clasificación y los algoritmos de reducción de dimensionalidad, se obtiene que el clasificador SVM obtiene mejores resultados que la red neuronal, incluso en un tiempo menor de entrenamiento.

La red neuronal requiere una mayor cantidad de parámetros a configurar para una mayor exactitud en el problema específico, por lo tanto requiere una mayor cantidad de combinaciones a probar, esto produce que la cantidad de tiempo requerido para entrenar una red neuronal sea mayor que el tiempo que requiere el entrenamiento de una máquina de soporte vectorial.

La combinación del algoritmo de reducción de dimensionalidad LDA junto con el algoritmo de extracción de características de los momentos de Zernike, y el clasificador SVM aumentó la exactitud aún con 30 % de los datos como entrenamiento.

Otra combinación buena es el algoritmo de extracción de características HOG junto al algoritmo de reducción de dimensionalidad PCA y el algoritmo de clasificación SVM, obtuvieron una mejor exactitud que las demás combinaciones, pero a un costo de entrenamiento y memoria mayor a las demás combinaciones de algoritmos.

6. Conclusiones

En esta investigación se propuso un sistema de reconocimiento de las señas estáticas del LSM basado en imágenes a color. Se generó una base de conocimiento (dataset) el cual consiste de 21 señas con 300 imágenes cada una, dando un total de 6300 imágenes en las cuales están representadas las señas estáticas.

Se muestra la comparación de dos clasificadores: redes neuronales y SVM junto con una base de datos que consta de 6300 imágenes en las cuales se consideraron variaciones de escala, rotación y traslación.

Se consideraron varios conjuntos de características obtenidas: Momentos de Hu, momentos de Zernike e Histogramas de orientación del gradiente. Así mismo, se consideraron LDA y PCA.

La tasa de reconocimiento para las imágenes estáticas en las condiciones descritas exhibieron una exactitud del 98.7 % considerando los momentos de Zernike y LDA para SVM con $gamma = 0{,}05$ y $C = 19$.

Los pasos para resolver la detección del alfabeto de lenguaje de señas son dependientes uno del otro. Por lo tanto, si un paso no se realiza correctamente puede afectar el rendimiento de los demás pasos. La segmentación basada en el color resulta aceptable, ya que remueve gran parte del fondo de una manera muy eficiente, sin embargo, cuando en el fondo se encuentran objetos de color parecidos a la piel, surgen problemas, para solucionar esto, se debe ajustar el rango de valor para la detección del color o usar alguna manera de diferenciar el fondo.

Se pretende que un un futuro se reconozca todo el abecedario del lenguaje de señas mexicano incluyendo los números, esto implica que se debe implementar el seguimiento de la mano y el reconocer los signos dinámicos no considerados en esta investigación.

Referencias

1. Galicia, R., Carranza, O., Jiménez, E., Rivera, G.: Mexican sign language recognition using movement sensor. In: Industrial Electronics (ISIE), 2015 IEEE 24th International Symposium on. pp. 573–578. IEEE (2015)
2. Hackeling, G.: Mastering Machine Learning with scikit-learn. Packt Publishing Ltd (2017)
3. Ibarra, E.: Generación de dataset para problema de visión computarizada (2017), `https://medium.com/inteligencia-artificial-itesm-cq/generaci%C3%B3n-de-dataset-para-problema-de-visi%C3%B3n-computarizada-a90c77a0dc9a`
4. Joshi, G., Gaur, A., et al.: Interpretation of indian sign language using optimal hog feature vector. In: International Conference on Advances in Computing and Data Sciences. pp. 65–73. Springer (2018)
5. Montaño, L.A.F., Rodríguez-Aguilar, R.M.: Automatic translation system from mexican sign language to text. Advances in Computational Linguistics p. 53 (2011)
6. Otiniano-Rodríguez, K., Cámara-Chávez, G., Menotti, D.: Hu and zernike moments for sign language recognition. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV). p. 1.

The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp) (2012)
7. Paris, S., Kornprobst, P., Tumblin, J., Durand, F., et al.: Bilateral filtering: Theory and applications. Foundations and Trends® in Computer Graphics and Vision 4(1), 1–73 (2009)
8. Patil, A., Subbaraman, S.: A review on vision based hand gesture recognition approach using support vector machines. Journal of Electronicsl and Communication Engineering (2012)
9. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review vol. 43(1), 1–54 (2015)
10. Shaik, K.B., Ganesan, P., Kalist, V., Sathish, B., Jenitha, J.M.M.: Comparative study of skin color detection and segmentation in hsv and ycbcr color space. Procedia Computer Science 57, 41–48 (2015)
11. Solís, F., Martínez, D., Espinoza, O.: Automatic mexican sign language recognition using normalized moments and artificial neural networks. Engineering 8(10), 733 (2016)
12. Sosa-Jiménez, C.O., Ríos-Figueroa, H.V., Rechy-Ramírez, E.J., Marin-Hernandez, A., González-Cosío, A.L.S.: Real-time mexican sign language recognition. In: Power, Electronics and Computing (ROPEC), 2017 IEEE International Autumn Meeting on. pp. 1–6. IEEE (2017)

ISSN 1870-4069

Reconstrucción tridimensional de objetos mediante el uso de luz estructurada

Ignacio Martínez Rojas[1], Mayra Pérez Robledo[2], Francisco Javier Cuevas de la Rosa[3], Pedro Arguijo[1]

[1] Tecnológico Nacional de México/Instituto Tecnológico Superior de Misantla, Misantla, Veracruz, México
[2] Universidad Iberoamericana, León, Guanajuato, México
[3] Centro de Investigaciones en Óptica, A.C, León, Guanajuato, México
grecoromano940@gmail.com, Maypr@outlook.com, fjcuevas@cio.mx, pedro_arguijo@excite.com

Resumen. En este trabajo se propone una metodología mediante el uso de técnicas basadas en proyección de luz estructurada, permitiendo la medición de puntos 3D pertenecientes al objeto bajo prueba. El sistema metrológico se calibra a partir de un conjunto de imágenes que presentan un patrón de desplazamiento. El cálculo de profundidad se realiza utilizando un ajuste por mínimo cuadrados en el que se calibra por medio de una serie de imágenes referentes a planos. Siendo los primeros dos los correspondientes a anchura y altura, el tercero es el que determinara la profundidad.

Palabras clave: reconstrucción tridimensional, mínimos cuadrados, luz estructurada, cambio de fase, flood fill.

Three-Dimensional Reconstruction of Objects through the Use of Structured Light

Abstract. In this work a methodology is proposed through the use of techniques based on structured light projection, allowing the measurement of 3D points belonging to the object under test. The metrological system is calibrated from a set of images that have a displacement pattern. The depth calculation is carried out using a minimum square adjustment in which it is calibrated by means of a series of images referring to planes. The first two being those corresponding to width and height, the third is the one that determines the depth.

Keywords: three-dimensional reconstruction, least squares, structured light, phase shifting, flood fill.

1. Introducción

Actualmente, la metrología óptica es uno de los campos de investigación por demás activos debido a las diversas aplicaciones industriales y científicas que tiene, por

ejemplo, la reconstrucción 3D de objetos, la medición y caracterización de deformación de superficies, entre otras.

La proyección de luz estructurada, en la metrología óptica, permite la captura de valores procedentes de imágenes de forma discreta, la cual proporciona información cuantitativa del objeto bajo prueba. A partir de este conjunto de valores se generan un modelo matricial que contiene implícita información de forma, tamaño, volumen, etc.

En el ámbito de la reconstrucción tridimensional se realiza el uso de proyección de líneas o de una nube de puntos láser [1, 2], lo cual permite realizar la medición automática del área seleccionada. En caso de la proyección de puntos se capturan miles de puntos por segundo [3]. Se han reportado trabajos que realizan proyección y detección de puntos para obtener trayectorias, como es el caso del movimiento de un objeto, una mano o una pierna permitiendo determinar la dirección en la que se desplaza [4], estos tipos de medias con respecto al desplazamiento, de manera general, se realizan en una escena con cierta profundidad la cual puede ser medida [5-7].

Por otro lado, las técnicas basadas en el desplazamiento de fase (Phase Shifting) se han estudiado a profundidad, realizando la replicación de objetos de interés o la medición de estos a partir de un patrón base como control de calidad industrial [8]. Existen variantes en cuanto a las reconstrucciones tridimensionales basadas en proyección de franjas como lo es el uso de técnicas de tres, cuatro pasos o incluso mediante la transformada de Fourier donde esta técnica proporciona información simultánea de alturas para todos los puntos de la superficie de un volumen observado [9]. Una venta clara de este tipo de técnicas es la gran calidad en cuanto a los detalles superficiales de un objeto por lo que una implementación usual es la metrología facial para aplicaciones biomédicas [10].

Una característica del uso de este tipo de técnicas es la reconstrucción tridimensional de objetos físicos en un entorno computarizado de manera completa en alta resolución, de forma no invasiva y mediante el desplazamiento de una secuencia de patrones consecutivos. En este trabajo se utiliza tanto la proyección de luz estructurada junto con un ajuste polinómico para obtener la profundidad de un objeto bajo prueba y realizar su reconstrucción tridimensional. Se muestran resultados de ambos casos, el primero basado en Proyección de Puntos y el segundo mediante técnicas de desplazamiento de Fase (Phase Shifting), se realiza una comparación respecto a la medición de error, así mismo se determinar cuál de estos métodos requiere menor tiempo de procesamiento. En la sección 2 se describe la calibración de la cámara para el método de proyección de nube de puntos utilizando ajuste por mínimos cuadrados. El desenvolvimiento de la fase envuelta por proyección de franjas se analiza en la sección 3. La implementación de ambos métodos se describe en la sección 4. En la sección 5 se muestran los resultados finalizando las conclusiones del presente trabajo.

2. Teoría

2.1. Calibración de la cámara

En todo proceso de reconstrucción tridimensional es imprescindible, para la obtención cuantitativa los datos, realizar un proceso de calibración. La calibración de la cámara se realiza proyectando un patrón de luz estructurada (puntos) sobre un plano de referencia el cual se desplaza sobre un arreglo optomectatrónico a razón de n

centímetros. En la Fig. 1 se muestra dicho arreglo que consta de un proyector, una cámara y un plano de referencia. Cada imagen adquirida representa un desplazamiento en el eje *Z*, *n* desplazamientos representar la profundidad en la escena, iniciando en Z_0 y concluyendo en Z_n.

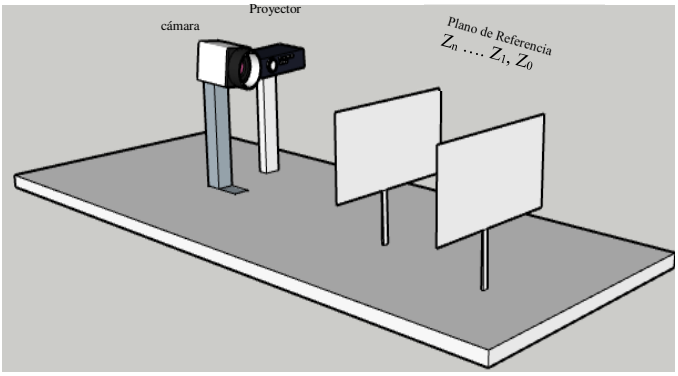


Fig. 1. Arreglo Optomecatrónico.

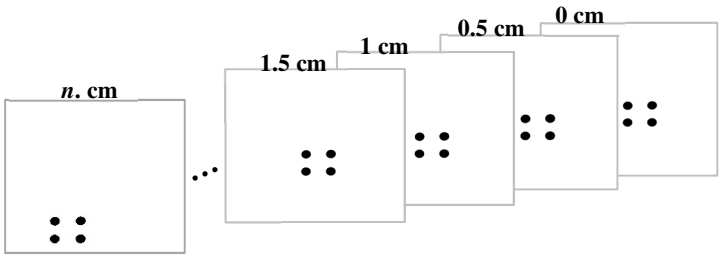


Fig. 2. Representación de un patrón de desplazamiento correspondientes en valores de 0 a n cm a razón de 0.5 cm.

La Fig. 2 visualiza estos **n** desplazamientos del plano de referencia, se puede apreciar un cambio de posición del patrón de luz proyectado.

Se utiliza el algoritmo de Otsu para la umbralización de cada una de las imágenes [11]. En las imágenes segmentadas se establece un orden de agrupamiento que permite diferenciar cada uno de los puntos de los demás y a su vez asociarlo en las demás imágenes pertenecientes a la calibración, esto se logra utilizando Flood Fill [12] con recursividad de cuatro pixeles. Con Flood Fill se localizan los centroides correspondientes a los puntos. Cada uno de los valores *xy* donde *x* representa una coordenada del centroide y *z* indica el desplazamiento en centímetros sobre el eje.

Para encontrar la relación entre las coordenadas de los centroides y los desplazamientos se ajustan los datos a una curva de calibración empleando regresión por mínimos cuadrados [13].

En el proceso de calibración de los ejes *x* e *y* se hace uso de una cuadrícula de puntos colocada sobre el plano de referencia, cada punto se localiza a una distancia de un centímetro de sus homólogos verticales y horizontales [14], tal como se muestra en Fig. 3.

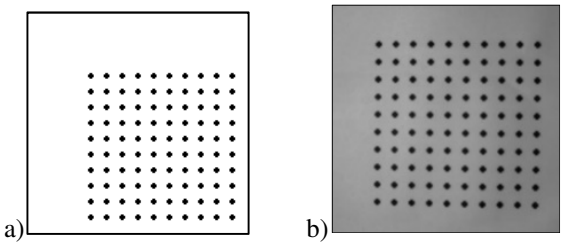


Fig. 3. a) Cuadricula utilizada para calibración, b) Imagen capturada del plano *Z* donde la cuadricula superpuesta sobre el plano de referencia.

Para realizar una calibración con mayor precisión se subdivide el mallado en cuadrantes con una distancia constante de 1 cm en vertical y horizontal respecto de cada punto. Si los puntos que se proyectan posteriormente se localizan dentro del algún cuadrante a este se le asigna un valor de posición en *x* e *y* que corresponde con los cuatro puntos que determinan cada cuadrante. En la Fig. 4 se muestra el primer cuadrante y la distribución de distancias entre los puntos, marcando en verde los puntos de referencia para calibrar si un centroide se encuentra o no dentro del cuadrante. Existe una correlación de la imagen de calibración con su homóloga obteniendo una coincidencia óptima respecto una de otra [15].

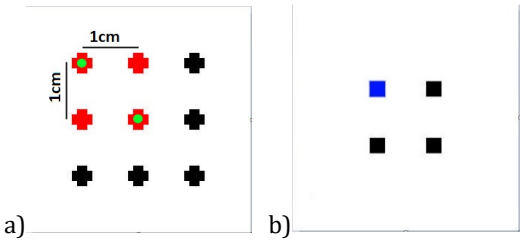


Fig. 4. a) Patrón con puntos de calibración, cuadrante no. 1, b) Patrón con punto en tonalidad azul a localizar mediante de la imagen de calibración.

De los valores obtenidos de la Fig. 4 se obtuvo la Tabla 1. Se puede observar que si un centroide se localiza dentro de uno de los cuadrantes se obtienen medidas reales de este por medio de los valores referencia previamente calibrados para *x* e *y* en dicho cuadrante. Los valores obtenidos de la calibración de *x* e *y* se almacenan (.DAT) en una serie de vectores al igual que las posiciones de cada cuadrante.

Tabla 1. Resultados de calibración de las imágenes.

X, Y	Coordenada X Calibrada	Coordenada Y Calibrada	Valor esperado X	Valor esperado Y	Error Absoluto	Error porcentual
Punto 1	0.5	0.5	0.5	0.5	0	0%
Punto 2	0.5	1.5	0.5	1.5	0	0%
Punto 3	1.5	0.5	1.5	0.5	0	0%
Punto 4	1.5	1.5	1.5	1.5	0	0%

2.2. Regresión por Mínimos Cuadrados

A partir de una serie de valores correspondientes a x y z, donde x representa una coordenada del centroide y z representa el valor del desplazamiento n, se puede obtener un polinomio de grado m, el cual se expresa como ecuación (1):

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m + e. \tag{1}$$

El ajuste por mínimos cuadrados es una técnica ampliamente conocida la cual se obtiene mediante una sumativa de derivadas que se igualan a cero y se reordenan [16] ecuación (2):

$$\begin{aligned}
&(n)a_0 + (\sum x_i)a_1 + (\sum x_i^2)a_2 + \ldots + (\sum x_i^2)a_n = \sum y_i \\
&(\sum x_i)a_0 + (\sum x_i^2)a_1 + (\sum x_i^3)a_2 + \ldots + (\sum x_i^n)a_n = \sum x_i y_i \\
&(\sum x_i^2)a_0 + (\sum x_i^3)a_1 + (\sum x_i^4)a_2 + \ldots + (\sum x_i^n)a_n = \sum x_i^n y_i.
\end{aligned} \tag{2}$$

El uso de la eliminación gaussiana permite la solución del conjunto de forma matricial [17]. La resolución de la eliminación gaussiana se obtiene en valores de a_0, a_1, a_2, …, a_n, permitiendo generar la función de sustitución en la ecuación (3).

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m. \tag{3}$$

Tabla 2. Errores relativos por desplazamiento, fila gris distancias *z* intermedias consideradas como datos de prueba y promedio general de orden de polinomio, los valores en azul indican un error mínimo en relación distancia-grado, el valor en rojo representa el menor promedio óptimo.

Orden.	Desplazamiento en centímetros de Z.									Promedios.
	0	0.5	1	1.5	2	2.5	3	3.5	4	
1	0	15.87	0.97	3.77	5.97	1.37	0.94	0.82	1.96	3.518 %
2	0.02	13.73	1.94	0.9	1.59	1.88	0.89	0.74	2.20	2.654 %
3	0.12	14.36	2.35	1.26	2.20	1.77	1.18	0.55	0.22	2.667 %

En estas fórmulas se sustituyen en los valores de posición de los Centroides de las imágenes. Realizando la sustitución de valores *x* e *y* en la función se obtiene un valor en *y* procediendo a encontrar el error absoluto.

La Tabla 2 indica las distancias *Z* en cm del primer punto y el orden de polinomio que se utilizó, así como resultados los errores relativos por grado, mediante el promedio de estos puntos el mejor polinomio es el de segundo orden.

3. Desplazamiento y desenvolvimiento de fase

La calibración de la cámara en el método de cambio de fase se implementa de manera inicial con la calibraron los ejes *xy* usando una malla de puntos, Fig, 5 malla de calibrado de la figura obtenida. Obteniendo valores de calibración a partir de una regla

de tres, haciendo uso de la distancia real entre los puntos y la distancia de estos en pixeles. A partir de la calibración determino que cada pixel equivale a 0.0147 centímetros.

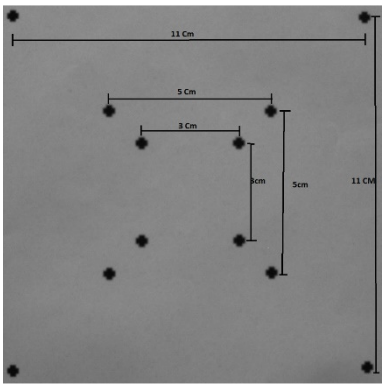


Fig. 5. Imagen de Calibración de ejes *xy.*

En el área de metrología óptica se proyecta una serie de patrones de franjas periódicas desplazadas (phase-shifting) sobre el objeto bajo prueba, las cuales ofrecen información cuantitativa de la profundidad o deformaciones topográficas de la superficie del objeto [18]. La información se obtiene de la deformación que sufren las franjas al proyectarse sobre el objeto. Existen diversos algoritmos de corrimiento de fase entre los que destacan los de tres pasos y cuatro pasos [19]. El método implementado en este trabajo es el de cuatro pasos. Las franjas que se proyectan se generan mediante una función sinusoidal que presenta un desplazamiento de fase [20] en la ecuación (4):

$$I(x, y) = A(x, y) + B(x, y)\cos\left(\phi(x, y) + \delta\right), \tag{4}$$

donde *A(x, y)* indica la intensidad de fondo en la imagen, *B(x, y)* es un factor que determina la visibilidad de las franjas observadas, *f(x, y)* es la información a reconstruir y δ es el valor que señala los desplazamientos en la fase de las franjas que iluminan el objeto.

La topografía del objeto es directamente proporcional al término de fase *f(x, y)* de (4). Existen diversos para encontrar *f(x, y)*, por ejemplo, el método de Takeda que se basa en la transformada Fourier [21,22]. En nuestro caso se genera un sistema de ecuaciones que permite hallar las tres variables desconocidas: *A(x, y), B(x, y)* y *f(x, y)*. Como se mencionó se utiliza el método de cuatro pasos, donde los desplazamientos tienen valores de *0, π/2, π y 3π/2*, los cuales se expresan en la ecuación (5) como:

$$\begin{aligned} I_1(x, y) &= A(x, y) + B(x, y)\cos\left(\phi(x, y)\right), \\ I_2(x, y) &= A(x, y) + B(x, y)\cos\left(\phi(x, y) + \pi/2\right), \\ I_3(x, y) &= A(x, y) + B(x, y)\cos\left(\phi(x, y) + \pi\right), \\ I_4(x, y) &= A(x, y) + B(x, y)\cos\left(\phi(x, y) + 3\pi/2\right). \end{aligned} \tag{5}$$

La obtención de la fase se representa en la ecuación (6):

$$\varphi(x, y) = \arctan \frac{I_4(x, y) - I_2(x, y)}{I_1(x, y) - I_3(x, y)}. \quad (6)$$

Se toma en cuenta que los valores obtenidos en la fase están entre los rangos desde $-\pi$ a π. El resultado obtenido con la función arco tangente se denomina como fase envuelta y la cual tiene discontinuidades.

Como se mencionó, las discontinuidades de 2π observadas no son parte de la fase real medida; en cambio, son valores introducidos por la función arco tangente. Por lo tanto, se utiliza un proceso de desenvolvimiento para eliminar estas discontinuidades no deseadas y obtener una fase continua [23]. El método de desenvolvimiento de fase utilizado es la integración en línea de las diferencias de fase envueltas descrito por Itoh [24]:

$$\varphi(x) = \varphi(x-1) + W[\varphi_w(x) + \varphi_w(x-1)], \quad (7)$$

$$\varphi(x) = \varphi_w(0) + \sum_{n=0}^{x-1} W\{D[\varphi_w(n)]\}. \quad (8)$$

Las ecuaciones (7) y (8) describen el método de Itoh [24] como la estimación de la fase desenvuelta al integrar las diferencias envueltas de la fase envuelta.

4. Implementación

La implementación de los dos métodos que se describen se realiza según el orden del presente en la Fig. 6.

4.1 Proyección de puntos y desplazamiento de fases

Tanto el método de proyección de puntos como el método de desplazamiento de fase se implementan para la reconstrucción tridimensional del objeto que se muestra en la Fig. 7a. Tal como se puede apreciar el objeto tiene forma sinusoidal, dicho objeto fue generado en una impresora 3D con Ácido poliláctico (PLA) el cual permite una superficie regular de características lambertianas siendo ideal al reflejar la radiación incidente de manera uniforme en todas las direcciones.

El uso de proyección de luz se llevó acabo en un ambiente controlado con iluminación exterior mínima permitiendo un menor índice de contaminación lumínica que generase ruido en forma espectral de luz. Una vez calibrada la cámara, tal como se describió anteriormente, se obtiene la proyección de los puntos sobre el objeto, ver Fig. 7b.

Los datos para la reconstrucción del objeto, después de ser procesada la imagen, se muestran en la Fig. 8.

El método de integración de fase, desarrollado por Itho, se aplica a las imágenes con desplazamiento de fase, ver Fig. 9.

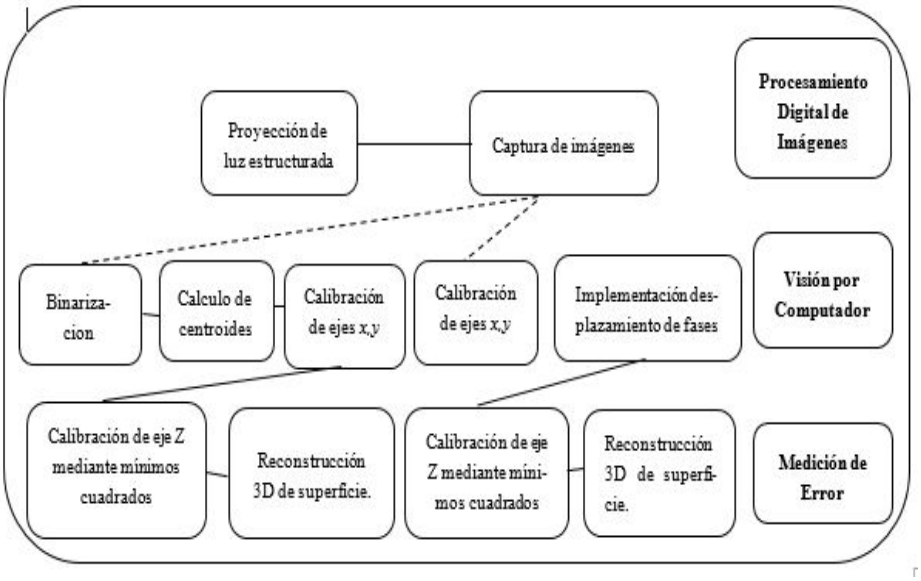


Fig. 6. Metodología utilizada para la calibración del Sistema de reconstrucción 3D.

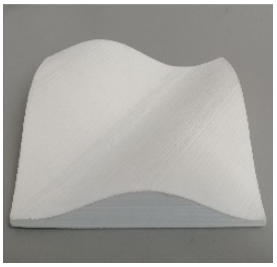
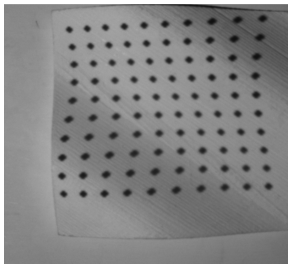
a)  b) 

Fig. 7. a) Objeto sinusoidal a reconstruir. b) Captura de proyección de puntos sobre el objeto.

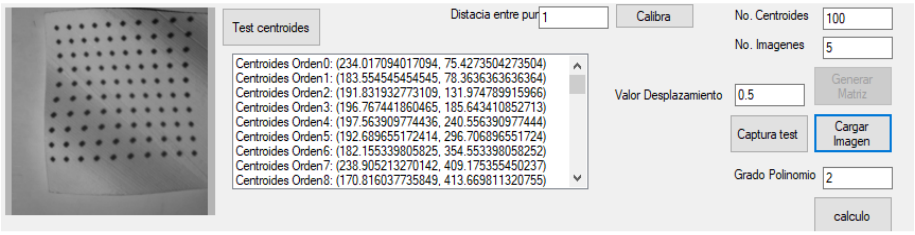


Fig. 8. Cálculo de valores del patrón de puntos proyectados de la Fig. 7a.

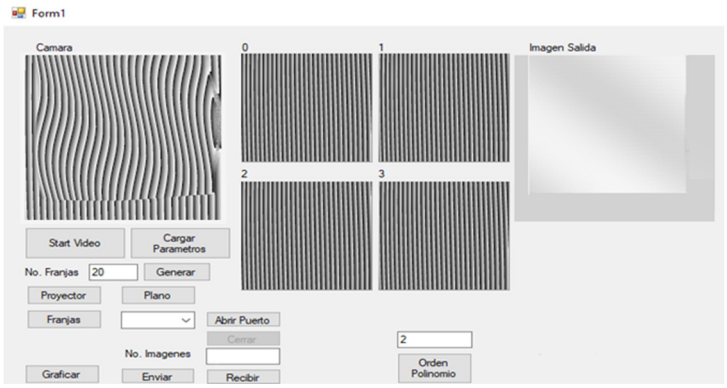


Fig. 9. Interfaz de la proyección de franjas sobre el objeto.

5. Resultados

5.1. Proyección de puntos

A partir de los datos obtenidos de la Fig. 8 se generó la representación tridimensional del objeto sobre el que se proyectaron 100 puntos, la profundidad, z, se obtuvo con un polinomio de grado 2 siendo el mejor ajuste por mínimos cuadrados. En la Tabla 3 se realiza una comparativa de los primeros 16 valores de la profundidad en centímetros y los valores de la función original correspondientes a las mismas posiciones, también se muestra el error existente entre ambos, representado como error absoluto. Así mismo, en la Tabla 4 se muestran los valores obtenidos de primeros 10 puntos en los ejes x, y.

Tabla 3. Comparación del error de valores con respecto a los valores reales y los obtenidos.

# Punto	V_Real	V_Obtenido	V_ Absoluto	ERM
1	0.92290132	0.89377323	0.02912809	3.20%
2	0.12138852	0.12078633	0.00060219	0.49%
3	0.21499043	0.20881489	0.00617554	2.91%
4	1.42394646	1.41481196	0.0091345	0.64%
5	1.67704467	1.58492765	0.09211702	5.64%
6	0.78158729	0.73474917	0.04683812	6.17%
7	0.05043044	0.04596907	0.00446136	9.25%
8	0.38299994	0.38064532	0.00235462	0.61%
9	1.99908386	1.83346247	0.16562139	8.64%
10	1.56997916	1.48824021	0.08173895	5.34%
11	0.64170195	0.58469729	0.05700466	9.29%
12	0.00894218	0.00820267	0.00073951	8.62%
13	1.51099759	1.46274964	0.04824795	3.24%
14	1.99509015	1.92887567	0.06621448	3.37%
15	1.45349709	1.37495217	0.07854492	5.55%
16	0.50556361	0.48784653	0.01771709	3.56%
Promedio				4.78%

Tabla 4. Valores de la calibración de los ejes *X* e *Y* en milímetros de los diez primeros puntos.

# Punto	Eje X	Eje y
1	1.37333258	7.32546299
2	1.55825212	5.69311618
3	1.47549931	4.00627923
4	1.14250375	2.24526106
5	2.91596252	7.39986265
6	3.1892382	5.74901527
7	3.32874351	4.09687728
8	3.18053391	2.37082333
9	4.56097929	7.47089811
10	4.78258029	5.79401085
11	5.04118147	4.14067282
12	5.13320665	2.46587429
13	6.40510438	7.57526862
14	6.45364506	5.83848352
15	6.69266647	4.1577533
16	6.92059466	2.5049999

La Fig. 10 muestra la reconstrucción tridimensional del objeto a partir del patrón de puntos mostrado en la Fig. 7b.

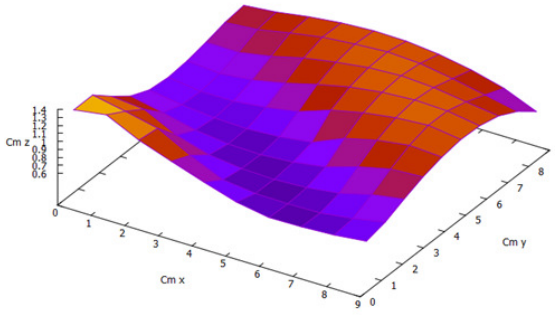


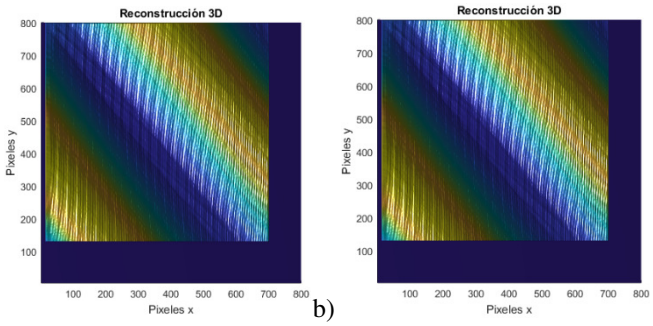
Fig. 10: Reconstrucción tridimensional a partir de patrón de puntos de la Fig. 7b.

5.2. Desplazamiento de fase

Se empleó una frecuencia de 60 franjas de acuerdo a la ecuación (4).

Se realizó una calibración polinomial mediante el agrupamiento de los planos como referencia y el valor de cada pixel para realizar un ajuste polinomial pixel a pixel, en una imagen de 800 x 800 se obtiene un polinomio por pixel que definen el eje *z* como la profundidad del objeto en centímetros.

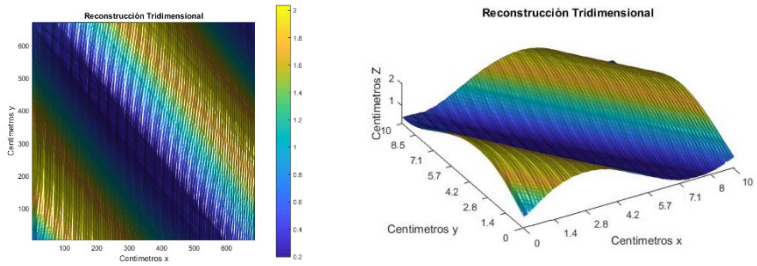
La figura obtenida mediante la reconstrucción polinómica se observa en la Fig.11a donde existe cierto ruido, se rectifica mediante un filtro pasa bajos obsérvese Fig. 11b,



a) b)

Fig. 11. Filtrado de Ruido. a) Figura con ruido, b) Figura sin ruido.

Se re-escaló la figura utilizando los valores de equivalencia de pixel y obteniendo una conversión espacial de *xy* en centímetros, la profundidad en *z* a su vez obtiene un máximo local de 2 cm y un mínimo de 0.4 cm. Obsérvese Fig. 12. Las medidas correspondientes al objeto original oscilan entre un máximo local de 2 cm y un mínimo de 0.2 cm.



a) b)

Fig. 12. Figura 3D reconstruida a partir del objeto original con las medias obtenidas y escalada a tamaño real. a) Visa superior de la figura. b) vista 3D del objeto.

La Tabla 5 muestra la diferencia entre la figura real y la reconstrucción encontrando el error absoluto de la reconstrucción a partir de una matriz de 16 puntos tomados de la parte central de la figura reconstruida.

Tabla 5. Comparación del error de valores reales y los obtenidos de la reconstrucción tridimensional a partir del método de desplazamiento de fase.

# Punto	V_real	V_obtenido	V_absoluto	ERM
1	1.9733	1.943	0.0303	1.54%
2	1.0059	1.019	0.0131	1.29%
3	0.0295	0.028	0.0015	5.21%
4	0.5375	0.5157	0.0218	4.13%
5	1.3328	1.264	0.0688	5.29%
6	0.1617	0.1573	0.0044	2.75%
7	0.2729	0.2548	0.0181	6.85%
8	1.4962	1.403	0.0932	6.42%

# Punto	V_real	V_obtenido	V_absoluto	ERM
9	0.3863	0.3605	0.0258	6.90%
10	0.0883	0.0865	0.0018	2.05%
11	1.1848	1.1392	0.0456	3.92%
12	1.9987	1.868	0.1307	6.76%
13	0.0126	0.012	0.0006	4.87%
14	0.8531	0.7952	0.0579	7.02%
15	1.9268	1.835	0.0918	4.88%
16	1.583	1.5205	0.0625	4.02%
Promedio				4.62%

6. Conclusiones

Las técnicas de metrología óptica basadas en la proyección de patrones de luz estructurada, muestran resultados factibles en la obtención de las medidas superficiales de objetos permitiendo una reconstrucción basada en datos parciales del objeto de interés como es el caso de una proyección de puntos, obteniendo a pesar de la dispersión de los puntos sobre el objeto una reconstrucción moderada de su superficie, en el caso del método basado en el desplazamiento de fase la reconstrucción es adecuada considerando que tiene un conjunto de valores más amplio de toda la superficie del objeto.

De manera objetiva el método basado en el desplazamiento de fase es más adecuado en cuanto a una reconstrucción integra de un objeto pudiendo obtener aquellos valores que son omitido en el primer método porque se encuentran intermedios en la proyección de puntos.

De acuerdo con el promedio de los errores porcentuales en ambos casos el método basado en desplazamiento de fase es el más adecuado en cuanto a un menor error de medición, por otro lado, en la búsqueda de obtener un método más rápido en cuanto al procesamiento para la generación de resultados el basado en la proyección de puntos es el más adecuado debido a que una vez calibrado el sistema solo se requiere una imagen para obtener una medida acorde a la superficie de un objeto, mientras que en el caso del método de desplazamiento de fase se requieren 3, 4 o más, imágenes a procesar según sea el caso, realizando toda la serie de pasos mencionados en la sección 3 por cada grupo de imágenes que se capturen

Agradecimientos. Ignacio Martínez Rojas agradece al CONACYT por la beca de movilidad recibida para la realización del presente trabajo y al Centro de Investigaciones en Óptica, A.C. por el apoyo brindado durante el periodo de estancias realizado. Ignacio Martínez Rojas es estudiante del posgrado de Maestría en sistemas Computacionales Instituto Tecnológico Superior de Misantla, Misantla Veracruz México.

Referencias

1. Leal, E., Leal, N., Sánchez, G.: Estimación de Normales y Reducción de Datos Atípicos en Nubes de Puntos Tridimensionales. Información Tecnológica 25(2), 39–46 (2014)

2. Leal-Narváez, E.A., Branch, J.W., Ortega Lobo, O.: Estimación de curvaturas y direcciones principales en nube de puntos no organizados. Dyna 74(153) (2007)
3. De Mesa Gisbert, A., Regot Marimón, J., Núñez Andrés, M., Buill Pozuelo, F.: Métodos y procesos para el levantamiento de reconstrucción tridimensional gráfica de elementos del patrimonio cultural. La iglesia de Sant Sever de Barcelona (2009)
4. González, J.A., Castillo, H.M.: Análisis comparativo de algoritmos de reconstrucción 3D basados en visión para la obtención de trayectorias. En: XVI Congreso Internacional anual de la Sociedad Mexicana de Ingeniería Mecánica (SOMIM), México (2010)
5. Caro, J.L.: Fotogrametría y modelado 3D: un caso práctico para la difusión del patrimonio y su promoción turística (2012)
6. Salcedo, A.F.C., Martínez, A.B., Salazar, E.A.Q.: Procesamiento de nubes de puntos por medio de la librería PCL. Scientia et Technica 2(52), 136–142 (2012)
7. Roca, J., Marambio, A., Moreno, M.: Modelos digitales de nubes de puntos de la Habana vieja, Cuba. En: Congreso internacional ciudad y territorio virtual, pp. 1–7 (2006)
8. Martínez, A., Rayas, J.A., Flores, M., Rodríguez-Vera, R., Donato Aguayo, D.: Técnicas ópticas para el contorneo de superficies tridimensionales. Revista mexicana de física 51(4), 431–436 (2005)
9. Oliveros, D.M.Q., Mejía, R.V., Mendoza, E.Y.: Reconstrucción tridimensional de deformaciones mecánicas superficiales en platinas, basada en la proyección de franjas. Revista Facultad de Ingeniería (59), pp. 169–180 (2012)
10. Gómez, A.L.G., Fonseca, J.E.M., Téllez, J.L.: Proyección de franjas en metrología óptica facial. INGE CUC 8(1), 191–206 (2012)
11. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, man, and cybernetics 9(1), 62–66 (1979)
12. Levoy, M.: Area flooding algorithms. Two-Dimensional Computer Animation. Course Notes 9 for SIGGRAPH, 82 (1981)
13. Chapra, S. Canalé, R.: Métodos Numéricos Para Ingenieros. 5ta edición, McGraw-Hill (2007)
14. Sosa-Balderas, J.V.: Proyección de luz estructurada aplicada a la biometría. CIO repositorio: https://cio.repositorioinstitucional.mx/jspui/handle/1002/88 (2018)
15. Ackermann, F.: Digital image correlation: performance and potential application in photogrammetry. The Photogrammetric Record 11(64), 429–439 (1984)
16. Scheid, F., Di Constanzo Lorencez, R.E., Nagore Cázares, G.: Métodos numéricos (1991)
17. Nieves Hurtado, A., Domínguez Sánchez, F.: Métodos numéricos aplicados a la ingeniería (2002)
18. Malacara, D. (ed.): Optical Shop Testing, 3rd ed., John Wiley and Sons (2007)
19. Vargas, J., Quiroga, J.A., Belenguer, T., Servín, M., Estrada, J.C.: Two-step self-tuning phase-shifting interferometry. Optics express 19(2), 638–648 (2011)
20. Quan, C., He, X.Y., Wang, C.F., Tay, C.J., Shang, H.M.: Comunicaciones Ópticas (Shape measurement of small objects using LCD fringe projection with phase shifting. Optics Communications), 189(1-3), 21–29 (2001)
21. Takeda, M., Mutoh, K.: Fourier transform profilometry for the automatic measurement of 3-D object shapes. Applied optics 22(24), 3977–3982 (1983)
22. Takeda, M., Ina, H., Kobayashi, S.: Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. JosA 72(1), 156–160 (1982)
23. Servin, M., Quiroga, J.A., Padilla, M.: Fringe pattern analysis for optical metrology: theory, algorithms, and applications. John Wiley & Sons (2014)
24. Itoh, K.: Analysis of the phase unwrapping algorithm. Applied optics 21(14), 2470–2474 (1982)

ISSN 1870-4069

Actividades colaborativas en sistemas distribuidos construidos con Erlang

Gerardo Carreño-García[1], Manuel Hernández-Gutiérrez[2]

[1] Banco de México, Ciudad de México, México
[2] Universidad Tecnológica de la Mixteca, Huajuapan de León, Oaxaca, México
gcarreno@banxico.org.mx, manuelhg@mixteco.utm.mx

Resumen. Este escrito presenta una formulación gradual de un sistema distribuido para la realización de labores que utilizan varios posibles nodos computacionales. El sistema es analizado a nivel del paradigma de concurrencia de *paso de mensajes* e implementado en el lenguaje de programación funcional y concurrente Erlang. El diseño del sistema resultante se basa en una metodología que comienza con la identificación de una partición de la labor original, continúa con la posible asignación de sublabores a nodos computacionales (aprovechando posibles oportunidades de colaboración inteligente) y, finalmente, por una estructuración de una solución final basada en subsoluciones.

Palabras clave: cómputo distribuido, colaboración, Erlang, paso de mensajes.

Collaborative Activities in Distributed Systems Built in Erlang

Abstract. This paper presents a gradual formulation of a distributed system for the accomplishment of tasks that use several possible computational nodes. The system is analyzed at the paradigm level of concurrent message passing and implemented in Erlang, a functional and concurrent language programming. The design of the resulting system is based on a methodology that begins with the identification of a partition of the original work, continues with the possible assignment from subtasks to computational nodes (taking advantage of possible opportunities for intelligent collaboration) and, finally, by structuring a final solution based on sub-solutions.

Keywords: distributed computing, collaboration, Erlang, message passing.

1. Introducción

Los sistemas distribuidos representan una gran oportunidad para realizar labores complicadas en paralelo o concurrentemente, siendo así que se abre la

puerta al tratamiento de labores que pueden desarrollarse en diversos nodos computacionales tanto por razones de eficiencia como de formulación inherentemente distribuida (como sería el caso de un conjunto de robots que realizan labores de salvamento en regiones geográficamente dispersas). En este artículo presentamos un sistema distribuido que realiza ejecuciones distribuidas de varios programas que incorporan algunas técnicas de subdivisión de una labor en sublabores. Estos programas, que pueden considerarse *agentes*, son implementados en el lenguaje de programación funcional y concurrente Erlang [1,2,7].

Enmarcaremos nuestras formulaciones téoricas en el paradigma de paso de mensajes para cómputo distribuido (particularmente, modelado matemáticamente como un *cálculo de actores*; ver [19]). Para algunas codificaciones se ha utilizado una metodología transformacional de programas funcionales [6,17], aprovechando el aspecto funcional de los programas de Erlang. Para el caso de programación distribuida entre agentes se sigue una filosofía del establecimiento de *protocolos*, *mensajes* y *diálogos* ([2]). Los protocolos indican qué se puede válidamente comunicar entre agentes, los mensajes seguirán estos protocolos y contendrán información relevante, y los diálogos serán sucesiones de mensajes entre diversos emisores y receptores.

Aplicaremos nuestros ejemplos generales a la especificación (descripción) e implementación de un sistema distribuido basado en agentes que, en una parte, gestionan tareas asignadas, y en otra, colaboran entre sí para llevar a cabo algunos objetivos (apoyándonos en algunos puntos presentados en [8] y en [4]. Se considera que una aportación medular de este trabajo es pasar de los diseños teóricos a implementaciones concretas vías programas de Erlang, así logrando que Erlang sea un de laboratorio computacional que hace de puente entre las conceptualizaciones teóricas y las técnicas de programación concurrente. Siendo Erlang por naturaleza *asíncrono*, enfatizaremos qué hacer cuando por alguna razón se requiene sincronía. Debido a las primitivas de concurrencia ya existentes *de facto* en Erlang, manejamos la hipótesis de que éste lenguaje de propósito general es preferible a otros.

Se supondrá cierto conocimiento básico de programación funcional, de teoría de grafos, y las características generales de los problemas de concurrencia, aunque en la parte distribuida con Erlang se entrará prácticamente de lleno (para lo cual ayudaría ver los artículos divulgativos [12,13]). A continuación, aplicaremos nuestros desarrollos parciales a la puesta en marcha de un sistema distribuido que trata con agentes inteligentes como procesos hospedándose en *nodos-e*, en donde la parte de *inteligencia* será colocada en la manera de distribuir las labores asignadas, promoviendo un estilo colaborativo de solución entre los agentes participantes.

2. Erlang: concurrencia y programación funcional

En esta sección presentamos algunas ideas que hacen del lenguaje de programación Erlang una adecuada herramienta de pensamiento aplicado a los sistemas distribuidos. Veremos que el punto de apoyo principal teórico es el de *sistemas*

distribudos con procesos asíncronos y paso de mensajes, la creación de procesos ligeros, y la posibilidad tersa de escalabilidad.

El punto de entrada. Erlang es un lenguaje de programación de propósito general con algunas primitivas directamente relacionadas con la teoría de un modelo distribuido de cómputo basado en paso de mensajes. Estas primitivas pueden en principio realizar tres tareas básicas: generar *procesos* de tipo ligero (ajenos y en convivencia con los procesos del sistema operativo en uso), *enviar mensajes*, y *recibir mensajes*. Los procesos se hospedan en un *nodo-e*, que es un un intérprete del lenguaje Erlang ejecutándose en una computadora.

Erlang tiene un estilo de *programación funcional*, pero a la fecha no se ha logrado que encaje en algún modelo teórico existente funcional y distribuido (aunque un intento digno de citarse es [15], basándose en el *cálculo pi*). De todas formas, con una metodología disciplinada que comience con una adecuada especificación, un desarrollo de programación apegado a esta especificación, y una implementación que satisfaga también la especificación, el problema de conocer e identificar la semántica del programa final se aminora.

La especificación de un problema y su solución en Erlang. Modelar sistemas distribuidos que se apeguen a una especificación de la manera más fiel posible, es algo preferible antes que una directa implementación [9,19]. No obstante, una de las banderas enarboladas desde hace tiempo por la comunidad de programadores funcionales es que frecuentemente los programas funcionales *actúan* como especificaciones [5]. De requerirse que estas especificaciones funcionales se apeguen de manera más directa a una ejecución en cierto modelo computacional, se pueden utilizar herramientas del tipo de *transformación de programas* [16]. Estas transformaciones son preservadoras de las semánticas iniciales, y con ello puede lograrse adecuadas implementaciones en el sentido de eficiencia. Cuando tales técnicas no puedan aplicarse, las herramientas de construcción de modelos (*model checking*, [10,11]) pueden ser oportunamente atraídas.

El núcleo de conceptos de Erlang es invariante a diversa escala. Erlang tiene una propiedad para realizar investigación de sistemas distribuidos: Es *tersamente* escalable y puede ser utilizado como un puente entre teoría y práctica, haciendo el papel de *laboratorio de prueba* (*sandbox*) para percibir tempranamente problemas de implementación, y una vez resueltos, facilitar la construcción de programas solución, para hacer pasar estos programas prácticamente sin cambios al campo industrial. Para ello, se comienza con un sistema de procesos sobre un nodo; posteriormente, en sistemas operativos tales como Linux, es posible abrir varias consolas, en donde cada consola es tratada como un nodo (computadora) independiente. En un último paso, la escalabilidad queda asegurada ya que se podrían ahora, con pequeños cambios, utilizar el sistema sistema distribuido (en apariencia, de "juguete") hecho en una sola computadora sobre varias otras (eso sí, de preferencia con conexiones que se suponen estables). Por lo demás, este artículo sigue la investigación de relacionar agentes con Erlang [18].

3. Vectores para rastreo de cómputo distribuido

Consideramos ahora un problema que requiere para su solución una subdivisión de labores, en subtareas. Si preliminarmente trabajamos con un sistema centralizado, supongamos que requerimos sincronía, monitoreada por un agente en particular, encargado de monitorear qué subtareas se han terminado y cúales no. Para el monitoreo de la realización de subtareas, necesitaremos un concepto auxiliar de *vector de terminación*. Definimos un *vector de terminación de tamaño n* que está puesto en 0s al inicio de toda computación distribuida, con cada entrada i ($1 \leq i \leq n$) indicando el estado de terminación de un proceso (llamado en adelante *trabajador*) i-ésimo. Estos vectores pueden tener un índice de terminación, indicando históricamente quién ha terminado su labor y en qué momento. Además, pueden servir para llevar un rastreo también histórico de qué tan eficaz y eficiente es un trabajador dado. Tales actividades son por el momento asignadas a un proceso especial que llamaremos en adelante `agenteOrd` (transitando de lleno, en un futuro, hacia el concepto de *agente*).

Por ejemplo, en la secuencia de vectores de terminación:

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1_2 & 1_1 & 1_3 \end{bmatrix}$$

se hizo un registro de qué trabajadores terminaron y cuando. Para el caso de una vigilancia parcial, se tendría:

$$\begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1_2 & 1_1 & 0 \end{bmatrix}$$

indicando que todavía se espera la respuesta del trabajador número 3. Cabe notar que, por diseño, es posible que agentes descentralizados estén equipados con estas capacidades de monitoreo, si así fuera necesario.

La función que apoyaría al proceso A (`agenteOrd`) (en posible compañía de un *operador*, que puede ser un humano) a verificar la terminación de los asignamientos dados para el caso de dos trabajadores sería, para una posible implementación, así:

```
procesoA({Tarea,{X,Y}}) ->
{_,V}={_,{X,Y}},
if {X,Y}=={1,1} -> io:format(''Tarea terminada ~n'',[]),
      %notificación al operador
receive
                    {terminada,ProcesoB} -> X=1,
                              procesoA({Tarea,{X,Y}});
                    {terminada, ProcesoC} -> Y=1,
                     procesoA({Tarea,{X,Y}})
                     %otros mensajes a recibir...
end.
```

en donde se supone de cada proceso B y C tiene la capacidad para enviar el mensaje de terminación; es posible con la construcción de Erlang `after` también enviar mensajes al operador de la *no terminación* (hasta ese punto del tiempo)

de la tarea intermitente, para un intervalo dado. La sincronía programada aquí es *flexible*: no significa que los procesos no puedan continuar con otras labores, solo que el `agenteOrd` puede dar diversos usos a sus vectores de terminación. Por ejemplo, si se programa para tal efecto, el `agenteOrd` puede proceder, bajo presión de tiempo, con una siguiente tarea aún con procesos que no hayan reportado terminación.

4. Ejemplo de cómputo en paralelo con apoyo en dos esclavos

En aras de un ejemplo que ponga en práctica el concepto de vector y los de sublabores, y a realizarse en tres computadoras remotas, tendremos una tarea o labor que se asigna a un proceso ubicado en un nodo-e `nodoA` con nombre `oso@tosh`, y otros dos procesos con nodos-e como sigue: `nodoB` con nombre `osina@tosh` y `nodoC` con nombre `osona@tosh`. El programa que se cargará en todos los nodos es nombrado `merge.erl`. Para el `nodoA` tendremos un nodo llamado `oso` y la activación de nodo inicial con `merge:startA()`, quedando en activo un proceso en este nodo-e llamado `agenteOrd`, mismo que es controlado localmente por el usuario `operadorA`, con usuarios `operadorB` y `operadorC` para los demás nodos, con su correspondencia a cada nodo-e B y C .

Realizamos ahora las siguientes acciones para activar nodos-e y procesos participantes: en un nodo llamado `osina` ejecutamos `merge:startTrab(oso@tosh)` y en otro llamado `osona` ejecutamos `merge:startTrab(oso@tosh)`. Por medio del usuario `operadorB` interactúamos con el proceso `trabajador` definido sobre el `nodoB`, ejecutando `trabajador ! disponible`. Similarmente, por medio del usuario `operadorC` y el proceso `trabajador` definido sobre el `nodoC`, ejecutamos `trabajador ! disponible`.

Para este momento, el nodo-e que hospeda al proceso `agenteOrd` ya está preparado para realizar asignaciones de labores. Mostraremos esta parte por medio de una lista que deberá ser dividida en dos componentes, ordenada cada subcomponente aparte, y *cuando ambos subcomponentes estén listos*, las subsoluciones armadas darán una solución en total. Notemos que esto podría verse como una formulación del algoritmo de ordenamiento por fusión (*mergesort*) en la parte de armado total, con un trabajo ya sea intermedio o final de *fusión* o *intermezclado*, pero se deja en libertad a cada nodo-e auxiliar la manera de ordenar su parte: en este ejemplo, localmente se puede utilizar el algoritmo de ordenamiento rápido (*quicksort*), para enfatizar tal libertad.

La siguiente orden local que *agenteOrd* recibe por medio de su usuario `operadorA` es como sigue, dejando las demás partes en modo automático:

```
(oso@tosh)3> agenteOrd !
          {[3,2,4,5,2,1,2,3,7,8,9,10,2,6,8,2,3],comenzar}.
Comenzando cómputo en paralelo...
{[3,2,4,5,2,1,2,3,7,8,9,10,2,6,8,2,3],comenzar}
Trabajadores en activo...
        [{trabajador,osona@tosh},{trabajador,osina@tosh}]
```

```
Trabajador C terminó
Trabajador B terminó
Tareas terminadas
Resultado distribuido obtenido y armado:
        [1,2,2,2,2,2,3,3,3,4,5,6,7,8,8,9,10]
```

Por lo demás, el sistema aquí expresado en el estado actual sigue funcionando y nuevas solicitudes son posibles:

```
(oso@tosh)4> agenteOrd ! {[21,32,12,43,54,65],comenzar}.
Comenzando cómputo en paralelo...
{[21,32,12,43,54,65],comenzar}
Trabajadores en activo...
         [{trabajador,osona@tosh},{trabajador,osina@tosh}]
Trabajador C terminó
Trabajador B terminó
Tareas terminadas
Resultado distribuido obtenido y armado: [12,21,32,43,54,65]
```

En resumen, para solucionar este problema se ilustró con un par de agentes trabajadores. No obstante, con tres procesos trabajadores, y aún con una subdivisión irregular en el tamaño de las sublistas, puede perfectamente darse la ejecución concurrente ilustrada en las Figuras 1 y 2, donde el sentido de las flechas indica, de arriba hacia abajo, "se asigna", y de abajo hacia arriba, "se resuelve".

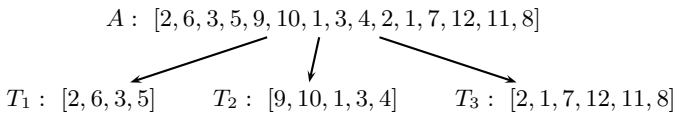


Fig. 1. Subdivisión que el agente A asigna los trabajadores T_1, T_2 y T_3.

4.1. Distribución de trabajo

En esta implementación notaremos que los trabajadores reciben sus órdenes de labor de forma vertical (ver Fig. 3), solo aceptando las directivas de trabajo sin ningún tipo de réplica; además, el conocimiento que tienen entre sí (a nivel de trabajadores) es nulo, así que esto hace imposible algún tipo de organización para enfrentar de manera horizontal las labores asignadas (coordinándose, posiblemente, entre sí). De darse la comunicación, esta puede darse, por un lado, mediante el `agenteOrd` como intermediario, o bien fungiendo como un servidor centralizado; esto puede orillar a que la terminación abrupta del agente

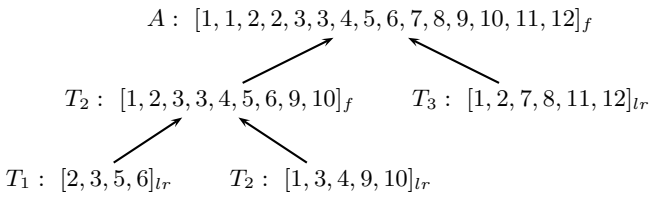


Fig. 2. Labores resueltas (lr), y ejecución del intermezclado (f). Note que el trabajador T_2 también está equipado para realizar intermezclado.

`agenteOrd` conlleve la terminación también abrupta de la realización distribuida de la labor en total; por otro lado, tomando un punto de vista más flexible, el `agenteOrd` podría terminar sin previo aviso, pero las subtareas asignadas todavía lograrse por medio de los trabajadores ya organizados y colaborando entre sí, ya sea surgiendo un trabajador como lider emergente o bien, aún sin ese liderazgo emergente, con las tareas ya finalizadas y almacenadas, para que cuando un nuevo agente lider surja sea notificado de qué se realizó mientras no había liderazgo "oficial". Si se utiliza tal almacenamiento, es necesario que la relevancia de la tarea completada (o lista por completarse por el nuevo `agenteOrd`) sea permanente en el tiempo.

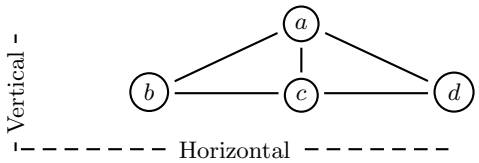


Fig. 3. Caracterizaciones de colaboración.

Ahora, el punto de vista horizontal se relaciona con *agentes colaborativos y cooperativos* [4], lo que por un lado debe formularse en un aspecto de paridad entre todos los agentes trabajadores, y por otro debe automatizarse en la posible reorganización de la labor aún sin los reportes finales, pero manteniendo un estado de resultado latente.

La primera parte para el conocimiento mutuo entre los agentes trabajadores es por medio de una solicitud del operador para conocer con nombre de referencia de nodos-e (ya que los procesos están identificados mediante `trabajador`) a quien tiene estos datos: el agente `agenteOrd`. Este conocimiento, cuando se haya autorizado, será solo de los agentes actualmente interconectados.

Procedemos como a continuación se indica:

Por la parte de un trabajador, quien inicia una solicitud:

```
solicitudConocer ->
        {agenteOrd,NodoR} ! {listaTrabajadores,node()},
        [reactivación de agente trabajador];
```

Por la parte del agente `agenteOrd`, quien procesa la solicitud del trabajador:

```
 {listaTrabajadores,Nodo} ->
            {trabajador,Nodo} ! {trabajadoresActuales,Ts},
                    [reactivación de agente agenteOrd];
```

Nuevamente, por la parte del trabajador quien recibe una respuesta a su solicitud:

```
{trabajadoresActuales,Ts} ->
io:format("Autorizado, trabajadores restantes:~p~n",[Ts]),
        [reactivación de agente trabajador];
```

Por ejemplo, suponiendo ya en funcionamiento un (pequeño) sistema distribuido con dos trabajadores además del agente `agenteOrd`:

```
(osito@RieGau)5> trabajador ! solicitudConocer.
solicitudConocer
Autorizado, trabajadores restantes:[{trabajador,osito@RieGau,
                                     ["1","Silvestre",29]},
                                    {trabajador,osita@RieGau,
                                     ["0","Piolin",19]}]
```

`map(fun(X) -> filtrar(X) end,Ts)` procedería a obtener `NTS=[{trabajador,osito@RieGau}, {trabajador,osita@RieGau}]` donde `filtrar({T,N,Ls}) -> {T,N}.` es una función de biproyección. En este caso desde el mismo proceso se tiene un conocimiento del propio nombre indirectamente, que bien podría servir para otro propósito distinto al de conocer los trabajadores (tal como darse a conocer en una *coalición* de `agenteOrd`s, posibilidad contemplada en el formalismo de SCEL). De momento es mejor eliminar el propio nombre: `Dest=eliminar({trabajador,node()},NTS)` lo que desde `osito@RieGau` obtenemos que `Dest=[{trabajador,osita@RieGau}]`, el nombre del otro trabajador, y simétricamente para `osita@RieGau`.

5. Colaboración vertical

La *colaboración vertical* que ahora estudiamos es un tipo de colaboración de jefe a subordinado. Para esta parte se establece que el proceso A existente en un nodo-e tenga un papel especial, que de momento llamamos de gestoría, y será nombrado como `gestor`. Ahora existirán dos procesos B y C con sendos nodos-e que tendrán un papel de realizar cierta clase de labores, y será nombrados cada uno de ellos `trabajador`, notando que no hay problema en la confusión de los procesos ya que los nodos-e determinan exactamente qué proceso es referido mediante la tupla `{trabajador, node()}`.

Mediante este esquema, es posible que los trabajadores acuerden realizar una labor mediante una distribución inteligente de trabajo. Este tipo de colaboración será nombrado *horizontal*. En ocasiones, solo algunas condiciones pueden ser cubiertas, tales como una distribución equitativa de trabajo (sin considerar las capacidades de los trabajadores), y se supone a todos los trabajadores con las mismas cualidades de labor, sin especificar tipos de trabajo. Esto ayuda a simplificar la repartición equitativa de trabajo sin caer en problemas complejos de calendarizaciones.

Para una posible aplicación, cada trabajador debería, además de su identidad, dar a conocer su capacidad mediante un número (unidades de labor) y su disponibilidad temporal (si no está realizando una labor ya o tiene un compromiso de hacerlo).

Con la función `partes/2` tendremos una repartición de trabajo válida solo para las labores son múltiplos exactos del número de trabajadores (ver Fig. 4); los trabajadores se ven beneficiados por tener una carga de trabajo equitativa, pero cuando las unidades de labor no son exactas, el programa tratará la parte residual, como un caso especial, asignado a ser gestionado por el agente `agenteOrd`.

```
asignacion(Ntrabajadores,Labor) ->
                                Len=length(Labor),
                                Div=Len div Ntrabajadores,
                                PartesExactas(Div,Labor).
```

Fig. 4. Una política parcialmente correcta y justa de distribución de labor.

Con la función `exactasYno/2` de la Fig. 5 la labor restante se reubica para el primer trabajador, aún siendo cercana al número de trabajadores (pero sin igualarla o sobrepasarla), lo que ejemplifica una política de distribución injusta; la formulación original de `partes/2`, por otro lado, solo aceptaba la división exacta del número de unidades de labor entre el número de trabajadores, o de otra manera conducía a un *crash* del agente `agenteOrd`, y aunque era correcta y justa cuando tal división era exacta, era incorrecta en la división inexacta (con residuo diferente de cero). En esta ocasión, el *crash* es evitable, pero con política injusta.

```
asignacion(Ntrabajadores,Labor) ->
                                Len=length(Labor),
                                Div=Len div Ntrabajadores,
                                exactasYno(Div,Labor).
```

Fig. 5. Una política injusta de distribución de labor.

```
asignacion(Ntrabajadores,Labor) ->
                                  Len=length(Labor),
                                  Div=Len div Ntrabajadores,
                                  distribuir(Div,Labor).
```

Fig. 6. Una política justa de distribución de labor.

Otra política más estaría dada por el cambio de `exactasYno/2` a `distribuir/2` (ver Fig. 6): ésta última función reparte equitativamente el excedente unidad por unidad entre los trabajadores participantes. Esta es una de las políticas que no requieren negociación implementadas, pero que son justas al repartir equitativamente el excedente de trabajo entre los trabajadores participantes. En esta ocasión, se reparte el excedente unidad por unidad entre los trabajadores participantes, repitiendo varias rondas si es necesario, hasta que se agotan todas las unidades del excedente. Algunos trabajadores pueden trabajar menos, pero definitivamente, no se sobreexplotará a uno solo ni tampoco se tendrán casos de terminación anormal (aunque hay que anotar que para todos los casos, la cantidad de labor debe ser al menos un múltiplo exacto mayor que el cuadrado de los trabajadores participantes, una característica de la implementación actual).

Para cuando se tengan tratamientos justos de excedentes se podrían considerar *rondas* que permitirían terminar el trabajo en mayor tiempo pero justamente (ver Fig. 7). Es necesario enfatizar que, en otros escenarios, la *capacidad* de labor de un trabajador podría tomarse en consideración al repartir las sublabores, lo que haría que, en una posible política, aquellos trabajadores que no tienen la suficiente capacidad, digamos k de ellos, sean exentos de la asignación de labores, y solo se considere a los $n - k$ restantes; esto originaría el problema de que obligaría a reasignar las sublabores a menos trabajadores, quizás presentándose nuevamente el problema de rebasar la capacidad de los parte de los $n - k$ trabajadores disponibles; bajo el mecanismo de rondas, suponiendo sea éste implementado, se puede encontrar el mínimo de entre las capacidades de todos los trabajadores para comenzar globalmente a realizar la labor asignada, pero a cambio de tardar varias rondas.

Consideremos por ejemplo a tres trabajadores con una labor de 12 unidades. Es posible una división justa: $(4, 4, 4)$ (en donde cada entrada del vector corresponde a una asignación del i-ésimo trabajador, $i = 1.,3$); otras injustas: $(12, 0, 0)$, $(6, 0, 0)$ y así.

Este sistema en términos de programación tiene otras características: ha sido especialmente diseñado para evitar generar nombres dinámicamente y puede tomar n (con $n \geq 1$) trabajadores para realizar n sublabores. Además, a propósito no se han utilizado algunas funciones de Erlang para el tratamiento de diccionarios o registros (*records*), pues las listas, como estructura de datos en la programación funcional, tienen una buena teoría transformacional de apoyo; el cambio a otras estructuras de datos estaría justificado cuando fuera necesario almacenar en una base de datos los registros de los (muchos) posibles trabajado-

```
asignacion(Ntrabajadores,Labor) ->
                                    Len=length(Labor),
                                    Div=Len div Ntrabajadores,
                                    distribuir(Div,Labor).
```

Fig. 7. Una política justa de distribución de labor.

res; llegado el caso, los datos a tratar como labor pueden ser enormes en cuanto a espacio y pueden requerir de accesos a un medio no volátil de memoria para su almacenamiento; si el procesamiento de las sublabores requiere tiempo variado, será deseable quizás solicitar "estados parciales" de parte del usuario `operadorA` para monitorear el curso global de cómputo. Otro tema dejado de lado es que hay una pobre tolerancia a fallas en el formato de las sublabores: si una parte de la sublabor no la puede elaborar un trabajador, éste debería realizar lo que sí puede y dejar aparte lo que no, quizás reportándolo.

En principio, estas limitantes son posibles de superar, aunque nuestro objetivo es (y será con los siguientes sistemas) ejemplificar los sistemas distribuidos con un núcleo básico de primitivas de Erlang, usando este núcleo básico de Erlang como una herramienta computacional adecuada para ilustrar aportaciones de índole distribuida.

5.1. Labor segmentada

Para los casos en que una labor pueda segmentarse, pongamos por ejemplo que tenemos 3 trabajadores, cada uno con capacidad 6 y una labor de 17 unidades. En un reparto equitativo, cada trabajador toma 6 unidades de labor, así que queda 1 unidad todavía por asignar. Esta asignación es dada directamente por el agente `agenteOrd`, y es solo por imposición (sin negociación de por medio); de la cadena de mando, esta encomienda es *vertical*. Órdenes que surgen de esta manera no son de ninguna manera puestas en duda por los trabajadores de más bajo nivel jerárquico. Dado el problema de ordenamiento que estamos elaborando, es posible que haya n trabajadores, así que algunas funciones de trabajo propias de `agenteOrd` deben generalizarse a tratar este caso general de tamaño n (uno por cada trabajador participante). Por ejemplo, `fusionGeneral/1` toma n listas ya ordenadas y genera una nueva lista ordenada:

```
fusionGeneral([]) -> [];
fusionGeneral([A]) -> A;
fusionGeneral([A,B|Ls]) -> C=fusionar(A,B),
fusionGeneral([C|Ls]).
```

(Una versión general con función de alto orde es la del módulo `lists`: `foldl`.)

Por medio de una imposición directa, el trabajo quedaría repartido como `trab1:7`, `trab2:6`, `trab3:6`. Esto es arbitrario y rompe la regla de no dar más labor a un trabajador de la permitida por su capacidad. La consecuencia es que `trab1` podría o no hacer la labor (y si la hace, podría ser motivo de falla de

operación), sin responsabilidad de su parte. Si, por otro lado, se respeta la regla de la capacidad, tenemos el reparto: `trab1:6`, `trab2:6`, `trab3:6`, quedando con una unidad para repartir posteriormente (en otros posibles turnos globales o *rounds*). Notemos que la labor devuelta por cada trabajador es válida, pero que debe abrirse otra ronda de labor, esta vez del tipo: `trab1:0`, `trab2:1`, `trab3:0`, aunque esto requiere guardar el resultado de la ronda previa (que ya está ordenado), y esperar a recibir el resultado nuevo (que en este caso, es fácil de realizar), para finalmente armar el resultado final:

```
(oso@tosh)4> mergeAg05:fusionGeneral([[1,2,3],[4,5,6],[-3,-2,-1]]).
[-3,-2,-1,1,2,3,4,5,6]
```

Por ahora, nos enfocaremos a generalizar también el vector de terminación. Para dos trabajadores, el vector debe estar en el estado `{1,1}` indicando que se ha terminado la labor. Mencionábamos que esto es anti-declarativo, pero es efectivo en esta etapa de implementación (en la parte de diseño no hay dificultad en indicar que los trabajadores emitan una "labor terminada" unánime para indicar que todo el trabajo asignado —globalmente— se ha llevado a cabo). Junto con este vector, otro más mantiene las tareas o por completar o ya completadas. A continuación mencionamos algunas otras formulaciones que se involucrarían en la cooperación para la repartición de labores, planeadas como trabajo a futuro.

5.2. Coordinación entre agentes

En un sistema distribuido donde los agentes tienen algunos aspectos de inteligencia, es clave mencionar el aspecto de la coordinación entre los agentes, de tal manera que tal coordinación favorezca el objetivo para el cual, en principio, el sistema está diseñado. Para el tema la coordinación para los agentes propuestos, existen varias alternativas a explorar, considerando también cuando se involucren *conflictos*. Veamos algunas opciones.

Ruptura de conflicto por azar. La ruptura de conflicto por azar es un sorteo de trabajo, realizado por los mismos trabajadores. Por ejemplo, todos los agentes pueden tener la misma posibilidad de ser elegidos para la unidad inicial de labor, y o bien, en otra ronda el que ya fue asignado con una unidad participa o no. No hay posibilidad de desacuerdo, ya que en principio todos acordarían estar conformes con los resultados obtenidos.

Ruptura de conflicto por altruismo. Para la ruptura de conflicto por altruismo, aquellos agentes que se vean con posibilidades de realizar una labor la aceptarán de inmediato, siendo inclusive varias posibles unidades; el componente no determinístico está presente, pero no causa ningún problema, ya que por definición los agentes candidatos estarían en una cola de espera, con algún tipo de asignación de orden; la asignación de orden implica que durante un intervalo de tiempo prudente el agente altruista todavía estará de acuerdo en aceptar la labor solicitada, pero no implica que realmente la labor se realice, dejando la terminación como parte de las características de la asincronía del sistema, o informando de partes realizadas, o reportando la terminación global de labor.

Ruptura de conflicto por votación. El caso de ruptura de conflicto por votación se inspira en la política democrática de algunas sociedades humanas. Es de tipo horizonal. Primero, debe surgir un conjunto de candidatos. Estos candidatos son seleccionados al azar o por otro agente o por auto-propuesta. Los candidatos, para empezar, aceptan su candidutra, y promocionan sus mejores atributos (como una función de beneficio social, *welfare function*). Surgen varias posibilidades de ordenamiento de preferencia del electorado. Finalmente, en varias posibles rondas de votaciones se decide quién será el candidato ganador, que en este caso es aquel que tenga el trabajo a realizar. De realizarse una implementación de algún sistema de votación (ya que hay varios, cfr. [20], cap. 12), sería deseable suponer una capacidad de procesamiento básico de cada trabajador, ya que es posible *arreglar* las rondas de votación para manipular los resultados finales, aunque hay evidencia de que esto tiene casos que caen en tratamientos difíciles NP [3]. No hemos implementado ningún tipo de votacion horizontal que logre una ruptura de conflicto.

Ruptura de confilicto por registro histórico y distribución justa posterior. Este tipo de ruptura de conflicto pasa de ser imposibilitad de resolverse horizontalmente a intentar ser resuelta verticalmente, y en este caso el agente `agenteOrd` mantiene una posición salomónica o justa. Por ejemplo, la ruptura de conflicto se puede dar por registro histórico y la distribución justa primero se basaría en observar qué ha hecho un agente. Si para tal agente no existe carga de trabajo previo o ha sido ligera, se le asigna una labor; de lo contrario, se considera otro agente, y así. Si todos tienen registros similares, se procede a romper esta simetría por azar, degenerando en un caso de ruptura de conflicto por azar.

Argumentaciones. Consideremos la situación ahora en donde un agente A de capacidad C recibe una asignación de D unidades de labor; si $D \leq C$, todo procedería sin problemas; la parte conflictiva viene cuando $D > C$. En una primera aproximación, el agente A puede rechazar de tajo la asignación, pero en otro cauce se podría tomar la siguiente decisión (de tipo vertical): De $D > C$, la cantidad $D - C$ es positiva, e irrealizable. Con una posibilidad de solventar aquella parte que el agente sí puede hacer, en lugar de rendirse ante la imposibilidad de la carga asignada puede hacer una parte y devolver la otra sin tocar. El agente A entonces recibe una labor Ds de tamaño de D unidades y devuelve $\{Cs, Ds - Cs\}$, en donde Cs es la labor que fue realizada y $Ds - Cs$ es la labor que fue imposible de realizar.

Alicientes o estímulos. Si fuera necesario cuantificar la proclividad de un agente se puede disponer de una entidad de valoración de labor parecida a la moneda que se utiliza en sociedades humanas, siendo una de las formas de "recompensar" la labor humana. Opcionalmente, podría ser que existan jerarquías de trabadores en donde el aliciente al llevar cabo algún tipo de labores sea escalar en tal jerarquía (recategorizaciones). Bajo ésta circunstancia, es posible también implementar *políticas de negociación*, de tal forma que se involucren conceptos de economía en la forma de ejecutarse el sistema distribuido propuesto.

La implementación de estas conductas inteligentes de colaboración es un tópico que al momento estamos abordando. Al momento, se tiene la convicción de que Erlang provee de un ambiente general de programación de temáticas en inteligencia artificial (dura) que es tan bueno como algunos lenguajes tradicionales en este tipo de aplicaciones, tales como Lisp, con la ventaja de tener ya incorporadas algunas primitivas robustas y escalables de concurrencia asíncronas, con un modelo teórico fiable de fondo. fiable

6. Conclusiones

En este artículo se han presentado algunos programas concurrentes y funcionales que están en el complemento práctico de algunas teorías de colaboración para la realización concurrente de labores.

Utilizando Erlang, se ha logrado ver la interrelación que existe entre la identificación de un conjunto de mensajes apropiados, su envío y recepción, y mediante esta comprensión se ha presentado una metodología basada en diálogos, mismos que fundamentan la colaboración y las políticas de repartición de trabajo. Nuestro problema a trabajar ha sido el de ordenar paralelamente una lista de números mediante un algoritmo de fusión, pero notemos que este problema puede no tener un tratamiento paralelo con otros algoritmos (por ejemplo, el de selección). Otros problemas, tales como los de grafos, pueden requerir de estrategias cuidadosamente planeadas para aprovechar la tecnología distribuida [14].

Frecuentemente en el área de la Inteligencia Artificial aparecen tareas que son exigentes en cuanto a cálculo computacional se refiere, o bien son por necesidad distribuidas. Ejemplos de la parte de exigencias de procesamiento serían: el reconocimiento de imágenes, la segmentación de videos, las redes neuronales y los algoritmos de geometría computacional; ejemplos de la parte distribuida serían los equipos de labores constituidos por varios robots y los agentes inteligentes de la Internet. El planteamiento de este escrito parte de un caso de estudio particular, ejemplificando las bondades de Erlang para el procesamiento distribuido de datos dentro del mismo lenguaje Erlang, pero se menciona como nota final que Erlang puede ser parte medular de un ecosistema donde diversos programas especialistas interactúan entre sí mediante la gestión de coordinación afianzada en los procesos de Erlang.

Agradecimientos. Gerardo Carreño agradece al Banco de México el apoyo económico otorgado para sus labores de investigación, y a la Universidad Tecnológica de la Mixteca por brindarle un agradable ambiente académico. Manuel Hernández agradece a la Universidad Tecnológica de la Mixteca las facilidades otorgadas para la realización de este trabajo. Particularmente, agradece a la Mtra. Mónica García, coordinadora de la Maestría Virtual de esta Casa de Estudios, su confianza y apoyo para obtener este artículo.

Referencias

1. Armstrong, J.: Programming Erlang: Software for a concurrent world. The pragmatic programmers (2007)
2. Armstrong, J.: Erlang. Communications of the ACM 53(9), 68–75 (2010)
3. Bartholdi, J.J., Tovey, C.A., Trick, M.: The computational difficulty of manipulating an election. Social choice and welfare 6, 227–241 (1989)
4. Bedrouni, A., Mittu, R., Boukhtouta, A., Berger, J.: Distributed Intelligent Systems, a Coordination Perspective. Springer (January 2009)
5. Bird, R., Wadler, P.: Introduction to Functional Programming. Series in Computer Science, Prentice Hall (1988)
6. Burstall, R.M., Darlington, J.: Some transformations for developing recursive programs. SIGPLAN Not. 10(6), 465–472 (Apr 1975), http://doi.acm.org/10.1145/390016.808470
7. Cesarini, F., Thompson, S.: Erlang programming. O'Reilly (2008)
8. Dunen-Keplicz, B., Verbrugge, R.: Teamwork in multiagent systems. A formal approach. Wiley series in agent technology, Wiley (2010)
9. Fokkink, W.: Modelling Distributed Systems. Springer (2007)
10. Fredlund, L.Å., Earle, C.B.: Model checking Erlang programs: the functional approach. In: ERLANG '06: Proc. of the 2006 ACM SIGPLAN workshop on Erlang. pp. 11–19. ACM (2006)
11. Fredlund, L.Å., Svensson, H.: McErlang: a model checker for a distributed functional programming language. In: ICFP '07: Proc. of the 12th ACM SIGPLAN Intl. Conf. on Functional programming. pp. 125–136. ACM (2007)
12. Huch, F.: Learning programming with Erlang. In: ERLANG '07: Proc. of the 2007 SIGPLAN workshop on ERLANG Workshop. pp. 93–99. ACM (2007)
13. Larson, J.: Erlang for concurrent programming. Queue 6(5), 18–23 (2008)
14. Lenharth, A., Nguyen, D., Pingali, K.: Parallel graph analytics. Commun. ACM 59(5), 78–87 (Apr 2016), http://doi.acm.org/10.1145/2901919
15. Noll, T., Roy, C.K.: Modeling erlang in the pi-calculus. In: ERLANG '05: Proceedings of the 2005 ACM SIGPLAN workshop on Erlang. pp. 72–77. ACM, New York, NY, USA (2005)
16. Partsch, H.: Specification and transformation of programs. Springer-Verlag (1991)
17. Pettorossi, A., Proietti, M.: Rules and strategies for transforming functional and logic programs. ACM Comput. Surv. 28(2), 360–414 (Jun 1996), http://doi.acm.org/10.1145/234528.234529
18. Varela, C., Abalde, C., Castro, L., Gulías, J.: On modelling agent systems with Erlang. In: ERLANG '04: Proc. of the 2004 ACM SIGPLAN workshop on Erlang. pp. 65–70. ACM (2004)
19. Varela, C.A.: Programming Distributed Computing Systems: A Foundational Approach. The MIT Press (2013)
20. Wooldridge, M.: An Introduction to MultiAgent Systems. Wiley Publishing, 2nd edn. (2009)

ISSN 1870-4069

Diseño de interfaces de inspección de calidad utilizando redes neuronales y visión artificial para la industria de manufactura

Luis Arturo Medina Muñoz, Samuel González-López,
Jesús Antonio Mayorquín Robles, Gabriel Antonio López Valencia

Instituto Tecnológico de Nogales, División de Estudios de Posgrado e Investigación,
Nogales, Sonora, México
mmarturo756@gmail.com, samuelgonzalezlopez@gmail.com,
ing.mayo@hotmail.com, glopez09@hotmail.com

Resumen. La inspección de componentes con sistemas de visión es una tarea que se puede lograr con diferentes métodos. La inspección en la mayoría de los procesos de fabricación depende principalmente de operadores humanos cuyo rendimiento es generalmente inadecuado y variable [4]. Las áreas de manufactura donde la visión juega un papel importante son la inspección, las mediciones y algunas tareas de ensamblaje [5]. Este artículo aborda el problema implementando redes neuronales artificiales. Utilizamos la técnica de erosión de imagen contra una imagen maestra (conector) y el reconocimiento de patrones de redes neuronales después de la erosión. La aceptabilidad de los conectores o cables se logra utilizando el reconocimiento de elementos mediante visión artificial. En este trabajo, se explica cómo podemos extraer las características clave de la imagen, posición de las chavetas del conector, y las convertimos en números y alimentamos la red neuronal para determinar si un conector o cable cumple con los requisitos o es un elemento defectuoso. El sistema se validó en el área de inspección de recibo en una fábrica local de manufactura, donde se realiza la primera inspección de los componentes, en contra de su dibujo y en la línea de producción. Los resultados de la validación del sistema de visión muestran una aceptación del 100 % para la identificación de conectores y una exactitud de 83% para la identificación de cable.

Palabras claves: sistemas de inspección, industria de manufactura, redes neuronales, sistemas de visión.

Design of Quality Inspection Interfaces using Neural Networks and Artificial Vision for the Manufacturing Industry

Abstract. The inspection of components with vision systems is a task that can be achieved with different methods. This article addresses the problem by implementing artificial neural networks. We use the technique of image erosion against a master image (connector) and the recognition of neural network patterns after erosion. The acceptability of the connectors or cables is achieved using the recognition of elements by artificial vision. In this paper, we explain how we can

extract the key features of the image, position the connector keys, and convert them into numbers and feed the neural network to determine if a connector or cable meets the requirements or is a defective element. The system was validated in the area of receipt inspection in a local manufacturing factory, where the first inspection of the components is carried out, against their drawing and in the production line. The results of the validation of the vision system show a 100% acceptance for the identification of connectors and an accuracy of 83% for the identification of the cable.

Keywords: inspection systems, manufacturing industry, neural networks, vision systems.

1. Introducción

La inspección industrial es una de las tareas cruciales para garantizar la calidad de los productos [6]. El menor costo de mano de obra en los países en desarrollo en comparación con el de las economías desarrolladas a menudo resulta difícil para justificar la maquinaria de automatización. Si bien esta tecnología puede mejorar la calidad del producto y la productividad, así como reducir el costo de los productos, en las economías en desarrollo y emergentes, a menudo es necesario que las empresas manufactureras soliciten apoyo a centros de investigación, ya que las universidades se dirigen a los estudiantes para apoyar a las industrias locales [1].

La Visión Artificial (AV) utiliza imágenes que son capturadas por una cámara de visión industrial y luego procesadas a través de software específico.

De esta manera, AV puede medir, contar, verificar, seleccionar e identificar fallas y anomalías y, con base en este procesamiento, la tecnología también tiene la capacidad de tomar decisiones como expulsar un determinado producto de la línea de producción cuando éste no cumpla con los estándares de calidad exigidos. Estas decisiones se toman en base a parámetros preestablecidos, de modo que todo lo que no cumpla con estos parámetros se considera incorrecto [2].

El precio de un sistema de visión en el mercado tiende a oscilar entre 2,000 y 100,000 USD. La industria de fabricación local tiene áreas de inspección de recibos y de producción, por lo que el costo para un sistema de visión podría ser tan alto como 60,000 USD, además de otros 15,000 USD para trabajos de calibración y entrenamiento de personal por año debido a la naturaleza cambiante del entorno de trabajo.

El objetivo de este artículo es proponer el desarrollo de un sistema de inspección automática de bajo costo utilizando técnicas de visión y redes neuronales artificiales entrenadas con el algoritmo de propagación hacia atrás, capaz de reducir los errores humanos. El sistema de inspección deberá ser rápido y confiable, capaz de identificar conectores con las características deseadas correctas y el despoje adecuado de los cables, además de otras características importantes en conectores y cables.

2. Marco referencial

Las redes neuronales artificiales (RNN) se han desarrollado como una generalización de los modelos matemáticos de la cognición humana y se han mostrado

prometedores para resolver problemas difíciles en áreas como el reconocimiento de patrones y la clasificación. Una de las principales dificultades que enfrentan los investigadores que usan redes neuronales es la selección del tamaño y la topología adecuados de las redes [7].

La obtención de buenos resultados depende en gran medida de los parámetros de aprendizaje, de los pesos iniciales, así como la topología de la red [10]. El uso de redes neuronales como clasificador requiere una fase de entrenamiento y una fase de prueba. En la fase de entrenamiento, la red neuronal realiza el ajuste apropiado para sus pesos (W) para producir la respuesta deseada. Cuando la respuesta de salida real se aproxima a la deseada, la red ha completado el entrenamiento. En la fase de prueba, se solicita a la red neuronal que clasifique un nuevo conjunto de imágenes y se evalúa su éxito [3].

En [9] los autores abordaron la tarea de identificar defectos o imperfecciones de la superficie en los procesos de fundición, como la inclusión, la vuelta fría y los errores. Los autores utilizaron un método de segmentación que marca la región afectada por algunos de los defectos y luego aplicaron técnicas de aprendizaje automático para clasificar las regiones incorrectas.

Los investigadores utilizaron la herramienta Weka (Weka es una plataforma de software para el aprendizaje automático y minería de datos) para realizar la tarea de clasificación con los algoritmos NaiveBayes, SVM, J48 KNN y RandomForest (RF). Con RF y una muestra de 50 los autores lograron una precisión de aproximadamente el 95%. De manera similar, nuestro trabajo busca identificar defectos en materiales diversos para luego predecir con el uso de una red neuronal.

El mantenimiento de los estándares de calidad es importante en cualquier empresa debido a la rotación de los empleados en el área de inspección y producción. En [10], los autores proponen un modelo de red neuronal para ayudar a los profesionales de la industria a tomar decisiones en el área de inspección con entradas binarias. La experimentación se llevó a cabo utilizando el 70% de los datos para el entrenamiento y el 30% para las pruebas. Los autores realizaron una evaluación en dos etapas. En la primera, a una red neuronal se le permitió decidir si el producto era un rechazo. En la segunda etapa, un segundo modelo de red neuronal decidió si la pieza podía ser reelaborada o si era definitivamente desperdicio.

A diferencia de nuestro trabajo, los autores utilizaron los comentarios de expertos, mientras que este trabajo toma información de las piezas a través de una cámara y hace un análisis similar al que hicieron en [8] donde se muestran datos importantes como la precisión de las redes neuronales, su estructura en términos de la cantidad de entradas, nodos ocultos y salidas. En la fase 1, alcanzaron el 97% de las piezas correctamente clasificadas y en la fase 2, la precisión fue del 98%.

3. Desarrollo

Debido al estricto requisito de calidad y la falta de inspección por visión artificial, es necesario inspeccionar el 100% de los productos. La empresa utiliza doble inspección en conectores. Durante el proceso de fabricación y la inspección final por control de calidad, pero aún con la doble inspección, la compañía rechazó 16,000 conectores y 300 arneses defectuosos que fueron aceptados por la fabricación y control de calidad en enero y marzo de 2017. El defecto en los conectores, se refiere a un

corrimiento de entre 1 o 2 grados de la posición original de las chavetas indicadas con el color blanco en la figura 7 como resultado de la binarización de la imagen. El costo de la mala calidad se ha estimado en 50,000 USD. Además, se revisa el despoje de cables blindados utilizados en antenas, que se maquilan en máquinas despojadoras especiales para ello, aun así, existen problemas de calidad.

El sistema propuesto utiliza la cámara de inspección Mighty Scope. La cámara tiene integrado 6 LEDs de alta intensidad; los LED ayudan en el 85% de los casos para evitar el bajo factor de iluminación. Por otra parte, la cámara fue diseñada para la tarea de inspección. Por esta razón, la lente está dedicada a capturar detalles en la imagen, esto es útil para obtener imágenes para procesar. Para obtener la imagen del conector, la cámara debe permanecer estable, fue necesario utilizar el soporte para cámara "Migthy Scope View Stand". La figura 1 muestra la cámara y el soporte juntos.



Fig. 1. Cámara de inspección Mighty Scope.

3.1. Procesamiento de conectores

La figura 2 muestra el diagrama de flujo del sistema para inspección de conectores.

En el proceso de captura de imagen, aun y cuando existen varios tipos de estos dispositivos, se utilizó una cámara capaz de tomar video y fotografías digitales con iluminación integrada para minimizar los efectos de una iluminación pobre (Fig. 3).



Fig. 2. Diagrama de flujo del Sistema.

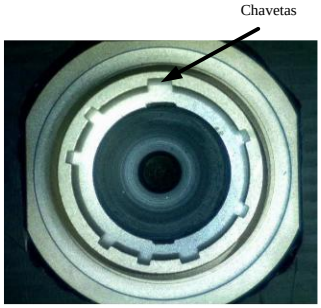


Fig. 3. Imagen de conector a inspeccionar con 8 chavetas.

El proceso de erosión se logra mediante 3 etapas distintas:

1. Dibujo de un círculo interior dentro de las chavetas del conector.

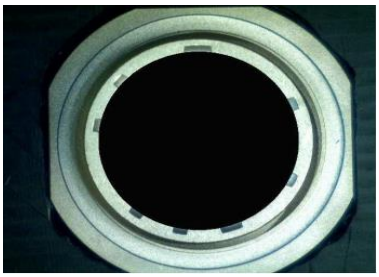


Fig. 4. Dibujo del círculo interior.

2. La figura 5 muestra el dibujo del círculo exterior que cubre el área de interés y deja el área exterior del conector.

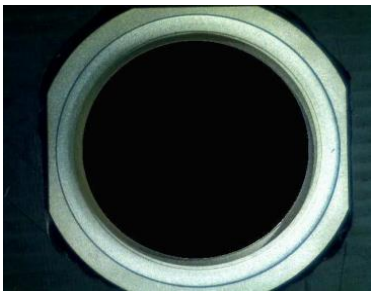


Fig. 5. Dibujo del círculo exterior.

3. La erosión de imágenes se logra comparando las figuras 3 y 4, los pixeles iguales se muestran en color negro dejando solamente los pixeles que son diferentes, como lo muestra la figura 6.

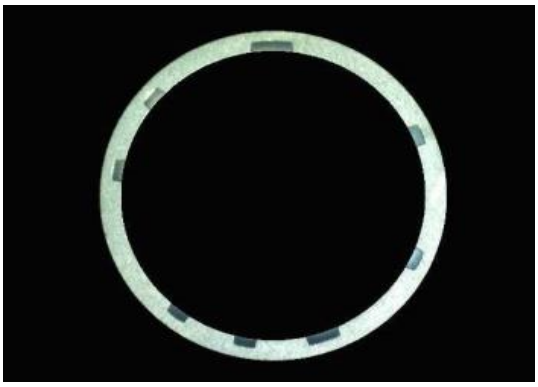


Fig. 6. Imagen erosionada.

El proceso de monocromatización y obtención de características claves se logra haciendo un recorrido de 360 grados desde el centro del conector hasta la altura del radio conocido y los pixeles encontrados como blancos corresponden a la ubicación de la chaveta a inspeccionar. Los datos de salida de extracción como características claves, son expresados como 1, donde se detecta que se encuentra la chaveta, que corresponde al color blanco y un 0 donde no existe la chaveta y que corresponde a un color negro. Esto se muestra en la figura 7 y tabla 1.

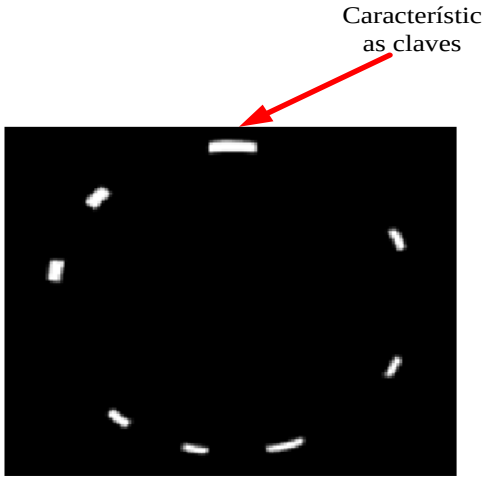


Fig. 7. Imagen procesada.

Los datos de la tabla 1 serán utilizados como parte del entrenamiento de la RNA, la imagen muestra un arreglo de 30 renglones por 12 columnas, de izquierda a derecha son el recorrido de 12 grados del círculo . de arriba hacia debajo de 12 en 12 hasta completar el recorrido de los 360 grados. El 1 significa que se encontró un color blanco que corresponde a la posición de una chaveta y el 0 corresponde a un color negro que indica la ausencia de la chaveta.

Tabla 1. Imagen binarizada.

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

3.2. Procesamiento de cables

La figura 8, nos muestra el diagrama de flujo para la inspección del despoje en cables blindados, indicando el procedimiento a realizar en cada una de ellas en el proceso de la inspección de calidad.

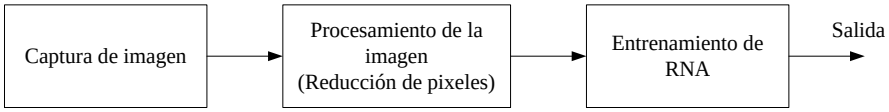


Fig. 8. Diagrama de flujo para la inspección de cables blindados.

Las imágenes fueron procesadas para reducir el tamaño, esto es, la imagen tenía el tamaño de 2,907 x 5,168 Pixeles, la figura 9 muestra la captura original del cable, así que se desarrolló una aplicación para capturar la imagen del cable y reducirla a un tamaño de 600 x 5,168 Pixeles.



Fig. 9. Imagen del cable a procesar.

De la imagen original se eliminó el fondo verde a través de un filtro, esto redujo el tamaño de la imagen de 1.6M a 178K, figura 10, a esta imagen se le aplicó la escala de grises y se sacó el histograma, dando como resultado un archivo con 256 columnas a

escala de grises, al cual se le agregó otra columna, a las piezas aceptables se le dio el valor de "1" y para las piezas defectuosas el valor de "0".



Fig. 10. Imagen procesada.

El entrenamiento de la red neuronal de hizo con los datos del histograma de 190 cables aceptables y con 54 histogramas de cables defectuosos, dejando 10 cables aceptables fuera del entrenamiento, así como 10 cables defectuosos para utilizarse como vector de pruebas.

Se propone implementar una estructura de RNA, como la mostrada en la figura 11, entrenada con el algoritmo de propagación hacia atrás, para realizar el reconocimiento de los patrones a identificar en ambas pruebas, posición de las chavetas en conectores y despoje de cables.

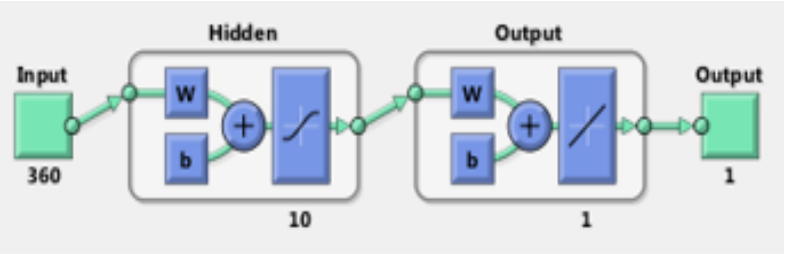


Fig. 11. Estructura de RNA propuesta.

3.3. Experimentación

Se utilizó una RNA perceptrón multicapa con un algoritmo de propagación hacia atrás para el entrenamiento, con una cantidad fija de neuronas en la capa oculta para la inspección de los conectores y una estructura de RNA con variación en la cantidad de neuronas en la capa oculta para la inspección de cables. La decisión sobre el uso de este tipo de estructura neuronal se basa en un análisis de prueba de diferentes topologías con un conjunto de datos de entrenamiento y pruebas que satisfacen la política de calidad de la compañía. La Tabla 2 muestra el número de conectores para el entrenamiento y el vector de prueba.

3.4. Resultados de la inspección de conectores

Tabla 2. Conjunto de entrenamiento de la RNA.

Descripción	Cantidad
Conectores aceptables	230
Conectores no aceptables	50

Finalmente, se usó un lote completo del producto como vector de prueba. En este caso, se utilizaron los conectores bajo prueba, que correspondían a la cantidad de 2,412 conectores en el área de entrada. Cabe mencionar que el lote completo era de conectores válidos y no se tenía control sobre el lote para incluir señuelos. La tabla 3 muestra los resultados de la inspección de prueba.

Tabla 3. Resultados de la inspección.

Lote de conectores	Resultados de la prueba
2,412	2,412 reconocimientos exitosos
	0 rechazos

El tiempo de inspección requerido se redujo drásticamente, mostrando una gran mejora cuando se implementa este proceso automático propuesto, como se muestra en la Tabla 4.

Tabla 4. Comparativo de inspección.

Tipo de inspección	Tiempo de inspección
Manual	150 segundos
Automatico	8 segundos

Después de entrenar la RNA propuesta con los datos mostrados en la tabla 2, podemos ver que el modelo propuesto funciona bien, clasificando sin errores de inspección el número de conectores que se muestra en la Tabla 3, con una tasa de éxito del 100%. Los resultados que se muestran en la Tabla 4 muestran un proceso rápido que reduce el tiempo de inspección en un 95%.

Uno de los objetivos propuestos se ha logrado porque tenemos un sistema de inspección rápido y confiable. Este modelo también muestra que es posible reducir los costos de implementación del sistema en un 94%, comparando lo que cuesta un sistema industrial y el desarrollo de la aplicación.

Este dispositivo de inspección propuesto, es posible que, se pueda utilizar para verificar la calidad de otros productos utilizando solo los datos de entrenamiento apropiados para la RNA y los resultados se pueden comparar entre sí.

3.5. Resultados de la inspección de cables

Para la inspección de cables blindados se realizaron pruebas con la misma estructura de RNA propuesta en la figura 11 del tipo perceptrón multicapa con un algoritmo de propagación hacia atrás para el entrenamiento, variando la cantidad de neuronas en la capa oculta. Se utilizaron los datos mostrados en la tabla 5 para el entrenamiento de la RNA.

Tabla 5. Conjunto de entrenamiento de la RNA.

Descripción	Cantidad
Cables aceptables	190
Cables con defectos	54

Tabla 6. Matriz de confusion de la inspección con 4 neuronas en capa oculta.

Aceptables	Defectusos	
179	11	Aceptables
19	35	Defectuosos

La medida de exactitud BAC(Balanced Accuracy) obtenida para la configuracion con neuronas en la capa oculta fue de 83.24%.

Tabla 7. Resultados de la inspección con 9 neurona en capa oculta.

Aceptables	Defectusos	
163	27	Aceptables
9	45	Defectusos

La medida de exactitud BAC(Balanced Accuracy) obtenida para la configuracion con neuronas en la capa oculta fue de 78.63%.

Tabla 8. Matriz de confusion de la inspección con 10 neurona en capa oculta.

Aceptables	Defectusos	
137	53	Aceptables
1	53	Defectusos

La medida de exactitud BAC(Balanced Accuracy) obtenida para la configuracion con neuronas en la capa oculta fue de 74.63%.

Después de entrenar la RNA variando la cantidad de neuronas en la capa oculta, con la misma cantidad de datos, en el reconocimiento de cables, los resultados tienen variaciones muy representativas para poder definir una estructura apropiada para este tipo de tareas.

En la matriz de confusión de la Tabla 6, podemos observar que la estructura es aceptable para identificar cables con el despoje correcto, pero es pobre en la identificación de cables con defectos. La Tabla 7, muestra que, al modificar el número de neuronas en la capa oculta a 9, la tasa de éxito para cables con despoje correcto es bajo, pero el éxito en la identificación de cables con defecto aumenta, comparado con 4 neuronas. La tabla 7, muestra un equilibrio en el porcentaje éxitos para aceptación y rechazo, pero el porcentaje es bajo para ser tomado como un sistema confiable. La tabla 8, la cual se obtuvo variando a 10 el número de neuronas en la capa oculta, el porcentaje de éxito es bajo, 74.63 %, mostrando que tiene un mejor logro de identificación de cables defectuosos, pero con un pobre desempeño en la clasificación de cables correctos. De acuerdo a estos resultados obtenidos, no es posible definir una estructura de RNA adecuada que cumpla con los criterios de calidad definidos por la empresa de manufactura.

4. Conclusiones

Después de entrenar diferentes modelos de RNA con la misma cantidad de datos, podemos ver que el modelo propuesto funciona bien y clasifica sin errores de

inspección el número de conectores que se muestra en la Tabla 3, con una tasa de éxito del 100%. Los resultados que se muestran en la Tabla 4 muestran un proceso rápido que reduce el tiempo de inspección en un 95%. Este modelo también muestra que es posible reducir los costos de implementación del sistema en un 94%, comparando lo que cuesta un sistema industrial y el desarrollo de la aplicación.

La inspección de cables muestra un desempeño pobre en la clasificación de patrones de las estructuras de RNA, es probable que se necesite modificar la cantidad de muestras de entrenamiento o utilizar otros algoritmos para lograr equilibrar el porcentaje de aceptación de cables con las características deseadas y de rechazar los cables que no sean consistentes con esto.

Este dispositivo de inspección propuesto se puede utilizar para verificar la calidad de otros productos utilizando solo los datos de entrenamiento apropiados para la RNA y los resultados se pueden comparar entre sí.

El uso de lenguajes de programación de alto nivel facilita la elaboración de interfaces que coinciden o superan las características de los productos del tipo industrial utilizado para visualizar información. Además, con el desarrollo de la informática, facilitan la incorporación de algoritmos de inteligencia artificial y técnicas de procesamiento de visión que, cuando se utilizan con cámaras de video comerciales, podemos diseñar sistemas de inspección automática. El trabajo futuro implica ampliar el aprendizaje y el reconocimiento con conectores y cables que correspondan a otros modelos e implementar un conjunto de equipos de inspección automatizados en toda la planta de fabricación.

Referencias

1. Oppenheimer.: El impacto de la automatización en países de ingresos medios. www.semana económica.com. Último acceso: 2018/01/21
2. https://recursos.infaimon.com/hubfs/ebooks/MON_-_eBook. Último acceso: 2018/02/05
3. Banumathi, P., Nasira, G.M.: Fabric Inspection System using Artificial Neural Networks. International Journal of Computer Engineering Science (IJCES), 2(5) (2012)
4. Saravanan, A.: Artificial Neural Intelligent Visual Inspection for Process Improvement International. Journal of Innovative Research & Development, Annamalai, pp. 768–778 (2013)
5. Mogatti, M.: Automatic PCB Inspection algorithms: A Survey. Elsevier, Rolla, pp. 287–313 (2007)
6. Diab, S.L.: Designing a Visual Inspection System for Quality Characteristics Dimensions. In: 9th International Proceedings on Proceedings. pp. 1–2 (2010)
7. Hunter, D., Hao, Y., Pukish, M.: Selection of proper neural network sizes and architectures. IEEE Transactions on Industrial Informatics 8(2) (2012)
8. Patel, V.C.: Color Computer Vision and Artificial Neural Networks for Detection of Defects in Poultry Eggs. Artificial Intelligence Review, pp. 163–176 (1998)
9. Kesharwani, R., Dagli, C., Sun, Z.: Application of Neural Network in Shop Floor Quality Control in a Make to Order Business. Procedia Computer Science vol. 95, pp. 209–216 (2016)
10. Castillo, P.A., Castellano, J.G., Merelo, J.J., Prieto, A.: Diseño de Redes Neuronales Artificiales Mediante Algoritmos Evolutivos. Revista Iberoamericana de Inteligencia Artificial no. 14, pp. 2–32 (2001)

ISSN 1870-4069

Variante eficiente de red neuronal pulsante con auto-onda dinámica para resolver el problema del camino más corto

Oliver Espinosa, Manuel Mejía-Lavalle, José Ruiz, Gerardo Reyes, Alicia Martínez

Tecnológico Nacional de México / CENIDET,
Cuernavaca, Morelos, México
{oliver.espinosa18ce, mlavalle,
josera,greyes,amartinez}@cenidet.edu.mx

Resumen. Desde hace aproximadamente 10 años se ha estudiado el comportamiento de las Redes Neuronales Artificiales Pulsantes en el campo de la Optimización de Trayectorias, específicamente en el problema de la ruta más corta. Sin embargo, existen modelos de Redes Neuronales Pulsantes que necesitan un gran número de iteraciones antes de encontrar la ruta más corta entre un punto y otro. En este artículo se presenta una variante de Red Neuronal Pulsante para resolver el problema de la ruta más corta de manera eficiente. Esta variante cuenta con una velocidad dinámica de propagación de auto-onda, la cual se ajusta de manera heurística para evitar iteraciones donde no hay cambios en la Red. Para mostrar la eficiencia del modelo, se realizan experimentos y se comparan los resultados contra los obtenidos por otros modelos de Redes Neuronales Pulsantes. Para la comparativa se utilizan paradigmas que emplean una velocidad de auto-onda estática y modelos con una velocidad de auto-onda dinámica.

Palabras clave: red neuronal pulsante, problema del camino más corto, optimización, auto-onda dinámica.

Efficient Variation of Pulsed Neural Network with Dynamic Auto-Wave to Solve the Shortest Path Problem

Abstract. For about 10 years the behavior of Pulsed Neural Network has been studied in the optimization field of Shortest Path Problem. However, there are models of Pulse Coupled Neural Network that need a large number of iterations before to find the shortest path between two points. This paper presents a variation of Pulse Coupled Neural Network to solve the shortest path problem in an efficient way. This variant has a dynamic auto-wave propagation speed, which adjusts in a heuristic way to avoid iterations where there are no changes in the network. To show the efficiency of the model, experiments are performed and the results are compared against two other models of Pulsating Neural Networks. In the comparison, paradigms that use a static auto-wave speed and models with a dynamic auto-wave speed are used.

Keywords: pulsed neural network, shortest path problem, optimization, dynamic auto-wave.

1. Introducción

El campo de optimización de trayectorias tiene como propósito encontrar la mejor trayectoria que le permita resolver una tarea específica. Uno de los problemas más estudiados de la optimización de trayectorias es el problema del camino más corto o *Shortest Path* (SP, por sus siglas en inglés). El objetivo de resolver el problema del camino más corto es encontrar el camino de menor costo entre una fuente dada y un destino en una red determinada, es decir, encontrar la ruta más corta en un grafo de un vértice a otro. Aquí, el "más corto" significa que el costo total que se correlaciona con la ruta debe ser el menor [1].

La teoría de grafos siempre se utiliza como la base matemática para la solución de este tipo de problemas. En un grafo $G = (V, E)$, donde la distancia entre los vértices v_i y v_j es w_{ij}, se requiere encontrar el camino más corto entre un vértice de inicio s y un vértice meta g. En grafos con pocos nodos la solución podría resultar sencilla, pero con grafos de mayor magnitud, la complejidad computacional se vuelve alta. Los algoritmos tradicionales generalmente tienen complejidad exponencial, siendo un problema NP.

Existen diversas soluciones algorítmicas para resolver el problema del camino más corto. Desde las soluciones exhaustivas como la Búsqueda en anchura o la Búsqueda en Profundidad, hasta las soluciones con mayor eficiencia como el *Branch & Bound* [2], el algoritmo A* [3], Dijkstra [4] y el algoritmo Bellman-Ford [5]. El problema del camino más corto tiene diversas aplicaciones en la robótica, las telecomunicaciones, el transporte, la teoría de juegos, las redes de computadoras, búsquedas *web*, entre otros.

Desde el trabajo original de Hopfield [6] en 1985, se han realizado diversas investigaciones utilizando las Redes Neuronales Artificiales (RNA) para resolver problemas de optimización combinatoria. En el presente artículo se propone utilizar el enfoque de las RNA para resolver el problema SP, en particular de las *Pulse-Coupled Neural Network* (PCNN, por sus siglas en inglés). Las PCNN son redes no supervisadas, forman una red auto organizada que no requiere entrenamiento. Las PCNN son capaces de emular el comportamiento de las neuronas corticales como se observa en las cortezas visuales de los mamíferos [7].

Ha aumentado el interés en el uso de PCNN para diversas aplicaciones, como el reconocimiento de objetivos [8], el procesamiento de imágenes [9], el reconocimiento de patrones [10] y reducir el efecto de ruido gaussiano [11]. Algunas investigaciones recientes muestran que la dinámica espacio temporal de las PCNN proporciona una buena capacidad computacional para resolver algunos problemas de optimización. En 1999, Caulfield y Kinser [12] presentaron la idea de utilizar la onda automática en PCNNs para encontrar la solución al problema del laberinto. Su modelo puede encontrar la ruta más corta rápidamente y con el mínimo esfuerzo, donde la solución está relacionada con la longitud de la ruta más corta e independiente de la complejidad del gráfico de la ruta. Sin embargo, se necesitan muchas neuronas para encontrar la ruta más corta en grandes laberintos o gráficos, ya que una neurona corresponde a una unidad de longitud de ruta. Posteriormente Qu [13] presenta el modelo *Modified Pulse-*

Coupled Neural Network (MPCCN), el cual tiene un costo computacional menor. El número de iteraciones de la MPCNN es proporcional a la longitud del camino más corto. Ma [1] publicaron el modelo *Auto-Wave Neural Network* (AWNN), el cual aprovecha la característica de auto-onda de las PCNN para resolver el problema de SP. Paredes-Cano [14] presentaron una mejora del modelo AWNN además de una técnica para explicitar el conocimiento generado por la red neuronal. Los modelos mencionados anteriormente sugieren que la velocidad de la onda debe mantener una velocidad constante. Posteriormente se presentaría un modelo llamado *Self-adaptive autowave pulse-coupled neural network* (SAPCNN) [15]. Este modelo puede ajustar la velocidad de propagación de la onda automáticamente de acuerdo con el estado actual de la red. Adicionalmente los resultados indican que el modelo logra calcular el camino más corto con menor cantidad de iteraciones.

En este artículo se propone el modelo *Self-adaptive autowave modified pulse-coupled neural network* (SAM-PCNN), el cual es un modelo modificado de PCNN que es capaz de adaptar la velocidad de onda de manera más eficiente que la SAPCNN. Los resultados de la experimentación muestran que el modelo SAM-PCNN tiene mayor eficiencia que los modelos que cuentan con una velocidad constante de onda, además de que también supera al modelo SAPCNN, el cual cuenta con una velocidad dinámica auto-onda.

El articulo se organiza de la siguiente manera. En la Sección 2 se hace una breve revisión de los modelos de PCNN para optimización de trayectorias. En la Sección 3 se hará una descripción del modelo propuesto SAM-PCNN. En la Sección 4 se describen los experimentos y los resultados obtenidos. En la Sección 5 se discuten los resultados y en la sección 6 se concretan las conclusiones y el trabajo futuro.

2. Modelos de PCNN para optimización de trayectorias

2.1. Auto-Wave Neural Network

Se analizaron diferentes modelos de PCNN para su implementación, el primero de ellos es el modelo AWNN (*Auto-Wave Neural Network*, por sus siglas en inglés) presentado en [1]. Dicho modelo está diseñado para resolver el problema de la ruta más corta. Utiliza como entrada la matriz de adyacencias de un grafo ponderado. En la Figura 1 se muestra una neurona de este modelo, el cual está compuesto por la capa de recepción, de enlace y de pulsos. La capa de enlace está formada por la combinación de W y F. La matriz U representa la capa de enlace mientras que la capa de pulsos está compuesta por el umbral E y la salida Y.

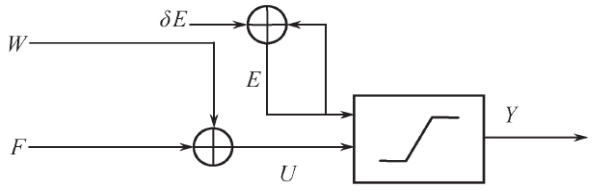


Fig. 1. Modelo de la neurona *Auto-Wave Neural Network* [1].

Cada neurona recibe como entrada información de la matriz de adyacencia W y además recibe un valor F del costo adicional en caso de continuar por cierta ruta: sumando W y F se obtiene el valor U que es el costo total incurrido por algún camino hasta un momento dado determinado por el umbral dinámico T. En cada época el parámetro E se incrementa por un valor constante δE: si el costo acumulado en U es igual o supera al umbral T, entonces la neurona se activa tomando la salida Y el valor de "1", de otra manera la salida de la neurona será cero.

Dado un problema con n nodos, la red neuronal tendrá entonces tantas neuronas como $n \times n$. El efecto de auto-onda se obtiene precisamente al ir realizando incrementos en el tiempo del umbral T. Las siguientes ecuaciones resumen el comportamiento de la neurona AWNN.

$$F_i[n] = F_h[n] + W_{hi}, \tag{1}$$

$$U_{ij}[n] = F_i[n] + W_{ij}, \tag{2}$$

$$Y_{ij} = \begin{cases} 1\ T[n] \geq U_{ij}[n] \\ 0\ de\ otra\ manera \end{cases}, \tag{3}$$

$$T[n] = T[n-1] + \delta E. \tag{4}$$

Se seleccionó el modelo AWNN por que cuenta con un mecanismo sencillo y fácil de comprender. Dicho modelo es un buen candidato para comenzar a entender la manera en que las Redes Neuronales Pulso Acopladas son empleadas en problemas de optimización de trayectorias.

2.2. Auto-Wave Neural Network explicitada

Este modelo de PCNN surge a partir de un artículo realizado en CENIDET [14]. Es una extensión del modelo AWNN presentado en [1]. Para hacer más eficiente el proceso los autores proponen un valor inicial de E como el costo menor del nodo de inicio a sus siguientes nodos. Está modificación permite que desde la primera interacción pulse por lo menos una neurona.

Adicionalmente se propone un método para recuperar la ruta calculada por la red neuronal. La explicitación del conocimiento es un sub-tema dentro de las RNA y se debe a que, en el surgimiento de la tecnología, existía la crítica de que aunque los paradigmas neuronales daban buenos resultados, no era posible acceder al conocimiento que de manera automática el algoritmo había generado. Se decía entonces que las RNA eran algoritmos de caja negra. Sin embargo, con el tiempo, han surgido propuestas exitosas para precisamente transparentar el conocimiento oculto entre las neuronas, de tal manera que el usuario tenga acceso a, por ejemplo, las reglas de producción que se generan de manera automática como resultado de la actividad interna de la RNA. En ese sentido, en [14] se logra rescatar el camino óptimo o de menor costo, extrayendo el conocimiento que queda oculto en la capa de salida Y de la RNP propuesta.

La capa de salida Y es básicamente una matriz de tamaño $n \times n$ con valores de ceros y unos. La ruta se recupera de manera inversa, saltando del nodo final al nodo inicial. En la matriz de salida Y, el nodo final se encontrará en la columna i con el mismo número que el nodo meta. En esa columna se buscará el valor "1" y ahí donde aparezca

se tomará el valor del renglón j. Con ese valor de renglón se entrará nuevamente a la matriz Y pero ahora buscando en la columna con valor j, el valor "1" y repitiendo el proceso hasta llegar al nodo inicial. Al final los valores j así recuperados definirán la ruta de costo mínimo.

Para ilustrar este proceso de explicitación de conocimiento se presenta un ejemplo. En un grafo de cinco nodos se busca la trayectoria más corta del nodo A al nodo D. Una vez que la red neuronal ha calculado la trayectoria que se muestra la Figura 2, la matriz de salida Y es tal cómo se muestra en la Tabla 1.

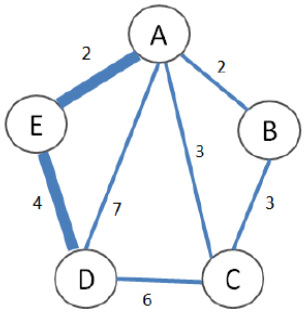


Fig. 2. Grafo del ejemplo que muestra la solución de la RNA [14].

Tabla 1. Matriz de salida del ejemplo [14].

Y	A (i1)	B (i2)	C (i3)	D (i4)	E (i5)
A (j1)	1	1	1	0	1
B (j2)	0	0	0	0	0
C (j3)	0	0	0	0	0
D (j4)	0	0	0	0	0
E (j5)	0	0	0	1	0

El nodo final D se encuentra en $i = 4$ y $j = 5$. Usando el valor de j como referencia, vamos a la columna $i = 5$, correspondiente al nodo E, y en donde el elemento marcado con "1" se encuentra en el renglón $j = 1$. Usando el valor de j una vez más como referencia, vamos a la columna $i = 1$, correspondiente al nodo A y en donde el elemento marcado con "1" se encuentra en el renglón $j = 1$. En este momento se termina la explicitación pues se alcanzó el nodo inicial. De esta manera se recupera el camino de menor costo que en este caso es $A - E - D$.

2.3. Self-Adaptive Autowave Pulse-Coupled Neural Network

El modelo SAPCNN [15] propaga la onda de forma dinámica según el estado de la red. Esto representa una mejora, pues la mayoría de los métodos sugieren que la onda

automática de los modelos PCNN debe mantener una velocidad constante para encontrar los caminos más cortos. En la Figura 3 se muestra el diagrama del modelo.

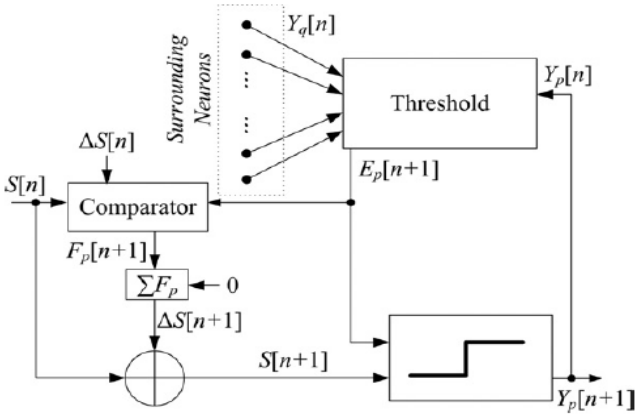


Fig. 2. Diagrama del modelo SAPCNN [15].

En el diagrama se puede ver que las salidas de la matriz de pulsos Y son las entradas de la neurona, las cuales se reciben el umbral $E[n]$. La actividad interna $S[n]$, la cual se incrementa por $\Delta S[n]$ en cada iteración.

$\Delta S[n]$ puede adquirir diferentes valores dependiendo del *comparator* como aparece en el diagrama de F_P que a su vez depende de la relación que existe entre el umbral $E[n+1]$ y $S[n]+\Delta S[n]$. En las ecuaciones se puede encontrar una descripción más detallada del comportamiento de este modelo.

$$Ep[n+1]=\begin{cases} V_E \; if \; Y_P[n]=1 \\ \min\left(S[n]+W_{qp}[n], Ep[n]\right) if \; Y_P[n]=0\,, Y_q[n]=1 \text{ y } p, \in R_q \\ E_p \; de \; otra \; manera \end{cases} \tag{5}$$

$$F_p[n+1]=\begin{cases} N \; if \; E_p[n+1] < S[n]+\Delta S[n] \\ -1 \; if \; E_p[n+1]=S[n]+\Delta S[n] \\ 0 \; de \; otra \; manera \end{cases}, \tag{6}$$

$$\Delta S[n]\begin{cases} \gamma \; if \; F_m > 0 \\ \Delta S[n]+\gamma \; if \quad F_m = 0, \\ \Delta S[n] \; if \; de \; otra \; manera \end{cases} \tag{7}$$

$$S[n+1] < S[n]+\Delta S[n], \tag{8}$$

$$Y_p[n+1]=\begin{cases} 1 \; if \; S[n+1] \geq Ep[n+1] \\ 0 \; de \; otra \; manera \end{cases}. \tag{9}$$

Este modelo está diseñado para resolver el problema del camino más corto, sin embargo, los autores reportan que también puede ser usado para la planeación de trayectorias para robots móviles.

3. Variante propuesta Self-Adaptive Autowave Modified Pulse-Coupled Neural Network (SAM-PCNN)

El objetivo del modelo SAM-PCNN que aquí se propone es resolver el problema del camino más corto o *Shortest Path* (SP, por sus siglas en inglés) en un grafo ponderado no dirigido en la menor cantidad de iteraciones. Cuenta con una velocidad de onda dinámica que puede adaptarse dependiendo del estado de la red. Sin embargo, a diferencia de la SAPCNN [15], el modelo SAM-PCNN logra adaptar la velocidad de onda de manera exacta en cada iteración, logrando eliminar las iteraciones donde no hay cambios en la Red.

El diagrama del modelo propuesto se presenta en la Figura 4. Se puede apreciar que la neurona se alimenta de las salidas del resto de las neuronas y se retroalimenta de su propia salida para calcular el umbral. Posteriormente se calcula la velocidad dinámica de la auto onda, la cual ayuda a la actividad interna de la Red a alcanzar el umbral de por lo menos una neurona en cada iteración.

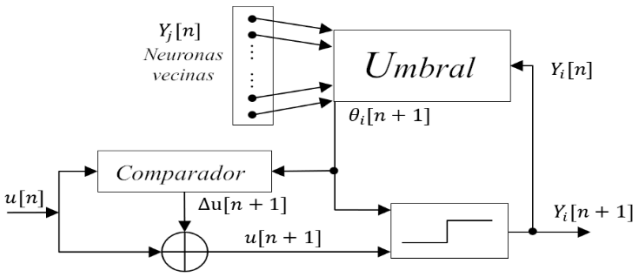


Fig. 3. Diagrama de la neurona del modelo propuesto SAM-PCNN.

Para explicar con mayor claridad el modelo propuesto, es necesario definir las notaciones más relevantes del tema. Los índices i y j se refieren a diferentes neuronas. Para hacer referencia a los vecinos de la neurona i se utiliza R_i. W_{ij} es la distancia positiva diferente de cero que existe entre la neurona i y la neurona j sólo sí $j \in R_i$, de lo contrario, W_{ij} es infinito. N es el número total de neuronas.

La descripción matemática del modelo SAM-PCNN es la siguiente: cada una de las neuronas tiene una salida binaria Y_i la cual se calcula de acuerdo con la ecuación (10).

$$Y_i[n+1] = \begin{cases} 1 \; if \; u[n+1] > \theta_i[n+1] \\ 0 \;\; de \; otra \; manera \end{cases}. \tag{10}$$

Si $Y_i[n] = 1$, se dice que la neurona i ha pulsado en la iteración n, u es la actividad interna de la red y θ_i es el umbral de la neurona, el cual se calcula con la ecuación (11).

$$\theta_i[n+1] = \begin{cases} V_\theta & if \; Y_i[n] = 1 \\ \min\left(u[n] + W_{ji}[n], \theta_i[n]\right) & \text{if } Y_i[n] = 0, Y_j[n] = i \; y \; i \; \in R_j, \\ \theta_i[n] & de \; otra \; manera \end{cases} \tag{11}$$

$$V_\theta = NW_{max}. \tag{12}$$

El umbral θ_i puede tomar tres valores diferentes dependiendo de las salidas de la Red. V_θ es un valor alto que el umbral toma cuando la neurona tiene una salida positiva. Se calcula con la Ecuación 12 multiplicando el número total de neuronas N por la máxima distancia de $W.V_\theta$ evita que la neurona pulse nuevamente. Si la neurona vecina j pulsa, se busca el valor mínimo entre $u[n] + W_{ji}[n]$ y el valor actual del umbral $\theta_i[n]$.

La parte novedosa de este modelo es la manera en que se comporta la actividad interna u. El mecanismo que permite que u incremente de manera dinámica de tal forma que pueda alcanzar el umbral de por lo menos una neurona en cada iteración, es el que se muestra en la ecuación (13) y se muestra en el diagrama como Comparador.

$$\Delta u[n+1] = \min(x) - u[n] \qquad (\forall x \in \theta[n+1] | x > u[n]). \tag{13}$$

En la expresión anterior se muestra que el incremento de u es Δu. Se calcula con la diferencia entre el valor mínimo de las x y u. Siendo x un elemento de los umbrales θ mayores que u. El cálculo de $u[n+1]$ se muestra en la ecuación (14).

$$u[n+1] = u[n] + \Delta u[n+1]. \tag{14}$$

En la ecuación (15) se muestra el cálculo de W la representa la matriz de distancias. Cuando una neurona ha pulsado se modifican las distancias que llevan a esa neurona por el valor de V_θ.

$$W_{ji}[n+1] = \begin{cases} V_\theta \; if \; Y_i[n+1] = 1 \text{ y } i \in R_j \\ W_{ji}[n] \; de \; otra \; manera \end{cases}. \tag{15}$$

Si una neurona i es alcanzada por la auto-onda proveniente de una neurona j, se guarda la neurona j como el padre de la neurona i utilizando la notación R_i^P. Cuando la onda de disparo alcanza la posición de la neurona meta, la neurona meta se disparará y se determinará una ruta desde la meta a la neurona inicio moviéndose sucesivamente hacia el padre de cada neurona. Este será el camino mas corto desde la neurona de inicio a la neurona meta.

3.1. Pseudocódigo SAM-PCNN

Para tener mayor claridad del funcionamiento del modelo propuesto, se presenta el siguiente pseudocódigo para mostrar los pasos de la red propuesta cuando se utiliza para resolver el problema del camino más corto. Aquí la neurona de inicio se representa con una s, la neurona meta se representa con una g. Uno de los aspectos importantes de este algoritmo y de las PCNN en general, es que se puede hacer en paralelo, lo que resulta en un cálculo más eficiente, especialmente con búsquedas de gran magnitud.

---

1) Inicializar la red: V_θ de acuerdo con la Ecuación 12, $n = 0$, $Y_i[0] = 0$, $u[0] = 0$, $\Delta u[0] = 0$, $Y_s[0] = 1$, $\theta_s[0] = V_\theta$,
2) Para cada neurona i en la SAM-PCNN:
    a. Calcular $\theta_i[n+1]$ de acuerdo con la Ecuación 11.
    b. Calcular la velocidad de la auto-onda $\Delta u[n+1]$ de acuerdo con la Ecuación 13.
    c. Calcular la actividad interna $u[n+1]$ de acuerdo con la Ecuación 14.

d. Calcular $Y_i[n+1]$ de acuerdo con la Ecuación 10.
e. Si $Y_i[n+1]=1$.
   i. Actualizar $W_{ji}[n+1]$ con forme la Ecuación 15.
   ii. $\theta_i[n+1]=V_\theta$.
   iii. Guardar los padres de las neuronas que han sido estimuladas
f. Si la neurona g ha pulsado ir a paso 3, de lo contrario $n=n+1$ y regresar al paso 2.

3) Explicitar la ruta generada buscando en el vector de neuronas padre desde la neurona g hasta que la neurona padre $R_i^P=s$.

3.2. Caso de ejemplo ilustrativo

Tomando en cuenta el grafo que se presenta en la Figura 5, se busca el camino más corto del nodo de inicio A al nodo meta C. El algoritmo SAPCNN [15] tiene el siguiente comportamiento:
Al inicializar la red en la iteración $n=0$, se hace pulsar el nodo A

1. Cuando $n=1$: El valor de la actividad interna calculado por las Ecuaciones 6, 7 y 8 es $S[1]=2$, y no le permite alcanzar el umbral de ninguna de las neuronas vecinas.
2. Cuando $n=2$: La actividad interna toma el valor $S[2]=3$, permitiéndole a la neurona B generar un pulso en su salida $Y_B[2]=1$.
3. Cuando $n=3$: $S[3]=5$ y no alcanza al umbral de la neurona C.
4. Cuando $n=2$: La actividad interna es $S[4]=8$ y $Y_C[4]=1$, dando por finalizada la búsqueda.

En comparación con el modelo SAPCNN [15], el modelo SAM-PCNN es capaz de encontrar esta trayectoria en menos iteraciones. En la iteración $n=0$ los valores de la red son: $\theta_s[0]=V_\theta, Y_s[1]=1, u[0]=0$. A partir de ahí, la Red SAM-PCNN se comporta de la siguiente manera:

1. Cuando $n=1$:
   Ecuación 11 cálculo de umbrales: $\theta_B[1]=3, \theta_C[1]=10$
   Ecuación 13 cálculo de velocidad de auto-onda: $\Delta u[1]=3$
   Ecuación 14 cálculo de actividad interna: $u[1]=3$
   Ecuación 10 se obtiene salida: $Y_B[1]=1$
   Actualizar umbrales $\theta_B[2]=V_\theta$
   Guardar padres: $R_B^P=s$
2. Cuando $n=2$:
   Ecuación 11 cálculo de umbrales: $\theta_C[2]=8$
   Ecuación 13 cálculo de velocidad de auto-onda: $\Delta u[2]=5$
   Ecuación 14 cálculo de actividad interna: $u[2]=8$
   Ecuación 10 se obtiene salida: $Y_C[2]=1$
   Actualizar umbrales $\theta_C[2]=V_\theta$
   Guardar padres: $R_C^P=B$

Se terminan las iteraciones de la Red ya que $Y_C = 1$. Se obtiene la ruta calculada recorriendo el vector de padres empezando por la neurona meta: $C \rightarrow R_C^P = B \rightarrow R_B^P = A$. Se hace notar que, para este caso, AWNN [14] requeriría ocho iteraciones.

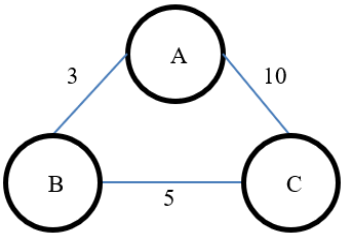


Fig. 4. Grafo del ejemplo.

4. Experimentación y resultados

En esta Sección se presentan experimentos que permiten ilustrar de mejor manera la eficiencia del modelo SAM-PCNN. Para realizar los experimentos se estudiaron cuidadosamente y se implementaron los modelos de PCNN propuestos por Ma [1], Paredes-Cano [14] y Li [15]. La implementación de los algoritmos se hizo en C++ de manera secuencial, en un equipo de cómputo con un microprocesador i5-7200U de séptima generación @ 2.50 GHz 2.70 GHz con 8 GB de memoria RAM. Para el registro del tiempo de procesamiento se presenta un tiempo promediado de 100 repeticiones.

4.1. Experimento 1

Para el primer experimento se utilizó un grafo asimétrico con diez nodos y veinte aristas presentado en Li [15] el cual se muestra en la Figura 6. Se buscaron 10 caminos diferentes dentro del grafo. Las búsquedas se originaron desde todos los nodos al nodo 10 y del nodo 10 al nodo 1. En la Tabla 2 se presentan los resultados de cada uno de los algoritmos al realizar estas búsquedas.

En la tabla 2 se puede apreciar que los algoritmos AWNN [1] y AWNN Explicitada [14], necesitan una mayor cantidad de iteraciones para el cálculo de la ruta. Al modelo AWNN [1] le toma una iteración por cada unidad de distancia. Es decir, si la distancia del camino más corto es 15 como en el ejercicio 1, requerirá de 15 iteraciones para resolver el problema. Al modelo AWNN Explicitada [14] le toma menos iteraciones debido a que la velocidad de la auto onda es dinámica en la primera iteración y posteriormente se vuelve constante.

Al analizar los resultados del modelo SAPCNN [15], es notorio que puede encontrar el camino más corto en menos iteraciones, ya que cuenta con una auto-onda que se propaga de manera dinámica dependiendo del estado de la Red, lo que le permite a su actividad interna cumplir fácilmente con las condiciones necesarias para generar pulsos en las neuronas de la Red.

Sin embargo, a pesar del mecanismo dinámico de este modelo, aún existen iteraciones donde no hay cambios en la Red, es decir, iteraciones donde ninguna de sus neuronas pulsa.

Tabla 1. Resultado de las iteraciones por algoritmo del Experimento 1.

			Iteraciones			
Experimento	Ruta	Costo	AWNN [1]	AWNNE [14]	SAPCNN [15]	SAM-PCNN
1	1-6-10	15	15	11	9	**6**
2	2-7-10	16	16	16	10	**6**
3	3-8-10	20	20	17	13	**8**
4	4-9-10	14	14	8	8	**4**
5	5-9-10	10	10	8	5	**3**
6	6-10	10	10	6	6	**4**
7	7-10	14	14	13	9	**6**
8	8-10	16	16	13	11	**8**
9	9-10	7	7	5	4	**2**
10	10-6-1	15	15	9	9	**4**
		Total	137	106	84	**51**
Diferencia con respecto a SAM-PCNN			+86	+55	+33	**0**

Por otro lado, el modelo propuesto SAM-PCNN presenta una menor cantidad de iteraciones para encontrar el camino más corto. El modelo cuenta con un mecanismo de auto onda que adapta su velocidad de manera heurística, lo cual le permite optimizar el número de iteraciones necesarias para el cálculo de la ruta.

En total el modelo propuesto SAM-PCNN hizo 86 iteraciones menos que el modelo AWNN [1], 55 iteraciones menos que el modelo AWNN Explicitada [14] y 33 iteraciones menos que SAPCNN [15].

En la Figura 7 se muestra de manera gráfica la comparativa de las iteraciones realizadas por cada modelo en cada uno de los experimentos. Las figuras en negro representan los resultados del modelo AWNN [1] con respecto de los demás modelos.

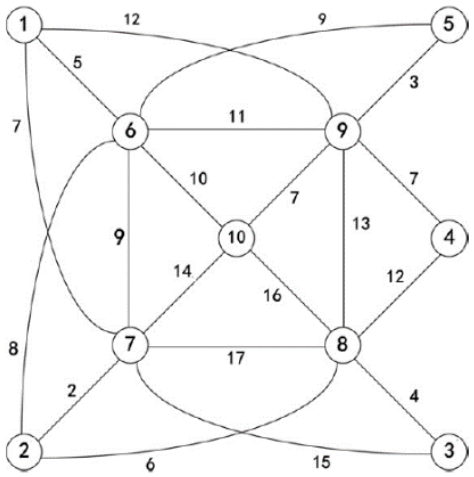


Fig. 6. Grafo asimétrico con 10 nodos y 20 aristas [15].

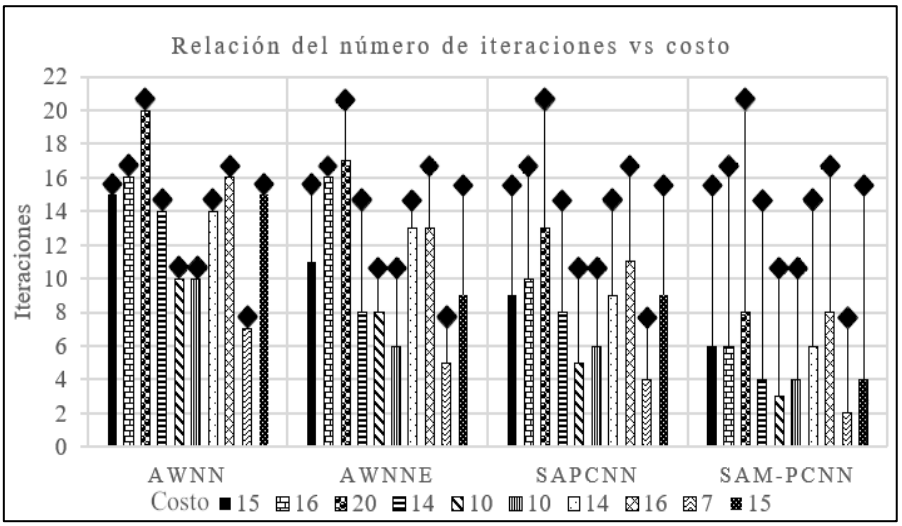


Fig. 7. Gráfica comparativa del número de iteraciones contra el costo o la distancia óptima en el Experimento 1.

4.2. Experimento 2

Para el segundo experimento se generaron aleatoriamente 4 grafos de 30, 50, 200 y 500 nodos respectivamente. Dichos grafos fueron generados tal como se especifica en [16]. El objetivo del experimento fue buscar el camino más corto entre el nodo 1 y el último nodo de cada grafo. El resultado de las iteraciones de cada algoritmo en cada grafo se muestra en la figura 8.

En la tabla 3 se presentan los tiempos de procesamiento en segundos obtenidos en cada una de las búsquedas.

Los resultados de la Tabla 2, muestran que, con grafos con pocos nodos, la diferencia en tiempo es apenas perceptible, pero cuando se trata de grafos de mayores dimensiones, la diferencia es considerable. Por otro lado, se puede apreciar que el tiempo de procesamiento del modelo SAM-PCNN tiene un comportamiento lineal con respecto al número de nodos.

Tabla 2. Comparativa de tiempos de procesamiento en segundos del Experimento 2.

	Nodos			
Algoritmo	30	50	200	500
AWNN	0.005914	0.002860	0.272663	13.8336
AWNN E	0.006121	0.002719	0.268951	13.7898
SAPCNN	0.002203	0.001512	0.111357	4.20460
SAM-PCNN	**0.002172**	**0.001152**	**0.111235**	**1.69298**

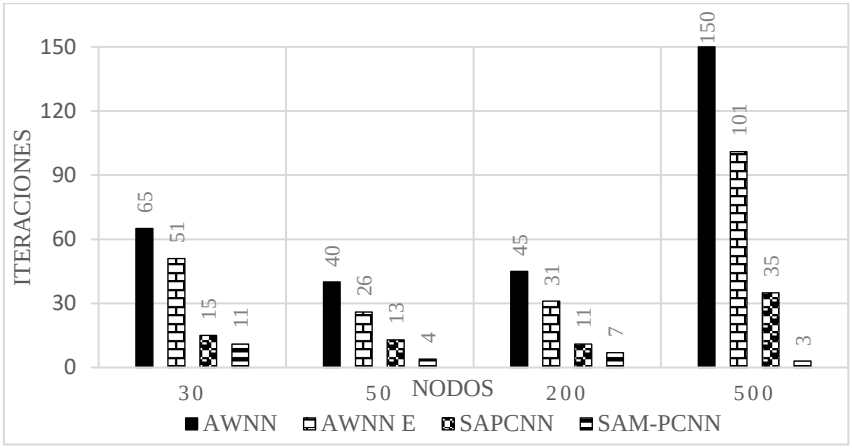


Fig. 8. Comparativa de iteraciones del Experimento 2.

5. Discusión

Encontrar la solución de la ruta óptima es un problema NP completo, incluso el muy citado algoritmo Dijkstra tiene complejidad algorítmica de $O(N^2)$, donde N es el número de nodos de un grafo G. Para los modelos de PCNN como el modelo AWNN [1], el cual cuenta con una velocidad constante de propagación, la complejidad algorítmica es $O(LN)$, donde L es equivalente a la distancia entre un nodo de inicio y el nodo más lejano. En el caso de los modelos SAPCNN [15] y SAM-PCNN, los cuales cuentan con una velocidad dinámica de propagación, la complejidad algorítmica es $O(N)$, debido a que para cada iteración es necesario calcular los umbrales, la velocidad de la auto onda y la actividad interna. Tomando en cuenta que M es el número de iteraciones necesarias para encontrar la solución, la complejidad de ambos modelos es $O(MN)$. Sin embargo, para el modelo propuesto SAM-PCNN, el número de iteraciones es proporcional al número de neuronas cuya distancia al nodo de inicio sea menor que la distancia del nodo objetivo. Es decir, si se desea calcular el camino más corto del nodo A al nodo C, pero el nodo B es más cercano al nodo A, el número de neuronas más cercanas que C es 1 que se puede representar como P. En este caso M es igual a $P + 1$, pues en la primera iteración pulsará la neurona B y en la siguiente iteración pulsará la neurona C. Por lo tanto, la complejidad algorítmica para el modelo propuesto SAM-PCNN puede re-escribirse como $O((P + 1)N)$.

6. Conclusiones y trabajo futuro

En este artículo se presentó el modelo SAM-PCNN para resolver de manera eficiente el problema del camino más corto. Para poder realizar la experimentación, fue necesario estudiar e implementar los modelos AWNN [1], AWNN Explicitada [14] y SAPCNN [14]. Se mostró que el modelo propuesto no sólo es una buena alternativa para resolver

este tipo de problemas, sino que además resulta ser más eficiente que otros modelos de PCNN que persiguen el mismo objetivo.

De acuerdo con los resultados del experimento 1, el modelo SAM-PCNN es hasta 39.28% más eficiente con respecto del modelo SAPCNN [15] y hasta un 62.77% más eficiente que el modelo AWNN [1].

En esta investigación los algoritmos fueron implementados de manera secuencial, por lo que los resultados presentados pueden mejorar considerablemente con una implementación paralela. Esto último se tiene contemplado como parte del trabajo futuro relacionado con esta investigación, además de probar el modelo SAM-PCNN en otros ámbitos como la Planeación de Trayectorias Discretas para Robots Móviles.

Referencias

1. Ma, Y., Zhan, K., Wang, Z.: Pulse-Coupled Neural Networks. Applications of Pulse-Coupled Neural Networks. Springer Heidelberg Dordrecht London New York, pp. 447–465 (2011)
2. Morrison, D. R., Jacobson, S.H., Sauppe, J.J., Sewell, E.C.: Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization, 19, pp. 79–102 (2016)
3. Thulasiraman, K., Arumugam, S., et.al.: Handbook of graph theory combinatorial optimization, and algorithms. CRC Press (2016)
4. Dijkstra, E.W.: A note on two problems in connection with graphs, Numer. Math. 1(1), 269–271 (1959)
5. Cherkassky, B.V., Goldberg, A.V., Radzik, T.: Shortest paths algorithms: theory and experimental evaluation, Math. Programming 73 (2), 129–174 (1996)
6. Hopfield, J.J., Tank, D.W.: Neural computation of decisions in optimization problems. Biol. Cybern, vol. 52, pp. 141–152 (1985)
7. Eckhorn, R., Reitboeck, H.J., Arndt, M., Dicke, P.W.: Feature linking via synchronous among distributed assemblies: Simulations of results from cat visual cortex. Neural Comput, vol. 2, pp. 293–307 (1990)
8. Ranganath, S.H., Kuntimad, G.: Object detection using pulse coupled neural networks. IEEE Trans. Neural Netw 10(3), 615–620 (1999)
9. Gu, X., Yu, D., Zhang, L.: Image shadow removal using pulse coupled neural network. IEEE Trans. Neural Netw 16(3), pp. 692–698 (1999)
10. Muresan, R.C.: Pattern recognition using pulse-coupled neural networks and discrete Fourier transforms. Neurocomputing, vol. 51, pp. 487–493 (2003)
11. Ortiz, E., Mejía-Lavalle, M., Sossa, H.: Uso de redes neuronales pulsantes para mejorar el filtrado de imágenes contaminadas con ruido Gaussiano. Research in Computing Science, Vol 114, pp. 45–58 (2016)
12. Caulfield, H.J., Kinser: Finding M.: The shortest path in the shortest time using PCNNs. IEEE Trans. Neural Netw 10(3), 604–606 (1999)
13. Qu, H., Yi, Z.: A new algorithm for finding the shortest paths using PCNNs Chaos. Solutions Fractals 33(4), 1220–1229 (2007)
14. Paredes-Cano, J.J., Mejía-Lavalle, M., Mújica-Vargas, D., Reyes-Salgado, G.: Red Neuronal Pulsante Explicitada para el Problema del Camino más Corto, IEEE (2018)
15. Li, X., Ma, Y., Feng, X.: Self-adaptive autowave pulse-coupled neural network for shortest-path problema. Neurocomputing, vol. 115, pp. 63–71 (2013)
16. Qu, H., Yi, Z., Yang, S.X.: Efficient Shortest-Path-Tree Computation in Network Routing Based on Pulse-Coupled Neural Networks. IEEE Transactions on Cybernetics, 43(3), 995–1010 (2013)

ISSN 1870-4069

Planificación de movimientos basada en sensores para robots móviles en ambientes desconocidos

Valerio Galiote T., Abraham Sánchez L., José F. Texcucano D.,
Alfredo Toriz P., Rogelio González V., Elberfeld Pérez G.

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
Laboratorio MOVIS, Puebla, Pue., México

{val.gali,faustinotexcucano,alfredot,enrique5001}@hotmail.com
abraham.sanchez@correo.buap.mx,rgonzalez@cs.buap.mx

Resumen. La navegación basada en sensores permite a un robot explorar un ambiente desconocido y construir un mapa de él, utilizando su información del sensor. Este es uno de los temas de investigación tradicionales para robots móviles y se han propuesto muchos algoritmos a lo largo de los últimos años. En este trabajo, desarrollamos algunas estrategias para la planificación de movimientos basada en sensores en entornos desconocidos, basados en el enfoque de árboles aleatorios de exploración rápida (RRT). Estas estrategias exploran el área desconocida de manera similar al método del árbol aleatorio basado en sensores (SRT) y adaptan el enfoque RRT a la planificación del camino local en el espacio de estados libre asociado con la región segura. Nombramos nuestras estrategias Local SRRT y Local B SRRT. Se presentan resultados experimentales para mostrar el desempeño de las estrategias propuestas.

Palabras clave: RRT, SRT, planificación de movimientos, robots móviles.

Sensor-Based Motion Planning for Mobile Robots in Unknown Environments

Abstract. Sensor based navigation enables a robot to explore an unknown environment and build a map of it, using its sensor information. This is one of the traditional research subjects for mobile robots and many algorithms have been proposed. In this work, we developed some strategies for sensor-based motion planning in unknown environments, based in the rapidly-exploring random trees (RRT) approach. These strategies explore the unknown area in a similar way than the Sensor-based Random Tree (SRT) method and adapt the RRT approach to the local path planning in the free space state associated with the safe region. We named our strategies Local SRRT and Local B SRRT. Experimental results are presented, in order to show the performance of the proposed strategies.

Keywords: RRT, SRT, motion planning, mobile robots.

1. Introducción

La navegación ambiental está comprendida de dos comportamientos principales, llamados exploración y planificación de caminos. Explorar un ambiente desconocido y buscar un punto objetivo está más allá de las tareas básicas de un robot móvil autónomo. Ambos problemas han recibido una atención considerable en geometría computacional y robótica; ver [2,3].

El enfoque RRT, presentado en [1], se ha convertido en el referente más popular en los últimos años. Los algoritmos basados en RRT fueron primero desarrollados para problemas de planificación no holonómica y cinemática [4], donde el espacio a ser explorado es el espacio de estados. Un método para exploración basada en sensores de ambientes desconocidos fue presentado en [5], este método procede de la construcción de una estructura de datos llamada SRT. El SRT representa un roadmap del área explorada asociada a su región segura y estima el espacio libre percibido por el robot durante la exploración. El método SRT original trata con robots holonómicos, Espinoza et al., han presentado algunas alternativas para tratar con robots móviles no holonómicos [6,7]. En este trabajo se emplearon algunas estrategias, basadas en el enfoque RRT para la navegación basada en sensores en ambientes desconocidos.

Las estrategias de exploración denominadas Local SRRT y Local B SRRT, con dos variantes diferentes cada una de ellas, adaptan a los planificadores de RRT para que se ejecuten localmente en el área segura construida con la información proporcionada por los sensores. De manera incremental, estas estrategias construyen un RRT a medida que el sensor informa el espacio libre del entorno.

El documento está organizado de la siguiente manera. La Sección II explica los detalles de las estrategias propuestas para la exploración y la planificación de movimientos. La Sección III analiza el desempeño de los dos planificadores propuestos, Local SRRT Local y Local B SRRT. Finalmente, las conclusiones y el trabajo futuro se presentan en la Sección IV.

2. Exploración y planificador de movimientos

El objetivo principal del presente trabajo es tomar ventaja del potencial de un método como RRT [1,4] en temas de exploración y el manejo de restricciones no holonómicas y expandirlo a exploración de ambientes desconocidos usando robots móviles. La idea es construir un árbol aleatorio local (del inglés, LRT, Local Random Tree) que representa la conectividad de un espacio de estado libre asociado con una región segura detectada por el sensor. El árbol puede ser expandido incrementalmente con las nuevas percepciones del ambiente obtenidas con el sensor, antes de que el estado meta sea un nodo en la LRT, cuando el procedimiento encuentra un camino factible libre o un máximo número de iteraciones se alcanza y el proceso concluye.

Se representa la construcción de una estructura llamada LRT, donde cada nodo contiene un estado libre de colisiones, una entrada de control y una descripción de la región segura local asociada con el estado. Cada arista en el árbol significa que el robot puede moverse de un estado a otro usando el control de entrada. Los nodos del árbol pueden ser de dos tipos: 1) Intermedio, un nodo utilizado para el movimiento del robot

dentro de la región segura local (LSR, Local Safe Region) y 2) Terminal, un nodo donde se puede realizar un nuevo proceso de percepción del entorno.

```
CONSTRUIR(x_inic, K_max, I_max)
1   x_act = x_inic;
2   k ← 0;
3   mientras ( NO(META_EN_RSL(x_act)) o NO(x_act = NULL) o k < K_max))
4     S ← PERCEPCION(x_act);
5     AGREGA(T, (x_act, S));
6     i ← 0;
7     repetir
8       x_cand ← EDO_CANDIDATO(x_act);
9       ExisteCamino ← PLANIFICADOR_RRT(x_act, x_cand);
10      i ← i + 1;
11    hasta que (ExisteCamino o i = I_max)
12    si ExisteCamino entonces
13      MOVER_A(x_cand);
14      x_act ← x_cand;
15    sino
16      MOVER_A(x_act.padre);
17      x_act ← x_act.padre;
18    k ← k + 1;

19  si (META_EN_RSL(x_act)) entonces
20    ExisteCamino ← PLANIFICADOR_RRT(x_act, x_meta);
21    si ExisteCamino entonces
22      Regresa EXITO;
23  si (x_act=NULL)) entonces
24    "Robot en la posición inicial ";
25  Regresa FALLO;
```

Fig. 1. Algoritmo básico del SRT.

Opcionalmente, este nodo puede incluir la referencia a una lista de estados candidatos para un nuevo proceso de percepción del entorno. Esta lista es generalmente referenciada en los nodos terminales. La descripción del robot, los obstáculos y el LSR son poligonales [7]. El procedimiento comienza en el estado inicial con un proceso de percepción del entorno, para determinar la región segura local asociada al estado. Después de eso, verifica si el estado del objetivo está en el LSR; de lo contrario, un planificador RRT alcanza un estado candidato y el ciclo se repite, expandiendo el LRT y aumentando el entorno explorado. Cuando se agota la lista de candidatos, el robot volverá al estado anterior. Este proceso de movimiento hacia atrás puede mover el robot al estado inicial; En este caso el algoritmo termina con fallo. El procedimiento también terminará con fallo después de un número máximo de iteraciones. El algoritmo finaliza con éxito cuando el estado objetivo está en la región segura local actual y hay una ruta que conecta los estados inicial y final.

El algoritmo explora el área desconocida de manera similar al método SRT y adapta el enfoque RRT a la planificación del camino local en el espacio de estados libre asociado con la región segura local. En cada iteración *k* del algoritmo, ver la Figura 1, se realiza un proceso de percepción para obtener la región segura local circundante S del robot, en el estado actual, X_{act}. Se crea un nuevo nodo que contiene el estado y la descripción de la LSR y se agrega al árbol como un nodo terminal. El algoritmo verifica

si el estado objetivo X_{goal} está en la región segura local de X_{act}, si no, un procedimiento determina un estado candidato válido dentro de la región segura para llevar a cabo un nuevo proceso de percepción.

Las restricciones diferenciales del robot tipo carro se manipulan con el planificador RRT que se ejecuta localmente en el espacio de estados libre para encontrar un camino factible entre los estados actual y candidato. Cada nodo creado por el planificador local se agrega al árbol como un nodo intermedio. Si el proceso de planificación local es exitoso, el nodo candidato se agrega al árbol como un nodo terminal y el ciclo se repite; de lo contrario, el algoritmo busca otro estado candidato válido de X_{act}, que el planificador pueda vincular con éxito con el estado actual, a través de un camino factible, o hasta que se alcance un número de iteraciones máximas I_{max}.

Si este es el caso, el robot regresa al nodo principal de X_{act} y el ciclo comienza de nuevo. El proceso hacia atrás puede llevar al robot hasta el estado inicial, en este caso el algoritmo termina con un error, es decir, no es posible ejecutar más percepciones. De este algoritmo surgen diferentes estrategias de exploración, que están relacionadas con la elección de los estados candidatos y la construcción del árbol de exploración. En las siguientes subsecciones se describirán las estrategias Local SRRT (RRT basado en sensores) y Local B SRRT. Hay dos variantes del Local SRRT que adaptan la ejecución local del planificador RRT en el proceso de exploración basado en sensores. Las variantes difieren en la forma en que construyen el árbol de exploración.

EDO_CANDIDATO(X_{act}). Los estados candidatos para un nuevo proceso de percepción del ambiente son generados con la rutina EDO_CANDIDATO(X_{act}). En el SRT local, la elección de candidatos es simple y sigue los siguientes pasos:

1. Una dirección aleatoria se selecciona para el primer estado candidato, X_{cand} y debe de estar dentro de la LSR dejando un margen en la frontera para asegurar la movilidad del robot, como en el algoritmo presentado en [7] (Figura 2).
2. El estado es validado si está localizado a una distancia mayor de un mínimo establecido y no está localizado dentro de la LSR de un nodo.
3. Los estados candidatos siguientes pueden ser generados en direcciones aleatorias o se puede usar un método más determinista, localizarlos en direcciones equidistantes de X_{act}. También deber ser validados de acuerdo al paso 2.

Los estados candidatos validos son almacenados en una lista de nodos, donde se toma la percepción del ambiente. La descripción de la LSR se debe calcular y opcionalmente almacenarse. Cuando el robot se mueve de X_{act} a X_{cand}, puede incrementar la información del espacio libre añadiendo o calculando la descripción de las LSRs de los nodos previos. El número de candidatos previos fue seleccionado experimentalmente con buenos resultados.

En este punto se realizaron dos maneras de generar nodos candidatos, ambas formas fueron mencionadas previamente, ver las Figuras 2 y 3. La primera forma consiste en generar cinco nodos candidatos distribuidos en una dirección aleatoria a la misma distancia de X_{act} entre sí en un área desconocida. La segunda forma consiste en generar los cinco nodos candidatos para que sean distribuidos en una distancia equidistante de cada nodo cercano a la frontera de la LSR en un área desconocida. Con ambos

planteamientos se busca que los candidatos cerca de la frontera de áreas no conocidas sean válidos, y con la distribución homogénea en cinco direcciones se busca eliminar la posibilidad de no visitar una zona no explorada. Una vez seleccionado, el estado candidato es marcado como visitado para no volver y hacer una exploración por segunda vez en esa zona.

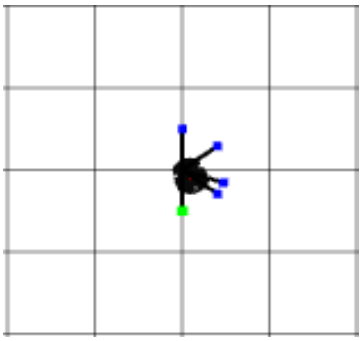


Fig. 2. Generación de nodos candidatos en direcciones aleatorias.

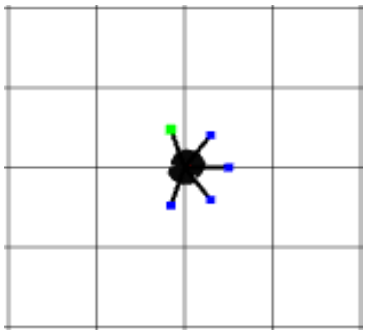


Fig. 3. Generación de nodos candidatos en direcciones equidistantes.

PLANIFICADOR_RRT. El enfoque del algoritmo RRT esta formulado en termino de 6 componentes:

1. Espacio de estados: Un espacio topológico, X.
2. Valores límite: $X_{ini} \in X$ y $X_{meta} \subset X$.
3. Detector de colisión: Una función, D: $X \rightarrow$ {verdadero, falso}, que determina si las restricciones globales son satisfechas desde el estado X_{act}. Esta podría ser una función binario o real.
4. Entradas: Un conjunto, U, que especifica el conjunto de controles o acciones que pueden afectar el estado.
5. Simulador incremental: Dado X_{act} y las entradas aplicadas sobre un intervalo de tiempo, $\{u\ (t') \mid t \leq t' \leq\ t + \Delta t\}$, calcular $X(t + \Delta t)$.

6. Métrica: Una función real, ρ: $X \times X \rightarrow [0, \infty)$, la cual especifica la distancia entre pares de puntos en X.

La planificación de caminos generalmente es vista como una búsqueda en el espacio de estados, X, para un camino continuo desde un estado inicial X_{ini} a un estado final X_{meta}. Se asume que se tiene un conjunto complicado de restricciones diferenciales sobre X y cualquier solución debe mantener al estado dentro de este conjunto. Un detector de colisión reporta si un estado dado, X, satisface las restricciones globales. Generalmente, se utiliza la notación, X_{libre} para referirse al conjunto de estados que satisfacen las restricciones globales. Se asignan restricciones locales y diferenciales a través de un conjunto de entradas (o controles) y de un simulador incremental. Estos dos componentes especifican los posibles cambios en el estado. El simulador incremental puede definirse por integración numérica de una ecuación de transición de estado. Finalmente, se define una métrica para indicar la cercanía de pares de puntos en el espacio de estados.

El planificador se desarrolló de acuerdo con el modelo de movimiento de velocidad (ver figura 4) el cual asume que podemos controlar al robot mediante 2 velocidades, una velocidad rotacional y una velocidad traslacional, donde denotamos la velocidad traslacional en tiempo t por *vt* y la velocidad rotacional por *wt*.

$$u_t = \begin{pmatrix} v_t \\ \omega_t \end{pmatrix}$$

Fig. 4. Modelo de movimiento de velocidad.

META_EN_LSR(X_{act}). Esta función regresa verdadero si X_{meta} se encuentra dentro de la LSR de X_{act}. Esto se puede validar calculando la distancia entre la posición de X_{act} con la posición de X_{meta} y comparándola con la longitud del rayo que une la posición actual con un punto en la frontera de la LSR de X_{meta}.

MOVER_A(X_{cand}). Esta rutina simplemente sigue las entradas de control generadas por el planificador local para llegar el estado X_{cand} desde el estado X_{act}. Opcionalmente en cada movimiento se actualiza la región segura con nuevas percepciones del ambiente.

EXISTE_CAMINO(). Esta función determina si existen nodos X_{cand} pertenecientes al nodo X_{act} y que no hayan sido visitados, en caso de existir nodos que aún no han sido visitados el nodo X_{act} se convierte en el nodo X_{cand} seleccionado, en caso contrario al ya no haber nodos por visitar el nodo X_{act} se convierte en el nodo X_{act}.*padre* generando así un encadenamiento hacia atrás.

PERCEPCIÓN(X_{act}). El ambiente es percibido mediante un sensor láser con un campo de visión de 240°, como ya se ha mencionado el número máximo de nodos X_{cand} es 5, se dividió el campo de visión en 5 regiones iguales, dando lugar a las 2 posibles maneras de generar un EDO_CANDIDATO(X_{act}), ver la figura 5.

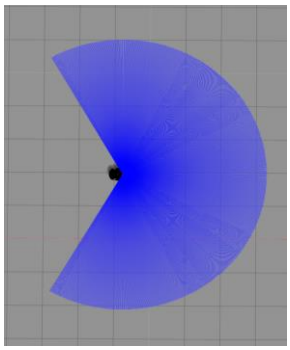


Fig. 5. Vista aérea del robot Pionner 3DX con el Laser Hokuyo en el simulador Gazebo.

3. Resultados experimentales

Esta sección presenta resultados experimentales del robot Pioneer 3DX. SRT fue implementado en la distribución de ROS Kinetic Kame, ROS agrega valor a muchos proyectos y aplicaciones en robótica. ROS fue diseñado para ser distribuido tan modular como sea posible, permitiendo al usuario utilizar mucho o poco de ROS como lo deseen o necesiten. La modularidad de ROS permite seleccionar que partes usar y que partes del proyecto desarrollar e implementar.

Las pruebas fueron desarrolladas en un procesador Intel CORE i7 7th Gen con 2.7 GHz con 16 GB RAM. Las variantes de estrategias de exploración son referenciadas como variante 1 y variante 2 de Local SRRT y Local B SRRT. Se utilizaron diferentes ambientes para poner a prueba la eficiencia de las variantes, ver las Figuras 6, 7, 8, 9, 10, y 11. Se utilizaron diferentes ambientes (Ambiente 1, Ambiente 2, y Ambiente 3) para poner a prueba la eficiencia de ambas variantes de Local SRRT, en las tablas 1 y 2 se muestran los resultados obtenidos respecto al número de nodos y el tiempo de ejecución. El tiempo de ejecución proporcionado por las simulaciones corresponde al tiempo total, incluyendo el tiempo de percepción del sensor.

Los caminos planificados con la variante uno de la estrategia de exploración Local SRRT, son factibles para los movimientos del robot móvil de tipo carro, pero son difíciles de seguir por que requieren muchas maniobras de conducción por parte del robot.

Esto sucede por la unión de pequeños arboles bidireccionales y la elección aleatoria del siguiente estado para un nuevo sensado, ambas características obligan al algoritmo a encontrar un camino factible en un espacio reducido que se traduce en muchas maniobras de conducción. La segunda variante de Local SRRT surgio motivada por este problema, ¿cómo obtener caminos que cumplan con las restricciones globales del ambiente y del robot y que además sean un poco menos complicados de seguir?

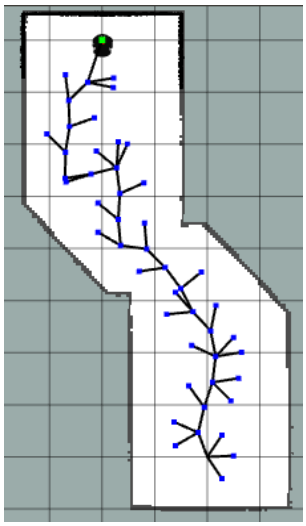


Fig. 6. Estrategia Local SRRT en el escenario 1.

Tabla 1. Resultados de Local SRRT.

	Ambiente 1	Ambiente 2	Ambiente 3
Nodos (mín.)	40	31	34
Nodos (máx.)	51	58	46
Tiempo (mín.)	119.12 segs.	77.12 segs	103.95 segs.
Tiempo (máx.)	240.01 segs.	371.56 segs.	331.78 segs.

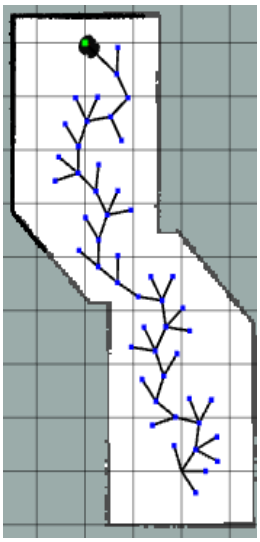


Fig. 7. Estrategia Local B SRSRT en el escenario 1.

Tabla 2. Resultados de Local B SRRT.

	Ambiente 1	Ambiente 2	Ambiente 3
Nodos (mín.)	38	27	31
Nodos (máx.)	51	44	49
Tiempo (mín.)	110.79 segs.	71.26 segs.	92.7 segs.
Tiempo (máx.)	238.8 segs.	262.2 segs.	348.3 segs.

El hecho de que la variante dos utilice al árbol aleatorio de exploración como árbol de planificación local da mayores oportunidades al planificador local de encontrar un nodo que conecte al estado candidato, sin obligarlo a construir caminos factibles en espacios reducidos, sino de aprovechar caminos ya construidos a lo largo de la región segura. En ocasiones el algoritmo ocupa un porcentaje de tiempo ligeramente mayor que la variante uno, en su búsqueda del nodo más cercano al cual conectar el nodo aleatorio producido por el planificador local.

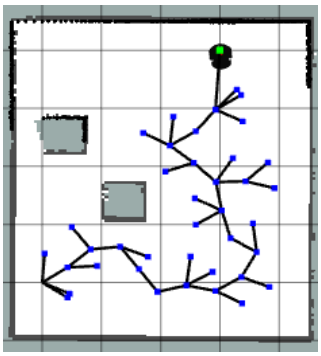


Fig. 8. Aplicación de la variante Local SRRT en el escenario 2 registrando un tiempo de 234.6 segundos.

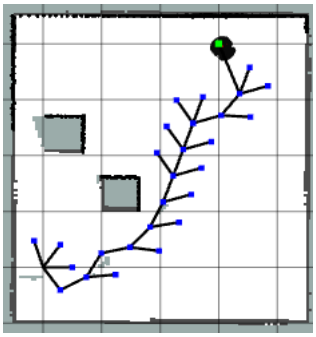


Fig. 9. Aplicación de la variante Local B SRRT en el escenario 2 registrando un tiempo de 75.98 segundos.

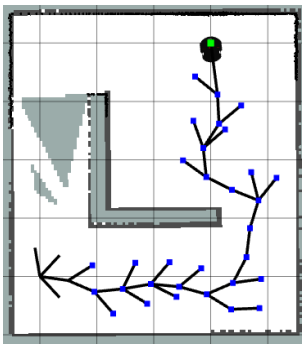


Fig. 10. Aplicación de Local SRRT en el escenario 3 registrando un tiempo de 120.6 seg.

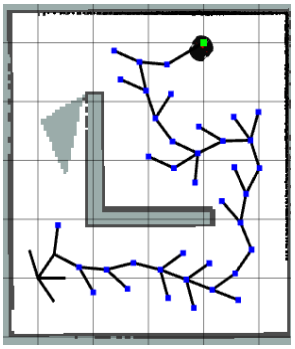


Fig. 11. Aplicación de Local B SRRT en el escenario 3 registrando un tiempo de 184.74 seg.

En la tabla número 3, se describen los resultados y puntos meta de las figuras 6, 7, 8, 9, 10, y 11; se visualizan los árboles aleatorios de exploración usando sensores (SRT), mediante la percepción del sensor se pueden generar y validar estados candidatos a través de espacios reducidos o explorados.

Tabla 3. Descripción general de las figuras.

	Tiempo segs.	Nodos	Meta (X, Y)
Figura 6	198.2	49	(-2, 8)
Figura 7	183.5	50	(-2, 8)
Figura 8	234.6	40	(3, 4)
Figura 9	75.98	28	(3, 4)
Figura 10	120.6	36	(3, 4)
Figura 11	184.74	43	(3, 4)

La ventaja que presentan las variantes de la estrategia Local B SRRT sobre la estrategia Local SRRT es que no arrastran con el problema original de los planificadores RRT–bidireccionales, que necesitan conectar varios vértices para unir un árbol al otro. En Local B SRRT la expansión del árbol de exploración se da al

construir un nuevo árbol local a partir de un nodo ya existente en el árbol de exploración y de conectar los nodos aleatorios que produce el planificador local al árbol de exploración con raíz en el estado meta o al árbol local con raíz en el estado actual.

Las estrategias de exploración conservan la habilidad de escapar de mínimos locales al realizar el proceso de retroceso hacia el estado inicial, al excederse el número máximo de intentos para encontrar un nodo valido o después de evaluar todos los nodos en la lista de estados candidatos válidos para el siguiente proceso de percepción del ambiente. De igual forma, los algoritmos mantienen la característica de terminar con fallo después de un tiempo máximo, e implementan otra forma de conclusión derivada del proceso de retroceso. El algoritmo termina cuando el robot regresa al estado inicial, asumiendo que no es posible incrementar la información del espacio libre a través de los sensores.

La siguiente Figura muestra unas instantáneas que ilustran la ejecución en tiempo real con el robot Pioneer 3DX.

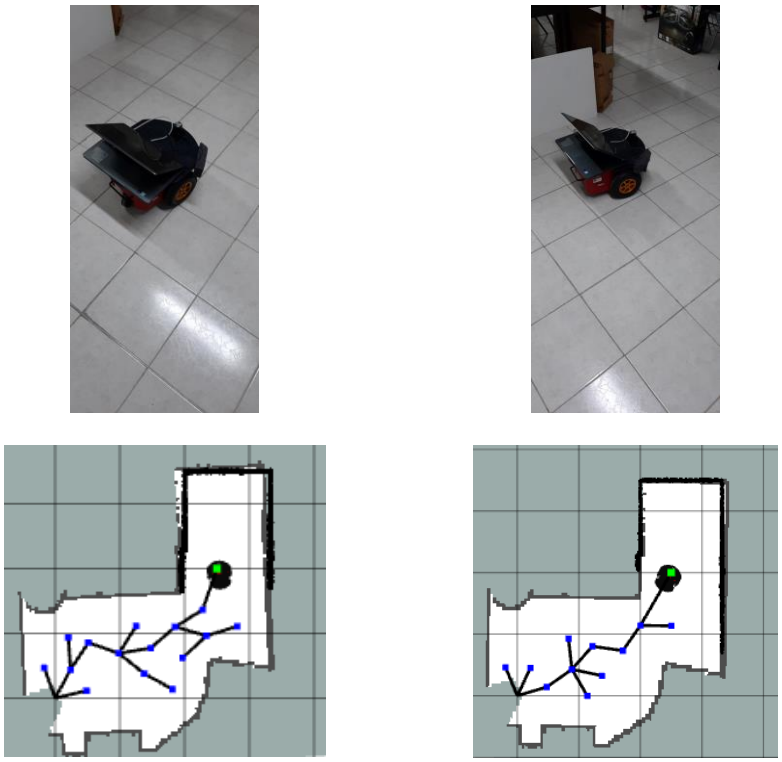


Fig. 12. Ejecución en tiempo real con el robot Pioneer 3DX.

4. Conclusiones y trabajo a futuro

En este trabajo se desarrollaron dos estrategias basadas en el enfoque RRT aplicados en ambientes desconocidos. Las estrategias son llamadas Local SRRT y Local B SRRT

con su determinada variante para generar nodos candidatos generando así una región segura construida a partir de las lecturas del sensor laser.

Los resultados obtenidos a través de las simulaciones muestran que las estrategias aplicadas resuelven el problema principal que es encontrar una meta, usando como apoyo la información sensorial, encontrando así un camino factible del estado inicial al estado meta (considerando las restricciones globales del ambiente y la información obtenida por los sensores del robot) en ambientes simples y complejos. Adaptamos al método, la estrategia de percepción SRT Radial [7], para aprovechar lo más posible la información proporcionada por los sensores.

Como ya hemos visto en la sección de resultados, ambas estrategias consiguen llegar a la meta, no obstante, en términos computacionales el método Local B SRRT arroja mejores resultados que el Local SRRT. Estas estrategias de exploración adaptan los planificadores RRT para ejecutarse localmente en el área segura construida por los sensores.

Las estrategias construyen incrementalmente un árbol aleatorio de exploración como el sensor reporta el espacio libre del ambiente. El árbol refleja la conectividad del espacio libre conocido donde el robot móvil se mueve para reconocer mayores porciones del espacio físico. Para explorar el ambiente, las estrategias utilizan una simple elección de estados candidatos para un nuevo proceso de sensado, dependiendo de la estrategia, guiados con una heurística sencilla sobre la distancia euclidiana que separa al estado candidato del estado meta. En el árbol, los nodos se distinguen por dos clases, nodos intermedios, aquellos donde se lleva a cabo un proceso de percepción del ambiente; y nodos terminales, nodos que sirven para darle movilidad al robot que opcionalmente tienen una descripción de la región segura local y principalmente contienen la entrada de control.

Como trabajo futuro, sería probar las estrategias en robots reales, situación que estamos concretando.

Referencias

1. LaValle, S.M., Kuffner, J.: Rapidly-exploring random trees. In: Progress and prospects. Workshop on Algorithmic Foundations of Robotics (2000)
2. Deng, X., Kameda, T., Papadimitrion, C.: How to learn an unknown environment I: The rectilinear case. Journal of the ACM 45(2), 215─245 (1998)
3. Mitchell, J.S.: Shortest paths and networks. In: Goodman, J.E., O'Rourke, J. (eds.): Handbook of Discrete and Computational Geometry, Charpter 24, pp. 445─466 (1997)
4. LaValle, S.M., Kuffner, J.: Randomized kinodynamic planing. International Journal of Robotics Research 20(5), 378─400 (2001)
5. Oriolo, G., Vendittelli, M., Freda, L., Troso, G.: The SRT method: Randomized strategies for exploration. In: IEEE Int. Conf. On Robotics and Automation, pp. 4688─4694 (2004)
6. Espinoza Leon, J.: Estrategias para la exploración de ambientes desconocidos en robótica móvil. Tesis de Maestría, FCC-BUAP (2006)
7. Espinoza, L.J., Sanchez, L.A., Osorio, L.M.: Exploring unknown environments with mobile robots using SRT-Radial. In: IEEE Int. Conf. On Intelligent Robots and Systems, pp, 2089─2094 (2007)

ISSN 1870-4069

Navegación de robots móviles utilizando algoritmos Bugs extendidos

Mario Serna H., Abraham Sánchez L., Martín Estrada A.,
Ma. Beatríz Bernábe L., Elberfeld E. Pérez G.

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación,
Laboratorio MOVIS, Puebla, Pue., México
mario_sh95@hotmail.com, abraham.sanchez@correo.buap.mx,
mestrada@cs.buap.mx, beatriz.bernabe@gmail.com,
enrique5001@hotmail.com

Resumen. Los algoritmos de navegación le permiten al robot móvil tener un cierto grado de autonomía en una tarea que requiere de un desplazamiento a un punto específico, planificación punto a punto. En este trabajo se realizó un estudio comparativo de los algoritmos Bug0, Bug1, Bug2, DistBug, Intelligent-Bug, Intensity-based Bug y Tangent Bug. Los algoritmos fueron implementados y probados tanto en ambientes reales como simulados en el robot Pioneer P3-DX con las herramientas provistas por ROS (Robot Operating System). Se presenta un estudio comparativo del rendimiento de los algoritmos, tomando como referencia el tiempo de ejecución y el tamaño total de la trayectoria.

Palabras clave: navegación, algoritmos bug, robots móviles.

Navigation of Mobile Robots Using Extended Bugs Algorithms

Abstract. The navigation algorithms allow the mobile robot to have a certain degree of autonomy in a task that requires a displacement to a specific point, i.e., point-to-point planning. In this work a comparative study of the Bug0, Bug1, Bug2, DistBug, Intelligent-Bug, Intensity-based Bug and Tangent Bug algorithms was carried out. The algorithms were implemented and tested in real and simulated environments in the Pioneer P3-DX robot with the tools provided by ROS. A comparative study of the performance of the algorithms is presented, taking as a reference the execution time and the total size of the trajectory.

Keywords: navigation, bug algorithms, mobile robots.

1. Introducción

Uno de los aspectos más relevantes en la robótica móvil es la navegación. Un robot móvil debe resolver en todo instante las preguntas: ¿Dónde estoy?, ¿Dónde voy? y ¿Cómo llego? La primera pregunta se resuelve con los sensores instalados en el robot y/o con sensores

ubicados en el ambiente de operación [1,2]. Existen muchas técnicas para hacerlo, como utilizar sensores isométricos, sensores de proximidad, cámaras de video o marca en el camino. La segunda pregunta se responde con una estrategia global de navegación y la tercera pregunta se responde con una estrategia de navegación local, que puede corresponder a una acción puramente reactiva o incorporar capacidades deliberativas que optimicen la evasión de obstáculos.

Para una navegación, con robots móviles, segura y efectiva es necesario un algoritmo de planificación de movimientos eficiente debido a que la calidad de la trayectoria afecta de gran manera a la aplicación robótica. Comúnmente, el principal objetivo de esto ha sido minimizar la distancia recorrida en el proceso de la navegación influenciando de esta forma a otras métricas como el tiempo de procesamiento y el consumo de energía. El problema de la planificación de movimientos puede ser clasificado de acuerdo con la naturaleza del entorno, el conocimiento del robot sobre este y el enfoque usado para resolverlo, es decir, de acuerdo con la naturaleza del entorno, conocimiento del mapa y completitud.

2. Planificación de movimientos con información local y algoritmos Bug

La planificación de movimientos puede ser divida de acuerdo con la información que el robot posee de su entorno. La planificación de movimientos con información local, o navegación reactiva, se destaca por la suposición de que la información sobre el espacio de trabajo es incompleta, es decir, la localización de los obstáculos se desconoce. El robot necesita contar con sensores (contacto, ultrasónico, láser, etc.) que le permitan la detección de los obstáculos y de esta forma evitar su colisión mientras avanza hacia el objetivo. En las primeras propuestas de este método [1], se considera al robot como un punto que se mueve en una escena bidimensional, en la cual existen obstáculos desconocidos con una forma arbitraria y de tamaño finito. El robot cuenta con información limitada a sus coordenadas actuales y a las coordenadas de la posición de objetivo. Para poder detectar los obstáculos en su camino, el robot hace uso de un sensor de contacto. Con esta información es suficiente para que el robot determine si es capaz de llegar a la posición de objetivo o en caso contrario, reportar que dicha posición no es alcanzable. Para lograr lo anterior, se definen dos comportamientos básicos que debe realizar el robot dependiendo de la situación que se presente:

1. Avanzar con dirección a la posición de objetivo.
2. Seguir la frontera del obstáculo encontrado hasta que una condición de abandono se cumpla.

El robot inicia la navegación en el ambiente ejecutando el primer comportamiento. Este comportamiento le permite llegar robot desde su posición inicial hasta la posición objetivo siempre y cuando no se encuentre obstáculos que le impidan continuar con su recorrido. Cuando el robot, por medio del sensor de contacto, detecta que hay un obstáculo entre él y la posición de objetivo procede a cambiar su comportamiento actual por el de seguir la frontera de dicho obstáculo encontrado. La condición de abandono se encarga de decidir cuando el robot tiene que dejar de seguir el obstáculo actual y retomar su camino con dirección a la posición objetivo. Los algoritmos Bug se propusieron por varios investigadores desde los años 80s para resolver el problema de la planificación de

movimientos. Estos algoritmos están enfocados en el sensado minimalista, es decir, necesitan poca información para poder dotar a los robots móviles de la capacidad de navegar en entornos con obstáculos.

La familia de algoritmos Bug se caracteriza por tener dos modos principales de movimiento: seguir la frontera de los obstáculos y moverse hacia la meta. A partir de estos dos modos se han derivado varias propuestas en las que se incluyen mejoras para poder reducir la distancia recorrida por el robot o disminuir la cantidad de información necesaria para lograr la meta principal, que es, ir de un punto inicial a otro punto en el entorno. En [3] se resume la evolución de los algoritmos de la familia Bug, en [4] se presenta una comparación de su rendimiento y en [5] se realiza un estudio más a fondo de estos algoritmos. Para este trabajo se consideraron los algoritmos Bug1, Bug2, DistBug y TangentBug por ser los más representativos en la familia Bug. Además de los anteriores, se incluyen los algoritmos:

- Bug 0, o también conocido como Com [5], el cual es usado para resaltar por qué es importante definir una condición de abandono para asegurar la terminación de los algoritmos.
- Intelligent-Bug [6], el cual es una versión orientada a la meta del algoritmo DistBug.
- I-Bug (Intensity-Based-Bug) [7], es una propuesta que utiliza la intensidad de una señal para permitirle al robot navegar sin necesidad de saber su localización en el entorno.

Las notaciones utilizadas en los en los trabajos presentados por los investigadores suelen ser similares, pero en algunos casos esto no es así, por tal motivo, se utiliza la siguiente notación para generalizar las descripciones y propiedades de los algoritmos Bug. La siguiente sección presenta a detalle algunos aspectos generales de los algoritmos BUG.

- X – la posición actual del robot.
- S – la posición inicial del robot.
- G – la posición meta, también llamada objetivo o final.
- H_i – el i-ésimo punto de contacto. Cada vez que el robot encuentra un obstáculo en su trayectoria y pasa del modo "moverse hacia la meta" al modo "seguir la frontera del obstáculo".
- L_i – el i-ésimo punto de abandono. Cuando se cumple alguna condición para dejar de seguir la frontera del obstáculo y pasar al modo "moverse hacia la meta".
- $d(a, b)$ – la distancia Euclidiana entre dos puntos arbitrarios a y b.
- $d_{trayectoria}(a, b)$ – el tamaño de la trayectoria entre dos puntos arbitrarios a y b.
- R– el rango máximo de detección del sensor.
- $r(\theta)$ – la distancia proporcionada por un sensor de distancia, en dirección de θ.
- F – el espacio libre en la dirección de la meta. Cuando θ es igual a la dirección de la meta, $F = r(\theta)$

3. Algoritmos Bug en detalle

Algoritmos Bug0, Bug1 y Bug2. En [8] se presentaron los algoritmos Bug1 y Bug2 como estrategias para la planificación de trayectorias de robots móviles en entornos con

obstáculos desconocidos. Estos algoritmos utilizan dos modos simples de movimiento para lograr su objetivo: moverse hacia la meta y seguir la frontera de los obstáculos. El robot tiene conocimiento de su posición X (coordenadas x, y) y su orientación (ángulo θ) en el entorno, así como de la localización del punto al que debe dirigirse G. Además, cuenta con un sensor de contacto que le permite detectar la presencia de un obstáculo. Esta es toda la información que necesita el algoritmo para funcionar.

El algoritmo Bug1 parte de la idea de moverse en línea recta desde S hacia G hasta que encuentre un obstáculo i. Cuando el obstáculo es detectado se define H_i, el robot ejecutará la rutina de seguir el borde hasta completarlo, mientras realiza esto, se calcula el punto L_i más cercano a la meta y se almacena temporalmente. Una vez finalizado el seguimiento del borde, el robot se dirige al punto Li almacenado siguiendo nuevamente el borde. Al posicionarse en dicho punto, el robot cambia al modo de moverse hacia la meta. Este procedimiento se repite si un nuevo obstáculo es encontrado hasta que el robot llegue a G o se complete de rodear la frontera del obstáculo actual, es decir, el punto H_i es encontrado nuevamente.

El algoritmo Bug2 funciona de forma similar que el algoritmo Bug1, pero presenta algunas variaciones. La trayectoria que se genera desde S hasta G se basa en la idea de seguir una línea recta imaginaria, llamada línea-m, entre estas dos posiciones. Si un obstáculo es encontrado se define H_i, el robot seguirá la frontera del obstáculo hasta encontrarse nuevamente sobre la línea-m. Para esto se define temporalmente un punto P, que representa a X sobre la línea-m, y se comprueba que $d(P, G)$ sea menor a $d(H_i, G)$. De ser esto así, se define L_i y se continua el trayecto cambiando al modo moverse hacia la meta; en caso contrario, se ignora la línea-m hasta que la condición anterior se cumpla [1].

El algoritmo Bug0 [5], también llamado Com, no pertenece oficialmente a la familia de algoritmos, pero es utilizado como base para el desarrollo de los algoritmos Bug. Esto se debe a que es una variación de los algoritmos originales anteriormente descritos y tiene como propósito ilustrar qué sucede si no se define una condición de abandono adecuada que asegure la terminación del algoritmo.

Algoritmo DistBug. El algoritmo DistBug [9], utiliza un sensor de distancia. La idea general de este algoritmo parte de lo siguiente: el sensor de distancia que proporciona los datos tiene una detección máxima cuyo rango es R. Los dos modos básicos de movimiento se mantienen. Inicialmente, el robot se mueve a hacia G hasta que se encuentra con un obstáculo. En esa posición se define H_i y se cambia al modo de seguir la frontera del obstáculo. Durante este modo, se registra la distancia mínima hacia la meta, denominada $d_{min}(G)$, alcanzada desde el punto H_i y se censa F a partir de las lecturas del sensor de distancia. F es la distancia en el espacio libre desde X hasta el obstáculo más cercano con dirección a G, si no se detecta un obstáculo entonces F es igual a $r(\theta)$. El robot deja de seguir la frontera del obstáculo sólo cuando el camino a G está despejado o la ecuación, $d(X, G) - F \leq d_{\min}(G) - Paso$, es satisfecha, donde $d(X, G)$ es la distancia de X a G y Paso es una constante predefinida. La variable Paso es un factor muy importante a la hora de decidir cuándo abandonar la frontera del obstáculo. Kamon y Rivlin asumen dos casos particulares para la elección del valor de esta variable. En el primer caso se asume el conocimiento de la distancia mínima entre los obstáculos, definida como M. El valor de Paso es establecido de acuerdo con el valor mínimo entre M y R.

En el segundo caso se asume que M es desconocida, por lo tanto, establecer un valor muy grande a Paso puede ocasionar que el robot no pueda dejar de seguir la frontera del

obstáculo y mucho menos alcanzar la meta. Para solucionar esto, los autores propusieron una versión modificada de la condición de abandono anterior donde se incorpora la condición del algoritmo Bug2 y es definida como CROSS (cruzar la línea, del inglés, line crossing). Mediante una relación booleana "O" se define la nueva condición de abandono de la siguiente manera: sea $C1 = CROSS$ y $d(X,G) < d(H_i,G)$, $C2 = d(X,G) - F \leq d_{min}(G) - Paso$, tal que, C1 o C2 se cumpla.

Algoritmo 1: DistBug con conocimiento a priori del entorno.

Entrada: Un robot en forma de punto con un sensor de distancia
Salida: Una trayectoria a G o la conclusión de que tal trayectoria no existe.

1. $L_0 \leftarrow S; Paso \leftarrow \min(M,R); i \leftarrow 1$
2. **mientras** Verdadero **hacer**
3. **repetir** moverse en línea recta desde L_{i-1} hacia G
4. **hasta** G es alcanzada o un obstáculo es encontrado en H_i
5. **si** G es alcanzada entonces salir con éxito
6. **repetir** seguir la frontera
7. Almacenar la distancia mínima hacia G $d_{\min}(G)$
8. Censar la distancia en el espacio libre F
9. **hasta**
10. (a) G es visible, $d(X,G) - F \leq 0$ o
11. (b) La condición de abandono basada en el rango se cumple, $d(X,G) - F \leq d_{min}(G) - Paso$ o
12. **si** G es alcanzada **entonces** salir con éxito
13. **sino** H_i es reencontrado **entonces** concluir que G no es alcanzable y salir con fallo
14. **sino** $L_i \leftarrow X; i \leftarrow i + 1$

El algoritmo Intelligent-Bug (IBA [11] es una versión orientada hacia la meta del algoritmo DistBug. En [12] se resalta que el algoritmo DistBug considera el costo de la trayectoria para la toma de decisiones, lo cual puede provocar que el robot se aleje de la meta aún en presencia de un camino libre hacia esta. Lo anterior es tomado como referencia para la mejora de este algoritmo. El procedimiento que sigue este algoritmo es el mismo que el presentado para DistBug. La diferencia radica en que el punto de abandono L es seleccionado con base en el camino libre hacia la meta.

Algoritmo I-Bug. Los algoritmos Bug, anteriormente citados, persiguen un enfoque donde se trata de reducir la trayectoria que sigue un robot desde una posición S hasta una posición G. Mientras que el algoritmo I-Bug [7] parte de la idea de reducir la cantidad de información que necesita el robot para navegar exitosamente hacia una fuente de intensidad en un entorno desconocido. El robot usa un sensor de contacto, un sensor de intensidad y un sensor de alineamiento para lograr la tarea de alcanzar la meta, la cual representa a la fuente de intensidad. El robot no tiene acceso a ninguna información que le permita inferir cualquier coordenada, su orientación o la distancia total recorrida. Un punto llamado torre existe en algún lugar del entorno. La torre emite una señal que es modelada como una función de intensidad. Además, se asume que la intensidad máxima es 1, cuando la torre es alcanzada. La localización de la torre es irrelevante para realizar los cálculos porque cualquier coordenada puede ser trasladada a la posición actual sin cambiar los valores de intensidad. Por lo tanto, se asume que la localización de la torre es (0, 0). Una restricción importante es

que únicamente se permite un máximo local, que es la torre. Las señales emitidas por la torre tienen una forma de círculos concéntricos para el caso de intensidad radialmente simétrica. La intensidad decae cuadráticamente con la distancia, sin importar la dirección. Si los conjuntos de nivel no son círculos concéntricos, entonces este caso es llamado asimétrico.

Anteriormente se hacía mención de los sensores con los que cuenta el robot, a continuación, se da una breve explicación de cada uno:

- Sensor de contacto: indica si el robot está tocando el borde del entorno.
- Sensor de intensidad: indica la intensidad de la señal emitida por la torre relativa a la posición del robot.
- Sensor de alineación: indica si el robot está mirando hacia la torre.

Generalmente para los algoritmos Bug se definen dos modos de movimiento: moverse hacia la meta y seguir la frontera del obstáculo. Sin embargo, para este algoritmo se agrega otro modo: alinearse a la torre. Para cualquier otro algoritmo Bug, este modo se puede ver como una subrutina del modo moverse hacia la meta, debido a que el robot conoce en todo momento su posición, su orientación y la posición de la meta. Por tal motivo, no es necesario utilizar un sensor externo para realizar esta tarea y se asume que el robot tiene que alinearse hacia la meta antes de dirigirse hacia ella, cosa que es diferente en este caso. Dicho lo anterior, a continuación, se describen los tres modos de movimiento adaptados para este algoritmo:

- Moverse en línea recta. Sólo se detiene si hace contacto con algún obstáculo, alcanza la torre o detecta un máximo local a lo largo de su línea de movimiento.
- Alinearse a la torre. El robot rota en dirección contraria a las manecillas del reloj, deteniéndose cuando esté alineado con la torre.
- Seguir la frontera del obstáculo. El robot mantiene contacto con la frontera del obstáculo a su izquierda todo el tiempo, sólo deteniéndose si alcanza un máximo local en la intensidad obtenida desde la torre.

Antes de mencionar el proceso de funcionamiento para este algoritmo se definen las siguientes intensidades:

- i_X es la intensidad actual recibida por el robot.
- i_H es la intensidad observada cuando el robot hace contacto con el obstáculo y cambia al modo de seguir la frontera.
- i_L es la intensidad obtenida justo antes de cambiar al modo moverse en línea recta.

De acuerdo con lo anteriormente mencionado, el procedimiento general del algoritmo es el siguiente: inicialmente se asigna a i_L el valor de i_X. Seguido de esto se ejecuta el modo de alinearse a la torre hasta que el sensor de alineación reciba la señal de que el robot está orientado hacia la torre. Se cambia al modo de moverse en línea recta hasta que i_X sea igual a 1, esto quiere decir que el robot alcanzó la torre, o hasta que el sensor de contacto detecte un obstáculo bloqueando el paso. Cuando el sensor de contacto detecta dicho obstáculo se asigna a i_H el valor de la intensidad obtenido en ese momento desde la torre y se cambia al modo de seguir la frontera del obstáculo. Mientras el robot está ejecutando este modo se comprueba que i_X sea igual a 1, indicando que se llegó a la torre, y en caso de que esto no ocurra, se busca detectar un máximo local mediante el muestreo de las tres últimas

intensidades de i_X de tal forma que $i_{X-2} > i_{X-1} \geq i_X$. Si se detecta este cambio en las intensidades, entonces se repite nuevamente el proceso desde el inicio.

Algoritmo TangentBug. El algoritmo TangentBug [10], al igual que el algoritmo DistBug, utiliza un sensor de distancia para permitirle al robot navegar en el entorno de una forma más ágil y recorriendo una trayectoria más corta. En la propuesta original se propone el uso de un grafo que representa los alrededores del robot para minimizar el tamaño de la trayectoria mediante una búsqueda realizada en él. Este grafo es llamado Grafo Tangente Local (LTG, Local Tangent Graph), el cual es una versión que utiliza la información local disponible para ser construido, en comparación a su contraparte global llamada Grafo Tangente (TG, Tangent Graph). El grafo LTG es generado a partir de la información obtenida del sensor de distancia, representada por la función $\boldsymbol{r}(\boldsymbol{\theta})$ que devuelve la distancia en dirección de θ desde la posición X al punto más cercano de un obstáculo. Al procesar $\boldsymbol{r}(\boldsymbol{\theta})$ se tienen en cuenta los siguientes casos:

- Se crea un nodo, llamado nodo-T, cuando G es visible o no hay obstáculos visibles en dirección de G, para esto se comprueba que $d(X,G) - F \leq 0$ o $F \geq R$, respectivamente.
- Se crea un nodo si es detectada una discontinuidad en la dirección de θ.

Se crea un nodo si $r(\theta) = R$ y $r(\theta)$ comienza a disminuir subsecuentemente. De la misma forma, si $r(\theta) \neq R$ y $r(\theta)$ comienza a incrementar tal que $r(\theta) = R$.

Al tener los nodos identificados, se determina cuál de estos tiene la distancia y dirección óptima hacia G y es seleccionado como el objetivo temporal mientras el robot se mueve en dirección a él y actualiza continuamente el grafo tangente local en busca de un nuevo nodo más cercano. Regresando al proceso que sigue el algoritmo, el robot sólo cambia al modo de seguir la frontera del obstáculo cuando detecta un mínimo local. Mientras se encuentra en este modo, se actualizan dos variables que permiten determinar la condición de abandono. $d_{seguimiento}$ registra la distancia mínima hacia G lograda por el robot durante este modo. Mientras $d_{alcanzada}$ es la distancia mínima desde un punto m, que se encuentra dentro del rango de detección, hasta G. Cuando $d_{alcanzada} < d_{seguimiento}$ el robot retoma su trayectoria hacia G. Si el robot completa un ciclo mientras recorre la frontera del obstáculo y encuentra nuevamente el punto m, se determina que la meta no es alcanzable. Simplificando el proceso del algoritmo, utilizando la forma heurística en lugar del grafo tangente local, puede ser expresado de la siguiente manera:

Algoritmo 2: TangentBug

Entrada: Un robot en forma de punto con un sensor de distancia
Salida: Una trayectoria a G o la conclusión de que tal trayectoria no existe.

1. **mientras** Siempre **hacer**
2. **repetir**
3. Tomar lecturas del sensor de distancia y determinar los puntos de discontinuidad
4. Moverse hacia el punto $p \in \{G, P1, P2, \dots\}$ que minimiza $h(X,p) = d(X,p) + d(p,G)$
5. **hasta**
   (a) G es alcanzada o
   (b) El valor que minimiza $h(X,p)$ empieza a incrementar $d(X,G)$, es decir, se detecta un "mínimo local"

6. Seguir la frontera eligiendo la dirección que continúe en la misma dirección del más reciente moverse hacia la meta
7. **repetir** actualizar continuamente $d_{alcanzada}$ y $d_{seguimiento}$
8. **hasta**
   (a) G es alcanzada
   (b) El robot completa un ciclo alrededor del obstáculo, determinar que G es no alcanzable
   (c) $d_{alcanzada} < d_{seguimiento}$

El rango máximo del sensor de distancia es un factor muy importante para este algoritmo. Debido a que puede afectar en el comportamiento del robot a la hora de elegir la trayectoria a seguir, causando que el tamaño final del recorrido aumente o disminuya.

4. Resultados experimentales

En esta sección presentamos los resultados experimentales, en simulación y en el robot móvil Pioneer P3-DX. Las herramientas que se utilizaron fueron: el sistema operativo ROS (Robot Operating System), las librerias Gazebo, el visualizador de datos rviz, y un sensor láser Hokuyo modelo URG 04LX-UG01.

Con el objetivo de tener más control a la hora de ejecutar los algoritmos se optó por una estructura que implementa la lógica del modelo cliente-servidor. Mediante el uso de las características de ROS se crearon los nodos correspondientes y se agruparon como se muestra en la Figura 1. En la parte superior se encuentra el nodo llamado Servidor Bug responsable del control de los nodos de los algoritmos. Para esto se utilizan los servicios de ROS, permitiendo así que un cliente externo envíe una petición para la ejecución de algún algoritmo en concreto o para cambiar su estado actual. El nodo Servidor Bug recibe estas peticiones y se encarga de validar los parámetros enviados en ellas. Después, envía un mensaje al nodo correspondiente para modificar su estado.

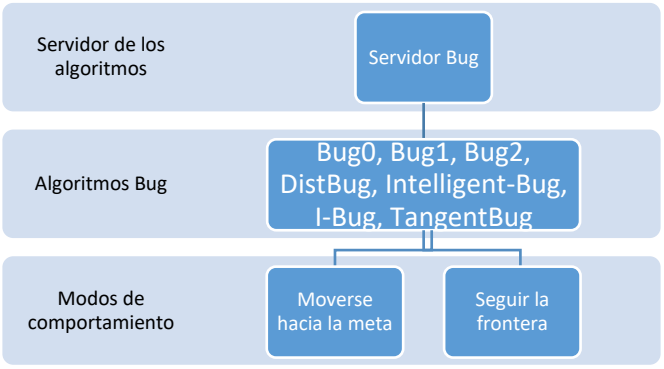


Fig. 1. Diagrama general de la propuesta para el control de los algoritmos.

Para que el nodo del Servidor Bug conozca el estado de los nodos de los algoritmos, se publican mensajes relacionados al estado interno de cada nodo en un tema de ROS. Los publicadores de estos mensajes sólo tienen permitido publicar el estado de su nodo

correspondiente cuando se produce un cambio en el estado del algoritmo, por ejemplo, cuando se pasa del modo de comportamiento "moverse hacia la meta" al modo de "seguir la frontera del obstáculo".

De esta forma se evita que los nodos que se encuentran en el estado de en espera sigan publicando su estado mientras otro nodo se encuentre en funcionamiento. El nodo Servidor Bug se encuentra suscrito a este tema y cada vez que se publica un nuevo mensaje se actualiza el estado global con la información del estado interno. El estado global es usado por el nodo Servidor Bug para comunicar de forma constante cada segundo qué es los que está ocurriendo a los clientes externos.

De la misma forma en que el Servidor Bug se comunica con los nodos de los algoritmos, estos se comunican con los nodos de los modos de comportamiento. Esto ocurre cuando hay un cambio en el estado del algoritmo, indicando que es necesario utilizar otro modo de comportamiento en ese preciso momento. Existieron otros pequeños inconvenientes para implementar los algoritmos, que se resolvieron con éxito y se pueden consultar con detalle en [13].

Las pruebas de los algoritmos se realizaron en el simulador Gazebo y para esto se construyeron algunos mapas diferentes (ver la Figura 2), haciendo uso del editor de construcciones. Esta herramienta permitió crear distintos tipos de obstáculos para su posterior uso en los mapas. La Tabla 1 presenta un resumen de la ejecución de los algoritmos en el mapa denominado "normal".

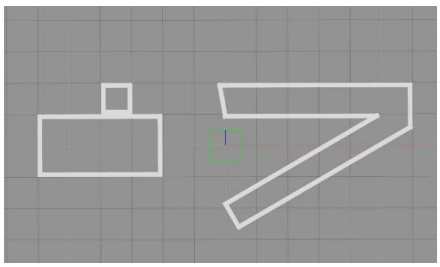


Fig. 2. Mapa tipo "normal".

Tabla 1. Resumen de la ejecución de los algoritmos implementados para el mapa tipo normal.

Algoritmo	Trayectoria final (Metros)	Tiempo (Segundos)	Moverse hacia la meta (seg)	Seguir la frontera del obstáculo (seg)
Bug0	21.041	81.927	22.276	59.651
Bug0 (E, R)	25.683	104.260	33.811	70.449
Bug1	82.382	318.491	21.791	296.700
Bug2	33.183	134.785	27.533	107.252
Bug2 (E)	33.124	135.502	27.603	107.899
Bug2 (R)	26.783	115.196	25.946	89.250
Bug2 (E, R)	26.983	116.751	25.602	91.149
DistBug	19.941	77.016	21.615	55.401
DistBug (E)	30.942	117.276	29.612	87.664
DistBug (R)	20.100	77.265	21.865	55.400
DistBug (E, R)	25.841	103.107	35.407	67.700
I-Bug	21.300	136.611	60.860	75.751
TangentBug	18.783	68.191	60.440	7.751

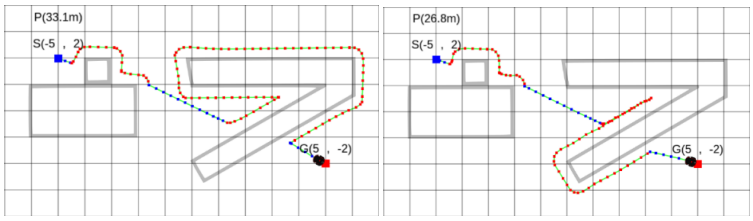


Fig. 3. Bug2 (E) y Bug2 (R) en el mapa de tipo normal.

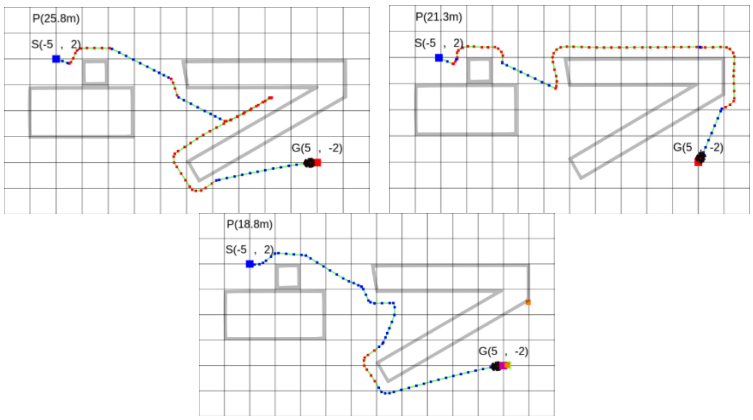


Fig. 4. DistBug (E, R), I-Bug y TangentBug en el mapa de tipo normal.



Fig. 5. Mapa del entorno real creado con el paquete Gmapping de ROS.

Las figuras 3 y 4 muestran las simulaciones realizadas con los diferentes algoritmos.

Para la realización de las pruebas con el robot real se adecuó el espacio en el laboratorio de Movis de la FCC-BUAP, creando de esta forma el entorno que se aprecia en la figura 5.



Fig. 6. Bug1 y Bug2 en el mapa del laboratorio de Movis.

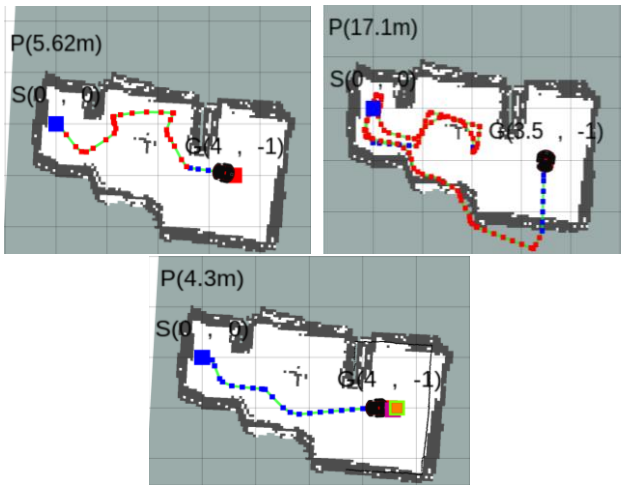


Fig.7. DistBug (E,R), I-Bug y TangentBug en el mapa del laboratorio de Movis.

Tabla 2. Resumen de la ejecución de los algoritmos implementados para el mapa real.

Algoritmo	Trayectoria final (mts)	Tiempo (seg)	Moverse hacia la meta (seg)	Seguir la frontera del obstáculo (seg)
Bug0	-	-	-	-
Bug0 (E)	5.466	55.637	22.587	33.050
Bug0 (R)	6.007	79.459	25.359	54.100
Bug0 (E, R)	5.683	54.238	23.588	30.650
Bug1	22.531	225.255	6.605	218.650
Bug2	10.083	108.254	7.754	100.500
Bug2 (E)	4.241	30.003	13.753	16.250
Bug2 (R)	6.683	92.060	13.059	79.001
Bug2 (E, R)	4.224	30.532	13.782	16.750
DistBug	13.900	152.188	20.288	131.900
DistBug (E)	3.920	29.130	14.988	14.149
DistBug (R)	6.600	84.140	12.640	71.500
DistBug (E, R)	5.620	41.850	5.050	36.800
I-Bug	17.107	227.307	80.675	146.650
TangentBug	4.300	32.080	32.080	0.000

Igualmente, las figuras 6 y 7 muestran las ejecuciones de los algoritmos. La tabla 2 presenta los resultados en el ambiente real.

De acuerdo con los resultados presentados, se resaltan los siguientes puntos:

- DistBug (E) fue el algoritmo que generó la trayectoria más corta, seguido de los algoritmos Bug2 (E) y Bug2 (E, R). El algoritmo TangentBug también realizó un buen trabajo en este escenario.
- De igual forma, los algoritmos anteriores presentaron los menores tiempos de ejecución para llegar de la configuración inicial a la final.
- Bug1 fue el algoritmo que presentó el peor rendimiento en el tamaño de la trayectoria y I-Bug fue el que requirió del mayor tiempo de ejecución para completar la trayectoria.
- I-Bug no obtuvo un buen tiempo de ejecución debido a que necesita dar un giro casi completo sobre su eje de rotación para volver a estar orientado hacia la meta.
- En la Figura 7 arriba se puede observar cómo afectan las rotaciones que realiza el robot a su odometría, haciendo que poco a poco su sistema de coordenadas iniciales pierda precisión con respecto al entorno.
- El algoritmo Bug0 fue el único que no pudo alcanzar la meta.
- Los comportamientos de elegir (E) y revertir (R) la dirección, tuvieron un impacto positivo en el rendimiento de los algoritmos originales.
- A pesar de que el algoritmo TangentBug consiguió un buen resultado, durante su ejecución se pudo observar que presentaba algunas dificultades a la hora de elegir el punto de discontinuidad a seguir, provocando que su avance hacia la meta se viera afectado un poco (Figura 7 abajo).

5. Conclusiones y trabajo futuro

A pesar de que en la actualidad existen técnicas muy avanzadas que resuelven el problema de la planificación de movimientos, por ejemplo, algunas de ellas utilizan un mapa para generar la trayectoria óptima o lo crean mediante la exploración del entorno, los algoritmos Bug siguen siendo una opción viable para resolver este problema, ya que combinan la planificación local con un criterio de convergencia global. En algunos casos, lo que se busca es minimizar el tamaño de la trayectoria final y en otros, el de simplificar el algoritmo de tal forma que requiera la mínima información para funcionar correctamente. Los algoritmos considerados para este trabajo permiten observar el panorama general de este enfoque y pueden ser vistos como una buena forma de introducirse al área de la robótica, especialmente para entender las bases de la navegación en entornos poblados de obstáculos por parte de robots autónomos.

El objetivo principal de la realización de este trabajo fue el de desarrollar estrategias para la navegación de robots móviles utilizando el enfoque de los algoritmos pertenecientes a la familia Bug. Lo anterior fue conseguido de forma exitosa al implementar los algoritmos más emblemáticos de esta familia. A través de su simulación fue posible pulir aspectos para su posterior utilización en un entorno real usando como sujeto de pruebas al robot móvil Pioneer P3-DX. Trabajar con ROS facilitó en gran medida el desarrollo y la implementación de los algoritmos, permitiendo que se distribuyeran las diferentes tareas y comportamientos

haciendo uso de sus nodos. De esta forma se tuvo un mejor control sobre lo que estaba pasando mientras los algoritmos se estaban ejecutando.

Existen muchas ideas para el trabajo futuro, pero creemos que el refactorizar el código es de los más importantes. La forma en que se realizó la implementación perseguía los propósitos de comprobar que todo funcionara correctamente y que cada archivo de código fuera entendible por sí solo, pero esta forma no es eficiente cuando se requieren incorporar nuevas funcionalidades o realizar cambios, ya que es necesario modificar la mayoría de los archivos. Por tal motivo, se plantea que el control de los nodos sea implementado en una clase que posea todas las propiedades y características propuestas en este trabajo. De igual forma, la implementación de la clase para los algoritmos Bug. Esto facilitará la inclusión de nuevas mejoras y el desarrollo de otros algoritmos.

Referencias

1. Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S.: Principles of robot motion: Theory, algorithms, and implementations. A Bradford Book (2005)
2. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. The MIT Press (2005)
3. Buniyamin, N., Wan Ngah, N.W., Mohamad, Z.: Points Bug versus TangentBug algorithm: A performance comparison in unknown static environment. In: IEEE Sensors Applications Symposium, pp. 278–282 (2014)
4. Yufka, A., Parlaktuna, O.: Performance comparison of the BUG's algorithms for mobile robots. In: INISTA 2009 International Symposium on INnovations in Intelligent SysTems and Applications, pp. 416–421 (2009)
5. Ng, J.: An analysis of mobile robot navigation algorithms in unknown environments. PHD Thesis University of Western Australia (2010)
6. Zohaib, M., Mustafa, P., S., Javaid, N., Iqbal, J.: IBA: Intelligent Bug algorithm – A novel strategy to navigate mobile robots autonomously. In: IMTIC 2013: Communication Technologies, Information Security and Sustainable Development, pp. 291–299 (2014)
7. Taylor, K., LaValle, S.: Intensity-based navigation with global guarantees. Autonomous Robots 36, pp. 349–364 (2014)
8. Lumelsky, V.J., Stepanov, A.A.: Path planning strategies for point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica 2, pp. 403–430 (1987)
9. Kamon, I., Rivlin, E.: Sensory-based motion planning with global proofs. IEEE Transactions on Robotics and Automation 13(6), 814–822 (1997)
10. Kamon, I., Rimon, E., Rivlin, E.: TangentBug: A range sensor-based navigation algorithm. The International Journal of Robotics Research (17)9, 934–953 (1998)
11. Zohaib, M., Iqbal, J., Mustafa Pasha, S.: A novel goal-oriented strategy for mobile robot navigation without sub-goals constraint. Revue Roumaine des Sciences Techniques - Serie Électrotechnique et Énergétique 63(1), 106–111 (2018)
12. Serna Hernández, M.: Navegación de robots móviles utilizando los algoritmos Bug extendidos. Tesis de Licenciatura – Facultad de Ciencias de la Computación – BUAP (2018)

ISSN 1870-4069

Mapeo de incidencia criminal en el Estado de México mediante procesamiento digital de imágenes

Tania Lira-Baca, Miguel Borboya Melchor, Roberto Hernández Sánchez, David Tinoco Varela, Fernando Gudiño Peñaloza

Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Cuautitlán, Departamento de Ingeniería, ITSE, México
{ITSE.LIRA, bm.mike07, roberto.her3130}@gmail.com, dativa19@hotmail.com, fernando.gudino@comunidad.unam.mx

Resumen. Los altos índices de criminalidad presentes en diversas regiones del Estado de México son un problema social de alto impacto el cual debe ser reducido. Asociado a los grupos criminales se encuentran una serie de marcas que pasando por grafitis se confunden como arte pero que son un indicativo de la zona de influencia de los grupos criminales. La aplicación de las técnicas de inteligencia artificial para resolver problemas de impacto social ha sido uno de los principales objetivos desde su concepción. Los sistemas de procesamiento de imágenes y reconocimiento inteligente de patrones son herramientas útiles para este fin. El presente trabajo presenta un sistema de alertas compuesto por dos etapas: por una parte, se desarrolla un sistema de reconocimiento de imágenes basado en redes neuronales artificiales bajo el enfoque de MultilayerPerceptron y un esquema de entrenamiento de Backpropagation; por otra parte, se desarrolla una aplicación móvil de alertas que identifica la zona por geolocalización e indica las marcas presentes que dan una idea del índice de peligrosidad de la zona, estos índices se obtienen de la correlación entre los grafitis presentes y los reportes de grupos criminales en la zona.

Palabras clave: red neuronal Artificial, Procesamiento de Imágenes, Abatimiento de criminalidad, Aplicación móvil, Web Services, Trabajo en proceso

Criminal Incidence Mapping in the State of Mexico through Digital Image Processing

Abstract. The high crime rates present in various regions of the State of Mexico are a high impact social problem which must be reduced. Associated with the criminal groups are a series of marks that, passing through graffiti, are confused as art, but they are indicative of the zone of influence of the criminal groups. The application of artificial intelligence techniques to solve problems of social impact has been one of the main objectives since its conception. The systems of image processing and intelligent pattern recognition are useful tools for this purpose. The present work presents an alert system composed of two stages, on the one hand an image recognition system based on artificial neural networks under the Multilayer Perceptron approach and a Backpropagation training scheme. On the other hand, a mobile application of alerts is developed that identifies the zone by

geolocation and indicates the present marks that give an idea of the danger index of the zone, these indexes are obtained from the correlation between the present graffiti and the reports of criminal groups in the zone.

Keywords: artificial neural network, image processing, criminal abatement, mobile application, web services, work in progress.

1. Introducción

En el Estado de México existe un alto índice de inseguridad tal como lo reflejan los datos de la tasa de incidencia delictiva por entidad federativa de ocurrencia por cada cien mil habitantes reportada por el INEGI [9], en donde se demuestra un incremento delictivo latente en los últimos 6 meses, por otra parte, se observa que en los municipios con mayor incidencia delictiva, son aquellos en los cuales la proliferación de pseudo-arte en forma de grafiti ha aumentado.

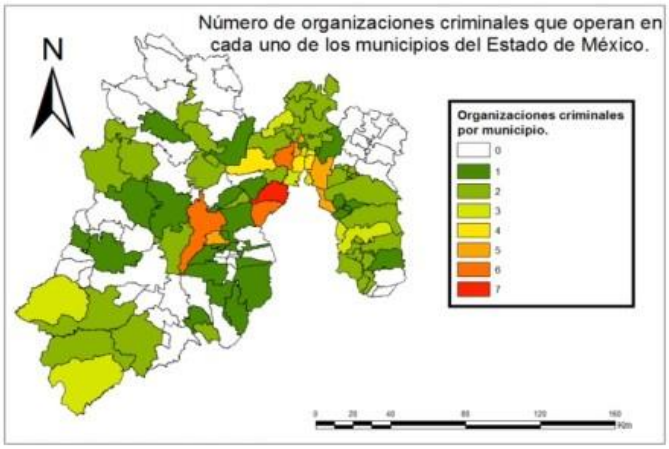


Fig. 1. Mapa de los grupos de crimen organizado por municipio en el Estado de México en 2015. Fuente: Animal Político [13].

Este tipo de arte clandestino se ha utilizado históricamente para delimitar los territorios de influencia de los diversos grupos delictivos. Con base a las diferentes marcas se podrá realizar un mapeo entre dichas marcas y los altos índices delictivos presentes en las diferentes regiones del estado y visualizar el crecimiento de las diferentes incidencias delictivas. Tal como se muestra en la Fig. 1.

El procesamiento digital de imágenes es una herramienta que nos permite detectar dentro de un área de interés en una imagen aquella información útil para la extracción de información relevante, por ejemplo, la determinación de profundidad de color nos permitirá calcular el tiempo de vida del grafiti, mientras que un sistema de mejoramiento de imágenes y localización de regiones de interés, nos permiten definir áreas en las cuales Podemos encontrar grafitis pertenecientes a diversos grupos criminales.

Así mismo, los sistemas de clasificación basados en técnicas de inteligencia artificial, tal como una red neuronal artificial, nos permitirá agrupar en categorías a los diferentes conjuntos de marcas para crear un mapa virtual de los grupos delictivos delimitando las zonas de influencia.

Por otra parte, el auge de los dispositivos móviles actuales nos permite acercar las técnicas de inteligencia artificial aplicadas en un problema de interés general [11], que puede ser atacado por diversos frentes, en este caso mediante la prevención de delitos.

2. Sistemas inteligentes de reconocimiento de caracteres

Un sistema de reconocimiento de imágenes típicamente está conformado por los siguientes partes [3,10].

Adquisición. Se refiere al proceso de convertir un documento a una representación apta para ser procesada por la computadora. Tal adquisición y conversión puede ser realzada por un escáner, cámara fotográfica o de video, etc. y el resultado puede variar dependiendo del proceso de digitalización usado y el método de codificación [10].

Binarización. La binarización es muy utilizada en la reproducción de imágenes, en la visión artificial y en la segmentación y reconocimiento entrópico de caracteres (OCR). Convierte la imagen recibida en una imagen binaria separando así el fondo de los objetos a analizar [10].

Normalización. Trata de ajustar la forma, tamaño y posición, esto con el objetivo de minimizar la variación entre imágenes de la misma clase [10].

Análisis de documento. Trata de analizar la estructura de un documento y enten-der la información contenida en sus componentes [10].

Segmentación. Se refiera a delimitar las regiones que nos interesan. En el caso del reconocimiento de caracteres se refiere a las líneas y caracteres individuales [10].

Extracción de características. Permite conocer las características como lo son el tamaño, perímetro, área, etc. Así como características topográficas como lo es la orientación de segmentos [10].

Reconocimiento. Convierte la imagen binaria en una representación electrónica. Misma que permite realizar operaciones de validación y análisis. Para nuestro caso se utiliza una red neuronal [10].

Prosproceso. Se refiere a mejorar el proceso de reconocimiento mediante el uso de información contextual [10].

2.1. Procesamiento digital de imágenes

Al conjunto de técnicas y procesos para descubrir o hacer resaltar información con-tenida en una imagen usando como herramienta principal una computadora se le conoce como procesamiento digital de imágenes (PDI) [12]. Los objetivos del PDI se basa en dos áreas de aplicación primordiales: El mejoramiento de la calidad de la información contenida en una imagen. El procesamiento de los datos contenidos en un escenario a través de una máquina de percepción autónoma [12].

Entre las aplicaciones del PDI están el reconocimiento de patrones, el cual se refiere al procesamiento de la imagen resultante del proceso de deconvolución. En esta imagen se manipulan para mejorar la lectura de la misma e identificar propiedades que pudiesen tener algún significado. La manipulación se lleva a cabo a través de tres subprocesos: resaltamiento de la imagen, reducción de ruido, segmentación y detección de bordes.

El propósito de las técnicas de realce de la imagen es mejorar la apariencia de la misma para el observador. La selección de los métodos apropiados y la elección de los parámetros adecuados dependen de la calidad de la imagen original y de la apli-cación. Entre las técnicas de realce están: Transformaciones puntuales [12]. Trans-formaciones locales [12], Transformaciones globales [12] y Transformaciones geomé-tricas.

Otro criterio de clasificación, del realce y la mejora de la imagen, tiene como base las aplicaciones que se van a efectuar. Este sistema de clasificación propone las tres categorías siguientes: Suavizado, definición y Corrección de efecto [12]. sos: resaltamiento de la imagen, reducción de ruido, segmentación y detección de bordes.

2.2. Redes neuronales artificiales

Una RNA (Red Neuronal Artificial) son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas [7]. Consisten en un gran número de elementos simples de procesamiento llamadas neuronas que están organizados en capas. Cada neurona está conectada con otras neuronas mediante enlaces de comunicación, cada uno de los cuales tiene asociado un peso. Dichos pesos representan la información que será usada por la red neuronal para resolver un problema determinado.

Así, las RNA son sistemas adaptativos que aprenden de la experiencia, esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Mediante este entrenamiento o aprendizaje, las RNA crean su propia representación interna del problema.

Entre las arquitecturas de RNA más conocidas están el perceptrón multicapa MLP [7], los mapas auto organizados de Kohonen [5,7], las redes de Hopfield [7], por mencionar solo algunos de ellos.

En general cualquier sistema de RNA puede utilizar distintos paradigmas para el aprendizaje de la red, al igual que distintos tipos de entrenamiento. Tal como se observa en la Fig. 2.

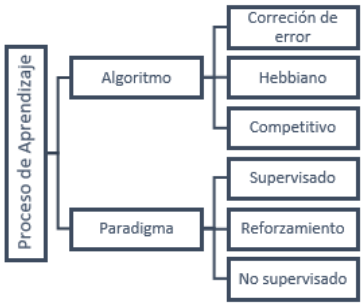


Fig. 2. Proceso de Aprendizaje en RNA.

En este Proyecto se utiliza el paradigma de aprendizaje supervisado y el algoritmo de corrección de error BackPropagation dentro de una arquitectura de MLP [4,7].

Back Propagation. Es un método de cálculo del gradiente utilizado en algoritmos de aprendizaje supervisado utilizados para entrenar redes neuronales artificiales [7]. El método emplea un ciclo propagación (adaptación de dos fases). Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas, Fig. 3. La importancia de este proceso consiste en que, a medida que se entrena la red, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que las distintas neuronas aprenden a reconocer distintas características del es-pacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

3. Sistema de mapeo de criminalidad basado en grafitis

La creación de un Sistema de mapeo de criminalidad se desarrolla en diversas etapas, tal como se muestra en la fig. 4. En este momento el Desarrollo del Proyecto se encuentra en la construcción del Sistema de la aplicación final y en una etapa de pruebas de geolocalización. La descripción complete del sistema se da a continuación.

3.1. Procesamiento digital de imágenes

El Sistema se desarrolla de acuerdo a las siguientes etapas. Primeramente, se realiza el procesamiento de las señales en 5 fases: adquisición, pre- procesado, mejoramiento de las imágenes, determinación de regiones de interés y extracción de características. En la segunda etapa, se realiza la implementación del sistema de clasificación por redes neuronales artificiales y finalmente en la última tapa se implementa la solución en un servidor y se explota un servicio web mediante un dispositivo móvil, tal como se observa en la Tabla 1.

Tabla 1. Implementación de un Sistema de mapeo de grafitis. A1 es la adquisición de imágenes, A2 es el pre procesado de las mismas, A3 es el mejoramiento de imágenes, A4 representa la selección de ROI y A5 es la extracción de características. Por otra parte, B1 representa el pre procesado de datos de entrada, B2 es el entrenamiento de la RNA y B3 es la ejecución de la misma. Finalmente, C1 es el desarrollo del servidor Web y C3. Es la aplicación móvil final.

Etapa	Descripción	Componentes				
1	Procesamiento Digital de Imágenes	A_1	A_2	A_3	A_4	A_5
2	Red Neuronal Artificial	B_1	B_2		B_3	
3	Sistema Móvil	C_1		C_2		

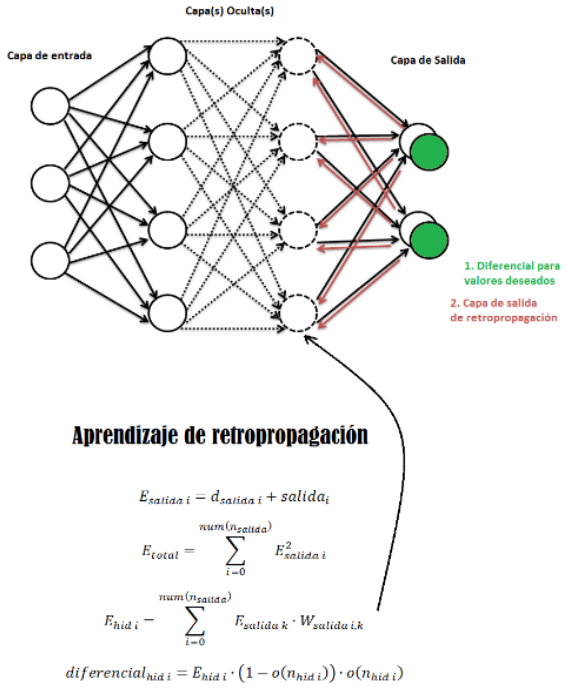


Fig. 3. Algoritmo BackPropagation.

Sistema de pre-procesamiento. En esta etapa se definieron las propiedades de captura de imagen, la resolución y profundidad de bit para las diversas tomas, así mismo se redimensionaron las imágenes para tener un estándar de calidad durante todo el proceso [6,8].

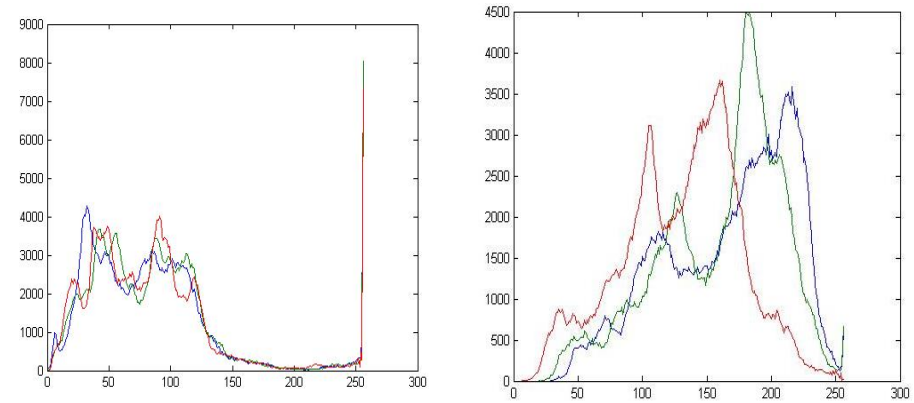
Las características estandarizadas de las imágenes adquiridas se encuentran en las Tabla 2.

Tabla 2. Características de las imágenes estandarizadas para el sistema.

Característica	Valor
Ancho (pixeles)	666
Alto (pixeles)	500
Resolución Horizontal (ppp)	96
Resolución Vertical (ppp)	96
Profundidad de color (BIT)	24

Cabe aclarar que las imágenes fueron tomadas con diversos dispositivos móviles de comunicación (smartphones) de diversas marcas y con calidades ópticas variadas, tal es el caso de iPhone 6 con cámara de 8 megapíxeles, el Nokia 2 con cámara de 2 megapíxeles o Zenek León de 13 Megapíxeles.

Sistema de mejoramiento de imágenes. Dentro de esta etapa se aplicaron diversas técnicas de realce. En primer lugar, se realizó un análisis de la composición de color de las imágenes, con el fin de determinar la profundidad de color y hacer un cálculo de la antigüedad de las diversas muestras: en grafitis antiguos el color es opaco y se visualizan intensidades bajas de la imagen, tal como se puede observar en la Fig. 4. Es importante mencionar que los índices de tonalidad dependen del fondo que se tiene, tal sea el caso de si la pared es blanca, gris, etcétera. Sin embargo, la intensidad es permanente en superficies iguales, por lo cual se puede establecer una correlación entre los colores sin importar el tipo de muro o tonalidad del mismo [12].



(a) (b)

Fig. 4. Intensidades de color para dos imágenes distintas. En (a) un grafiti con más de 5 años de antigüedad. En (b) un grafiti con 1 año aproximadamente de antigüedad.

El proceso de binarización es una técnica de segmentación que se emplea cuando existe una gran diferencia entre el objeto de análisis y el fondo de la imagen, evaluando la similitud y las diferencias entre los pixeles en escalas de grises, consiste entonces, en establecer un umbral óptimo que permita la mejor separación del objeto [12].

Como paso consecuente se convirtió a las imágenes de escala RGB a escala de Grises, y se ecualizó el histograma para mejorar las diferencias y lograr así una segmentación más clara. Posteriormente se convirtió a una imagen binaria. El resultado del procedimiento se visualiza en la Fig. 5.

Una región de interés (ROI), es un área dentro de una imagen delimitada por un contorno, sobre la que se evalúa un determinado parámetro. Dentro del desarrollo del proyecto se seleccionaron de manera automática regiones con alta conectividad entre pixeles, para lo cual las imágenes binarias fueron sometidas a un proceso de erosión y dilatación.

Estas operaciones morfológicas simplifican imágenes y conservan las principales características de forma de los objetos. Un sistema de operadores de este tipo y su composición, permite que las formas subyacentes sean identificadas y reconstruidas de forma morfología óptima a partir de sus formas distorsionadas y ruidosas. Cuando los objetos de la escena sean menores que el elemento estructurante, éstos desaparecerán.



(a) (b) (c)

Fig. 5. Procedimiento de conversión. En (a) imagen original. E (b) imagen en escala de grises. En (c) imagen binarizada después de ecualización de histograma.

Posteriormente se aplicó un filtro de detección de contornos tipo Sobel [12] a la imagen, esta decisión se tomó dado que experimentalmente y desde un punto de vista de subjetivo fue el filtro que mejor definición de las áreas contiguas tenia.

A continuación, se marcaron regiones de al menos 30X 50 pixeles y menores a 300X150 pixeles que fueran dominantes en la imagen, con esto se pretende eliminar regiones del fondo muy similares, por ejemplo, postes de luz, cortinas de negocios, o vidrios, cuyas dimensiones pueden compete con las posibles regiones de interés. De esta manera se garantiza que únicamente las áreas con probabilidad fe contener grafitis sean conservadas.

Para el etiquetado se utiliza conectividad de 8, es decir que los 8 pixeles contiguos al de interés tuvieran un valor igual y posteriormente se aglomeraron las regiones que tuvieran un coeficiente igual en regiones más grandes hasta obtener las adecuadas. Un ejemplo lo hemos en la Fig. 6.

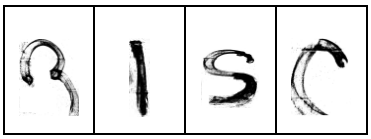


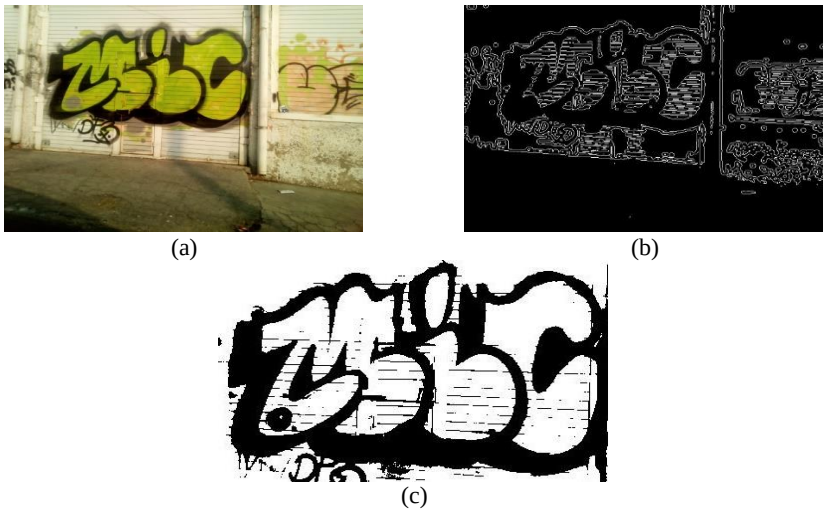
Fig. 6. Delimitación de una región de tamaño 38X53 pixeles.

Una vez definida las regiones se recortaron de la imagen original, dando como resultado una región con alta probabilidad de contener un grafiti. Los resultados finales del procedimiento se observan en la Fig. 7.

Una vez obtenido los grafitis estos se guardan como matrices de dimensión 360 por 150 pixeles que son alimentados a la red neuronal en forma de vectores de datos binarios.

3.2. Sistema de clasificación con MLP

Para la creación del sistema de red neuronal se realizaron 3 etapas: la primera es una etapa de pre-procesamiento, para reducir la dimensionalidad de los vectores de entrada; la segunda es la etapa de entrenamiento de la red neuronal; y por último se encuentra la etapa de evaluación del sistema. Para ello se estableció que el número de clases es de 180, correspondientes al número de muestras totales menos aquellas que se repetían determinado por un coeficiente de similaridad [1, 4,7].



(a) (b)

(c)

Fig. 7. Proceso para la obtención de grafitis. En (a) imagen Binaria de origen. En (b) filtro Sobel aplicado. En (c) recorte de la imagen por etiquetado de regiones.

Preprocesado de los vectores de entrada. Se generó un descriptor de características al aplicar PCA a los vectores de entrada y se aplicó un discriminante lineal [2,8]. El Análisis Discriminante Lineal (LDA) permite tomar dicho vector y reducir la dimensionalidad, seleccionando un número inferior de características formadas como combinación lineal de las originales, las cuales proporcionan la mayor información discriminante.

Luego de realizar LDA y aplicar una regresión lineal a las características obtenidas, para analizar el aporte de cada una de las características originales, se obtuvieron 179 funciones discriminantes, es decir, el número de clases menos uno, que separan linealmente cada una de las clases. Este análisis mostró que las características menos relevantes pueden ser omitidas y reducir los vectores originales de entrada de 360 X 150 a vectores de 120X70.

Por lo tanto, se construyó una red neuronal con 8400 entradas, equivalente al número de componentes creados en el PCA, con 8 neuronas de salida lo que genera 256 posibles salidas siendo inhabilitadas las 76 restantes, en la capa oculta se utilizaron 5600 neuronas.

Entrenamiento de evaluación de la red neuronal. El algoritmo Backpropagation [4,7] se utilizó como método de entrenamiento de la red. El primer paso en la aplicación de este método consiste en la generación aleatoria del conjunto de pesos de la red neuronal. El valor de inicialización está dado por una distribución uniforme, y cuyo rango de valores óptimos es un parámetro en estudio. Adicionalmente se consideraron los parámetros de la Tabla 3.

Para poder evaluar las características de la red obtenida se particionó el set de datos en dos subconjuntos: el subconjunto de entrenamiento y el subconjunto de evaluación. El primer subconjunto se utilizó para realizar el entrenamiento de la red, mientras que el segundo subconjunto se utilizó para evaluar la red obtenida. Se realizaron 3

experimentos, los dos primeros con 100 muestras cada uno y l ultimo con los 200 ejemplos.

Tabla 3. Parámetros de configuración de la Red Neuronal bajo el algoritmo Backpropagation.

Parámetro	Valor
Cantidad de ciclos (epochs)	1000
Error máximo admisible	0.05
Umbral de acierto	0.35
Tasas de aprendizaje (α)	{0.01, 0.05, 0.1, 0.25}
Taza de momento (β)	{0, 0.01, 0.05, 0.1}

El particionamiento se realizó de manera tal que los subconjuntos resultantes sean representativos, es decir, que posean las mismas características que el conjunto original. Para la primera y segunda serie de experimentos se utilizó el mismo particionamiento del conjunto de datos. Se toman aleatoriamente el 65% de los ejemplos para entrenamiento y se dejan un 35% de ejemplos para evaluación. Finalmente, se particionó nuevamente el conjunto de datos, tomando en este caso el 80 % de los ejemplos para entrenamiento.

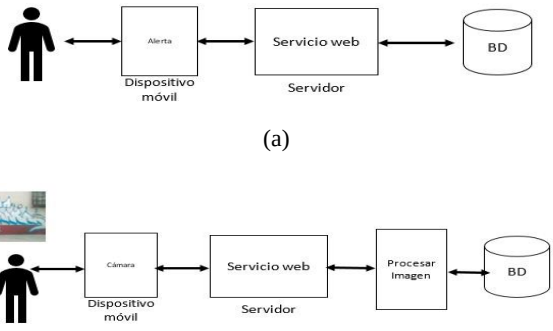


Fig. 8. Modos de operación del Sistema de usuario. En (a) en modo alerta. En (b) modo captura y actualización.

3.3. Implementación del sistema de mapeo y geolocalización

Actualmente esta etapa se encuentra en desarrollo, el sistema se ejecutará de dos modos, tal como se observa en la fig. 8: en el primero la idea es crear un sistema que tome la localización del dispositivo móvil y mande como referencia los grafitis que se pueden asociar a grupos delincuenciales, cabe recordar que no todos los grafitis son creados por grupos criminales, e esta manera el usuario podrá recordar aquellas imágenes que indiquen un peligro para él y tomar las precauciones debidas.

El otro modo de operación es la captura de una imagen de un grafiti y enviar la consulta al sistema, con lo cual el usuario podrá saber si este pertenece a un grupo criminal o no. Así mismo la base de datos se amplia y el sistema se mantiene actualizado [10].

4. Experimentos y resultados

Tal como se indicó anteriormente, el sistema se encuentra en la etapa final de desarrollo, por el momento únicamente se tiene los resultados del sistema de procesamiento de imágenes y del sistema de clasificación de la red neuronal.

Actualmente se cuenta con una base de datos de 500 imágenes de seis municipios correspondientes a diversas regiones del Estado de México: Atizapán, Cuautitlán, Cuautitlán Izcalli, Ecatepec, Naucalpan y Nicolas Romero. De los cuales se tienen datos de la probable creación de los grafitis en un 15% de los casos y se han identificado la participación de al menos 10 grupos delictivos, incluyendo algunos de cobertura tanto regional como nacional. Se omiten datos de los mismos por cuestiones de seguridad. En el proceso de la determinación de antigüedad con base a los datos de intensidad de color los resultados de aplicar la técnica se ven reflejados en la Tabla 4.

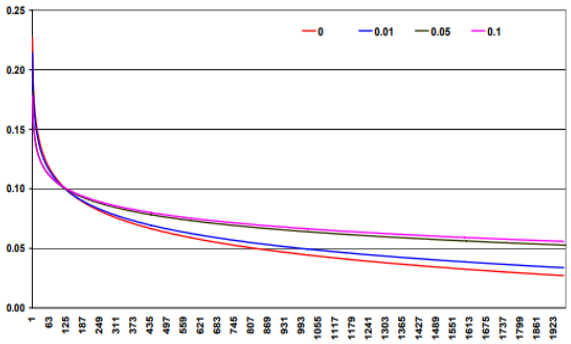


Fig. 9. Curva del error de entrenamiento para una tasa de aprendizaje = 0.1 y diversos valores de momentos, en el eje vertical se expresa el error cuadrático medio, en el eje horizontal el tiempo de entrenamiento en segundos hasta alcanzar las 1000 épocas.

El Sistema fue alimentado con 20 imágenes de las cuales se tiene la certeza de su fecha de creación. Para ello se tomaron algunas muestras y se normalizaron dichos valores para posteriormente crear una serie de rangos de valores asociados con la edad probable de los grafitis.

Tabla 4. Resultados experimentales de la determinación de antigüedad por intensidad de color.

Parámetro	Valor
Grafitis identificados correctamente (%)	87.5
Grafitis descartados erróneamente (%)	12.5
Total de muestras	20

De la tabla 4 podemos observar que la exactitud es de tan solo el 87.5% lo cual nos indica que el sistema no es tan confiable en la clasificación. Esto se entiende por el tamaño reducido de la muestra. Por otra parte, en cuanto la capacidad de identificación de los grafitis, se procesaron para este experimento 200 muestras y los resultados se ven reflejados en la Tabla 5.

De la Tabla 5 podemos observar que la precisión y exactitud del Sistema son altos con una baja razón de falsos positivos. Por último, los resultados del sistema de clasificación por medio de RNA se observan en la Tabla 6 y la figura 9. Este sistema se probó con las mismas 200 muestras utilizadas para el experimento anterior.

Tabla 5. Resultados de la segmentación e identificación de grafitis.

Parámetro	Valor
Promedio de grafitis por imagen	1.2
Grafitis identificados y etiquetados correctamente (%)	97
Grafitis descartados erróneamente (%)	3
Regiones clasificadas erróneamente (%)	2
Regiones descartadas correctamente (%)	98

Tabla 6. Resultados experimentales del entrenamiento y ejecución de la red neuronal.

Parámetro	Valor
Número de épocas	1000
Grafitis identificados correctamente en entrenamiento (%)	94.32
Desviación estándar en entrenamiento (%)	0.83
Grafitis identificados correctamente en validación (%)	83.2
Desviación estándar en validación (%)	2.0
Grafitis identificados correctamente en promedio global (%)	92.1
Desviación estándar global (%)	0.8

De los datos obtenidos en la Tabla 6 y la fig. 10 podemos ver un desempeño bajo en la etapa de validación. Pero un comportamiento global aceptable. El Sistema podría mejorar si se agrega una etapa de optimización, por ejemplo, mediante un Algoritmo Genético.

5. Conclusiones

El procesamiento de imágenes para geolocalización es, a grandes rasgos, software diseñado para recopilar información de fotografías de grafitis, con el propósito de generar delimitaciones de zonas de riesgo dentro del mapa del Estado de México. Esto logrado gracias a la codificación, en formato escala de grises y binario, de las fotos recopiladas con diversos modelos de Smartphone. Una vez que estén codificadas, con el fin de adquirir información lo más limpia posible, pasan por una etapa de filtración para eliminar datos que no sean de interés.

Utilizando un sistema de red neuronal, un proceso de aprendizaje y clasificación entra en marcha con el propósito de guardar tanto la información de origen como la obtenida del proceso dentro de una base de datos. Y para finalizar, la siguiente fase se encarga de procesar los datos para posicionar dentro del mapa las zonas de riesgo.

Tanto el software, así como cada una de las etapas que lo conforman, demuestra ser sólido en cuanto al tratado de la información base, puesto que los procesos son parte de una tecnología existente, estable y explorada a pleno, sin mencionar el gran auge en los últimos años.

Contextualizando dentro de la época, circunstancias y entorno social, es fácil orientar la idea hacia el usuario promedio, puesto que un alto porcentaje poblacional cuenta con

algún tipo de dispositivo móvil, el cual, optimizado mediante una app, será de utilidad en cuanto a reunir la información. El beneficio al usuario es el fácil acceso a los mapas, para su uso preventivo en primera instancia, facilitando su seguridad e integridad.

Agradecimientos. Proyecto apoyado por PAPIME PE111519, y PIAPIME 4.31.05.19.

Referencias

1. Behnke, S.: Hierarchical Neural Networks for Image Interpretation. Lecture Notes in Computer Science, 2766. Springer (2003)
2. Cruz, F.C., Simas Filho, E.F., Albuquerque, M.C.S., Silva, I.C., Farias, C.T.T., Gouvêa, L.L.: Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing. Ultrasonics, 73, 1–8 (2017)
3. Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., Blasco, J.: Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. Food and Bioprocess Technology 4(4), 487–504 (2011)
4. Fausett, L.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. New Jersey, USA: Prenctice-Hall (1994)
5. Fukushima, K.: Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202 (1980)
6. Hantson, S., Chuvieco, E., Pons, X., Domingo, C., Cea, C., Moré, G., Tejeiro, J.A. (et. al.): Cadena de pre-procesamiento estándar para las imágenes Landsat del Plan Nacional de Teledetección. Revista de Teledetección, 36, 51–61 (2011)
7. Haykin, S.S.: Neural networks and learning machines/Simon Haykin. New York: Prentice Hall (2009)
8. Hoyer & Hyvarinen: Independent component analysis applied to feature extraction from colour and stero images. Network: Computation in Neural Systems 11(3), 191–210 (2000)
9. INEGI: Información acerca del número de eventos individuales de victimización delictiva reportados. México. http://www.beta.inegi.org.mx/temas/incidencia/, último acceso: 2019/05/11 (2017)
10. Nieto, D.: Sistema de reconocimiento de kanjis japoneses basado en procesamiento digital de imágenes aplicado a dispositivos móviles. http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/nieto_b_d/capitulo2.pdf, último acceso: 2019/05/11 (2016)
11. Richardson, L., Ruby, S.: RESTful web services, O'Reilly Media, Inc. (2008)
12. Russ, J.C.: The image processing handbook. CRC press (2016)
13. Sánchez-Valdés, V.M.: ¿Por qué aumentó la violencia en el Edomex? Animal Político. https://www.animalpolitico.com/el-blog-de-causa-en-comun/por-que-aumento-la-violencia-en-el-edomex/, último acceso: 2019/05/11 (2014)

ISSN 1870-4069

Diseño e implementación de un algoritmo genético para la predicción de una variable

Hilda Avelar Uribe[1], Ludivina Gutiérrez Torres[1], Ismael Zúñiga Félix[2], Zindi Sánchez Hernández[1]

[1] Instituto Tecnológico de Nogales, Nogales, Sonora, México
[2] Instituto Tecnológico de Agua Prieta, Agua Prieta, Sonora, México
hilda.au@nogales.tecnm.mx, ludivina.gt@nogales.tecnm.mx, iazuniga@gmail.com, sanchezhz@gmail.com

Resumen. Este artículo describe la aplicación de un algoritmo genético para obtener una mejor precisión de pronóstico de cualquier variable utilizando datos históricos. Para medir esta precisión, se utilizó el tipo de cambio peso mexicano - dólar estadounidense en México en comparación con las predicciones calculadas por el método estadístico de promedios móviles. El problema consiste en obtener los valores que minimizan el error cuadrático promedio entre el valor real y el valor de pronóstico para obtener una predicción con un margen de error mínimo. La aplicación del algoritmo genético fue diseñada utilizando cromosomas de representación real, el operador BLX-0.5 de cruce y el operador de mutación no uniforme por ser los operadores que ofrecen una mejor capacidad de exploración y explotación, lo que da como resultado el algoritmo genético que proporciona un aumento de precisión del 14% en comparación con la precisión del método estadístico de media móvil.

Palabras clave: algoritmos genéticos, predicción, promedios móviles.

Design and Implementation of a Genetic Algorithm for the Prediction of a Variable

Abstract. This article describes the application of a genetic algorithm to obtain a better forecast precision of any variable using historical data. In order to measure this precision, the exchange rate Mexican Peso – US Dollar in Mexico was used in comparison with the predictions calculated by the statistical method moving averages. The problem consisted of obtaining the values that minimize the average quadratic error between the real value and the forecast value to obtain a prediction with a minimum error margin. The genetic algorithm application was designed using the following real representation chromosomes: the crossing BLX-0.5 operator as well as the non-uniform mutation operator. These operators offer a better capacity on exploration and exploitation resulting in the genetic algorithm that provides a precision increase of 14% in comparison to the precision of the statistical moving average method.

Keywords: genetic algorithms, prediction, moving averages.

1. Introducción

La predicción es un pronóstico de lo que ocurrirá en el futuro y esto implica un proceso incierto. A causa de esta incertidumbre, se puede deducir que la exactitud del pronóstico es un factor muy importante en adición a la importancia que tiene el resultado señalado por la predicción [1].

La elección e implementación de un método adecuado de pronóstico siempre ha sido de importancia relevante, pues un error significante en la precisión del pronóstico puede ocasionar problemas críticos y de diversas índoles cuando éste está involucrado en la toma de decisiones.

Actualmente existen básicamente tres tipos de métodos para el cálculo de predicciones los cuales son: los métodos cualitativos, los modelos causales y finalmente los modelos de series de tiempo.

Los métodos de carácter cualitativo se basan primordialmente en opiniones de expertos.

Los modelos de pronóstico causales parten del supuesto de que las variables que afectan al comportamiento a predecir permanecen estables, para luego construir un modelo que relacione ese comportamiento con la variable que se estima. Los modelos de series de tiempo, que son los métodos estadísticos, miden valores de una variable en el tiempo a intervalos espaciados uniformemente con el objetivo de determinar el patrón que existe en su comportamiento para así proyectar en el futuro la variable deseada [1].

El método más utilizado actualmente para el cálculo de pronósticos es el método estadístico de los promedios móviles, pues se considera uno de los mejores y más fáciles de utilizar.

Los métodos anteriormente mencionados presentan varias desventajas, entre las que se pueden señalar que utilizan un número determinado de datos, son más útiles para validar un modelo sugerido y preexistente pero no para generarlo y son restrictivos. Desventajas que pueden ser superadas aplicando un algoritmo genético, pues estos tienen como características principales que permiten manejar grandes cantidades de datos y además no requieren conocer con anticipación los datos que serán introducidos al algoritmo, estas ventajas ofrecen mayor efectividad y confiabilidad para las predicciones.

El presente artículo propone comparar la precisión de los pronósticos calculados con método estadístico promedios móviles contra la precisión obtenida en las predicciones realizadas mediante la aplicación de un algoritmo genético.

2. Antecedentes

El algoritmo genético es una técnica que está basada en la teoría de la evolución de Darwin que se fundamenta en los elementos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven al adaptarse a los cambios que se producen en su ambiente, esta adaptación ocurre, pues las formas de vida no son estáticas sino que evolucionan, lo que quiere decir que las especies cambian continuamente, unas se originan y otras más se extinguen irremediablemente [2].

Un punto importante de explicar es que la adaptación natural se lleva a cabo en dos fases, la primera de ellas es la producción de variabilidad, que es la generación de modificaciones espontáneas en los individuos, la segunda fase es la supervivencia en la lucha por la vida donde los individuos mejor dotados, que son los que han nacido con características favorables para hacer frente al medio ambiente, van a contar con más posibilidades de sobrevivir, reproducirse y dejar descendencia.

Los algoritmos genéticos usan una analogía directa con el comportamiento natural, trabajan con una población de ejemplares, cada uno de los cuales representa una solución factible a un problema dado, a cada ejemplar se le asigna un valor relacionado con dicha solución, cuanto mayor sea la adaptación de un ejemplar al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro ejemplar seleccionado de igual forma. Este cruce producirá nuevos ejemplares descendientes de los anteriores los cuales comparten algunas de las características de sus padres.

Cuanto menor sea la adaptación de un ejemplar, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones, de ésta manera se produce una nueva población de posibles soluciones. De tal forma que a lo largo de las generaciones las buenas características se propagan a través de la población, favoreciendo el cruce de los ejemplares mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda.

Si el algoritmo genético ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

El método estadístico de Promedios Móviles (PM) calcula pronósticos eficaz y eficientemente cuando las series de tiempo son estacionarias tanto en media como en varianza. La fórmula (1) es utilizada para encontrar los promedios móviles de orden *n*, PM(n) para un período t+1 [1]:

$$PM\ t+1 = [Dt + D\ t-1 + ... + D\ t-n+1] / N. \quad (1)$$

Donde N es el número de observaciones utilizadas en los cálculos y D es el dato observado en el tiempo t.

La principal desventaja que presenta es que se usa para hacer pronósticos a corto plazo, pero las ventajas que tiene es que requieren de pocos datos y los resultados son bastante aceptables. Las características de los promedios móviles son:

- El promedio móvil es calculado con cierto período de tiempo predefinido.
- Mientras más corto es el período, mayor es la probabilidad de una señal falsa.
- Mientras más largo es el período, menor es la sensibilidad del promedio móvil.

3. Planteamiento del problema

En la actualidad la información está afectando cada vez más a las sociedades, gobiernos, industrias, comunidades y personas. La era de la información está provocando transformaciones sorprendentes en la mayoría de las áreas de la actividad humana, como por ejemplo, ningún gerente o persona responsable de la toma de decisiones, puede ignorar la importancia de la información, ya que la mayor parte de

los éxitos provienen de ella, de hecho muchos de los fracasos empresariales son consecuencia de un uso indebido de la información, de una mala interpretación de la misma y en muchas ocasiones porque no se detecta información que se encuentra oculta dentro de los datos que se manejan día a día y que podrían ser de gran utilidad para la aplicación de nuevas estrategias o resolución anticipada de problemas que se pueden presentar en un determinado momento.

Uno de los aspectos más importantes es poder predecir cómo se desenvolverá una variable en el futuro, constituyendo una información muy valiosa y que representa una innegable ventaja competitiva, dicha información tiene un indiscutible interés, tanto desde el punto de vista comercial, como de servicios y control de procesos. El hecho de poder predecir el comportamiento de cierta variable puede ser útil en diversas áreas, que van desde la detección de oportunidades, prevención de problemas y gestión óptima del personal, por solo mencionar algunas.

Actualmente existen diversos métodos estadísticos para hacer predicciones, entre los más utilizados está el método de los promedios móviles que tiene como objetivo efectuar un cálculo estimativo sobre eventos futuros [3] pues se considera uno de los mejores y más fáciles de utilizar, las ventajas que tiene es que requieren de pocos datos y los resultados son bastante aceptables. Pero también presenta desventajas que son de gran peso al momento de realizar una predicción entre las que se pueden mencionar:

- Utilizan un número determinado de datos.
- Son más útiles para validar un modelo sugerido y preexistente, pero no para generarlo.
- Son restrictivos, es decir, puede ser que los datos no satisfagan los requerimientos del análisis.
- Sólo pronostica un período más.
- Si el comportamiento de las variables no es estable, los pronósticos calculados por este tipo de modelo dejan de ser acertados.

De esta forma surge el problema de cómo lograr conocer con anticipación y confiabilidad el comportamiento de una determinada variable que se encuentra en datos históricos, que indique una tendencia o problema que por el momento se encuentra oculto dentro de la gran cantidad de datos, pero que de una manera u otra, estos datos nos podrían guiar al descubrimiento de estas posibles tendencias y por lo tanto nos permita encontrar soluciones a tiempo o detectar oportunidades que nos ofrezcan una ventaja competitiva.

4. Estado del arte

Originalmente los algoritmos genéticos fueron diseñados para aplicaciones de búsqueda y optimización, pero en la actualidad existen algunos trabajos de investigación cuyo objetivo es mejorar la precisión en la predicción. Se plantea [4] que es posible aplicar algoritmos genéticos para predecir la variación en los precios del cobre, con el fin de mejorar el grado de certeza al incorporar la inversa del porcentaje de predicción de signo PPS.

En otra fuente [6], se sostiene que, aunque los algoritmos genéticos se diseñaron originalmente como método de optimización, también pueden ser utilizados, en el

contexto económico, como herramienta predictiva de los cambios en la composición de una población, en términos de las características individuales de los agentes que la componen. En este documento, se desarrolla un algoritmo genético específico capaz de predecir los cambios en las características de los turistas que visitan el sur de Tenerife. Los resultados obtenidos sugieren la conveniencia de sustituir los tradicionales operadores de cruce y mutación por la acción de una adecuada matriz de transición. Esta matriz dirige la dinámica de transformación de la población en el sentido de que permite introducir consideraciones económicas que otorgan mayores probabilidades a ciertas transformaciones en las características de los turistas que visitan la Isla.

La combinación de predicción de series temporales y la Programación Genética ha tomado fuerza debido a su capacidad de deducir la ecuación y aquellos parámetros que mejor aproximan la relación entre la variable de salida y el conjunto de variables de entrada [5], sin embargo, al ser esta aplicada en la predicción de series de tiempo presenta limitaciones que para ser superadas, es necesario modificar la estructura del algoritmo de programación genética original, la función de aptitud, los operadores de selección, intensificación, reproducción, mutación y cruce.

Inspirado en el proceso de evolución natural, el algoritmo genético está diseñado para resolver problemas de optimización y búsqueda mediante el uso de algunos operadores básicos, como la mutación, el cruce y la selección. Los algoritmos genéticos son usados como una herramienta de búsqueda global efectiva por lo que son usados en el campo de las ciencias aplicadas. Existen algunos estudios sobre los que se ha informado recientemente en el que se ha usado a los algoritmos genéticos para la predicción de la quiebra de empresas [11].

5. Metodología

El algoritmo genético desarrollado es de tipo elitista, es decir, incluye un proceso de selección que es mejorado con una estrategia con la cual se asegura la convergencia hacia el individuo con el valor más óptimo en cada generación.

Se utilizó un cromosoma formado por dos genes, donde cada gen es el valor A y el valor B de la función de la recta, la cual es utilizada para calcular los pronósticos, la representación utilizada fue la forma directa, consistiendo en un vector de números reales. Se utilizó la representación real, ya que, si se utilizaba la binaria, al momento de realizar conversiones se perdía precisión numérica y esto no es favorable ya que dicha precisión es un factor muy importante en las predicciones de este tipo, además que la representación de las soluciones potenciales del problema son puntos clave al trabajar con algoritmos genéticos.

Existen tres operadores básicos para el desarrollo de un algoritmo genético, el operador de selección el cual se asegura de la supervivencia de los individuos mejor adaptados dentro de la población, mientras que los operadores de cruce y mutación aseguran la combinación y la diversidad de los cromosomas seleccionados para generar nuevos individuos. El operador de selección utilizado en esta investigación fue el método de la ruleta mejorado, con este método la probabilidad que tiene un individuo de reproducirse es proporcional a su valor de función de evaluación, es decir, a su adaptación.

A causa de esta característica, se implementó este tipo de selección pues para realizar las predicciones se busca al individuo más apto de acuerdo a su ambiente. Una vez calculadas estas probabilidades, la selección de los individuos para reproducirse es aleatoria según estos valores. El método de selección se divide en dos pasos, en el primer paso se forma el espacio de selección acumulativa, utilizando el valor de aptitud de cada uno de los individuos de la población como se muestra en (2) [8]:

$$cul_j = \frac{\sum_{i=1}^{j} fit_{raw}}{\sum_{i-1}^{NPopSize} fit_{raw}}, 1 \le j \le NPopSize \tag{2}$$

donde:

fit: Valor de aptitud, el error de estimación calculado de cada cromosoma,

c*ul:* Valor acumulado del total de fits.

Posteriormente, un coeficiente de adaptación es establecido, el cual tiene la finalidad de crear diferentes niveles de presión a los individuos de la población conforme aumenta el número de generaciones, $fit_{\max}$ representa el máximo, es decir el cromosoma con menor error de estimación y $fit_{\min}$ representa el mínimo valor de aptitud, es decir le cromosoma con mayor error, dentro de la misma generación. La aptitud escalada es calculada como en (3):

$$fit_{Scale} = \frac{fit_{\max} - fit + \gamma}{fit_{\max} - fit_{\min} + \gamma'} \tag{3}$$

Utilizada como función de optimización para minimizar. El valor γ se encontrará entre el rango que se muestra en (4):

$$2.5 \bullet (fit_{\max} - fit_{\min}) \quad \text{hasta} \quad 0.25 \bullet (fit_{\max} - fit_{\min}), \tag{4}$$

conforme avance la generación desde 0 hasta N, para incrementar la presión de selección entre el mejor y el peor individuo dentro de una generación durante la evolución.

Después en el proceso de selección, se genera un valor aleatorio entre el rango de 0 y 1 que simulará el proceso de girar de la rueda, el cual está representado en (5):

$$r_k \in [0,1], 1 \le k \le NPopSize \tag{5}$$

El individuo X_n será seleccionado si satisface la regla (6):

$$cul_{n-1} \le r_k \le cul_n, 1 \le n \le NPopSize \tag{6}$$

El operador de cruce permite realizar una exploración de toda la información almacenada en la población y combinarla para crear mejores individuos. Existen diversos operadores de cruce que se utilizan según sea la representación que tenga el cromosoma. Para el desarrollo e implementación del algoritmo genético propuesto, se utilizó el operador de cruce BLX-0.5, debido a que con α=0.5, BLX ofrece un mejor

balance entre explotación y exploración [8], además si el cromosoma tiene más de un gen, este operador puede ser aplicado a cada uno de los genes con una probabilidad de $P_{BLX-\alpha}$.

El Cruce BLX-α establece que un descendiente es generado: $H = (h_1,\ldots,h_i,\ldots, h_n)$, donde hi es un número aleatorio uniforme escogido del intervalo $[c_{min} - I \cdot \alpha, c_{max} + I \cdot \alpha]$, donde $c_{max} = \max(c_{1i}, c_{2i})$, $c_{min} = \min(c_{1i}, c_{2i})$ e $I = c_{max} - c_{min}$.

El objetivo del operador de mutación es producir nuevas soluciones a partir de la modificación de un cierto número de genes de una solución existente, con la intención de fomentar la variabilidad dentro de la población. Existen muy diversas formas de realizar la mutación, desde la más sencilla conocida como puntual, donde cada gen muta aleatoriamente con independencia del resto de genes, hasta configuraciones más complejas donde se tienen en cuenta la estructura del problema y la relación entre los distintos genes [7].

En la presente investigación se implementó la mutación no uniforme por presentar la característica de que al inicio de la evolución explora de una manera más global en el espacio de búsqueda y de manera más local durante el avance de la evolución. Por el contrario, en la mutación uniforme la exploración siempre es de manera global, lo que puede ocasionar que la mutación nos provea un cromosoma que ya no es tan apto conforme el progreso del algoritmo genético avanza.

La mutación no-uniforme se define de la siguiente manera [8]:
Si este operador es aplicado en una generación t, y g_{max} es el número máximo de generaciones, entonces se aplica (7):

$$c'_i = \begin{cases} c_i + \Delta\,(t, b_i - c_i) \text{ si } \tau = 0, \\ c_i - \Delta\,(t, c_i - a_i) \text{ si } \tau = 1, \end{cases} \tag{7}$$

donde τ es un número aleatorio el cual puede tener el valor de cero o uno, y se calcula con (8):

$$\Delta(t, y) = y\left(1 - r^{\left(1 - \frac{t}{g_{max}}\right)^b}\right), \tag{8}$$

donde r es un número aleatorio del intervalo de [0, 1] y b es un parámetro que determina el grado de dependencia sobre el número de interacciones y usualmente tiene un valor entre 1 y 5. Esta función da un valor en el rango de [0, y] tal que la probabilidad de regresar un número cercano a cero incremente conforme avance el algoritmo. El tamaño del intervalo de la generación del gen será menor conforme pasan las generaciones. Dicha propiedad causa que este operador realice una búsqueda uniforme en el espacio inicial cuando t es pequeña y muy local en estados posteriores.

Se establecieron probabilidades para que el algoritmo genético sea capaz de determinar si debe o no aplicar estos operadores, cabe mencionar que ambos operadores utilizan probabilidades independientes y esta probabilidad fue aplicada a nivel gen.

Finalmente, como función de aptitud se utilizó el error cuadrático medio, ya que este permite calcular la precisión del grado de correspondencia entre un valor pronosticado y un valor observado, donde por valor observado se entiende que es aquel valor que es el real.

El error cuadrático medio, está definido como (9) [9]:

$$ECM = \sqrt{\frac{\sum_{i=1}^{N} (\phi_i - \phi_{iObs})^2}{N}} \tag{9}$$

donde ϕ_i = Valor Pronosticado,
ϕ_{iObs} = Valor Observado,
N = Número de valores analizados.

Cabe mencionar que el error cuadrático medio nos da la medida de las diferencias en promedio de los valores pronosticados y los observados.

6. Resultados

Para comprobar la certeza de las predicciones obtenidas con el algoritmo genético, se tomaron los datos históricos USD/MXN de Investing.com de febrero de 1976 a abril del 2019, por lo tanto, se obtuvo una población total de 514 datos [10].

Se tomaron dos muestras distintas en tiempo y cantidad de datos para comprobar la precisión del algoritmo genético diseñado, en ambas comprobaciones se tomaron los mismos 200 datos para entrenar al algoritmo, en la primera verificación se utilizaron 314 valores para comprobar los resultados de las predicciones y en la segunda verificación se tomó una muestra de 112 valores en un rango de fechas diferentes a la primera comprobación.

Los datos tomados para el entrenamiento del algoritmo genético se muestran en la tabla 1 y los pronósticos obtenidos en la primera comprobación se muestran en la tabla 2.

En la figura 1 se muestran los resultados del comportamiento de las predicciones realizadas por el algoritmo genético, los pronósticos calculados por el método de promedios móviles, así como los datos reales de la primera muestra utilizada de 314 valores para la comprobación de la precisión del algoritmo genético diseñado. En la figura 2 se muestran los resultados obtenidos con la segunda muestra de 112 datos.

En ambos casos es posible observar que la línea punteada color verde que representa los pronósticos calculados con los promedios móviles sigue solamente la tendencia de los datos originales que están representados por la línea punteada color negro, sin embargo, se puede observar claramente cómo la línea, color rojo, que representa las predicciones obtenidas con el algoritmo genético sigue muy de cerca la línea de los datos originales. Además, se puede notar que en los puntos de inflexión cuando las variaciones en el tipo de cambio son más abruptas, el algoritmo genético muestra mayor precisión en la predicción que las realizadas con el método estadístico de promedios móviles.

Las predicciones realizadas con el algoritmo genético diseñado con la muestra de 314 valores tienen un porcentaje de precisión del 91%, pues de los 314 datos que se predijeron se acertaron con un margen de error del ± 5% 285 datos, además estos resultados se compararon con las predicciones hechas utilizando el método estadístico de los promedios móviles, en el cual se obtuvo un porcentaje de precisión del 77% al predecir con un margen de error del ± 5% 241 datos de los 314.

En la muestra de 112 datos se obtuvo un porcentaje de precisión del 88%, pues de los 112 datos que se predijeron se acertaron con un margen de error del ± 5% 99 datos, en las predicciones calculadas utilizando el método estadístico de los promedios

móviles, se obtuvo un porcentaje de precisión del 68% al predecir con un margen de error del ± 5% 76 datos de los 112.

Tabla 1. Datos utilizados para el entrenamiento del algoritmo genético.

Fecha	Dato	Fecha	Dato	Fecha	Dato
7/3/1976	0.01250	**…….**	**…….**	**7/2/1990**	2.85990
8/1/1976	0.02050	**…….**	**…….**	**8/1/1990**	2.87730
9/4/1976	0.01980	**…….**	**…….**	**9/3/1990**	2.88830
10/2/1976	0.02541	**…….**	**…….**	**10/1/1990**	2.89670
11/1/1976	0.02125	**…….**	**…….**	**11/5/1990**	2.92070
12/1/1976	0.01998	**…….**	**…….**	**12/3/1990**	2.93270
1/2/1977	0.02263	**…….**	**…….**	**1/2/1991**	2.94470
10/1/1977	0.02274	**…….**	**…….**	**10/1/1991**	3.05350
……..	……	**…….**	**……..**	**……..**	……..
3/3/1987	0.02274	**…….**	**…….**	**3/2/1992**	3.06110
4/1/198	0.02274	**…….**	**…….**	**4/1/1992**	3.06090
5/2/1980	0.02273	**…….**	**…….**	**5/4/1992**	3.08160

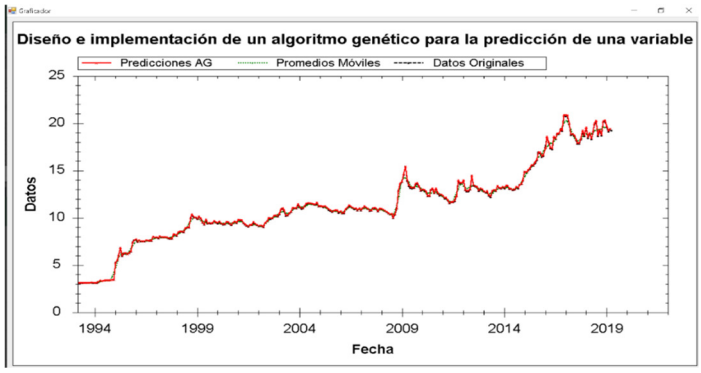


Fig. 1 Comportamiento del Algoritmo Genético comparado con el método de Promedios Móviles. Primera comprobación con una muestra de 314 datos.

Tabla 2. Datos utilizados para comparar los resultados del algoritmo genético y el método de los promedios móviles con la muestra de 314 datos.

Fecha	Real	AG	PM		Fecha	Real	AG	PM
3/1/1993	3.0939	0	0	⋯	7/2/2018	20.1453	19.98064955	18.90372
4/1/1993	3.094	3.118735155	0	⋯	8/1/2018	18.5899	20.26124078	19.2527
5/3/1993	3.1079	3.118835689	0	⋯	9/3/2018	19.2137	18.69753028	19.19848
6/1/1993	3.1264	3.132809957	0	⋯	10/1/2018	18.6531	19.32466325	19.38188
7/1/1993	3.1292	3.151408802	0	⋯	11/1/2018	20.1329	18.76106796	19.29364
8/2/1993	3.1162	3.154223763	3.11028	⋯	12/3/2018	20.2346	20.24877453	19.34698
9/1/1993	3.1137	3.141154304	3.11474	⋯	1/2/2019	19.5878	20.35101791	19.36484
10/1/1993	3.1163	3.138640946	3.11868	⋯	2/1/2019	19.1098	19.70076206	19.56442

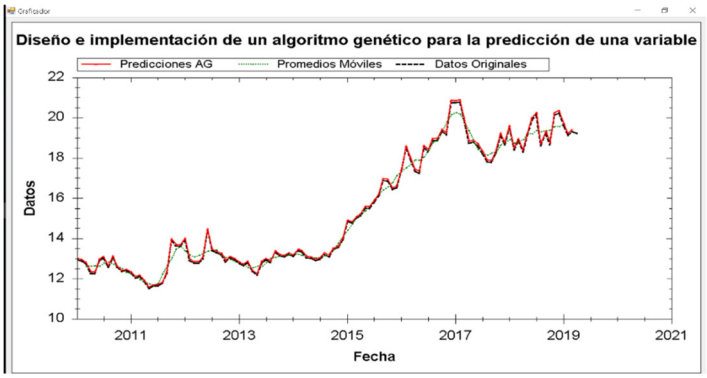


Fig. 2 Comportamiento del Algoritmo Genético comparado con el método de Promedios Móviles. Segunda comprobación realizada con una muestra de 112 datos.

7. Conclusiones

De acuerdo con los resultados obtenidos en la presente investigación, se puede concluir que el algoritmo genético permite obtener una mayor precisión en las predicciones para una variable en una colección de datos históricos, que el método estadístico promedios móviles.

Uno de los aspectos más importantes, aunado al logro del incremento en la precisión del pronóstico es, como puede verse en la Fig. 1 y Fig. 2, el hecho de que el algoritmo genético sigue a los datos reales mucho más de cerca que el método estadístico de promedios móviles. Ya que el algoritmo genético alcanza a las subidas y bajadas de los precios del dólar tan de cerca que a veces se confunde con los datos reales. Como resultado de lo anterior se presenta una mayor precisión en la predicción.

Para realizar la predicción se utilizaron dos poblaciones una de 314 datos y otra de 112 datos. En la primera la precisión fue de 91%, comparada con la precisión obtenida por promedios móviles que fue de 77%. Obteniéndose un aumento en la precisión por el uso del algoritmo genético del 14%.

Sin embargo, cuando se disminuyó el tamaño de la población se obtuvieron otros resultados, donde la precisión disminuyo a 88%, y para promedios móviles se obtuvo un valor de 68%.

Por lo tanto, el mejor resultado se obtuvo para una población de datos mayor como puede apreciarse. En investigaciones posteriores puede explorarse este aspecto del tamaño de la población de tal forma que pueda buscarse un tamaño de población óptimo. Esto constituye una información muy valiosa ya que representa una innegable ventaja competitiva. Lo anterior tiene un indiscutible interés, tanto desde el punto de vista comercial, como de servicios y control de procesos, pues el hecho de poder predecir el comportamiento de una cierta variable con un alto grado de precisión puede ser muy útil en diversas áreas, que tengan un especial interés en una acertada predicción.

Referencias

1. University of Baltimore, http://home.ubalt.edu/ntsbarsh/Business-stat/stat-data/Forecast.htm. Último acceso: 2019/05/07
2. Jimenez-Builes, J., Arango-Sanchez R., Jimenez-Pinzón, J.: Métodos de búsqueda usando los algoritmos de enjambre de partículas y genético. Lámpsakos, no. 16, pp. 52–60 (2016)
3. Contreras Juárez, A., Atziry Zuñiga, C., Martínez Flores, J., Sánchez Partida, D.: Análisis de series de tiempo en el pronóstico de la demanda último de almacenamiento de productos perecederos. Estudios Gerenciales (32), pp. 387–396 (2016)
4. Carrasco, R., Soto, I., Adasme, P., Alfaro, M., Vargas, E.: Pronósticos de precios del cobre usando algoritmos genéticos: En: XXI Congreso ACCA 2014 Asociación Chilena de Control Automático, pp. 1–7 (2014)
5. Martínez, C.: Generación Automática de Modelos de Pronóstico usando Bloques Funcionales y Programación Genética. Universidad Nacional de Colombia, Facultad de Minas, Doctorado en Ingeniería–Ingeniería de Sistemas, Medellín, Colombia (2016)
6. Hernández-López, M.: Predicción mediante algoritmos genéticos con matriz de transición. Una aplicación a la demanda turística en Tenerife. Facultad de Ciencias Económicas y Empresariales, Tenerife. Islas Canarias. España (2015)
7. Sosa, H., Villagra, S., Villagra A.: Operadores de mutación en algoritmos genéticos celulares aplicados a problemas continuos. En: ICT-UNPA-86-2014, Caleta Olivia (2014)
8. Herrera, F., Lozano, M., Verdegay, J.: Tackling Real-Coded Genetic Algorithms: Opera-tors and Tools for Behavioural Analysis. Artificial Intelligence Review 12(4), 265–319 (1998)
9. IArtificial.net Inteligencia Artificial y Machine Learning en Español, https://iartificial.net/error-cuadratico-medio-para-regresion/. Último acceso: 2019/05/07
10. Investing.com: https://es.investing.com/currencies/usd-mxn-historical-data. Último acceso 2019/05/07
11. Chou, C.-H., Hsieh, S.-C., Qiu, C.-J.: Hybrid genetic algorithm and fuzzy clustering for bankruptcy prediction. Applied Soft Computing, vol. 56, pp. 298–316 (2017)

ISSN 1870-4069

Aplicación de un algoritmo genético multiobjetivo para la replaneación de liberaciones en proyectos ágiles de software

Victor Hugo Escandon Bailon, Humberto Cervantes Maceda, Abel García Nájera

Universidad Autónoma Metropolitana
Ciudad de México, México
escvic@gmail.com, hcm@xanum.uam.mx,
agarcian@correo.cua.uam.mx

Resumen. El desarrollo de proyectos de software es un proceso que tiene una naturaleza dinámica, pues es común que en los proyectos de software los requerimientos y la composición del equipo presenten cambios. Estas modificaciones requieren de hacer cambios en la planeación del proyecto. A este proceso de adecuar la planeación para cumplir con la entrega a pesar de los eventos disruptivos que puedan ocurrir se le conoce como problema de replaneación de proyectos. Actualmente muchas organizaciones usan las metodologías ágiles, tales como Scrum, para desarrollar proyectos de software, en las cuales el software se desarrolla iterativamente y se entrega al cliente en incrementos llamados liberaciones. Para resolver el problema de replaneación en el contexto de las liberaciones se propone un modelo tomando en cuenta las características del desarrollo ágil usando Scrum. Para resolverlo, se propone un algoritmo genético multiobjetivo basado en NSGA-II. Se consideran los objetivos de tiempo, costo, estabilidad, desaprovechamiento de la capacidad de desarrollo y valor de la liberación. Todos los objetivos se describen con base en las definiciones de puntos de historia y velocidad de sprint usadas en el desarrollo ágil con Scrum. Los resultados muestran que el algoritmo presenta resultados coherentes.

Palabras clave: scrum, replaneación de proyectos, desarrollo de software dinámico, NSGA-II, optimización.

Application of a Multi-Objective Genetic Algorithm to the Release Replanning in Software Agile Projects

Abstract. Software project development is a process that has a dynamic nature, as it is common that in software projects the requirements and the team composition change. These modifications require making changes in the project planning. To the process of adapting the planning to comply with the delivery, despite the disruptive events that may occur, is known as the project rescheduling problem. Currently, many organizations use agile methodologies, such as Scrum, to develop software projects, in which the software is developed iteratively and it is delivered to the client in increments named releases. To solve the project replanning problem in the context of releases, a model is proposed taking into

account the characteristics of agile development using Scrum. To solve this problem, a multi-objective genetic algorithm based on NSGA-II is proposed. The objectives time, cost, stability, development capacity underuse and release value are considered. All objectives are described based on definitions of story points and velocity that are used in agile development with Scrum. The results show that the algorithm presents coherent results.

Keywords: scrum, project replanning problem, dynamic software development, NSGA-II, ptimization.

1. Introducción

En la actualidad muchas organizaciones han adoptado el desarrollo de proyectos de software con metodologías ágiles, particularmente Scrum, la cual tiene más de 20 años de desarrollo [1]. En estas metodologías, el software se desarrolla iterativamente y se entrega al cliente en incrementos llamados *liberaciones*. En las liberaciones se busca desarrollar funcionalidad del sistema que aporte valor al negocio del cliente de forma rápida. Al inicio del proyecto se realiza la planeación de una o más liberaciones. Durante la ejecución del proyecto, es común que se presenten eventos que afectan al plan, a los cuales llamaremos *eventos disruptivos*. Un ejemplo de un evento disruptivo es que un empleado se vaya o que se agregue un nuevo requerimiento (llamados historias de usuario en el contexto ágil). A la acción de ajustar o actualizar el plan original después de la ocurrencia de un evento disruptivo se le conoce como *replaneación*. Ya que el desarrollo de software es costoso y generalmente tiene fechas de entrega definidas, los administradores de proyectos deben realizar de inmediato una replaneación que minimice los impactos económicos y operativos, y que cumpla con las fechas de entrega que ya estaban definidas.

Además, existen mínimo otros tres objetivos importantes que se deben valorar en un proyecto real. Se desea que al realizar una replaneación ésta no difiera demasiado de la original, ya que se considera que la planeación inicial es la mejor opción para el desarrollo del proyecto. De igual manera se busca que las historias de usuario (HU) de mayor prioridad para la liberación se asignen a los primeros sprints y así desarrollar las HU más importantes primero. En el desarrollo de software los equipos son costosos y se debe usar de la mejor manera el tiempo de desarrollo de los empleados para evitar que este se desaproveche. Por lo tanto, al realizar una replaneación se podrían considerar cinco objetivos: tiempo, costo, estabilidad, valor de liberación y desaprovechamiento de la capacidad de desarrollo. Por lo anterior, el problema de replaneación es un problema de optimización multiobjetivo.

Este problema de replaneación se puede ver como una generalización del problema de asignación, el cual es bien conocido que pertenece a la clase NP-difícil. Además, algunos de los objetivos considerados están en conflicto, por ejemplo, si aumentamos el equipo de desarrollo buscando minimizar el tiempo de entrega, el proyecto será más costoso. Por esta razón, la solución que proponemos para este problema está basada en un algoritmo genético multiobjetivo, el cual considera a cada objetivo igualmente importante y busca dar como resultado un conjunto de propuestas de replaneación que sirvan como apoyo a los líderes de proyecto.

Los resultados obtenidos muestran que el algoritmo tarda un poco menos de 7 minutos para soluciones que proponen replaneaciones compuestas por 16 sprints, esto equivale a 240 días de proyecto. Muestran que, al aplicar un operador de reparación aumenta el indicador de calidad de hipervolumen en la población resultado. Esto mejora el desempeño del algoritmo, ya que el tiempo de ejecución es prácticamente el mismo.

Nuestro trabajo está estructurado de la siguiente manera: A continuación, en la Sección 2, se presenta la revisión bibliográfica. El problema de replaneación se describe en la Sección 3. En la Sección 4 se presenta la propuesta de solución al problema de replaneación. Después, en Sección 5, se muestran los resultados obtenidos y el análisis de los mismos. Finalmente, nuestras conclusiones se encuentran en la Sección 6.

2. Revisión bibliográfica

En esta sección presentaremos las ideas más importantes de algunos artículos, además de los enfocados en metodologías ágiles también tomaremos algunos que proponen soluciones al problema de replaneación.

De acuerdo a la revisión de la literatura hasta el momento, sólo Roque et al. [4] y Nigar [5] estudian el problema de replaneación en las metodologías ágiles, los cuales solo son introductorios y únicamente presentan las características del problema, por lo que no presentan una herramienta que sirva de apoyo a los administradores de proyectos de software en el contexto de las metodologías ágiles. En ellos usan heurísticas y se enfocan en las metodologías ágiles. En el primero [4], se propone un modelo sin replaneación por lo que presenta un modelo estático, es decir, sin cambios con el tiempo. Presentan una propuesta de solución al problema de planeación de proyectos, los autores proponen un algoritmo genético y sus objetivos son el tiempo y el costo del proyecto. Se enfocan en la fase de liberación, pero sin presentar eventos disruptivos y se realiza una planeación para cada sprint, pero no considera la replaneación a partir de un plan existente. En su modelo se busca realizar la asignación de empleados a las tareas. El segundo [5], sólo propone características para el modelo de planeación/replaneación de proyectos en metodologías ágiles ya que no presenta un algoritmo ni modelo para el problema. En este trabajo sólo se mencionan los objetivos que buscarán usar, los cuales son tiempo, costo, robustez, estabilidad y la fragmentación de HU (impacto de que una HU se retrase). Al ser sólo una propuesta, no tiene más detalle y sólo menciona eventos disruptivos generales.

La mayoría de artículos encontrados toman características de los modelos desarrollados para las metodologías tradicionales, tales como la asignación de empleados a las tareas o que la planeación se realiza solo al inicio del proyecto. Jahr [3] considera la planeación de proyectos en un contexto ágil, propone un método de programación entera para resolverlo y sólo toma como objetivo el tiempo que tarda el desarrollo. No encontramos replaneación, pero presenta asignación de los empleados a las tareas y no considera una estimación adecuada para las HU.

Los artículos en donde se modela el problema de replaneación hacen énfasis en que el desarrollo de software es un sistema dinámico. Los cambios se presentan durante la fase de ejecución debido a que ocurren eventos disruptivos. Pero, sin contar los trabajos de Jahr [3], Roque et al. [4] y Nigar [5], en todos los casos se enfocan a las metodologías tradicionales. Presentan propuestas que buscan resolver el problema de planeación de

proyectos de software, pero ahora en un entorno dinámico. En el trabajo de Ge [6] podemos encontrar un modelo para el problema de replaneación, como propuesta de solución implementa un algoritmo genético que busca minimizar el tiempo y costo de proyecto. Se introduce la estabilidad como objetivo del problema y los eventos disruptivos que considera son dos: llegada de nuevas tareas y el movimiento de los empleados (cuando llegan o se van).

Como antes se mencionó, el problema de replaneación es actual y en los últimos dos años podemos resaltar tres documentos dirigidos a las metodologías tradicionales, comenzando con el de Ge y Xu [7], en el cual los autores buscan minimizar el tiempo, costo y estabilidad del proyecto. Los eventos disruptivos que disparan una replaneación son: la llegada o retiro de empleados al proyecto, así como cuando llegan tareas que no se tenían en la planeación original. Los autores buscan modelar la productividad del equipo y cómo afecta ésta en el desarrollo de sus habilidades. Como en los casos anteriores la propuesta de solución está basada en un algoritmo genético.

En el estudio de Shen et al. [8] encontramos por primera vez una función multiobjetivo propuesta en el modelo, de esta manera cada objetivo se calcula de manera independiente. La propuesta presenta como solución a un conjunto de posibles propuestas de replaneación. Los objetivos que se buscan optimizar son: duración del proyecto, tiempo de desarrollo, estabilidad y robustez. Ellos consideran la llegada de nuevas tareas, el retiro de un empleado del proyecto y cuando un empleado se incorpora, como eventos disruptivos. Para validar su trabajo utilizaron tres proyectos del mundo real y los compararon con las propuestas que su algoritmo daba como solución. Como en el caso anterior, Shen et al. [9] también presenta una función multiobjetivo la cual se compone de: la duración del proyecto, el costo del desarrollo, la estabilidad entre planeaciones, la robustez para buscar un plan tolerante a eventos e introducen como objetivo la satisfacción de los empleados al ser asignados a tareas de su agrado. Como propuesta de solución usan un algoritmo compuesto de dos heurísticas. Como solución global proponen un algoritmo genético el cual da como resultado una propuesta de planeación, después, tratan de mejorar a esta propuesta aplicando un algoritmo AMDE (Angle Modulated Differential Evolution). Mencionan que para validar su algoritmo se compararon con proyectos reales, además de casos creados para prueba.

Nos damos cuenta que además de los objetivos tiempo, costo y estabilidad, algunos de los artículos consideran la robustez. Este objetivo se define como qué tan propensa es la replaneación a retrasarse en las fechas de entrega y costo cuando ocurre algún evento disruptivo. Es decir, la robustez es la capacidad de la replaneación a hacer frente a pequeños cambios.

Para proponer un modelo más cercano a la realidad, en los trabajos de Xiao et al. [10], Chen et al. [11] y Ge et al. [7], los autores modelan la comunicación necesaria entre los empleados para desarrollar una tarea. Esta comunicación afecta al proyecto si muchos empleados están asignados a la misma tarea, porque gastan mucho tiempo en comunicarse. Por último, encontramos que en trabajos más actuales se modela la productividad y la curva de aprendizaje de los empleados, podemos ver que esta característica se presenta en Ge et al. [7], Roque et al. [4] y mejoran el modelo presentado en Ge [6].

Como podemos darnos cuenta, pocos artículos hablan del problema de replaneación y tan solo tres están enfocados en las metodologías ágiles. Ninguno presenta

características del desarrollo ágil usando Scrum, ya que no utilizan conceptos como velocidad de sprint y puntos de historia. Por lo anterior, no existe una herramienta que presente propuestas para resolver el problema de replaneación en las metodologías ágiles.

Nuestro trabajo busca proponer un modelo de replaneación de proyectos desarrollados con metodologías ágiles. El modelo que presentamos toma en cuenta cinco objetivos a optimizar: tiempo, costo, estabilidad, aprovechamiento de la capacidad de desarrollo y valor de liberación. Estos dos últimos objetivos son parte de nuestras aportaciones y se explican más adelante.

3. El problema de replaneación de liberaciones en proyectos ágiles de software

En el presente trabajo proponemos un modelo para el problema de planeación en las metodologías ágiles (Scrum), en especial nos enfocamos en la replaneación de liberaciones. Llamaremos a este, problema de replaneación de liberaciones en proyectos ágiles de software (RLPAS). Después de un evento disruptivo se debe ajustar la planeación lo antes posible y con el menor número de cambios, de modo que el costo total y el tiempo de finalización no se vean afectados o, en su defecto, que el incremento sea mínimo, ya que el incremento en el costo debe ser absorbido por la empresa que lo desarrolla y esto se traduce en pérdida.

Las liberaciones están enfocadas a entregar funcionalidad específica y marcan las fechas de entrega de los incrementos al sistema. Cada liberación se divide en una o más iteraciones, llamadas *sprints*, como podemos ver en el ejemplo de la Fig.1. En la planeación de liberación se seleccionan las historias de usuario necesarias para cumplir con el objetivo de cada liberación y se asignan a algún sprint para calendarizar el orden en que serán desarrolladas. Al realizar una replaneación, se considera la asignación de HU al sprint que mejor convenga al plan, tomando en cuenta la duración, prioridad y dependencias de éstas.

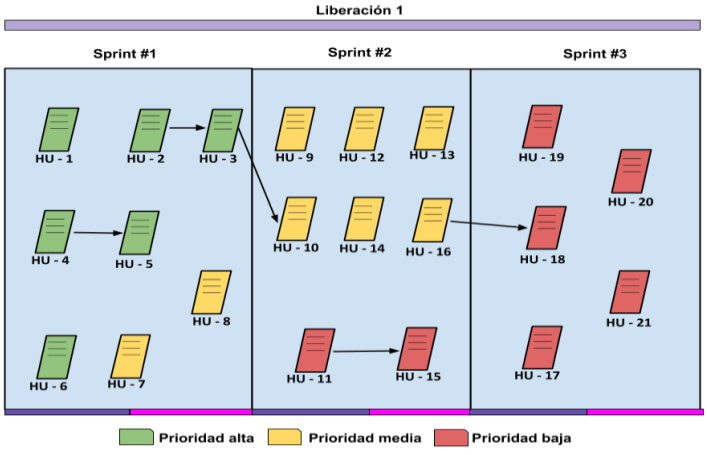


Fig. 1. Representación de un ejemplo de planeación de liberación, el cual consta de tres sprints.

La replaneación de liberaciones tiene características propias, las cuales, principalmente están orientadas a hacer más flexible y ágil la calendarización de las HU. Antes de presentar el modelo, es importante introducir algunos conceptos que son importantes en el contexto ágil, y más específicamente en Scrum.

Puntos de historia. En Scrum, la estimación del esfuerzo necesario para desarrollar las historias de usuario recae en el equipo de trabajo. Las HU se estiman con una unidad llamada *puntos de historia*, que representa el esfuerzo necesario para desarrollar una HU respecto a otra de referencia. Esta técnica se conoce como planning poker [2]. En este trabajo consideramos que el equipo ha hecho una estimación del esfuerzo de todas las HU antes de que se planee la liberación.

Velocidad de sprint. Por otro lado, la velocidad de sprint es una métrica histórica de la capacidad que ha mostrado tener el equipo para desarrollar de forma completa historias de usuario en un sprint.

Nuestro modelo contempla el tiempo extra que un equipo de desarrollo puede trabajar. Si un equipo trabaja horas extra puede tener más capacidad de desarrollo en un sprint (aumenta la velocidad de sprint) [5], pero el proyecto se volverá más costoso, pues es necesario pagar tiempo extra. En este trabajo consideramos un 22.5% de trabajo extra, en una jornada de 8 hrs este porcentaje representa 2 hrs extra[1].

Cuando se presenta un evento disruptivo y se tiene que hacer una replaneación, se intenta que estos eventos tengan el menor impacto sobre el proyecto, considerando los siguientes criterios.

Costo. Es el costo del equipo de trabajo más las horas extras que se necesiten en la replaneación. Para calcularlo se obtiene la velocidad de sprint regular y la velocidad de sprint considerando el trabajo extra.

Tiempo. Es la cantidad de sprints que son necesarios en la replaneación.

Estabilidad. Se refiere a las diferencias de asignación de historias de usuario a los sprints entre la planeación de liberación original y la que resulta de la replaneación. Se busca minimizar este objetivo.

Desaprovechamiento de la capacidad de desarrollo. Este objetivo mide la velocidad de sprint que la replaneación está "desaprovechando". Consideraremos como desaprovechamiento de igual manera que se use tiempo extra cuando aún hay tiempo regular disponible. En los proyectos se debe ocupar de la mejor manera el tiempo de desarrollo, ya que, como se mencionó antes, este es muy costoso. Una buena replaneación debe asegurar que en cada sprint se ocupe la cantidad máxima de tiempo disponible (velocidad de sprint). Al ajustar el plan después de un evento disruptivo, la nueva asignación de HU debe buscar la mejor combinación de acuerdo a la suma de los puntos de historia de cada sprint, para que la velocidad de sprint en cada uno sea ocupada, de preferencia, en su totalidad.

[1] El artículo 66 de la ley federal del trabajo en México, marca que una jornada extraordinaria de trabajo no puede exceder de 3 horas ni de 3 días a la semana. Lo que equivale aproximadamente a un 22.5% en una semana de trabajo de lunes a viernes.

Valor de liberación. Para que el modelo se ajuste adecuadamente, en una planeación o replaneación se deben desarrollar las HU con mayor prioridad en los primeros sprints. En una liberación buscamos entregar funcionalidad que aporte el mayor valor al negocio del cliente. Por ejemplo, en una tienda en línea las HU prioritarias son aquellas que permiten realizar una compra. Al realizar una replaneación después de un evento disruptivo, se deberían asignar las HU con mayor prioridad a los primeros sprints, con el fin de aumentar la probabilidad de tener el tiempo suficiente para poder desarrollarlas.

En nuestro trabajo, el problema de replaneación de liberaciones de proyectos ágiles de software se modela como un problema de optimización multiobjetivo, que busca minimizar el impacto de eventos disruptivos en todos los objetivos que se describieron anteriormente. A continuación, presentamos formalmente el modelo propuesto.

En un proyecto ágil de software usando Scrum, se tiene un conjunto $H = \{h_1, h_2, \dots, h_n\}$de n historias de usuario $h_i, \forall\, i \in \{1, \dots, n\}$, que se tienen que desarrollar. Cada historia de usuario h_i tiene una prioridad πh_i y ph_i puntos de historia estimados.

Definiremos un plan de liberación L, como el conjunto $L = \{S_1, S_2, \dots, S_m\}$de m sprints $S_i, \forall\, i \in \{1, \dots, m\}$. A su vez, cada sprint S_i es el conjunto $S_i = \{h_1^i, h_2^i, \dots, h_{n_i}^i\}$ de n_i historias de usuario $h_j^i \in H, \forall\, j \in \{1, \dots, n_i\}$, que se tienen que desarrollar en el sprint S_i. Así, los puntos de historia phS_i del sprint S_i se definen como:

$$phS_i = \sum_{j=1}^{n_i} ph_j. \tag{1}$$

Todos los sprints son conjuntos disjuntos, de tal forma que $S_1 \cup S_2 \cup \cdots \cup S_m = H = \{h_1, h_2, \dots, h_n\}$ contiene a las n historias de usuario h_j que se tienen que desarrollar en el plan de liberación L.

Para poder desarrollar las historias de usuario, se cuenta con un equipo de trabajo de k empleados y cada empleado $e_i, \forall\, i \in \{1, \dots, k\}$, aporta una velocidad de desarrollo ve_i a la capacidad de desarrollo del equipo. De esta forma, definimos la velocidad vS_i del sprint S_i como:

$$vS_i = \sum_{i=1}^{k} ve_i. \tag{2}$$

Habiendo definido formalmente el problema, ahora podemos plantear las cinco funciones objetivo que se consideran en este trabajo. Primero, el costo del proyecto se define como:

$$Costo = \sum_{i=1}^{m} (\, vS_i + veS_i), \tag{3}$$

en donde veS_i es la velocidad extra del sprint S_i: $veS_i = 0$ si $phS_i - vS_i \leq 0$ y $veS_i = phS_i - vS_i$ si $phS_i - vS_i > 0$.

El tiempo de entrega del proyecto es simplemente la cantidad de sprints considerados, es decir

$$Tiempo = m. \tag{4}$$

La estabilidad de la replaneación se define como:

$$Estabilidad = \frac{1}{(m-1)\cdot n}\sum_{i=1}^{n} |\, sprint(i,t) - sprint(i,t-1)|, \quad (5)$$

en donde $sprint(i,t)$ indica el sprint al cual está asignada la historia de usuario h_i al tiempo t, por lo que $sprint(i,t) \in \{1, \dots, m\}$. El factor $1/(m-1)\cdot n$ se incluye para obtener un valor normalizado.

El desaprovechamiento de la capacidad de desarrollo se define como:

$$Desaprovechamiento = \sum_{i=1}^{m} |\, vS_i - phS_i \,|. \quad (6)$$

Finalmente, el valor de liberación se define como:

$$Valor = \frac{1}{a}\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\sum_{k\in S_j}\sum_{l\in S_j}(\pi h_k - \pi h_l)(j-i), \quad (7)$$

en donde $a = 2(m-1)n^2$. Esta función objetivo da como resultado un número entre -1 y 1, en donde 1 significa que la planeación tiene el mejor valor de liberación y -1 el peor valor de liberación.

En los experimentos que se presentarán más adelante, se consideran dos tipos de eventos disruptivos: se agrega una nueva HU y se va un empleado. Sin embargo, existen otros tipos de eventos disruptivos, por ejemplo, no se termina o se elimina una HU, llega un empleado y un empleado es sustituido.

4. Algoritmo genético multiobjetivo para resolver el RLPAS

Con nuestra solución buscamos proveer a los líderes de proyecto un conjunto de propuestas de replaneación en unos pocos minutos. Ya que en proyectos reales un experto generalmente presenta una propuesta de replaneación como mínimo en 180 minutos [7], sin garantizar que sea una buena solución. Al modelar el problema de replaneación como uno multiobjetivo, el resultado es, idealmente, un conjunto de soluciones que presentan los mejores valores encontrados para cada uno de los objetivos y soluciones que presentan una compensación entre ellos.

Nuestro algoritmo calcula la aptitud de cada una de las propuestas de replaneación implementando el concepto de dominación. Se dice que una solución x *domina* a otra solución *y* si x es al menos tan buena como *y* para todos los objetivos y es estrictamente mejor en al menos uno de ellos. Las soluciones con mejor aptitud son aquellas que no son dominadas por ninguna otra, a esto se le conoce como *no dominancia*.

Para conocer el desempeño de los algoritmos multiobjetivo se usan indicadores de calidad, por ejemplo, el de hipervolumen, que dan como resultado un valor que representa el desempeño de un algoritmo respecto a otro. El indicador de hipervolumen (HV) [13] mide el espacio que cubren las soluciones no dominadas respecto a un punto de referencia y esto da cierta información de la convergencia y la diversidad de los individuos en la población. Por lo tanto, una población es mejor respecto a otra si su HV es mayor.

Existen varios métodos para resolver problemas de optimización multiobjetivo: exactos, heurísticos y metaheurísticos. Por las características del problema bajo estudio, hemos empleado un método metaheurístico, en particular, un algoritmo genético (AG) basado en NSGA-II [14].

Los algoritmos genéticos se basan en los principios de evolución mediante la selección natural. A grandes rasgos, un algoritmo genético considera una función a optimizar, una representación de la solución llamada cromosoma, operadores genéticos y una función que mide la aptitud del cromosoma. La búsqueda que realiza un AG debe tener un balance entre explotar y explorar el espacio de búsqueda. Un algoritmo para un AG consiste en:

1. Representación de la solución como un cromosoma.
2. Generación de una población.
3. Aplicación repetida de los operadores genéticos.

Para nuestra propuesta de solución al problema de replaneación se implementó NSGA-II [14], el cual es un algoritmo genético multiobjetivo que presenta elitismo y conserva la diversidad de la población. Cada individuo (cromosoma) de la población representa una replaneación. NSGA-II implementa una clasificación rápida basada en la no dominancia, en esta se comparan todos los individuos de la población y se les asigna un nivel. La aptitud de cada individuo depende de su nivel de no dominancia, donde cero representa la mejor aptitud. La segunda característica de NSGA-II es la distancia de amontonamiento de un individuo. Para calcular esta distancia los individuos se ordenan de forma creciente en cada uno de sus objetivos y se mide la distancia entre ellos. Ésta se obtiene calculando la distancia que tiene el individuo con sus dos vecinos más cercanos. Si el individuo se encuentra en un extremo del espacio objetivo, entonces se le asigna una distancia infinita, ya que a su lado no se encuentran más individuos. Después de calcular el nivel de no dominancia y la distancia de amontonamiento, se comienza con la selección de individuos que formarán parte de la población en la siguiente generación. Como se mencionó antes, NSGA-II implementa el elitismo por lo que generación a generación solo sobreviven los individuos con mejor aptitud, es decir, los de menor nivel de no dominancia. La diversidad en la nueva población depende de la distancia de amontonamiento. Cuando se está realizando la selección de individuos que sobreviven, si dos o más individuos presentan un mismo nivel de no dominación, entonces se selecciona a los que se encuentre en una zona menos poblada, es decir, al que tenga una distancia de amontonamiento mayor.

4.1 Algoritmo genético multiobjetivo propuesto

Hasta ahora hemos definido una función para cada uno de los objetivos que buscamos optimizar: tiempo, costo, estabilidad, desaprovechamiento y valor de liberación. De igual manera la función que calcula la aptitud de los individuos, la cual está basada en la no dominancia de soluciones. A continuación definiremos las partes del AG: representación de la solución como cromosoma, creación de la población y los operadores genéticos.

Cromosoma. Un cromosoma es una representación de la forma $a_1 a_2 a_3 \dots a_n$, en donde a cada posición i se le llama gen y a_i se le conoce como alelo. Cada gen

representa una HU y su valor es igual al sprint al que está asignada en la replaneación. El primer gen representa a la h_1, el segundo a la h_2 y así sucesivamente.

Creación de la población. Para la creación de la población, la primera vez que se ejecuta nuestro algoritmo, se crean los individuos a partir de la planeación inicial. Esto quiere decir que de la calendarización inicial se construye un cromosoma. Los demás miembros de la población son variaciones de este, los cuales se obtienen de mutar los valores al azar de la calendarización original.

Selección. Se realiza con un torneo aleatorio binario: se seleccionan dos individuos de la población al azar y gana el que tenga mejor aptitud. Si empatan en rango de no dominación, entonces se elige al que tenga una mayor distancia de amontonamiento.

Cruzamiento. El cruzamiento será en un solo punto y tendrá una probabilidad de 0.9. Se seleccionó este valor ya que presentó los mejores resultados preliminares, además que los autores de NSGA-II lo recomiendan. Cabe destacar que también se probó la cruza en dos puntos, pero los resultados no fueron satisfactorios.

Mutación. Para implementar este operador cada gen del cromosoma tendrá una probabilidad de ser mutado de 0.2. Se implementa una mutación gen a gen. Donde se recorren todo el cromosoma y en cada gen se obtiene una probabilidad entre 0 y 1 al azar. Si es menor o igual que 0.2, entonces se le asigna un nuevo sprint, distinto al que estaba asignado, el cual también obtenemos al azar. Se eligió este valor porque de igual manera presentó los mejores resultados en las pruebas preliminares.

Por la naturaleza probabilística de los AG, al aplicar alguno de los operadores genéticos a un individuo puede que este ya no sea factible. En nuestro algoritmo buscamos que todos los individuos de la población lo sean, por lo que implementamos un operador de reparación.

Reparación de cromosoma. En algunas ocasiones las soluciones dejan uno o más sprints vacíos. Por lo que, a pesar de que es un cromosoma válido, en los proyectos reales es algo indeseable. En esta etapa de reparación se valida al cromosoma y si presenta un sprint vacío, este se elimina. El procedimiento es el siguiente: Se identifica el sprint vacío. Se recorren las HU del sprint siguiente, al sprint vacío. Si hay más sprints vacíos se siguen recorriendo las HU para no dejar espacios. Si ya no hay sprints vacíos, se elimina el o los sprint vacíos que se recorrieron al final de la replaneación como resultado de la reparación.

5. Resultados

Al no encontrar trabajos que estudien la replaneación de proyectos enfocados en las metodologías ágiles, no se cuenta con casos de prueba públicos que permitan comparar nuestra propuesta. Por lo tanto, se crearon casos de prueba artificiales y los experimentos se dividieron en dos clases. La primera, con casos pequeños, 12 y 17 HU, para probar si el algoritmo ofrece soluciones coherentes. La segunda clase considera casos de prueba más grandes, desde 40 hasta 100 HU, para saber si el operador de reparación realmente aporta mejora al desempeño del algoritmo.

5.1. Experimentos con casos de prueba pequeños

Se crearon dos casos de prueba pequeños. El primero, 12 HU y 4 empleados, creado especialmente para mostrar el impacto de los objetivos. Los resultados para este caso de prueba fueron coherentes, al ser pequeños, visualmente pudimos observar el impacto de los objetivos y que el comportamiento fue el esperado al simular alguno de los eventos disruptivos. Los resultados fueron similares al segundo caso de prueba. El cual está formado de 17 HU y 4 empleados. Se muestra a continuación en la Fig. 2 el resultado después de simular un evento disruptivo que agrega una HU (HU 18).

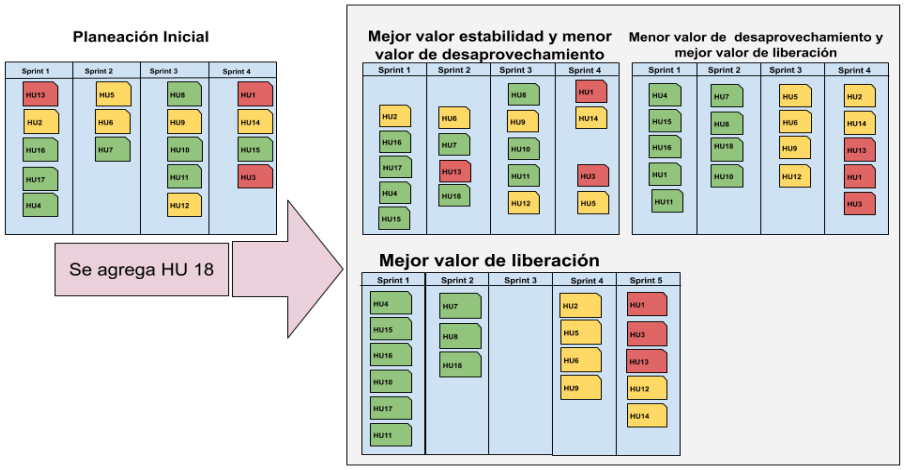


Fig. 2. A la izquierda se encuentra la planeación inicial, a la derecha tres resultados de replaneación: mejor valor de liberación, mejor estabilidad y mejor desaprovechamiento.

Como podemos observar en la parte inferior de la Fig. 2, la replaneación que presentó el máximo valor de liberación tiene un sprint vacío. La reparación, como antes se mencionó, ajusta las historias de usuario para que no queden sprint vacíos dentro de la replaneación. Al implementarla obtuvimos el resultado que se muestra en la Fig 3.

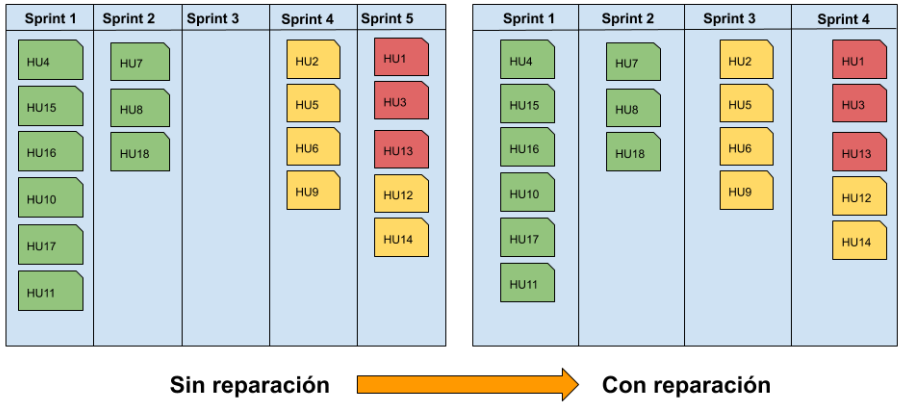


Fig. 3. Replaneación de liberación después de implementar la reparación de sprint vacío.

El algoritmo da como resultado un conjunto de propuestas de replaneación, las cuales presentan un equilibrio en sus objetivos o presentan el mejor valor encontrado para alguno de ellos. En la Fig. 2 se presenta, en la parte superior, dos ejemplos de equilibrio entre dos objetivos y en la parte inferior se muestra la planeación con el mejor valor de liberación. El valor de costo y tiempo son directamente proporcionales con el desaprovechamiento. Cuando el desaprovechamiento aumenta el tiempo también, ya que al desperdiciar mucho tiempo de desarrollo, el proyecto tardará más en desarrollarse. En el caso del costo es lo mismo, ya que al tardar más deberá gastarse en más sprints para terminar el proyecto.

5.2. Experimentos con casos de prueba grandes

Los casos de prueba para esta fase fueron creados con el generador de casos de prueba de Alba y Chicano [12], el cual se configuró para que creara aleatoriamente planeaciones con 40, 70 y 100 HU. Para el número de empleados, se consideraron 4, 5 y 6, respectivamente. Al ejecutar el generador de casos de prueba se obtuvieron planeaciones con hasta 16 sprints. En cada una de las pruebas se modelaron dos eventos disruptivos: se va un empleado y se agrega una nueva HU, ya que son los que tienen un mayor impacto negativo en la planeación.

Para comprobar si la reparación de sprint vacío realmente aporta al desempeño del algoritmo, Se obtiene un caso de prueba con un número de HU y de empleados con el generador. Después se simula un evento disruptivo y se ejecuta el algoritmo 10 veces. En cada ejecución tenemos la misma cantidad de HU, el mismo número de empleados, el mismo evento disruptivo y se aplica la reparación de sprint. De las 10 repeticiones se obtiene un promedio del tiempo de ejecución del algoritmo y los valores máximos de todas las ejecuciones para cada uno de los objetivos. Este procedimiento lo volvemos a realizar, pero ahora sin la reparación de sprint.

Ahora entonces tenemos 10 repeticiones con reparación y 10 sin ella, además de dos conjuntos de valores máximos. De los dos conjuntos volvemos a comparar y obtenemos los máximos globales de las 20 ejecuciones. Los máximos globales son los puntos de referencia con los cuales se calcula el hipervolumen de cada población resultado. Por último, calculamos el promedio del hipervolumen de las 10 poblaciones con reparación y el promedio de las 10 sin reparación. El resumen de los resultados de los experimentos se muestra en la Tabla 1. En la primera columna encontramos el número de HUs, la segunda muestra el número de empleados y en la tercera se muestra el evento disruptivo: **A** indica que se va un empleado y **B** representa que se agrega una nueva HU. Las últimas cuatro columnas representan, respectivamente, el promedio del tiempo de ejecución en segundos (t) y el promedio del hipervolumen para el algoritmo sin reparación y con reparación.

Los resultados muestran que la reparación de sprint vacío ayudó a que el algoritmo alcance un mayor hipervolumen. Podemos interpretar entonces que, cuando se aplica la reparación de sprint vacío, las replaneaciones de la población resultado presentan una mayor diversidad y mejores resultados en las funciones objetivo. Es decir, encuentran mejores replaneaciones, ya que, en general, la población converge a mejores resultados para los objetivos de tiempo, costo, estabilidad, desaprovechamiento y valor

de liberación. Podemos ver que, en la mayoría de los casos, el promedio del tiempo de ejecución aumenta al aplicarse la reparación, pero este no es considerable.

Tabla 1. Datos de los casos de prueba, el evento disruptivo y resumen de resultados obtenidos con el algoritmo genético multiobjetivo, sin y con operador de reparación.

No. HU	No. Emp.	Evento	Sin reparación		Con reparación	
			t	HV	t	HV
40	3	A	111.158	166716.55	109.904	184277.84
40	4	B	102.09	146908.69	104.49	177666.49
70	4	A	235.19	240939.09	243.59	301195.96
70	5	B	214.69	217033.16	212.79	254914.15
100	5	A	317.42	310596.38	339.34	315241.21
100	6	B	307.22	369386.78	313.48	463991.84

De hecho, para el primero y cuarto experimentos, el promedio del tiempo de ejecución con la reparación es menor que sin ella. Los resultados muestran que aplicar la reparación en el algoritmo da como resultado una población con un HV mayor, por lo tanto, el algoritmo tiene un mejor desempeño y no afecta significativamente el tiempo de ejecución.

6. Conclusiones

En este trabajo estudiamos la replaneación de liberaciones de proyectos ágiles de software como un problema de optimización multiobjetivo. La revisión bibliográfica indicó que no existen muchos trabajos sobre problemas de optimización en las metodologías ágiles de desarrollo de software. Aún más, no encontramos uno solo que desarrolle las características de éstas. Por lo que, como primera contribución, presentamos nuestra propuesta de un modelo para el problema de replaneación de liberaciones de proyectos ágiles de software. A diferencia de los trabajos encontrados en la revisión de la bibliografía, nuestro modelo presenta características específicas del desarrollo ágil como son: los conceptos de puntos de historia y velocidad de sprint.

Con este trabajo introducimos los objetivos de desaprovechamiento y valor de liberación. Además de implementar el tiempo extra de desarrollo de un equipo como un parámetro configurable, ya que cada equipo es distinto.

El problema de replaneación de liberaciones podemos considerarlo como una generalización del problema de asignación, el cual es un problema de la clase NP-difícil. Por lo tanto, para resolverlo, se propone un algoritmo genético multiobjetivo que implementa NSGA-II. Al ser un algoritmo basado en población, cubre la necesidad de obtener de manera rápida propuestas de replaneación de liberaciones después de ocurrir un evento disruptivo. En los resultados podemos ver que nuestro algoritmo presenta un conjunto de propuestas de replaneación de liberaciones en menos de 7 minutos para los casos más difíciles que fueron probados. El análisis de los resultados muestra que al aplicar la reparación en todos los casos el HV hipervolumen aumenta y por lo tanto el algoritmo tiene un mejor desempeño sin afectar apenas el tiempo de

ejecución. Estas soluciones presentan 100 HU, que abarcan 15 o 16 sprints, lo que podemos comparar con aproximadamente 240 días de desarrollo de un proyecto. En proyectos reales un experto generalmente presenta una propuesta de replaneación en mínimo 180 minutos [7], sin garantizar que es una buena elección.

Como trabajo futuro se propone obtener datos de proyectos reales y con esos datos ejecutar el algoritmo, simular los eventos disruptivos y verificar que las soluciones sean factibles en proyectos reales. De igual manera se buscará implementar otro tipo de heurísticas, como MOEA/D y SMS-EMOA. Se buscará implementar una segunda heurística para mejora de la población resultado (optimización local). Se investigarán otros indicadores de calidad que sean importantes para nuestros resultados y serán implementados. Por último, se buscará generar una herramienta de apoyo para los proyectos reales.

Referencias

1. Schwaber, K., Sutherland, J.: The scrum guide. The definitive guide to scrum: The rules of the game., vol. 268, Scrum org (2013)
2. Cohn, M.: Agile estimating and planning. Pearson Education (2005)
3. Jahr, M.: A hybrid approach to quantitative software project scheduling within agile frameworks. Project Management Journal 45(3), 35–45 (2014)
4. Roque, L., Araújo, A.A., Dantas, A., Saraiva, R., Souza, J.: Human Resource Allocation in Agile Software Projects Based on Task Similarities. In: Sarro F., Deb K. (eds) Search Based Software Engineering. Lecture Notes in Computer Science, vol. 9962, pp. 292–297. Springer, Cham (2016)
5. Nigar, N.: Model-based dynamic software project scheduling. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pp. 1042–1045. ACM (2017)
6. Ge, Y.: Software project rescheduling with genetic algorithms. In: Artificial Intelligence and Computational Intelligence, 2009 International Conference, vol. 1, pp. 439–443. IEEE, Shanghai (2009)
7. Ge, Y., Xu, B.: Dynamic Staffing and Rescheduling in Software Project Management: A Hybrid Approach. PloS one 11(6), e0157104 (2016)
8. Shen, X., Minku, L.L., Bahsoon, R., Yao, X.: Dynamic software project scheduling through a proactive-rescheduling method. Transactions on Software Engineering 42(7), 658–686 (2016)
9. Shen, X.N., Minku, L.L., Marturi, N., Guo, Y.N., Han, Y.: A Q-learning-based memetic algorithm for multi-objective dynamic software project scheduling. Information Sciences, vol. 428, pp. 1–29 (2018)
10. Xiao, J., Osterweil, L.J., Wang, Q., Li, M.: Dynamic Resource Scheduling in Disruption-Prone Software Development Environments. In: Rosenblum D.S., Taentzer G. (eds) Fundamental Approaches to Software Engineering. Lecture Notes in Computer Science, vol 6013, pp. 107–122. Springer, Berlin, Heidelberg (2010)
11. Chen, W.N., Zhang, J.: Ant colony optimization for software project scheduling and staffing with an event-based scheduler. Transactions on Software Engineering 39(1), 1–17 (2013)
12. Alba, E., Chicano, J.F.: Software project management with GAs. Information Sciences, vol. 177, pp. 2380–2401 (2007)
13. Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L., Vahrenhold, J.: On the complexity of computing the hypervolume indicator. Transactions on Evolutionary Computation 13(5), 1075–1082 (2009)

14. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Transactions on Evolutionary Computation 6(2), 182–197 (2002)

ISSN 1870-4069

Control de la distancia inter electrodos en un proceso de micro maquinado electroquímico por pulsos (PECMM)

Iván Gallegos Miranda, José Arturo Pérez Martínez, Everardo Efrén Granda Gutiérrez, Juan Fernando García Mejía

Universidad Autónoma del Estado de México,
Toluca, Estado de México, México
igallegosmiranda@outlook.es, japerezm@uaemex.mx,
eegrandag@uaemex.mx, fgarciam@uaemex.mx

Resumen. El micro maquinado electroquímico por pulsos (PECMM), es un proceso de manufactura no convencional, en donde la herramienta de trabajo no tiene desgaste, por lo que ofrece una gran ventaja para fabricar piezas de un tamaño reducido, en comparación con los procesos convencionales. El mantener las condiciones óptimas del proceso PECMM garantiza mayor eficiencia y rendimiento de maquinado, donde una de las condiciones críticas del proceso es mantener constante la distancia entre el electrodo y la pieza de trabajo (distancia entre electrodos) para garantizar la remoción homogénea del material. En este documento se presenta el diseño y aplicación de un controlador por lógica difusa (FLC) tipo SISO (del inglés, una entrada una salida) con tres conjuntos difusos de entrada y tres de salida, cuyo objetivo es asegurar que la distancia entre electrodos se mantenga en el orden de 240µm. La implementación de un controlador de este tipo demuestra ser una alternativa eficaz para esta tarea, principalmente debido a que la distancia entre electrodos no es una variable observable directamente, sino que se debe estimar a partir de otras variables del proceso. Lo anterior se realiza mediante la medición de la corriente de maquinado. Los resultados muestran que el controlador permite ajustar progresivamente la distancia del electrodo, por lo que se obtiene un canal de profundidad homogénea, inclusive si la superficie presenta irregularidades a lo largo de la trayectoria de desplazamiento del electrodo, con una precisión cercana a 10 µm.

Palabras clave: control por lógica difusa, micro-maquinado electroquímico por pulsos, procesos de manufactura.

Gap Control of a Process of Micro Electrochemical Machining by Pulses (PECMM)

Abstract. Pulsed electrochemical micro machining (PECMM) is a non-conventional manufacturing process where the tool does not present wear, offering a considerable advantage over conventional processes on the machining of small pieces. In order to guarantee the proper efficiency and performance, the process working parameters must be kept in optimal intervals. One of the critical constraints is to keep constant the distance between the workpiece and the tool

electrode (gap) to ensure a homogeneous material removal. With the aim to keep the gap in the order of 240 µm, in this work is proposed the design and application of a SISO (single-input single output) Fuzzy Logic Controller (FLC) with three fuzzy sets for both: input and output. The implementation of such a controller demonstrates to be a good alternative performing this task, mainly because the gap is not an observable variable, but it could be estimated from other process variables. Thus, the machining current has been used in this investigation. Results show that the FLC allows to continuously adjust the inter electrode distance, machining a channel with homogeneous depth, even though the surface presents irregularities along the electrode trajectory, reaching a precision close to 10 µm.

Keywords: fuzzy logic control, pulsed electrochemical micro-machining (PECMM), manufacturing process.

1. Introducción

La creciente demanda en la fabricación de componentes de tamaño reducido en áreas como robótica, biología, micro electrónica, material biomédico, entre otros; da auge a la investigación en el desarrollo de procesos de micro manufactura [1]. Como aplicaciones de este proceso se puede mencionar la fabricación de micro boquillas, micro engranes, instrumentos quirúrgicos (bisturíes), micro tubos, micro inyectores y micro bujes [2].

El proceso de micro manufactura, puede realizarse mediante métodos convencionales (se genera fricción entre la herramienta y la pieza de trabajo) y no convencionales (se realiza la remoción de material sin contacto ente la herramienta y la pieza de trabajo). El proceso de manufactura convencional, se genera mediante fricción, generando micro fisuras y residuos metálicos, esto es una desventaja con respecto a los métodos no convencionales, que se caracterizan por la ausencia de fricción y generación de residuos [3].

El proceso de micro maquinado electroquímico por pulsos (PECMM por sus siglas en ingles), es un proceso de manufactura no convencional donde la herramienta de trabajo no tiene contacto con la pieza de trabajo, por lo que siempre existe una separación entre ambas a esto se le denomina distancia inter electrodos (*gap*). PECMM es una técnica que aprovecha el efecto electrolítico y la corriente eléctrica para provocar la remoción de material por disolución anódica, en componentes conductores de la electricidad, principalmente materiales metálicos [4].

De acuerdo con los trabajos [5-8], el controlar el proceso de maquinado electroquímico tiene una alta complejidad debido a que los resultados obtenidos al término del maquinado, dependen de la configuración de los parámetros de entrada; voltaje de polarización, distancia inter electrodos, frecuencia y ciclo de trabajo, concentración del electrolito y el flujo del electrolito, mientras que los parámetros de salida resultantes son: la taza de remoción de material (MRR, por sus siglas en ingles Material Removal Rate), Diámetro del Sobre corte (DOC, por sus siglas en ingles Diametral Overcut) y profundidad del maquinado.

Goud en [9], resalta que el proceso de maquinado electroquímico (ECM por sus siglas en inglés) está sujeto a complicaciones, debido a que no existe instrumento para medir la distancia inter electrodos de manera directa, una opción es, estimarla mediante la corriente de maquinado debido a que es proporcional a dicha distancia [10]. Kolhi et al. en [11], resaltan que el controlar el proceso de ECM utilizando técnicas de control clásicas resulta complicado debido a la necesidad de obtener el modelo matemático e inclusive los resultados obtenidos presentaban diferentes características en cada prueba.

Paul y Hiremath en [12], presentan una alternativa para controlar el proceso EDM (Maquinado por Descargas Eléctricas), mediante el uso de una técnica de Inteligencia Artificial, específicamente con Lógica Difusa, donde el controlador es diseñado en base a parámetros proporcionados por un experto, sin requerir del modelo matemático de la planta. La ventaja que se tiene al utilizar un control por lógica difusa es que asemeja las acciones de acuerdo con el proceso cognitivo del ser humano.

Keasberry y otros en [13], reportan el desarrollo de un sistema de control por lógica difusa para mantener contante la remoción de material en un proceso de maquinado electroquímico (ECM), donde afirman que el controlarlo mediante voltaje y presión de inyección del electrolito, resulta insuficiente debido a que el control nunca llego a estabilizarse. Por tal motivo Morteza en [14], presenta un estudio para mantener constante la remoción de material en un proceso ECM basado en simulación, utilizando lógica difusa, donde se resalta que para mantener una remoción de material constante es indispensable conservar la distancia inter electrodos en todo el transcurso del proceso y con ello se garantiza una profundidad homogénea.

En este trabajo se presenta el desarrollo de un control basado en lógica difusa donde se mantiene constante la distancia inter electrodos a través del monitoreo de la corriente de maquinado para el maquinado de micro canales. Los parámetros de trabajo utilizados en este trabajo son: Voltaje de polarización de 5V, 40kHz como frecuencia de los pulsos, ciclo de trabajo del 50% y la conductividad del electrolito de 4.7 Ω*cm, mientras que la distancia inter electrodos inicial del proceso es de 240μm. El material utilizado es acero AISI1018, cuya composición química es 0.14 - 0.2 C, 0.6 - 0.9 Mn, Fe balance (% peso), que es el utilizado para la fabricación de micro boquillas, micro engranes y micro bujes.

2. Metodología

El proceso PECMM trata de la remoción de material metálico por disolución anódica evitando la generación de chispas. Para ello se polariza la pieza de trabajo como ánodo y la herramienta de trabajo como cátodo utilizando una fuente pulsada.

En la Fig. 1, se muestran un esquema de los componentes que conforman al sistema de micro maquinado electroquímico por pulsos, que consta de un recipiente en donde se coloca la pieza a maquinar, que estará inmersa dentro de un electrolito formado por una mezcla de H_2O y NaCl con una concentración de masa del 16.6% lo cual genera una conductividad de 217.1 mS/cm o 4.7 Ω*cm (medido con un conductímetro HANNA modelo HI5521-02). El electrolito fluye por un tubo de acero inoxidable de

700µm de diámetro externo y 490µm de diámetro interno con un flujo aproximado de 1 l/min.

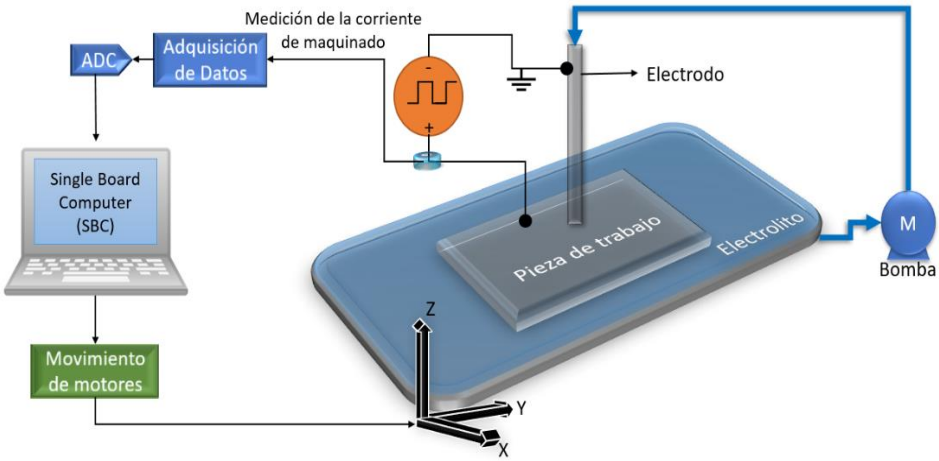


Fig. 1. Diagrama general del Sistema de Micro Maquinado Electroquímico por Pulsos.

La fuente de polarización del proceso es mediante la fuente reportada por [15], la cual se configura para proporcionar un voltaje de 5 V, con ciclo de trabajo de 50 % y la frecuencia de los pulsos es de 40 kHz. Mientras que la etapa de adquisición de datos se compone por el integrado AD536 que genera una señal de voltaje proporcional en CD del valor cuadrático medio (RMS) proporcionado por un sensor de efecto Hall (ACS712).

El electrodo es desplazado mediante un sistema mecánico de ejes coordenados (XYZ) acoplados a motores a pasos que son manejados mediante un controlador basado en el circuito integrado L297 y en un arreglo de Mosfets en una topología en puente completo para administrar la energía a los devanados de los motores. La SBC genera la acción de comandar el movimiento de los motores y monitorear el comportamiento de la corriente de maquinado, la cual permite estimar la distancia inter electrodos, dado que es proporcional, de acuerdo con la ecuación 1(Grover [1]):

$$g = \frac{A}{I_{RMS} * (R * c)}, \quad (1)$$

donde:

g = Distancia inter electrodos.

A= Área de exposición del electrodo.

I_{RMS}= Corriente RMS de Maquinado.

R= Resistividad del electrolito.

c= (Densidad^2*Velocidad de remoción específica) = $(7.87)^2$*$(3.0*10^{-2})$, constante que depende del material utilizado.

La acción de control por lógica difusa es realizada por la SBC, que es la encargada de comandar el movimiento en el eje Z; buscando mantener la distancia inter electrodos dentro de un intervalo similar a la distancia inicial.

2.1. Diseño del sistema de control difuso

En la Fig.2 se muestra el diagrama de flujo representativo del proceso de micro maquinado electroquímico por pulsos, donde es necesario establecer los parámetros iniciales del proceso, después inicia el proceso de PECMM y se comienza con el monitoreo de la corriente de maquinado para estimar la distancia inter electrodos y generar una acción de control mediante lógica difusa de acuerdo a la posición en que se encuentre, si durante el proceso se genera una remoción de material, la distancia entre los electrodos aumentará y la corriente de maquinado disminuye. Tal efecto es monitoreado y el sistema detectará el cambio en la corriente, la acción del control es compensar esa disminución en la corriente moviendo el electrodo hacia la pieza de trabajo con lo cual disminuye la distancia entre electrodos hasta que la corriente se incremente y se mantenga en un intervalo determinado por el propio control; de tal manera se pretende que la distancia entre los electrodos se mantenga constante.

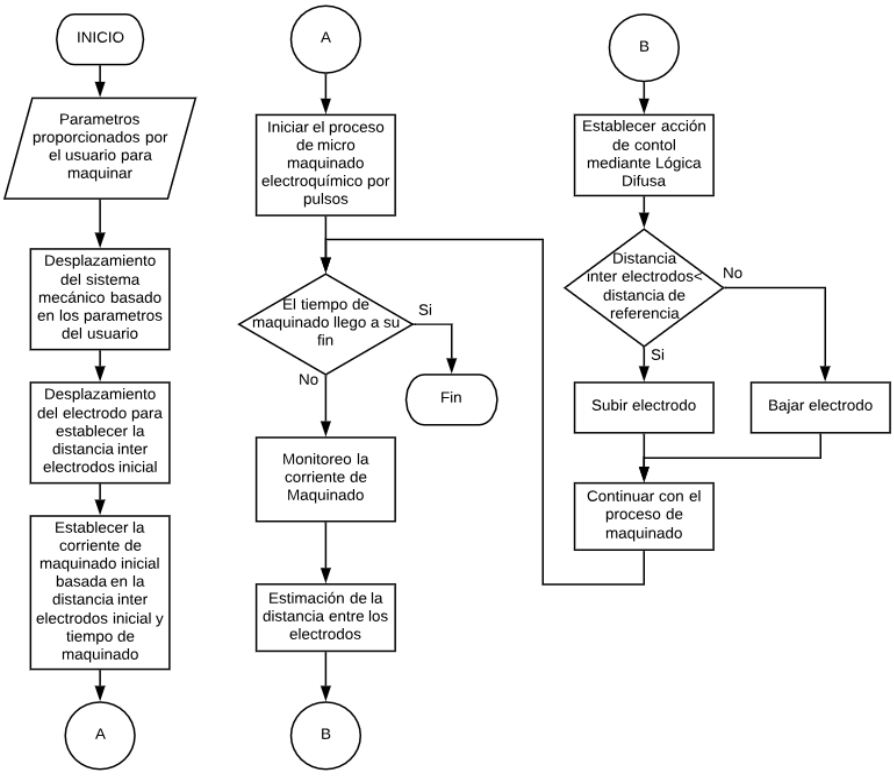


Fig. 2. Diagrama de flujo del proceso PECMM.

Durante el proceso PECMM, es necesario mantener la distancia inter electrodos en un intervalo cercano a los 240µm, a fin de obtener remoción de material constante al desplazar el electrodo de forma lineal y generar micro canales cuya profundidad sea la misma, esto es posible controlando la distancia inter electrodos mediante lógica difusa.

Por tal motivo para el diseño de las reglas de inferencia es necesario contar con la experiencia y el conocimiento de la operación del proceso PECMM, ya que son fuentes de información críticas que se requieren para identificar las condiciones óptimas de operación.

El desarrollo del sistema de control tuvo como base al diseño establecido por Mamdani, y el modelo mediante un sistema SISO (single input- single output), como el mostrado en la Fig. 3. La etapa de entrada contiene 3 funciones de membrecía, mientras que la salida cuenta con tres funciones de membrecía. La entrada al sistema de control es el error de la distancia inter electrodos, que es establecida a través del monitoreo de la corriente de maquinado, mientras que la salida es generar el movimiento del motor a pasos para mantener una la distancia cercana a los 240µm. En la Tabla 1 se muestran conjuntos difusos de la entrada o "Error" de la distancia inter electrodos, mientras que en la Tabla 2; se muestran los conjuntos difusos de la distancia requerida para desplazar el electrodo y mantener al punto de referencia.

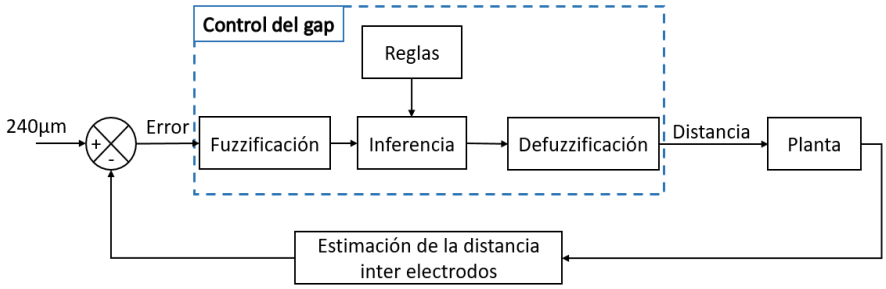


Fig. 3. Modelo SISO del sistema de control para mantener la distancia inter electrodos.

La elección de los conjuntos difusos utilizados se realizó mediante caracterización previa y definir los parámetros de trabajo idóneos del proceso PECMM [16], en la Tabla 3., se muestran las reglas de inferencia generadas. Los conjuntos difusos de entrada utilizados para el control del proceso PECMM, son representados en 3 intervalos, teniendo representaciones de una saturación abierta como μ_L y representaciones triangulares como μ_T, como se muestra en la Tabla 1.

Mientras que en la Tabla 2 se presentan los conjuntos difusos de la salida pertenecientes al universo de discurso "Distancia", definidos en tres intervalos contando con formas μ_L y μ_T. De acuerdo con Bonifacio Martín del Rio y Alfredo Sanz Molina en [17], el trabajar con conjuntos difusos con formas triangulares; requiere de menor peso computacional y la respuesta a la acción de control se realiza en un menor tiempo que al utilizar conjuntos difusos con formas de tipo campana.

Tabla 1. Conjuntos Difusos del universo de discurso error.

Variable Lingüística	Tipo de Función	Ponderación
Negativo	μ_L	$(-60, 60)$
Nulo	μ_T	$(-60, 0, 60)$
Positivo	μ_L	$(-60, 60)$

Tabla 2. Conjuntos Difusos del universo de discurso Distancia.

Variable Lingüística	Tipo de Función	Ponderación
Acércate	μ_L	$(120, 480)$
Estable	μ_T	$(120, 240, 480)$
Aléjate	μ_L	$(120, 480)$

Las reglas de inferencia relacionan los conjuntos difusos de entrada y salida para definir el comportamiento del sistema.

Tabla 3. Reglas de inferencia

Numero de Regla	Reglas de inferencia
1	Si (Error es negativo) entonces (Distancia es Aléjate)
2	Si (Error es nulo) entonces (Distancia es Estable)
3	Si (Error es positivo) entonces (Distancia es Acércate)

3. Resultados

La validación del sistema de control propuesto se realizó mediante tres pruebas, las cuales se presentan mediante el diagrama de la Fig. 4. La primera se realizó maquinado una pieza de prueba con el eje del movimiento del electrodo paralelo a la superficie de la pieza de trabajo, como se muestra en la Fig. 4(a), la segunda prueba se realizó inclinando la pieza de trabajo un ángulo de 3.5°, Fig. 4(b). Para la tercera prueba se maquinaron canales a lo largo de la superficie de la pieza de trabajo simulando un desnivel sobre la superficie, posteriormente se realizó un maquinado mediante PECMM en dirección perpendicular, Fig. 4(c), al finalizar las pruebas se realizó la medición de la profundidad. Si el sistema de control funciona de manera adecuada, la profundidad del maquinado a lo largo del canal generado mediante el PECMM debe ser de la misma magnitud.

En la Tabla 4, se muestra a manera de resumen la configuración inicial y la profundidad alcanzada al término cada una de las pruebas. La medición de la profundidad de los canales resultantes se realizó al término del maquinado utilizando un medidor de alturas digital marca FLOWLER, modelo 48976.

La configuración que se realiza al principio del proceso PECMM, es establecer el punto de referencia de la corriente de maquinado equivalente a 240µm, después el sistema genera una ventana que corresponde al cambio de un paso en sentido positivo y negativo del eje Z, con lo cual se establecen los límites de corriente de maquinado. Dado que el ajuste de altura se realiza mediante motores a pasos, este proceso permite establecer la resolución correspondiente al sistema mecánico.

Posteriormente el control efectúa la acción de mantener constante la distancia inter electrodos y en el momento que detecta que se puede realizar el ajuste de la distancia inter electrodos realiza el reposicionamiento del electrodo, para mantener constante la separación entre la pieza y la herramienta de trabajo a 240 µm.

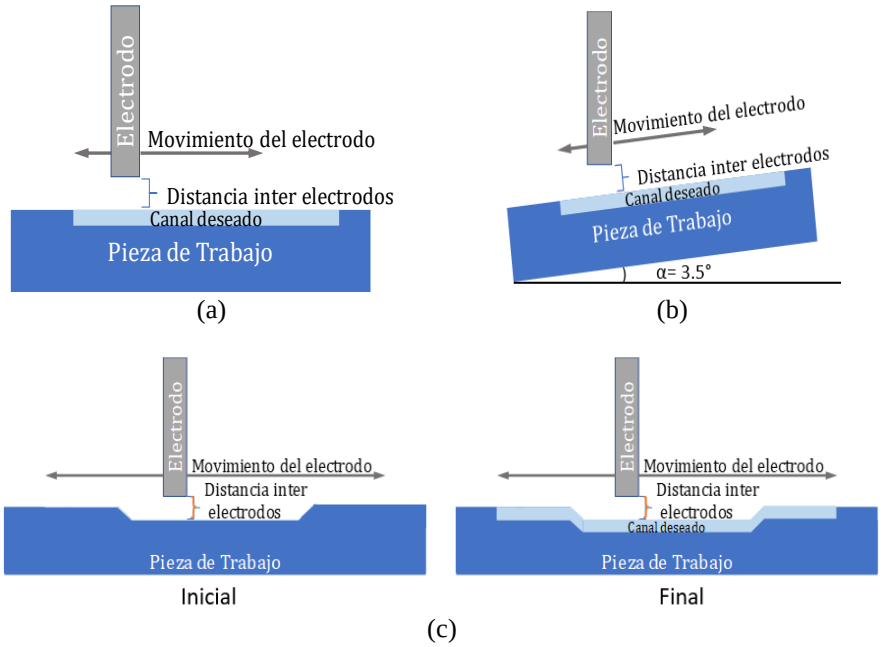


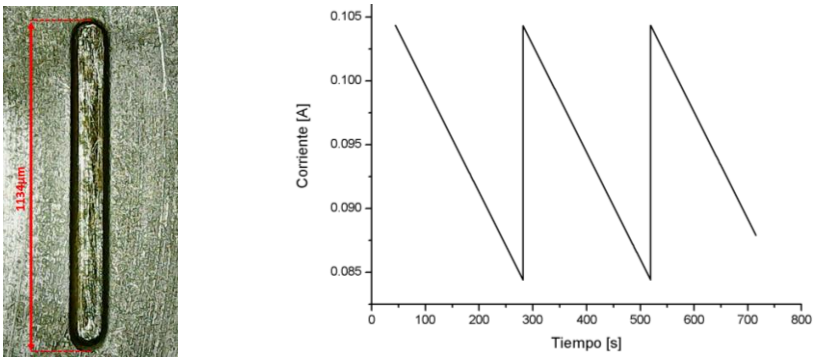
Fig. 4 . Pruebas propuestas para realizar la validación del sistema de control, (a) Prueba sin perturbaciones, (b) Prueba con la superficie de maquinado inclinada y (c) Prueba sobre una superficie con un canal como perturbación al sistema.

Para generar el canal, el electrodo hace un recorrido constantemente sobre la longitud total del canal, este movimiento se repite hasta obtener el canal deseado, en la Fig. 5(a), se muestra el resultado del maquinado de la prueba 1 y en la Fig. 5(b) el comportamiento de la corriente de maquinado; los resultados de esta prueba son considerados bajo las condiciones más apegadas a la idealidad.

La corriente de maquinado muestra una forma de onda triangular, que corresponde a los ajustes de profundidad que realiza el controlador, de acuerdo a la ecuación 1, la distancia es inversamente proporcional a la corriente RMS de maquinado por lo cual en términos de la distancia, la gráfica representa un aumento del gap dado que se ha removido material de la pieza metálica, hasta un punto que el controlador determina que requiere un reajuste de la distancia, lo que provoca acercar el electrodo a la pieza de trabajo.

En la Fig. 6(a), se muestran los resultados de la prueba 2, al someter el proceso de PECMM a una perturbación que consiste en una inclinación de 3.5°, mientras que la Fig. 6(b) muestra el comportamiento de la corriente de maquinado.

En la forma de onda de la corriente es posible apreciar que la acción de control obliga al proceso a reajustar la posición del electrodo para mantener constante la distancia inter electrodos dentro de la ventana correspondiente a la resolución del sistema mecánico, buscando la estabilidad del proceso.



(a) (b)

Fig. 5. (a) Resultados de maquinado sin perturbación y (b) forma de onda de la corriente de maquinado, sin perturbaciones.

Tabla 4. Configuración inicial y mediciones de longitud y profundidad.

Prueba	Duración del proceso PECMM [min.]	Longitud prestablecida en el sistema [mm]	Longitud resultante del proceso PECMM [mm]	Profundidad del canal resultante [µm]
1	15	1000	1134	200
2	15	1000	1136	190
3	30	2000	2136	150
4	25	2000	2135.6	120
5	40	2000	2132	300
6	40	2000	2131.5	300
7	33	2000	2130	250

En la Fig. 7(a), se muestra el resultado del sistema de control, sometido a la prueba 3, donde la profundidad de la perturbación 1(p1) es de 330µm, mientras que profundidad de la perturbación 2 (p2) es de 270µm. En la Fig.7 (b), se muestra la representación de los puntos en donde se realizaron las mediciones de profundidad de cada uno de los canales resultantes. Los resultados de las mediciones de profundidad se muestran en la Tabla 5 y en la Fig. 8, se muestran los puntos de medición al término del maquinado del canal 1 y 2.

El comportamiento de la corriente de maquinado de la prueba 3 se muestra en la Fig. 8, la forma de la corriente muestra picos abruptos que corresponde a los puntos donde el electrodo se encuentra con el borde de la perturbación, como hay un cambio de distancia entre los electrodos, hay un cambio en la magnitud de la corriente, el

controlador detecta tales cambios y ajusta la distancia hasta regresar a la magnitud de corriente en que se encontraba antes de presentarse la perturbación.

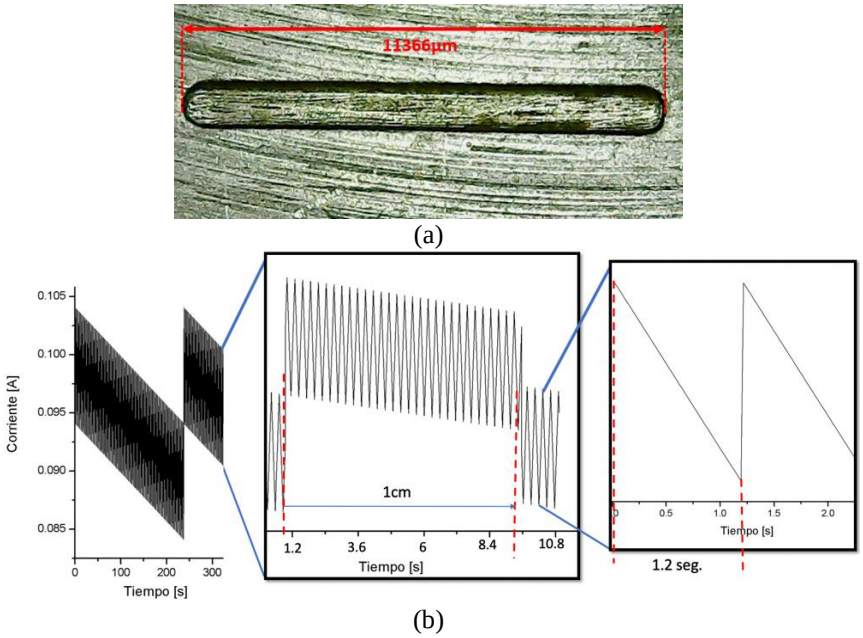


Fig. 6. (a)Resultados de maquinado con 3.5° de inclinación y (b) forma de onda de la corriente de maquinado, con una inclinación de 3.5º.

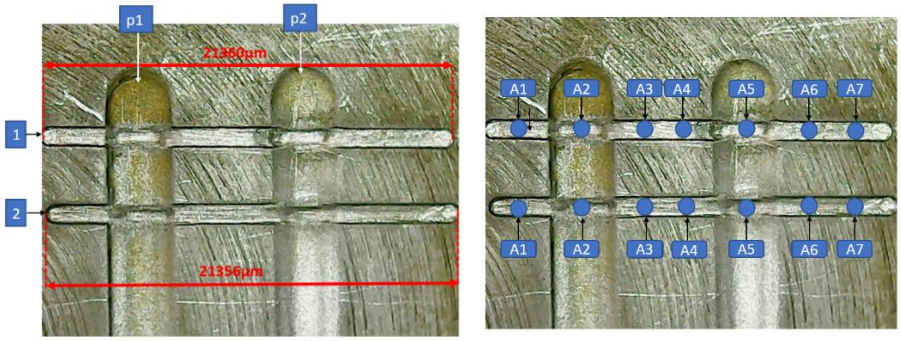


Fig. 7. (a) Resultado del sistema de control sujeto a perturbaciones, (b) Ubicación de puntos donde se realizaron las mediciones.

Tabla 5. Medición de la profundidad resultante en los canales 1 y 2.

Canal	Profundidad [µm]							Media [µm]
	A1	A2	A3	A4	A5	A6	A7	
1	150	140	160	150	150	150	150	150±2.9
2	120	110	120	120	120	120	120	118.6±2.4

Se presenta solo la forma de onda de la corriente de maquinado para la prueba 3, dado que las formas de onda son similares.

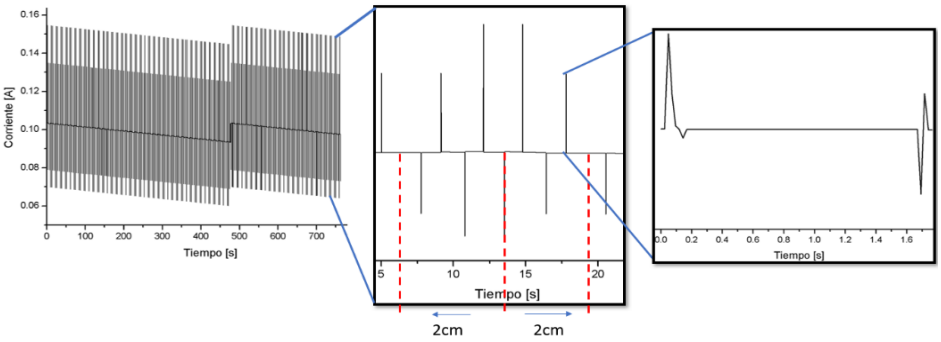
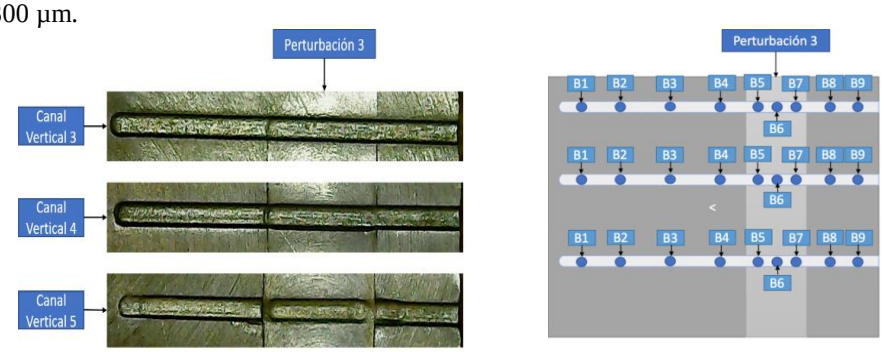


Fig. 8. Comportamiento de la corriente de maquinado del canal 1 (Prueba 3).

En la Fig. 9(a), se muestran los resultados del sistema de control, al ser sometido a una perturbación con un canal lineal vertical creciente, donde la profundidad de la perturbación 3(p3) es un canal con incremento en la profundidad. En la Fig. 9(b), se muestra la representación de los puntos en donde se realizaron las mediciones de profundidad de cada uno de los canales resultantes. Los resultados de las mediciones de profundidad se muestran en la Tabla 6.

La profundidad de la intersección del canal 3 con la perturbación 3, previamente al maquinado fue de 20µm, mientras que en la intersección del canal 4 con perturbación 3 la profundidad fue de 160µm y la profundidad del canal 5 con la perturbación 5 de 300 µm.



(a) (b)

Fig. 9. (a) Resultados del control del proceso PECMM con perturbación de un canal con profundidad incremental y (b) ubicación de los puntos en donde se realizaron las mediciones.

Tabla 6. Medición de la profundidad resultante en los canales 3, 4 y 5

Canal	Medición [µm]									Media [µm]
	B1	B2	B3	B4	B5	B6	B7	B8	B9	
3	310	300	300	300	320	320	320	300	300	301.1±2.5
4	310	300	300	300	470	460	460	300	300	302.2±4.4
5	260	250	250	250	540	540	540	260	250	248.9±7.6

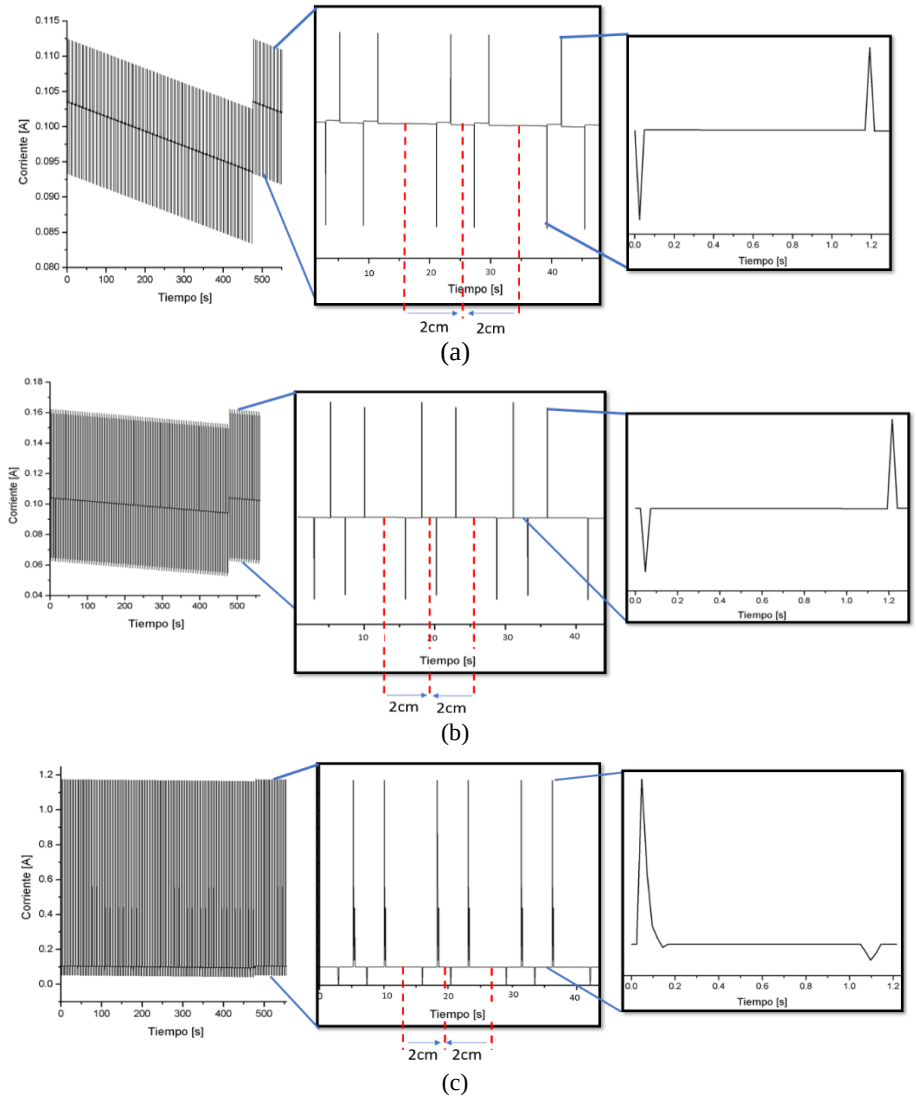


Fig. 10. Comportamiento de la corriente de maquinado; (a) del canal 3, (b) del canal 4 y (c) del canal 5.

La forma de onda de la corriente de maquinado es similar a las pruebas realizadas en la prueba 3 y 4, es decir, al recorrer el electrodo la longitud del canal que se desea maquinar, en el trayecto encuentra perturbaciones de diferente magnitud (el canal 3 con menor magnitud, mientras que el canal 5 con mayor magnitud).

Dichas perturbaciones se ven reflejadas en la forma de corriente y el distanciamiento entre cada pico de corriente coincide con la distancia recorrida por el electrodo. La acción de control se ve reflejada cuando el electrodo cae dentro del canal de

perturbación, el gap aumenta drásticamente y la corriente disminuye, el controlador ajusta la distancia hasta que la corriente regresa a la condición de estabilidad (magnitud de corriente de maquinado).

En caso opuesto, si el electrodo sale del canal de perturbación, el gap disminuye abruptamente y corriente aumenta; el controlador ajusta la distancia alejándose de la pieza de trabajo hasta que la corriente regresa a la condición de estabilidad.

En la Fig. 10, se muestra el comportamiento de la corriente de maquinado de los canales 3, 4 y 5. La diferencia entre las figuras es la magnitud de los picos de corriente, y es debido a las diferentes profundidades que se presentan en las perturbaciones. Para el canal 3, el pico de corriente es de 112 mA; para el canal 4 es de 164 mA y para el canal 5 es de 1.185A.

4. Conclusiones

El presente trabajo se muestra el desarrollo de un sistema de control difuso que mantiene constante la distancia entre electrodos. De acuerdo a la literatura y a experimentación previa, el PECMM es un proceso de maquinado complejo, en el cual intervienen fenómenos, físicos, químicos y eléctricos, donde una característica que se requiere cumplir es mantener constante la distancia entre los electrodos para establecer una remoción homogénea de material. Al momento no existe herramienta que permita realizar la medición de dicha distancia, una alternativa es estimarla mediante el monitoreo de la corriente de maquinado.

El controlar el proceso PECMM a través de sistemas clásicos requiere de conocer el modelo matemático de la planta, lo cual resulta complejo de obtener, ya que involucra diversas y diferentes variables. Una alternativa viable presentada en este trabajo es mediante un control inteligente, de forma específica por medio de lógica difusa. El diseño y aplicación del regulador difuso tipo SISO con inferencias tipo Mamdani permite mantener dentro de un intervalo adecuado de operación el proceso PECMM, los resultados permiten validarlo, inclusive con las limitantes mecánicas con que está construido el sistema pues el ajuste se realiza con motores a pasos. Se realizaron tres diferentes pruebas para validar el controlador, la primera es ajustando la pieza para considerar los parámetros idóneos.

En esta prueba se visualiza el ajuste continuo de la distancia conforme va removiendo material. La segunda prueba se realiza al inclinar la pieza de trabajo, donde el controlador ajusta continuamente la distancia, manteniendo el gap lo más cercano a una constante.

La tercera prueba fue generando irregularidades en la pieza de trabajo consideradas como perturbaciones, en la forma de onda de la corriente de maquinado se puede visualizar cuando el controlador ajusta la distancia hasta regresar a un valor de corriente previa a la detección de la perturbación.

El resultado en cualquier de las pruebas es un canal uniforme, con un error máximo de 7.6µm. Como trabajo a futuro se debe mejorar la respuesta mecánica del proceso; además de realizar una comparativa de método aquí presentado con otros métodos.

Referencias

1. Grover P, M.: Fundamentos de Manufactura Moderna. 4th edn. Mc Graw Hill, México (2010)
2. Kalpakjian, S., Schmid, S.R.: Manufactura, Ingeniería y Tecnología, 2nd edn. Prentice Hall, México (2005)
3. Hassan, A., Gawad, E.-H.: Advanced Machining Processes, 5th edn. McGraw-Hill, Alexandia (2005)
4. Katz, Z., Tibbles J.C.: Analysis of micro-scale ECM Proccess. The International Journal of Advance Manufacturing Technology 35(154), 923–928 (2012)
5. Sengottuvel, P., Satishkumar, S., Dinakaran, D.: Optimization of Multiple Characteristics of EDM Parameters Based on Desirability Approach and Fuzzy Modeling. Procedia Engineering 64, 1069–1078 (2013)
6. Shabgard, M., Badamchizadeh, M.G., Ranjbary, K.: Fuzzy approach to select machining parameters in electrical discharge machining (EDM) and ultrasonic-assisted EDM processes. Journal of Manufacturing Systems 32(1), 32–39 (2013)
7. Mohd, A., Sarkheyli, A., Mohd Zain, A., Habibollah, H.: Fuzzy logic for modeling machining process: a review. Artificial Intelligence Review 43(3), 345–349 (2013)
8. Katz, Z., Tibbles J.C.: Analysis of micro-scale ECM Proccess. The International Journal of Advance Manufacturing Technology 35(154), 923–928 (2012)
9. Klocke, F.: Results of Surface Integrity and Fatigue Study of PECM and PEO processed γ-TiAl for Turbine Applications. Procedia CIRP19 vol. 3, pp. 131–136 (2016)
10. Lilong Baoji, M., Cheng, P., Yun, K.: Effect of Power Signal Waveform on Shape Accuracy in Electrochemical Drilling. Materials Science and Engineering 392, pp. 1–8 (2018)
11. Kohli, A., Wadhwa, A., Virmani, T., Jain, U.: Optimization of Material Removal Rate in Electrical Discharge Machining Using Fuzzy Logic. World Academy of Science, Engineering and Technology International Journal of Mechanical and Mechatronics Engineering 12, pp. 2839–2844 (2012)
12. Hung Lea, V., Khanh Tran, D.: Extending fuzzy logics with many hedges. Fuzzy Sets and Systems 1(1), 45–52 (2018)
13. Keasberry, J.V., Labib, W.A., Atkinson, J., Frost W.H.: A Fuzzy Logic Control Approach to Electrochemical Machining (ECM). In: Proceedings of the 34th International, Formerly the International Machine Tool Design and Research Conferences 34, pp. 153–160 (2008)
14. Sadegh, A.M.: Controlling Electrochemical Machining by Using a Fuzzy Logic Approach. Journal of Modern Processes in Manufacturing and Production 6(2), 61–71 (2017)
15. Martínez Alvarado, R., Granda Gutiérrez, E.E., Zuñiga, L.M., Guerrero, R.: Pulsed Power Supply for Electrochemical Machining. In: 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC) 3(15), 1–6 (2015)
16. Gallegos Miranda, I., Pérez Martínez, J.A., Granda Gutiérrez, E.E., Flores Fuentes, A.A.: Caracterización de Proceso de Maquinado Electroquímico por Pulsos (PECM). En: Coloquio de Investigación Multidisciplinaria 6 (1), 414–421 (2018)
17. Martín del Río, B., Sanz Molina, A.: Redes Neuronales y Sistemas Borrosos. 3rd edn. Alfaomega, México (2007)

ISSN 1870-4069

Movimientos en pierna robótica para el pateado de un balón a través de reinforcement learning

Gerardo Franco Delgado, M. De La Rosa, C. E. De Los Santos, J. D. Carrillo

IXMATIX ROBOTICS, San Luis Potosí, S.L.P, México
gfranco@ixmatix.com, {m.delarosa, c.delossantos, dcarrillo}.ixmatix@gmail.com

Resumen. En este trabajo, se presentan los resultados obtenidos del aprendizaje de una pierna robótica para el pateado de un balón. El entrenamiento se realizó por medio de reinforcement learning (RL) utilizando el algoritmo Q-learning. Esto con el objetivo de probar la fiabilidad del aprendizaje para futuros trabajos que involucren caminado de robots cuadrúpedos y manipulación de objetos con actuadores. El entrenamiento fue realizado dentro de un entorno de simulación en Gazebo con la intención de reproducir de forma real el aprendizaje obtenido de la simulación, es decir, que el agente pueda realizar el pateado del balón en una pierna robótica real, a través de un entrenamiento simulado.

Palabras clave: reinforcement learning, pierna robótica, Q-learning, manipulación robótica.

Movements in Robotic Leg for Kicking A Ball Using Reinforcement Learning

Abstract. The results shown in this work were obtained from a robotic leg kicking a ball using reinforcement learning (RL), specifically the Q-learning algorithm. The goal of this study was to test the reliability of learning using RL, all this with the aim to apply these concepts in future work, like quadruped walking robots and object manipulation using electromechanical actuators. The training was carried out within a simulation environment in Gazebo with the intention of reproducing in a real way the learning obtained from the simulation, so in the end the agent can perform the kicking of the ball on a real robotic leg, through training simulated.

Keywords: reinforcement learning, robotic leg, Q-learning, robotic manipulation.

1. Introducción

En la literatura actual, se han visto avances sobresalientes en el campo de la inteligencia artificial mediante el uso de reinforcement learning. Una de las

principales características de este, es que imita la forma en cómo aprendemos los humanos, donde a través de la práctica se mejora en la actividad que realiza [11]. Este enfoque ha logrado dar soluciones a retos complejos, como ganarle al campeón mundial de Go [22] y la creación de un algoritmo que permite a un agente aprender a jugar StarCraft II [27], logrando vencer a humanos profesionales sin importar lo complejo del entorno.

En este trabajo se busca demostrar la factibilidad del uso de algoritmos de auto-aprendizaje con RL para el movimiento de una pierna robótica, donde esta sea capaz de patear un balón sin la necesidad de definir la posición de este ni los ángulos de las articulaciones.

Se han desarrollado diversas investigaciones para el pateado de un balón basados en el movimiento de una pierna humana, entre ellas podemos mencionar a Roboleg [21], un robot desarrollado para la evaluación de tenis y balones, y el robot de 9DOF (por sus siglas en inglés, Degrees Of Freedom) que utiliza el método Denavit-Hartenberg para el análisis de cinemática y un método recursivo langriano para definir la dinámica [26].

Por otra parte, nuevas investigaciones han tratado el pateado del balón en robots de tipo humanoide como HOAP-3 [23] y NAO [13], el cual utiliza movimientos basados en trayectorias como un conjunto de curvas de Bezier por medio de estabilización del centro de masa y un controlador PID. A diferencia de estas propuestas, nuestro trabajo se asemeja a lo realizado por Riedmiller et al. (2001) en la competencia RoboCup, creando un agente que utiliza RL para el pateado del balón, definiendo 536 acciones que el agente puede escoger por ciclo [19]. En nuestro trabajo definimos el pateado por medio de una pierna robótica de 3DOF que se define con un total de 1848 acciones disponibles por estado, el cual es entrenado en simulación y aplicado en la realidad.

2. Descripción del sistema

El entrenamiento para el pateado del balón se realiza únicamente en la simulación y los resultados obtenidos se muestran tanto en el entorno real como en el simulado. El algoritmo Q-learning [29] es con el cuál se busca maximizar una acción en un determinado estado.

En cuanto al reconocimiento del balón, para conocer sus coordenadas, se tiene un clasificador de objetos por medio de OpenCV [25] que hace uso del algoritmo de Viola-Jones. [8, 28]

2.1. Entorno

Las características para el pateado del balón pueden definirse de acuerdo a la clasificación de entornos para agentes [20] de la siguiente forma:

- Observable: La cámara observa el entorno completo, obteniendo la información necesaria para el agente.

- Determinista: Un entorno determinista es aquel en que el siguiente estado está completamente determinado por el estado actual y la última acción ejecutada por el agente.
- Episódico: La experiencia de la pierna se encuentra dividida por episodios independientes, donde al iniciar un nuevo episodio se empieza desde un nuevo estado.
- Semi-dinámico: Durante el entrenamiento el entorno se mantiene inmutable, mientras que el agente va aumentando su rendimiento.

Los objetos delimitados dentro del entorno se definen por la pierna robótica, una mesa, una cámara y un balón de fútbol estándar (diámetro entre 68 y 70 cm).

Se utiliza un ángulo en picado [12] para la cámara, de tal manera que pueda observar hacia abajo el entorno completo, con un ángulo aproximado de 45 grados bajo la horizontal. El balón tiene límites en su ubicación, definidos por el alcance de la pierna robótica.

2.2. Diseño mecánico

El diseño está basado en las patas traseras de un canino e intenta imitar su movimiento de locomoción, además de tener en consideración las especificaciones de los motores hidráulicos. Se tiene una estructura con cada una de las partes que conforman la pierna robótica, este diseño será usado posteriormente para la simulación.

El ensamblado de la misma, se realizó con impresiones en 3D y perfiles de aluminio; cabe mencionar que dicho diseño es original y es realizado por el equipo de trabajo.

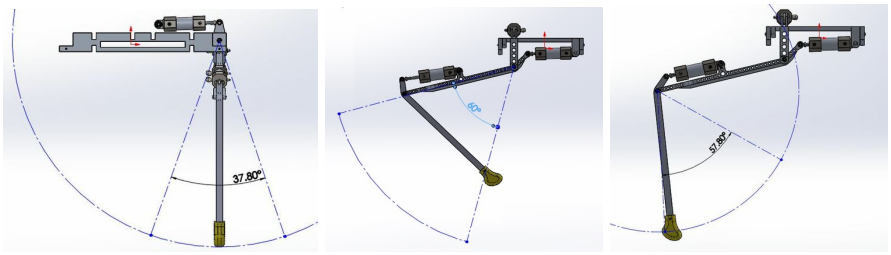
2.3. Configuración y medidas del robot

La pierna robótica consta de tres actuadores y articulaciones, que permiten tener 3DOF. Cada articulación se nombra como se muestra en la figura 2.3.

El movimiento de la pierna robótica se ve limitado debido a su estructura mecánica. Cada articulación tiene un ángulo de apertura que se calcula desde el punto máximo que permite el actuador, hasta el punto mínimo. Los ángulos de apertura para cada articulación son los siguientes: 37.8 grados para la coxa, 60 grados para el fémur y 57.8 grados para la tibia, tal como se muestra en la figura **??**.

3. Simulación

Para asegurar una aproximación acertada de la simulación de robots en entornos virtuales, es importante definir diversas características y propiedades físicas sobre cada pieza, en este caso, de la pierna del robot. Tales consideraciones incluyen el peso, centro de masa, la matriz diagonalizable del momento de



(a) Coxa
Vista lateral izquierda
37.8 grados de apertura.

(b) Fémur
Vista frontal
60 grados de apertura.

(c) Tibia
Vista frontal
57.8 grados de apertura.

Fig. 1. Vistas de los ángulos de apertura para cada actuador en la pierna robótica.

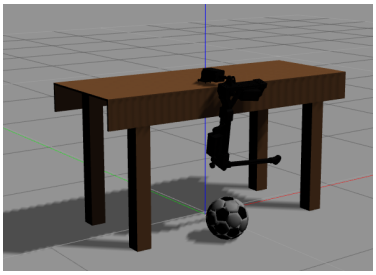


Fig. 2. Simulación de la pierna del robot.

inercia, así como una representación de mallas 3D para definir las colisiones y su visualización. En la figura 2, se muestra el entorno de simulación.

Estas propiedades se definen en un archivo con formato URDF [24] para poder utilizarlas en Gazebo [10] y ROS [18] simultáneamente. El entorno de simulación se muestra en el video [3] (accesible desde este url[1]).

3.1. Controladores

Los motores hidráulicos de cada articulación, cumplen con las características en esfuerzo y velocidad (rad/s) definidos en la ficha de datos del proveedor, con el modelo: 32TRCHEBU9A65 [14].

Estos límites se definen mediante la siguiente expresión: $-k_v(v - v^+)$. Donde k_v determina la escala del límite del esfuerzo, v expresa velocidad y v^+ el límite de la velocidad [9].

[1] `https://www.youtube.com/watch?v=F34zIs0cI-0`

4. Entorno real

4.1. Características del entorno

La simulación y el entorno real tienen las mismas condiciones para el aprendizaje del pateado del balón. La pierna robótica se encuentra en una mesa de 82 cm de altura, dejando un espacio de 20 cm desde el suelo hasta el extremo inferior de la tibia, como se muestra en la figura 3.

Cada articulación se controla mediante un sistema hidráulico como se muestra en el video [2] (accesible desde este url[2]). Los grados de movimiento coinciden con los descritos en la sección 2.3.

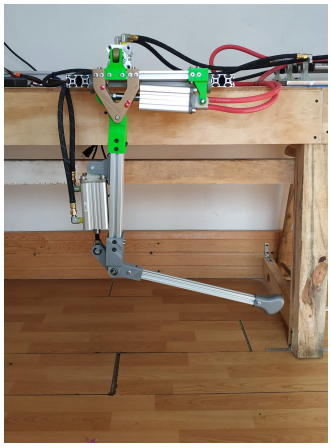


Fig. 3. Pierna robótica en entorno real.

4.2. Reconocimiento del balón en tiempo real

Se entrena un clasificador en cascada [30] para el reconocimiento del objeto mediante el algoritmo Viola-Jones utilizando OpenCV. En el entrenamiento se usan 644 imágenes positivas donde se incluye el objeto que se quiere detectar (en este caso, el balón), también se utilizan 3,019 imágenes negativas, es decir que no contienen el objeto deseado.

En cuanto a las imágenes positivas se indica la posición donde se localiza el objeto, finalmente con las herramientas proporcionadas de OpenCV se realiza el entrenamiento en cascada con un total de 20 episodios [15].

Los resultados del entrenamiento para la detección del balón y la obtención de sus coordenadas, se pueden observar en el video [4] (accesible desde este url[3]).

[2] `https://www.youtube.com/watch?v=qmeyC5y9fQE`
[3] `https://www.youtube.com/watch?v=_Cl9SZCBb0E`

5. Misión a ejecutar

El objetivo consiste en que el agente logre aprender a patear un balón dentro de un rango definido, a través de dos movimientos en sus articulaciones. En un posible caso, el agente podría hacer un movimiento para acomodarse y un segundo para generar la patada con la mayor fuerza posible.

El balón cambia su posición inicial en cada episodio; un episodio consiste en reiniciar la posición del balón y finaliza cuando la pierna ejecuta los movimientos en los motores hidráulicos. En las primeras pruebas la posición del balón se mantiene, con la intención de enfocarse y validar que es posible patear el balón a pesar del gran campo de acciones disponibles.

6. Mecanismos para el aprendizaje y control de motores

6.1. Estados

Los estados son representados por las posiciones del balón. Al cambiar sus coordenadas, las acciones que deben realizar los pistones cambian. El rango de la posición del balón (figura 4) en el eje x es de 0.1 a -0.1 m y en el eje y es de 0.1 a -0.1 m. Las posiciones cambian cada 0.05 m, por lo que se tiene un total de 25 estados, que representan las posibles posiciones del balón dentro del entorno. (El eje z siempre mantiene su mismo valor).

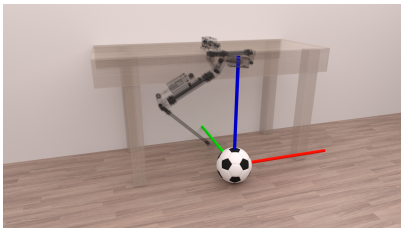


Fig. 4. Representación tridimensional dónde la línea roja representa el eje x, la verde el eje y y la azul el eje z.

6.2. Acciones

Por acciones se refiere a los movimientos que pueden ser ejecutados por los actuadores. Si las acciones se representan por cada grado que puede mover cada articulación de la pierna, existirían 126,540 posiciones diferentes, para reducir este número la articulación se mueve cada 5 grados lo que da un total de 924 posibles acciones para representar todas las posiciones. Tomando en cuenta que una acción conlleva dos movimientos, se estaría hablando de 853,776 acciones,

sin embargo, se utilizaron dos matrices de pesos donde se busca la mejor acción en determinado tiempo para el pateado del balón y el acomodo de la pierna, dando así un total de 1,848 acciones.

6.3. Q-learning

Q-learning es el algoritmo ocupado para el entrenamiento del pateado del balón, su funcionamiento se puede describir de la siguiente forma:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \lambda maxQ(s_{t+1}, a)]. \tag{1}$$

La función Q se va actualizando conforme pasa el tiempo, esta ayuda al agente a realizar la mejor acción con la experiencia adquirida durante el entrenamiento.

Dependiendo del estado (s) se ejecuta cierta acción (a), en un tiempo determinado (t). Conforme avanza el tiempo, existe una actualización entre la recompensa futura y la antigua al valor Q existente, esto se puede expresar como la siguiente equivalencia:

$$A^{new} = A + B. \tag{2}$$

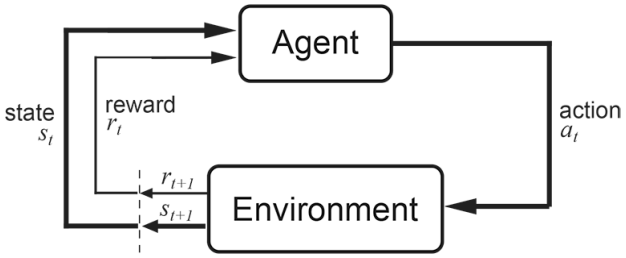


Fig. 5. Representación del entorno y el agente del algoritmo Q-learning [16].

Como se observa en el algoritmo Q-learning (figura 5), el agente recibe recompensas para indicar su desempeño en forma de retroalimentación. En este caso, las recompensas son determinadas por la magnitud que se tiene entre la posición final del balón (x_f^{ball},y_f^{ball}) con respecto a la posición en la que se inició el episodio (x_i^{ball},y_i^{ball}):

$$reward_t = \sqrt{(x_f^{ball} - x_i^{ball})^2 + (y_f^{ball} - y_i^{ball})^2}. \tag{3}$$

En cuanto a la selección de los mejores movimientos para el pateado del balón, se realiza con base a la experiencia adquirida de la primera y segunda acción, estas son determinas por medio de las ecuaciones 4 y 5:

$$W_1^{new}(s_t, a_t) \leftarrow W_1(s_t, a_t) + \alpha r_t^{reward} + \lambda max(W_2(s_{t+1}, a_t)), \tag{4}$$

$$W_2^{new}(s_t, a_t) \leftarrow W_2(s_t, a_t) + \alpha r_t^{reward}, \tag{5}$$

donde α representa el ritmo del aprendizaje de cada nueva experiencia, es decir, qué tanto el agente puede aprender de la acción realizada, y λ indica el factor de descuento que valora las recompensas de las acciones pasadas, donde 0 significa que sólo importa la recompensa por la acción realizada y no toma en cuenta la experiencia adquirida del agente. [20]

En las gráficas de la sección 7.3 se toma el promedio de los pesos de la primera y segunda acción ejecutada, se muestra que al pasar los episodios se van seleccionando solo las mejores acciones con base a sus pesos:

$$\frac{W_1(s_t, a_t) + W_2(s_t, a_t)}{2}. \tag{6}$$

7. Resultados

7.1. Estados base

En las primeras pruebas el balón siempre mantiene la misma posición. El objetivo del agente es buscar y seleccionar la mejor acción que pueda realizar cada uno de los motores hidráulicos, según el sistema de recompensas.

Actualmente se está entrenado para que sin importar la posición del balón (dentro de los 25 estados posibles como se describe en la sección 6.1) siempre pueda hacer contacto con este de la mejor manera posible y de esta forma, sin importar el estado, se ejecute una acción, que a través de los movimiento de los motores hidráulicos, realice una patada que pueda alejar al balón a una mayor distancia con respecto a las coordenadas iniciales del episodio.

En las secciones 7.2 y 7.4 se muestran los resultados obtenidos con el entrenamiento tanto de forma simulada como real.

7.2. Entrenamiento del agente en el entorno de simulación

En la figura 6, se observa la ponderación de las acciones realizadas (weights), conforme pasan los episodios se ejecutan las que tienen mayor peso (sección 6.3).

En la gráfica se puede observar que al principio del entrenamiento los resultados llegan a ser extremos entre puntos, esto se debe a que el agente se encuentra en fase de búsqueda, probando diferentes acciones hasta encontrar la mejor, en un episodio se puede tener una recompensa alta y en otro no hacer contacto con el balón.

El agente evalúa por medio de su matriz de pesos, el desempeño de la acción ejecutada y, de esta forma, considera cual acción futura sería la más conveniente. En dicha gráfica (figura 6), se muestra que conforme transcurren los episodios, se seleccionan los movimientos que tienen mayor peso con base a la experiencia adquirida. En este caso, a partir del episodio 250, se ejecutan las mejores acciones hasta el punto de llegar a ser constante con los movimientos que se realiza.

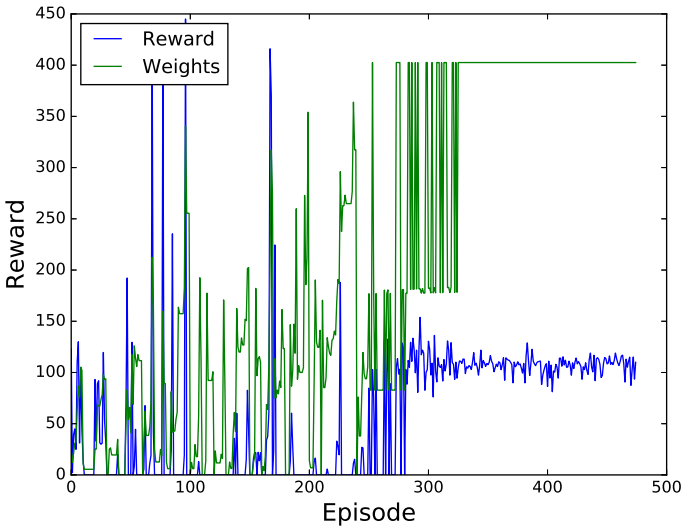


Fig. 6. El trazado con línea verde representa los pesos de las acciones y el azul indica la recompensa final que obtuvo el agente en cada episodio.

La recompensa (rewards) indica el desempeño del pateado del balón (el desplazamiento recorrido con respecto a su punto inicial); esta representación de datos es la más relevante para medir los resultados del agente, ya que muestra el aprendizaje e indica como mejora el pateado al transcurrir los episodios.

En la figura 7, se observa la recompensa por cada acción que realiza el agente. En los primeros 200 episodios se visualizan picos repentinos de recompensas mayores a 400 puntos, seguidos de caídas abruptas; esto se debe a que ciertas acciones llevan al agente a recibir puntuaciones altas, sin embargo, si se realiza la misma acción y se obtiene una recompensa baja, se genera poca probabilidad para volver a seleccionarla, por lo que el agente opta por los movimientos que siempre producen un contacto con el balón de forma constante. De esta forma se mantiene una recompensa constante de aproximadamente 100 puntos, lo que demuestra que siempre se está pateando el balón, variando el desplazamiento de este en cada episodio.

Se puede observar en el video [1] (accesible desde este url[4]) el entrenamiento realizado en el motor de simulaciones de Gazebo, se muestra como la pierna robótica ejecuta el pateado del balón al transcurrir los episodios, con el fin de ir mejorando su desempeño. Gazebo no cuenta con la propiedad de simular los coeficientes de fricción de rodadura para el balón por lo que cambia el desempeño dentro del entorno virtual con respecto al real.

[4] `https://www.youtube.com/watch?v=NlCBaihAQ9k`

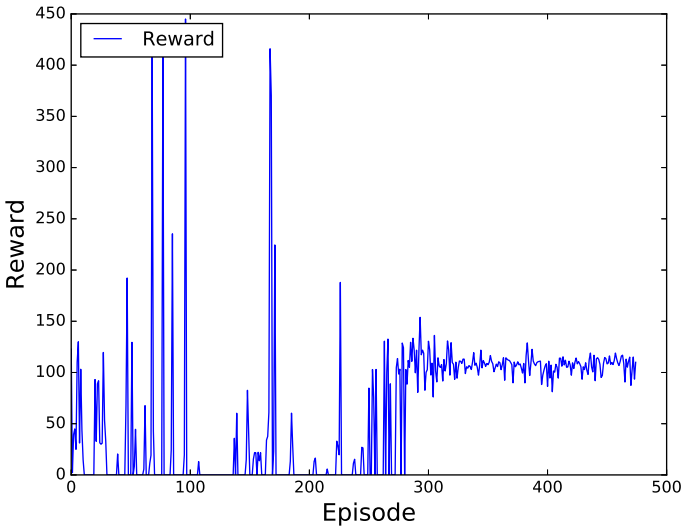


Fig. 7. Recompensas obtenidas por las acciones del agente.

7.3. Análisis del aprendizaje

Se ejecutaron 6 entrenamientos con 450 episodios cada uno. En estos ocurren variaciones, como se muestra en la tabla 1, dependiendo de las diferentes acciones ejecutadas por el agente.

Cada entrenamiento tiene una duración aproximada de una hora y 28 minutos, las características del equipo de computo utilizado son:

- Modelo del procesador: I5-7200u,
- Familia del procesador: 7ª generación de procesadores Intel Core,
- Número de núcleos del procesador: 2,
- Memoria: 8 GB DDR4-SDRAM,
- Almacenamiento: 100 GB HDD,
- Gráficos: Intel HD Graphics 620.

En las gráficas de aprendizaje (figura 9 a 14), se observa un comportamiento similar, donde el agente al finalizar su entrenamiento, siempre selecciona las mejores acciones con base a su experiencia de los episodios pasados y de esta forma se mejora el pateado del balón. La recompensa en todos los casos se mantiene sobre un rango, lo que indica que se hace contacto con el balón. Generalmente a partir del episodio 250 tiene una mejora notable, dónde el agente realiza movimientos en los actuadores que logran el pateado del balón de una forma casi constante.

El promedio de la puntuación fue de 97.33 unidades y, a partir del episodio 303.33, generalmente, se empezaba a tener un mejoramiento de forma continua.

Tabla 1. Datos de aprendizaje.

Episodio	Promedio desempeño del agente	Episodio con mejora continua
Entrenamiento 1	82	320
Entrenamiento 2	125	280
Entrenamiento 3	85	270
Entrenamiento 4	95	305
Entrenamiento 5	105	370
Entrenamiento 6	92	275

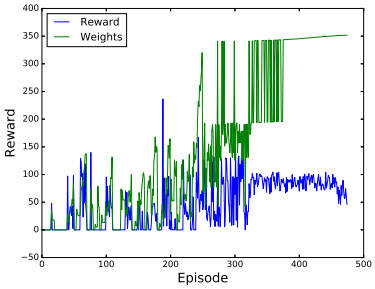


Fig. 8. Entrenamiento 1.

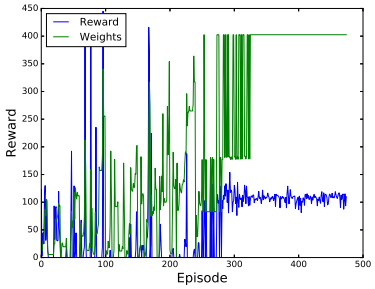


Fig. 9. Entrenamiento 2.

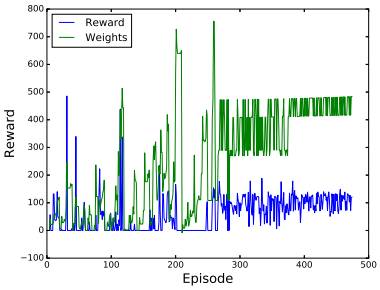


Fig. 10. Entrenamiento 3.

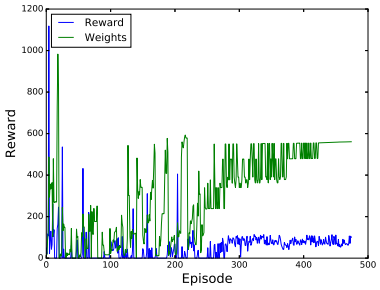


Fig. 11. Entrenamiento 4.

7.4. Resultados pateado del balón

Entorno simulado. Los resultados de la sección 7.3, para el entorno simulado se observan en el video [7] (accesible desde ete url[5]). En dicho video se muestran las ejecuciones de cada uno de los entrenamientos para el pateado del balón.

Entorno real. La matriz de pesos resultante de los entrenamientos, se trasladó a la pierna robótica real para evaluar si esta puede ejecutar el pateado del

[5] `https://www.youtube.com/watch?v=KItPI11y9cQ`

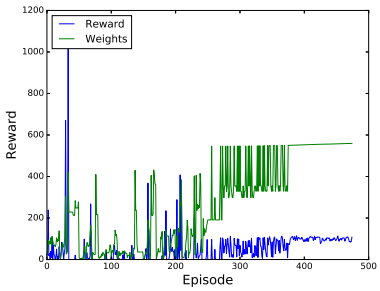


Fig. 12. Entrenamiento 5.

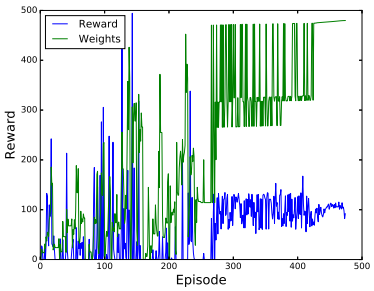


Fig. 13. Entrenamiento 6.

balón de manera efectiva, a partir del entrenamiento simulado y a través de la acción ejecutada por los motores hidráulicos. Estos resultados se pueden observar en video, [5] (accesible desde este url[6]), en el cual, se muestra los resultados del pateado del balón en el entorno real, lo que confirma que a través de la simulación, se puede entrenar de manera correcta el pateado del balón para poder ser aplicado en la realidad.

8. Conclusión

Los resultados de la sección 7.3 indican que en todos los casos, al finalizar los entrenamientos, la recompensa se muestra constante. Esto es debido a que el agente opta por movimientos que siempre generan contacto con el balón, en lugar de arriesgarse a tomar acciones que puedan llegar a tener recompensas más altas a costa de estabilidad. Un punto crítico en nuestra investigación es el uso de la matriz de pesos en el entorno real. Los resultados demuestran una viabilidad en el uso de simulaciones para obtener los pesos y, de este modo, utilizarlos en la pierna robótica real sin necesidad de un entrenamiento o adaptación; sin embargo, esto no es garantía de que sean los pesos óptimos a utilizar. Mejorar la simulación permitiría al agente tener recompensas cercanas a las esperadas en la realidad, rompiendo de este modo la brecha que pueda existir entre los pesos del entorno real y el simulado. Para mejorar la simulación se tendrían que considerar diversas cuestiones físicas como resistencia a la rodadura, y de esta forma, obtener comportamientos fidedignos. Además de esto, el aumento de calidad y realismo gráfico de la simulación, permitiría al algoritmo de detección de coordenadas un traslado rápido de aprendizaje del mundo simulado al mundo real.

Los resultados demuestran una viabilidad en el uso de estos algoritmos para movimiento de piernas robóticas. Futuros trabajos incluirían el uso de los algoritmos mencionados en este trabajo para el caminado de un cuadrúpedo y manipulación de objetos con robots industriales de 7DOF. Actualmente se

[6] `https://www.youtube.com/watch?v=tU4fGfTBfgI`

está trabajando para que la pierna robótica realice el pateado sin importar las posiciones del balón, los avances se pueden observar en el video [6] (accesible desde este url[7]), así como también se están generando cambios en el aprendizaje del agente para obtener un mejor desempeño de este.

Referencias

1. Entrenamiento en entorno simulado. `https://www.youtube.com/watch?v=NlCBaihAQ9k` (2019), último acceso: 2019/05/11
2. Pruebas de movimiento. `https://www.youtube.com/watch?v=qmeyC5y9fQE` (2019), último acceso: 2019/05/11
3. Pruebas de simulación. `https://www.youtube.com/watch?v=F34zIs0cI-0` (2019), último acceso: 2019/05/11
4. Reconociminto de balón con opencv. `https://www.youtube.com/watch?v=_Cl9SZCBb0E` (2019), último acceso: 2019/05/11
5. Resultados entorno real. `https://www.youtube.com/watch?v=tU4fGfTBfgI` (2019), último acceso: 2019/05/11
6. Resultados entorno real en diferentes posiciones. `https://www.youtube.com/watch?v=BUe4uNiYSR0` (2019), último acceso: 2019/05/11
7. Resultados entorno simulado. `https://www.youtube.com/watch?v=KItPI11y9cQ` (2019), último acceso: 2019/05/11
8. Castrillón, M., Déniz, O., Hernández, D., Lorenzo, J.: A comparison of face and facial feature detectors based on the viola–jones general object detection framework. Machine Vision and Applications 22(3), 481–494 (2011)
9. Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Tsouroukdissian, A.R., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., et al.: ros_control: A generic and simple control framework for ros. The Journal of Open Source Software 2(20), 456–456 (2017)
10. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). vol. 3, pp. 2149–2154. IEEE (2004)
11. Kormushev, P., Calinon, S., Caldwell, D.: Reinforcement learning in robotics: Applications and real-world challenges. Robotics 2(3), 122–148 (2013)
12. Lachat Leal, C., et al.: Didáctica de la traducción audiovisual: enseñar a mirar (2011)
13. Müller, J., Laue, T., Röfer, T.: Kicking a ball – modeling complex dynamic motions for humanoid robots. In: Ruiz-del Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010: Robot Soccer World Cup XIV. pp. 109–120. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
14. Parker: Compact hydraulic cylinders. `http://www.parker.com/literature/Industrial%20Cylinder/cylinder/cat/english/HY08-1137-7NA_che-chd.pdf` (2008), último acceso: 2019/05/11
15. Puttemans, S.: Cascade classifier training. `https://docs.opencv.org/3.3.0/dc/d88/tutorial_traincascade.html` (2017), último acceso: 2019/05/11
16. Qicui Yan, Quan Liu, Daojing Hu: A hierarchical reinforcement learning algorithm based on heuristic reward function. Reproducido como en la Figura 1 en [17] (2010), último acceso: 2019/05/11

[7] `https://www.youtube.com/watch?v=BUe4uNiYSR0`

17. Qicui Yan, Quan Liu, Daojing Hu: A hierarchical reinforcement learning algorithm based on heuristic reward function. In: 2010 2nd International Conference on Advanced Computer Control. vol. 3, pp. 371–376 (March 2010)
18. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3, p. 5. Kobe, Japan (2009)
19. Riedmiller, M., Merke, A., Meier, D., Hoffmann, A., Sinner, A., Thate, O., Ehrmann, R.: Karlsruhe brainstormers - a reinforcement learning approach to robotic soccer. In: Stone, P., Balch, T., Kraetzschmar, G. (eds.) RoboCup 2000: Robot Soccer World Cup IV. pp. 367–372. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
20. Russell, S.J., Norvig, P.: Inteligencia Artificial: un enfoque moderno. No. 04; Q335, R8y 2004. (2004)
21. Schempf, H., Kraeuter, C., Blackwell, M.: Roboleg: a robotic soccer-ball kicking leg. pp. 1314 – 1318 vol.2 (06 1995)
22. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature 529(7587), 484 (2016)
23. Sung, C., Kagawa, T., Uno, Y.: Planning of kicking motion with via-point representation for humanoid robots (11 2011)
24. Theobald, M., Sozio, M., Suchanek, F., Nakashole, N.: Urdf: Efficient reasoning in uncertain rdf knowledge bases with soft and hard rules (2010)
25. Turk, M.: Computer vision in the interface. Communications of the ACM 47(1), 60–67 (2004)
26. Vahidi, M., Moosavian, S.: Dynamics of a 9-dof robotic leg for a football simulator. pp. 314–319 (10 2015)
27. Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A.S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J., et al.: Starcraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782 (2017)
28. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision 57(2), 137–154 (2004)
29. Watkins, C.J., Dayan, P.: Q-learning. Machine learning 8(3-4), 279–292 (1992)
30. Wilson, P.I., Fernandez, J.: Facial feature detection using haar classifiers. Journal of Computing Sciences in Colleges 21(4), 127–133 (2006)

ISSN 1870-4069

Analítica de datos en simulador de redes para sistemas de medición inteligente

Juan C. Olivares-Rojas[1], Enrique Reyes-Archundia[1], José A. Gutiérrez-Gnecchi[1], Johan W. González-Murueta[1], Adriana Téllez-Anguiano[1], Jaime Cerda-Jacobo[2]

[1] Tecnológico Nacional de México / Instituto Tecnológico de Morelia,

Morelia, Michoacán, México

[2] Universidad Michoacana de San Nicolás de Hidalgo, Facultad de Ingeniería Eléctrica,
Morelia, Michoacán, México
{jcolivares, ereyes, agnecchi, jwalter,
adrianat}@itmorelia.edu.mx, jcerda@umich.mx

Resumen. La Red Eléctrica Inteligente es uno de los pilares fundamentales de la Transformación Digital producida por la cuarta revolución industrial. Muchos de los problemas de la red eléctrica inteligente como los sistemas de medición inteligente no pueden modelarse a través de pruebas reales de laboratorio debido a la densidad de componentes utilizados; para ello, es necesario utilizar simuladores de redes eléctricas. La gran mayoría de los simuladores de redes eléctricas no cuentan con muchas capacidades de comunicación a redes de datos ni permiten realizar analítica de datos. Este trabajo presenta la adecuación del simulador de redes PacketTracer para permitir el manejo de analítica de datos en sistemas de medición inteligente.

Palabras clave: analítica de datos, simulador de redes, medidor inteligente, sistema de medición inteligente.

Data Analytic in Networks Simulator for Smart Metering Systems

Abstract. The Smart Grid is one of the fundamental pillars of the Digital Transformation produced by the fourth industrial revolution. Many of the problems of the smart grid such as smart metering systems cannot be modeled through real laboratory tests due to the density of components used; for this, it is necessary to use simulators of electrical networks. The vast majority of electrical network simulators do not have many communication capabilities to data networks or allow data analytics. This paper presents the suitability of the PacketTracer network simulator to allow the management of data analytics in smart metering systems.

Keywords: data analytics, network simulator, smart meter, smart metering system.

1. Introducción

La red eléctrica constituye un sistema sumamente complejo e interconectado. En los últimos años se han agregado Tecnologías de Información y Comunicaciones (TICs) a la red eléctrica dándole capacidades de comunicación y procesamiento de datos [1]. Esto a llevado a darle algún grado de inteligencia a la red eléctrica tradicional, motivo por el cual a pasado a denominarse Red Eléctrica Inteligente (REI).

Para los usuarios finales, una de las aplicaciones más visible de la REI, son los Sistemas de Medición Inteligente (SMI) caracterizados por el reporte periódico de los datos de consumo por parte de los medidores de energía eléctrica hacia los centros de datos de las empresas eléctricas [2]. Para ello, el medidor ha sido dotado de capacidades de comunicación de datos a través de redes cableadas e inalámbricas, motivo por el cual ha recibido el nombre de medidor inteligente.

Los medidores inteligentes, son dispositivos empotrados que cuentan con una pequeña base de datos donde se guardan las lecturas de las mediciones de energía eléctrica. Recientemente, con la integración de los recursos de energía distribuidos principalmente de fuentes renovables como la fotovoltaica y eólica, los medidores inteligentes han integrado capacidades para medir la producción de energía eléctrica [1].

La implementación mejor conocida de los SMI es la infraestructura de medición avanzada (AMI, por sus siglas en inglés). Los sistemas AMI permiten además del monitoreo de consumo/producción de energía en tiempo casi real de algunas otras funcionales como cortes y reconexiones de forma automática, señalización de precios dinámicos, así como la automatización de ahorro energético y respuesta a la demanda de los dispositivos conectados a la REI [2].

Todas las funcionalidades existentes en AMI y de las próximas en inventarse, están generando una gran cantidad de datos que necesitan ser procesados y analizados para la toma eficiente de decisiones energéticas [3].

Muchos de los problemas existentes en la REI no pueden ser modelados y probados de forma física debido a diversas cuestiones como los grandes voltajes, así como a no disponer de gran cantidad de componentes debido en muchos casos a los altos costos de los mismos. Por este motivo, los simuladores de redes son altamente utilizados.

La mayoría de los simuladores de redes existentes no cuentan con muchas capacidades de integración con las TICs, por lo que algunas aplicaciones como comunicación de datos, ciberseguridad, pero sobre todo analítica de datos, no están disponibles o lo están de forma parcial.

El presente artículo muestra una revisión de algunos simuladores de red eléctrica inteligente que existen, así como la adecuación de uno de ellos para modelar problemas de analítica de datos en SMI.

2. Simuladores de redes eléctricas

Existen diversos simuladores para el manejo de redes eléctricas en temas referentes a potencia en transmisión, distribución, generación y pocos se enfocan a la parte de consumo en particular en la parte de medición inteligente.

La gran mayoría de estos simuladores son comerciales y de altos costos; y aunque la gran mayoría ofrecen versiones de prueba o licencias para uso académico, son generalmente más ligeras y con menos funcionalidades.

Algunos de los simuladores eléctricos más utilizados son CYME [4] de la empresa Cooper Power Systems, PowerFactory [5] de la empresa Digsilent, Neplan Smarter Tools [6], DSATools [7] de PowerTech Labs, etap Powering Success [8], SKM PowerTools [9], PowerWorld Simulator [10] de PowerWorld Corporation, EasyPower [11], entre otras. Estos simuladores carecen de buenas capacidades de comunicación y analítica de datos.

Por otra parte, otros de los simuladores existentes están enfocados en el manejo de circuitos eléctricos y el cálculo de valores. Entre los más destacados sobresalen: PSCAD/EMTC [12] de Manitoba Hydro International, Orcad [13], CircuitLab [14], entre otros. Dichos simuladores tampoco cuentan con buenas capacidades de comunicación y de análisis de datos.

Existen algunos entornos como Matlab [15] de la empresa MathWorks que permiten realizar simulaciones (a través de simulink [16]), los cuales se enfocan a diversas funcionales de simulación pero que cuentan con bibliotecas y componentes para realizar simulaciones en el campo eléctrico y de comunicaciones (Communication Systems Toolbox Support Package for RTL-SDR Radio [17], IoT [18], Comunicaciones Inalámbricas [19], Procesamiento Digital de Señales [20]).

En el ámbito de investigación se ha utilizado Matlab y otros simuladores de procesos tradicionales para simular el comportamiento de redes eléctricas inteligentes [21] y [22], con muy buenos resultados. De manera particular la comunicación y procesamiento de los datos se realizan de buena forma, aunque el modelar SMI aun no es posible de manera directa.

Además, existen simuladores que incluyen componentes de hardware especializado como RTDS [23], el cual permite simular no solo componentes de la red eléctrica sino aplicaciones como electrónica de potencia, micro redes, pruebas a sistemas de protecciones, Power Hardware In the Loop (PHIL) y pruebas de ciberseguridad. Otro buen simulador dentro de esta misma línea es HyperSIM de OPAL-RT [24] el cual incluye diversas aplicaciones de tiempo real no sólo para la industria eléctrica, en las últimas versiones incluyen elementos de ciberseguridad. Este producto puede integrarse con otros softwares como Matlab a través de la aplicación RT-LAB [25]. Esta clase de simuladores no cuentan con grandes capacidades de analítica de datos.

Por otra parte, existen simuladores de redes de datos como Cisco PacketTracer [26] enfocados al mundo de las telecomunicaciones y que recientemente han integrado elementos de Internet de las Cosas (IoT) entre los que se incluyen la REI. Desafortunadamente los simuladores de redes de computadoras no cuentan con una gran integración de componentes de la red eléctrica tradicional.

Como se ha podido observar en la presente sección, no existen como tales soluciones únicas que integren el mundo de la energía eléctrica, telecomunicaciones, sistemas embebidos, ciberseguridad y analítica de datos, entre otros componentes, para los problemas actuales de la REI y en particular SMI. Por este motivo, se han realizado en los últimos años diversos estudios comparativos y adecuaciones de simuladores [27, 28, 29, 30, 31] para reflejar de mejor forma las condiciones actuales de la REI y SMI.

También cabe destacar que año con año las actualizaciones de los principales simuladores de REI están incluyendo cada vez más y más capacidades de comunicación, ciberseguridad y analítica de datos.

3. Simulador de Sistema de Medición Inteligente con Analítica de Datos

Del análisis realizado en la sección anterior se notó que los dos mejores softwares de simulación para REI y SMI son Matlab y PacketTracer.

Debido a que la licencia de Matlab es de alto costo (y más si se agregan Toolboxes adicionales), así como a la experiencia previa con PacketTracer y a que no ha sido muy explotado por la comunidad de la Industria Eléctrica, se decidió explorar está herramienta, la cual es gratuita y sólo es necesario registrarse.

Además, PacketTracer presenta otras ventajas entre las que destacan que existe una gran variedad de dispositivos de IoT; las funciones de programación en IoT están basadas en Arduino. Aunque arduino está hecho en C se puede migrar fácilmente a JavaScript y Python que son los dos lenguajes base de este simulador. Adicionalmente se puede programar de forma visual a través de Blockly. Blockly es un lenguaje por bloques desarrollado por Google e implementado hoy en día en muchos Entornos de Desarrollo Integrados (IDEs). El lenguaje utilizado para la programación de las simulaciones fue Python debido a su versatilidad en el manejo de análisis de datos.

Aunque el simulador cuenta con varios componentes de interconexion a dispositivos de Internet de las Cosas, muy pocos son para REI. Por este motivo, se procedió a realizar una topología básica para simular un hogar inteligente, dado que la gran mayoría de componentes son para redes de área en el hogar (HAN por sus siglas en inglés).

La topología realizada se muestra en la Figura 1, la cual es una propuesta de un hogar inteligente, en donde existen distintos electrodomésticos cada uno de ellos con una pequeña base de datos empotrada, una placa Raspberry Pi, y se conectan a la nube o una base de datos centralizada. Además se añadió un panel solar, así como un medidor inteligente y una batería de emergencia.

Cabe señalar que todos los dispositivos en la misma red local pueden intercomunicarse entre sí, o bien se puede controlar un dispositivo mediante un Smart Phone, Tableta o Computadora.

De manera predeterminada, todos los dispositivos de IoT del simulador disponen de una programación básica. Referente al consumo de energía eléctrica, el único dispositivo que tiene esta funcionalidad es la batería, mientras que de producción tanto el panel solar como el aerogenerador la poseen.

Debido a la limitante de consumo de energía se agregó una función de consumo de energía, para lo cual se tomaron valores de consumo de referencia. Por ejemplo, a la lámpara inteligente se le puso un valor de 60w. Esta función es genérica y puede colocarse a cada tipo de dispositivo, midiendo su consumo por hora o fracción de hora. Los valores de consumo de energía pueden enviarse de forma automática al medidor inteligente o bien consultarse a través de un pequeño servicio Web desarrollado dentro del simulador. Esta funcionalidad de reporte de datos fue extendida a la batería, panel solar y aerogenerador.

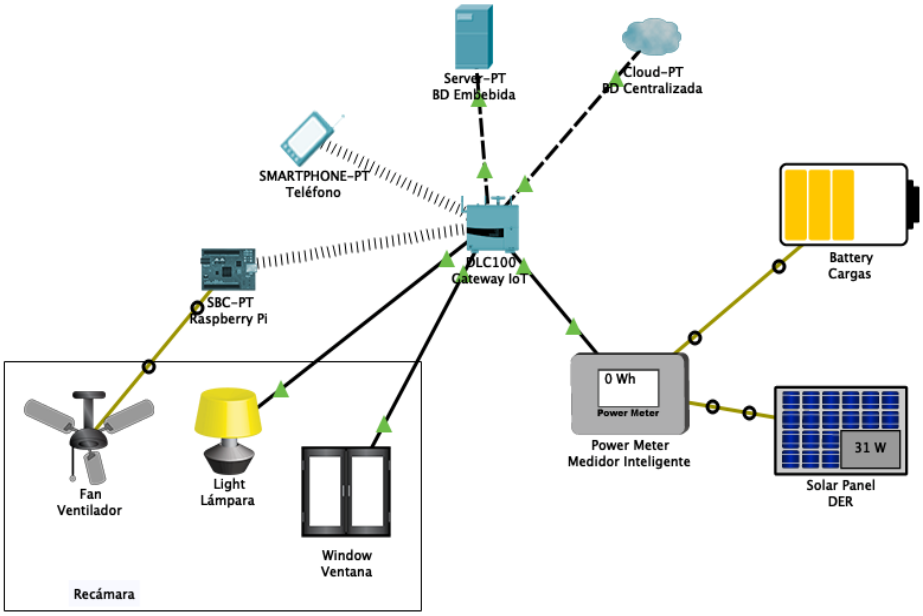


Fig. 1. Topología de simulación básica de un hogar inteligente.

Otra de las limitantes del simulador es el contar con pocas capacidades de almacenamiento de información, ya que, aunque no disponen de base de datos en forma, si se tiene la capacidad de almacenar información en archivos, motivo por el cual el medidor inteligente utiliza un archivo separado por comas (CSV) en donde se guarda la información proveniente de cada dispositivo con la siguiente estructura: *timestamp, id_dispositivo, tipo, valor*. Tal y como puede observarse en la Tabla 1.

Tabla 1. Ejemplo de Base de Datos del Medidor Inteligente

timestamp	Id_dispositivo	Tipo	Valor
20190301.00000015	0xAB990F	C	0.06
20190301.00000022	0xBB4312	P	1
20190301.00000037	0xBBCD23	C	0.14
20190301.00000049	0x1468BC	C	0.001
20190301.00000103	0x29FA1C	C	0.12

En la Tabla 1 se puede observar que el tipo C corresponde con el consumo de energía y P a la producción. Los valores están dados en Kw/h, mientras que los IDs de los dispositivos de la simulación son: 0xAB990F para lámpara, 0xBB4312 para panel fotovoltaico, 0xBBCD23 para la batería, 0x1468BC para la ventana inteligente y 0x29FA1C para el ventilador.

La funcionalidad de almacenamiento también puede integrarse a los dispositivos de IoT permitiendo tener una pequeña base de datos empotrada en cada uno de ellos.

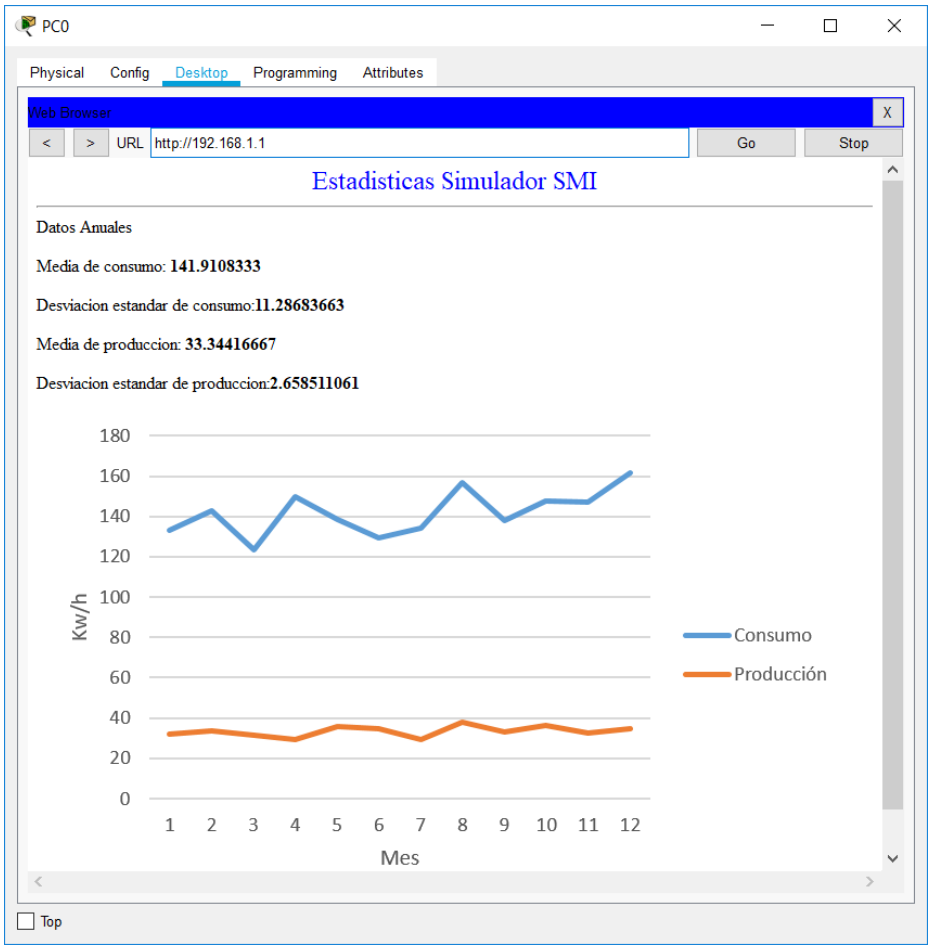


Fig. 2. Resultados de la simulación en forma anual.

Por otra parte, los dispositivos de producción de energía no toman en cuenta otras variables de la señal eléctrica importante en las simulaciones como son la frecuencia y el voltaje; motivo por el cual fue agregado esta variante al panel fotovoltaico y áerogenerador para tener estos datos.

Se agregó un dispositivo raspberry pi para simular un Concentrador de Datos (CD) dentro de la arquitectura AMI. Un CD permite almacenar las lecturas de diversos medidores inteligentes y reportarlos a los servidores de la empresa eléctrica.

La funcionalidad de monitoreo de energía implementada en el medidor inteligente también fue colocado en el CD y en el dispositivo Servidor.

Dentro del medidor inteligente se agregaron rutinas para el procesamiento estadísticos de los datos de medición implementando medidas de tendencia central. Esta funcionalidad también se migró a los CD y Servidor para brindar una plataforma de procesamiento distribuido edge-fog-cloud computing.

4. Resultados

En el presente trabajo, se realizó la simulación de la red presentada en la sección anterior durante un periodo de 30 días representando un año físico (el software permite adelantar los tiempos de la simulación).

Los datos de los dispositivos de consumo y producción de energía son reportados cada 15 minutos (el cual es el promedio de reporte en AMI) por lo que la información generada fue de un total de 175,200 registros para un tamaño aproximado de 8.75 MB.

Las funciones de analítica de datos fueron ejecutadas por hora, por día y por mes observando que los tiempos de consulta a mayor cantidad de datos son más lentos debido a que el acceso a los registros en el archivo es de manera secuencial.

La Figura 2 muestra los resultados finales de la simulación a un año con datos creados por los dispositivos de consumo/producción modificados.

Aunque el simulador trabaja con Python, muchas de las librerías populares para el manejo de datos como numpy, pandas, matplotlib, etc., no están disponibles en el simulador PacketTracert.

Se pueden reutilizar módulos desarrollados en otros proyectos, por lo que es posible utilizar la implementación de algoritmos de aprendizaje máquina sencillos como clasificadores y técnicas de agrupamiento.

5. Conclusiones y trabajo futuro

En este trabajo se demuestra que es posible realizar simulaciones de SMI de una manera integral considerando no solo elementos de energía eléctrica sino además considerando factores como la comunicación, ciber seguridad y minería de datos.

Aunque la simulación presenta buenos resultados aún falta realizar muchas tareas, muchas de las cuales se tienen que realizar desde cero, lo cual ha implicado invertir demasiado tiempo de desarrollo.

Además, se puede observar que los datos empiezan a crecer en una manera considerable particularmente en los CD y sobre todo en los Servidores.

Como trabajos futuros se encuentran: extender la simulación a múltiples HAN, medidores inteligentes y sobre todo a la implementación de algoritmos más robustos para la predicción de datos como series de tiempo. A su vez se pretende realizar ciberataques para medir el grado de confidencialidad, integridad y disponibilidad de los datos simulados.

Por otra parte, se tiene contemplado integrar funcionalidad de micropython y llevar el trabajo de simulación sobre medidores inteligentes realizados con placas como Raspberry Pi.

Agradecimientos. Este trabajo está parcialmente financiado por el Tecnológico Nacional de México dentro del proyecto con registro 6385.19-P.

Referencias

1. Morello, R., De Capua, C., Fulco, G., Mukhopadhyay, S.C.: A smart power meter to monitor energy flow in smart grids: The role of advanced sensing and IOT in the electric grid of the future. IEEE Sensors Journal (2017)
2. Alahakoon, D., Yu, X.: Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey. IEEE Transactions on Industrial Informatics (2016)
3. Wang, Y., Chen, Q., Kang, C., Xia, Q., Luo, M.: Sparse and Redundant Representation-Based Smart Meter Data Compression and Pattern Extraction. IEEE Transactions on Power Systems (2017)
4. Cooper Power Systems- Eaton: CYME Power Engineering Software. Disponible en: http://www.cyme.com/, último acceso: 2019/05/11
5. DigSILENT: PowerFactory. Disponible en: https://www.digsilent.de/en/powerfactory.html, último acceso: 2019/05/11
6. Neplan: Neplan Smarter Tools. Disponible en: https://www.neplan.ch/, último acceso: 2019/05/11
7. PowerTech Labs: DSATools. Disponible en: https://www.powertechlabs.com/dsatools-services/, último acceso: 2019/05/11
8. Electrical Power System Analysis & Operation Software: etap Powering Success. Disponible en: https://etap.com/18, último acceso: 2019/05/11
9. SKM System Analysis, Inc.: SKM PowerTools. Disponible en: http://skm.com/, último acceso: 2019/05/11
10. PowerWorld Corporation: PowerWorld Simulator. Disponible en: https://www.powerworld.com/products/simulator/overview, último acceso: 2019/05/11
11. EasyPower: EasyPower Power System Software. Disponible en: https://www.easypower.com/, último acceso: 2019/05/11
12. Manitoba Hydro International Ltd.: PSCAD/EMTC. Disponible en: https://hvdc.ca/pscad/, último acceso: 2019/05/11
13. Orcad: Orcad Professional PCB Design. Disponible en: https://www.orcad.com/, último acceso: 2019/05/11
14. CircuitLab: CircuitLab. Disponible en: https://www.circuitlab.com/, último acceso: 2019/05/11
15. MathWorks: Matlab. Disponible en: https://la.mathworks.com/products/matlab.html, último acceso: 2019/05/11
16. MathWorks: Simulink. Disponible en: https://la.mathworks.com/products/simulink.html, último acceso: 2019/05/11
17. MathWorks: RTL-SDR Support from Communications System Toolbox. Disponible en: https://www.mathworks.com/hardware-support/rtl-sdr.html, último acceso: 2019/05/ 11
18. MathWorks: MATLAB and Simulink for IoT Applications. Disponible en: https://la.mathworks.com/solutions/internet-of-things.html, último acceso: 2019/05/11
19. MathWorks: MATLAB para Comunicaciones Inlámbricas. Disponible en: https://la.mathworks.com/solutions/wireless-communications.html, último acceso: 2019/05/11
20. MathWorks: Digital Signal Processing (DSP). Disponible en: https://la.mathworks.com/solutions/dsp.html, último acceso: 2019/05/11
21. Shlebik, T., et al.: The Development of a Simulation-Based Smart Grid Communication Management System Using MATLAB. In: 2017 International Conference on Green Energy Conversion Systems (GECS). Hammamet, Tunisia. 23-25 de marzo (2017)
22. Pochacker, M., et al.: Simulating the Smart Grid. In: IEEE PowerTech 2013 Institute of Networked and Embedded Systems/ Lakeside Labs Alpen-Adria Universitat Klangenfurt, Austria (2013)

23. RTDS Technologies: Real Time Digital Power Systems. Disponible en: https://www.rtds.com/, último acceso: 2019/05/11
24. OPAL-RT Technologies: OPAL RT. Disponible en: https://www.opal-rt.com/, último acceso: 2019/05/11
25. OPAL-RT Technologies: RT-LAB. Disponible en: https://www.opal-rt.com/software-rt-lab/, último acceso: 2019/05/11
26. Cisco: Cisco Packet Tracer. Disponible en: https://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/Cisco_PacketTracer_DS.pdf, último acceso: 2019/05/11
27. Delamare, J., Bitachon, B., Peng, Z., Wang, Y., Haverkort, B.R., Jongerden, M.R.: Development of a Smart Grid Simulation Environment. Electronic Notes in Theoretical Computer Science, vol. 318, pp. 19–29 (2015)
28. Pang, C., Vyatkin, V., Deng, Y., Sorouri, M.: Virtual smart metering in automation and simulation of energy-efficient lighting system. In: 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, pp. 1–8 (2013)
29. Zhang, T., Nuttall, W.J.: An Agent Based Simulation of Smart Metering Technology Adoption. doi: https://doi.org/10.17863/CAM.5130 (2007)
30. Li, W., Zhang, X.: Simulation of the smart grid communications: Challenges, techniques, and future trends. Computers & Electrical Engineering 40(1), 270–288 (2014)
31. Sahu, A., Goulart, A., Butler-Purry, K.: Modeling AMI network for real-time simulation in NS-3. In: Principles, Systems and Applications of IP Telecommunications (IPTComm), Chicago, IL, pp. 1–8 (2016)

ISSN 1870-4069

Control basado en campos potenciales para un vehículo autónomo usando realimentación visual

Oscar González-Miranda, Santos M. Orozco-Soto, Juan M. Ibarra-Zannatha

CINVESTAV, Departamento de Control Automático, México
{ogonzalez,sorozco}@ctrl.cinvestav.mx, jibarra@cinvestav.mx

Resumen. En este trabajo se desarrolló un controlador para conducir de forma autónoma, al vehículo a escala 1:10 AutoNOMOS, desarrollado por la Freie Universität Berlin. El vehículo se condujo por una carretera blanca, dibujada sobre un fondo negro, sin obstáculos y a una velocidad de crucero constante de 0.6 $\frac{m}{s}$. Este controlador usa la información visual de la cámara y un enfoque similar al usado en el Control Automático Basado en Campos Potenciales; para regular el ángulo de dirección $u(t)$ del automóvil. Se probó su desempeño en el simulador Gazebo y se demostró que el vehículo podía mantenerse dentro de su carril durante todo el recorrido.

Palabras clave: vehículo autónomo, control de dirección, navegación reactiva, control basado en campos potenciales, control con realimentación visual.

Potential-Field Based Control for Autonomous Vehicle Using Visual Feedback

Abstract. This paper presents the design and implementation of a con-trol strategy for autonomous driving of the AutoNOMOS, a 1:10 scaledvehicle developed by the Freie Universität Berlin. The autonomous nav-igation problem is to drive the vehicle within a lane of a 2-way road,which is white painted over black ground, without obstacles but withcrossroads, and additionally, keeping a constant navigation speed of 0.6m/s. The presented controller uses visual feedback and a field potential-based approach in order to supply a steering angleu(t) for the auto-mobile. The performance of the controller was tested using the Gazebosimulator, and it was proved that the vehicle is capable to drive withinits corresponding lane during the whole road.

Keywords: autonomous vehicle, steering control·reactive navigation, potential field control, visual feedback control.

1. Introducción

En los últimos años las grandes empresas del sector automotriz han llevado a cabo una competencia para conseguir vehículos completamente autónomos y

aunque algunas ya han logrado introducir autos de este tipo en el mercado; aún existen problemas que exigen cierto nivel de robustez para conseguir una conducción más segura y confortable. Esta clase de problemas se han vuelto muy importantes en la investigación. Varias universidades de México están incursionando en esta clase de tecnologías gracias a las donaciones hechas por el Dr. Raúl Rojas de la Freie Universität Berlin [7]. Desde el 2016, una treintena de Instituciones de Educación Superior, organizadas por el Instituto Politécnico Nacional, han recibido prototipos de vehículos autónomos a escala 1:10. Este trabajo se enfoca en este vehículo en particular.

El problema de la condución autónoma, se suele separar en otros más específicos: Planeación de trayectorias[6,7,8], el cual consiste en elegir una trayectoria que sea óptima (bajo algún criterio) para el vehículo; seguimiento de trayectorias [4,13], el cual consiste en diseñar un controlador que le permita al automóvil mantenerse en la trayectoria deseada mientras avanza; mapeo del medio que rodea al auto a partir de la información sensorial [15]; realización de maniobras como la evasión de obstáculos o el revase de otros vehículos [12]; entre otros.

En particular, conducir un vehículo tipo Ackerman manteniéndolo en su carril se puede definir como el problema de calcular el ángulo de dirección del automóvil, para dar seguimiento a una trayectoria de referencia; mantieniendo el auto en el centro del camino y con una orientación apropiada [14]. Este problema se ha resuelto modelándolo de diferentes formas y estableciendo controladores, tanto lineales como no lineales[1,2,7,11,14]; sin embargo, existen fuerzas externas como la fricción estática, la fuerza debida a la interacción del frente con el aire o la fuerza del viento sobre el centro aerodinámico; que hacen dificil la identificación de los parámetros del modelo y por lo tanto las estrategias de control pierden robustez en ambientes externos [9].

En este trabajo se desarrolló una ley de control para conducir el AutoNOMOS por una carretera de dos carriles, con un par de cruces y sin obstáculos; a una velocidad constante de 0.6 $\frac{m}{s}$. La realimentación visual se consiguió usando la cámara RGBD a bordo del vehículo. A las imágenes obtenidas se les aplicó una corrección de perspectiva para calcular la distancia entre el centro del vehículo y la línea del carril derecho. La estrategia de control utilizada es una técnica basada en Campos Potenciales, diseñada para atraer al vehículo al centro del carril y mantener el vehículo en la orientación deseada.

El controlador se implementó en el sistema operativo para robots (ROS) y es ejecutado en tiempo real. Se probó su desempeño en el simulador Gazebo desarrollado por la Freie Universität Berlin. Este trabajo está organizado de la siguiente manera: la sección 2 presenta la descripción del hardware. El diseño del sistema de control se detalla en la Sección 3. La sección 4 describe las simulaciones y sus resultados. Finalmente, las conclusiones se exponen en la Sección 5.

2. Descripción del hardware

El vehículo autónomo usado en este trabajo es el AutoNOMOS, el cual es un automóvil tipo Ackerman a escala 1:10 que mide 40 *cm* de largo y 20 *cm* de ancho. Utiliza un motor sin escobillas de 24 Watts para mover las ruedas y una batería Li-Po de 14.8 V de 4 celdas. El ángulo de dirección es controlado por un servomotor de 5 V. A bordo se encuentra una computadora Odroid XU4 con el sistema operativo Ubuntu 16.04 y la distribución ROS Indigo instalada. El auto también está equipado con sensores como el sensor inercial MPU6050, sensores de efecto Hall embebidos en el motor sin escobillas, una cámara *fish-eye*, una cámara RGBD y un LIDAR infrarrojo. La figura 1 muestra al AutoNOMOS y algunos de su dispositivos a bordo.

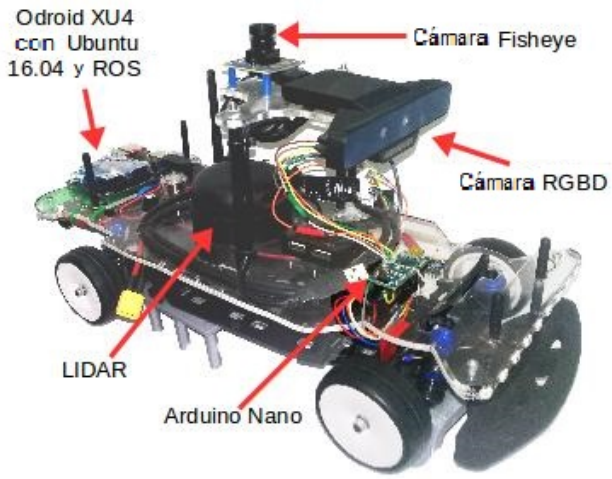


Fig. 1. Vehículo AutoNOMOS [3].

3. Desarrollo del sistema de Control

3.1. Definición del problema

El objetivo de control es poder conducir el auto a través de una carretera blanca dibujada sobre un fondo negro, con cruces, sin obstáculos, sin conocimiento *a priori* del mapa y sin salirse de su carril. Para llevar a cabo este objetivo se usa la cámara RGBD para reconocer la distancia x_1 (en centímetros), desde la línea del lado derecho del camino, hasta el centro de la parte inferior de la imagen denotado como $\{N\}$; tal y como se muestra en la figura 2. Esta imagen

presenta la vista desde arriba del camino; obtenida a través de la corrección de perspectiva de la imagen tomada por la cámara RGBD. Nótese que el punto $\{N\}$ representa la posición del vehículo y más adelante es usado como marco referencial de navegación. Por otro lado, x_2 también es la distancia entre el centro de la imagen hasta la línea de la derecha del camino; pero 15 cm por encima de $\{N\}$. Independientemente de la forma del camino si el vehículo está en el centro de su carril entonces $x_1 = x^d = 20cm$. Así pues, las variables de control en este sistema son: la velocidad de cruzero del vehículo v y el ángulo de dirección u del mismo. Por simplicidad se mantuvo constante $v = 0{,}6\frac{m}{s}$. Por lo tanto, de modo general podemos decir que:
Dado un conjunto arbitrario de distancias $X_1 \subset \mathbb{R}^+$ *conocido como carretera, se conduce al auto con un ángulo de dirección* u $|x_1 \in X_1 \rightarrow x^d$, $x^d = 20$ cm.

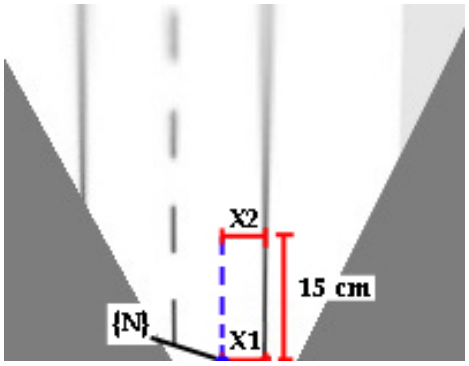


Fig. 2. Fragmento de la homografía de la carretera por la que circula el vehículo autónomo, cuando este se encuentra en el centro del carril derecho. Completa tiene un ancho de 200 cm y un largo de 300 cm. Cada carril mide 40 cm.

3.2. Realimentación visual

Para controlar ángulo de dirección del auto u, primero se obtuvo una imagen RGB capturada por la cámara RGBD (figura 3.a) y se recortó el fondo para que sólo se muestre la carretera. Después se transformó la imagen a escala de grises y se hizo la corrección de la perspectiva para tener una imagen del camino vista desde arriba (3.b). Este mapeo se obtiene usando la transformación:

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = H^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}. \tag{1}$$

De este modo cada pixel en la posición (u, v) de la imagen, se mapea a las coordenadas (x, y) en el suelo (medidas en centímetros); λ es un factor de escala

diferente para cada pixel y a H se conoce como la matriz de homografía de la cámara [5]. En este caso se calculó que:

$$H = \begin{pmatrix} -7{,}90 \times 10^{-2} & -4{,}19 \times 10^{-1} & 1{,}28 \times 10^{2} \\ 1{,}00 \times 10^{-2} & -1{,}46 & 4{,}00 \times 10^{2} \\ 0 & -4{,}15 \times 10^{-3} & 1 \end{pmatrix}. \tag{2}$$

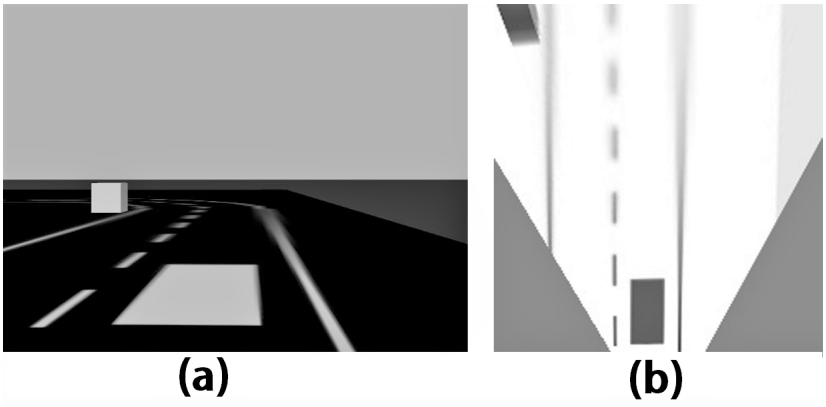


Fig. 3. Corrección de la perspectiva de la cámara usando la matriz de homografía inversa. Los colores se invierten de (a) a (b) para permitir una mejor visualización. Las esquinas del rectángulo que está dibujado en el suelo se usaron para calcular la matriz de homografía H de la cámara.

Suponiendo que la imagen tiene una resolución de $p \times q$ y dada una matriz $J \in (\mathbb{N} \cup \{0\})^{3\times n}$ que representa la posición de los n pixeles de la imagen; podemos definir M como:

$$M = \frac{1}{\lambda} H^{-1} J, \qquad M \in \mathbb{R}^{3\times n}. \tag{3}$$

Así pues, aplicando un algoritmo de búsqueda dentro de una región de interés $M_{ROI} \subset M$, la distancia x_1 puede ser calculada usando la expresión:

$$x_1 = M\left(\frac{p}{2}, q\right) - M_{ROI}(j, q), \tag{4}$$

donde $M\left(\frac{p}{2}, q\right)$ es el pixel que se encuentra en el centro de la parte inferior de la imagen (marco referencial de navegación) y $M_{ROI}(j, q)$ es el $j-$ésimo pixel cuya intensidad sobrepasa cierto umbral δ. Por otro lado, x_2 puede ser calculado como:

$$x_2 = M\left(\frac{p}{2}, q\right) - M_{ROI}(j, q - \epsilon), \tag{5}$$

donde $\epsilon > 0$ es un parámetro que depende de la velocidad del vehículo y los FPS de la cámara.

3.3. Control basado en Campos Potenciales

Para determinar el valor de una variable auxiliar de control θ se usa un enfoque similar al Control Automático de Sistemas Usando Campos Potenciales [10]; es decir, se le asignan potenciales atractivos y repulsivos a ciertos puntos sobre la homografía, se calcula la "fuerzaresultante sobre el vehículo y se determina la dirección que debe tomar. En la figura 4, a los puntos $\bar{r_1}$ y $\bar{r_2}$ (ambos medidos desde $\{N\}$) se les asignan los potenciales $\frac{K_0}{r_1}\hat{r}_1$ y $-\frac{K_0}{r_2}\hat{r}_2$, respectivamente. K_0 es un parámetro arbitrario que se simplifica más adelante. Los vectores $\hat{r}_1 = \cos(\theta_1)\hat{\imath} + \sin(\theta_1)\hat{\jmath}$ y $\hat{r}_2 = \cos(\theta_2)\hat{\imath} + \sin(\theta_2)\hat{\jmath}$ son unitarios con componentes vertical $\hat{\jmath}$ y horizontal $\hat{\imath}$; mientras que:

$$r_1 = \sqrt{15^5 + x_2^2} \qquad y \qquad r_2 = \sqrt{15^5 + (x_2 - 20)^2}, \tag{6}$$

$$\theta_1 = \arctan\left(\frac{15}{x_2}\right) \qquad y \qquad \theta_2 = \arctan\left(\frac{15}{x_2 - 20}\right), \tag{7}$$

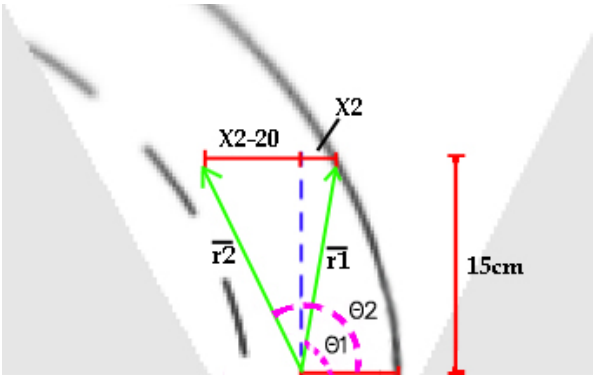


Fig. 4. Obtención de la variable θ mediante Campos Potenciales.

La dirección vertical $\hat{\jmath}$ siempre es la dirección de avance del vehículo; asi que, para que el vehículo se mueva hacia adelante se supone que sobre él actúa una fuerza $\overrightarrow{F_1} = K_1\hat{\jmath}$. Por lo tanto, la fuerza resultante en el vehículo esta dada por la ecuación:

$$\overrightarrow{F} = K_1\hat{\jmath} + \frac{K_0}{r_1^2}\hat{r}_1 - \frac{K_0}{r_2^2}\hat{r}_2. \tag{8}$$

De aquí se puede deducir que la dirección del vehículo, medida desde la vertical, esta dada por:

$$\theta = 90 - \arctan\left(\frac{Kr_1^2r_2^2 + r_2^2\sin(\theta_1) + r_1^2\sin(\theta_2)}{r_2^2\cos(\theta_1) + r_1^2\cos(\theta_2)}\right) \qquad con: \qquad K = \frac{K_1}{K_0}. \tag{9}$$

El signo de θ determinará si el giro es hacia la izquierda o hacia la derecha; mientras que K es un parámetro que se ajusta experimentalmente.

3.4. Ley de control

El diagrama de bloques de la figura 5 representa al sistema que se desea controlar. El tópico de ROS que conduce el auto está programado de tal modo que, si el ángulo de dirección u cumple que $0 \leq u < 90$, el vehículo girará a la izquierda; por otro lado si $90 \leq u < 180$ el giro es hacia la derecha. Por lo tanto, se usa como ley de control:

$$u = 90 + K_x e_x + K_\theta \theta, \tag{10}$$

donde $e_x = x^d - x_1$ es el error de posición del vehículo cuando este se aleja del centro de su carril, θ es una variable relacionada con la forma del camino y K_x y K_θ son ganancias que se ajustan de acuerdo al desempeño del controlador.

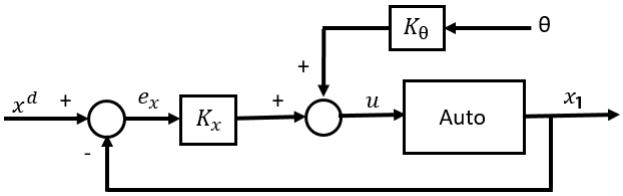


Fig. 5. Diagrama de bloques del sistema.

4. Simulaciones y resultados

Para probar el desempeño del controlador de la ecuación (10) con θ como en la ecuación (9), se le asignaron los valores a los parámetros: $K_x = 3{,}0$, $K_\theta = 0{,}25$ y $K = 0{,}01$; y se mantuvo una velocidad de crucero constante de 0.6 $\frac{m}{s}$. La ley de control se programó como un nodo en ROS y las pruebas se hicieron en el simulador Gazebo con una carretera como la mostrada en la figura 6. La carretera tiene un ancho de 80 cm por lo que cada carril mide 40 cm. Todas las curvas de la pista tienen un radio externo de 1.8 m y un radio interno de 1.0 m.

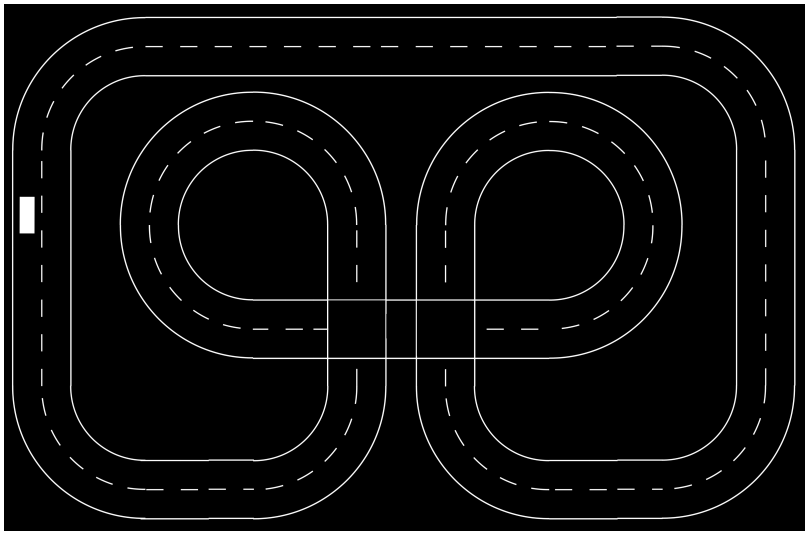


Fig. 6. Pista por la que se condujo el AutoNOMOS en el simulador de Gazebo.

La gráfica de la figura 7 muestra el cambio en el tiempo del error de posición del vehículo; mientras que la gráfica de la figura 8 muestra el ángulo de dirección del mismo. Como se puede observar, el error de posición aumenta cuando el vehículo toma una vuelta y disminuye hasta casi cero mientras el camino es recto. Dado que $-20 \leq e_x \leq 20$ en todo momento, entonces el vehículo se mantuvo dentro de su carril durante todo el recorrido.

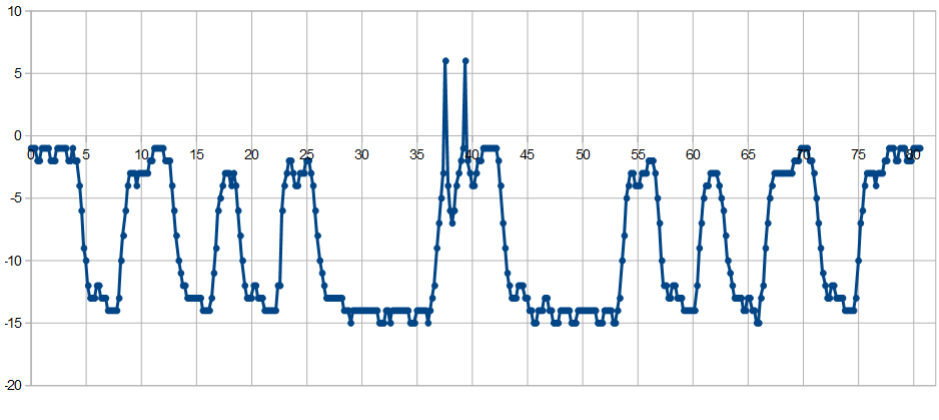


Fig. 7. Evolución del error e_x en el tiempo. Si se cumple que $-20 \leq e_x \leq 20$ el automóvil se mantendrá dentro de su carril. Esta cantidad está medida en cm.

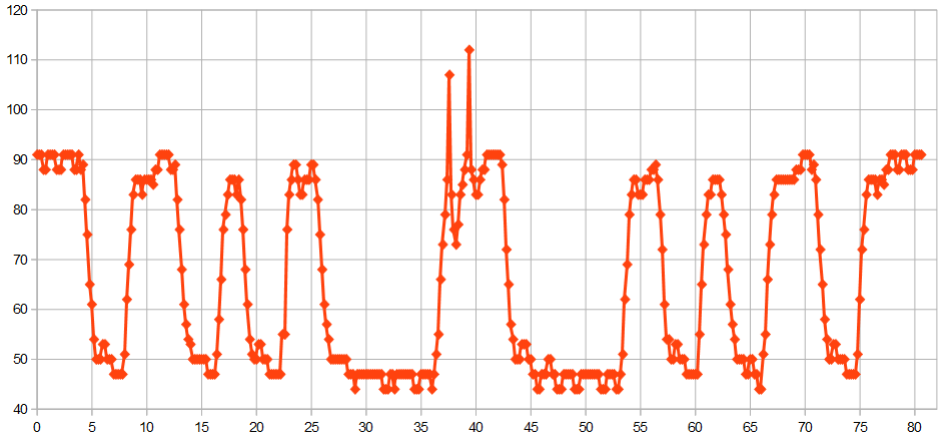


Fig. 8. Cambio del ángulo de dirección $u(t)$ en el tiempo. Este ángulo está medido en grados.

5. Conclusiones

A partir de las gráficas de las figuras 7 y 8 podemos concluir que el término que más peso tiene en la ecuación 10 es $K_x e_x$; pues es el término que modula el ángulo $u(t)$ cuando el vehículo se sale de su carril. Por otro lado, se observó que el término $K_\theta \theta$ mejora el desempeño del vehículo.

Así pues el controlador diseñado fue lo suficientemente robusto como para mantener el vehículo dentro de su carril en todo momento y por lo tanto, la metodología empleada ofrece una solución sencilla y fácil de implementar, al problema de la conducción autónoma de un vehículo tipo Ackerman; sin embargo, dicha robustez está limitada a la condición inicial de que $x_1(0)|e_x(0) \in [0, 28]$; de otro modo, se tomará mucho tiempo alcanzar el objetivo de control.

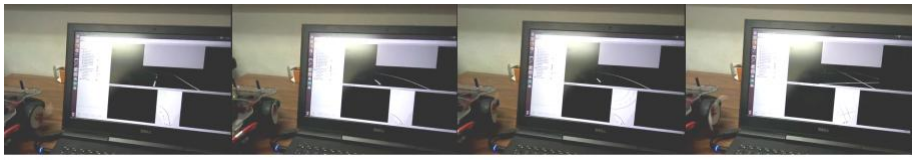


Fig. 9. Capturas del vehículo real conduciéndose por la pista virtual del simulador. Video disponible en: `https://www.youtube.com/watch?v=hiniVFp3mio`

Aunque aún no se tienen resultados experimentales, se han hecho pruebas con el vehículo AutoNOMOS como la que se muestra en la figura 9. Allí se puede observar cómo el auto cambia su ángulo de dirección adecuadamente; de acuerdo

a las imágenes de la cámara del simulador. En trabajos futuros se conducirá el vehículo real por una pista blanca con fondo negro.

Referencias

1. Ackermann, J., Guldner, J., Sienel, W., Steinhauser, R., Utkin, V.I.: Linear and nonlinear controller design for robust automatic steering. IEEE Transactions on Control Systems Technology 3(1), 132–143 (1995)
2. Ackermann, J., Sienel, W.: Robust control for automatic steering. In: 1990 American Control Conference. pp. 795–800. IEEE (1990)
3. Campbell, S.F.: Steering control of an autonomous ground vehicle with application to the DARPA urban challenge. Ph.D. thesis, Massachusetts Institute of Technology (2007)
4. Corke, P.: Robotics, vision and control: fundamental algorithms In MATLAB® second, completely revised, vol. 118. Springer (2017)
5. Freie Universität Berlin: AutoNOMOS Model. `https://github.com/AutoModelCar/AutoModelCarWiki/wiki` (2016), [Online; accessed 15-April-2019]
6. Hernandez Beleno, R.D., Bernardes Vítor, G., Vaqueiro Ferreira, J., Siqueira Meirelles, P.: Planeación y seguimiento de trayectorias de un vehículo terrestre con base en el control de dirección en un ambiente real. Scientia et technica 19(4) (2014)
7. Ibarra Zannatha1, J.M., Vera Bustamante, P., Orozco Soto, S.M., Cureno Ramírez, A., de la Rosa Picasso1, R., Ramírez Morales, L.E.: Generacion de trayectorias para vehículos autónomos (2018)
8. Katrakazas, C., Quddus, M., Chen, W.H., Deka, L.: Real time motion planning methods for autonomous on-road driving: State of the art and future research directions. Transportation Research Part C: Emerging Technologies 60, 416–442 (2015)
9. Kodagoda, K., Wijesoma, W.S., Teoh, E.K.: Fuzzy speed and steering control of an agv. IEEE Transactions on control systems technology 10(1), 112–120 (2002)
10. Lewis, F.L., Stingu, E., TTXYVVVRFFRX, Y.: Potential fields in cooperative motion control and formations (2013)
11. Marino, R., Scalzi, S., Netto, M.: Nested pid steering control for lane keeping in autonomous vehicles. Control Engineering Practice 19(12), 1459–1467 (2011)
12. Rastelli, J.P., De Pedro, T., Santos, M.: Controladores borrosos para la dirección de vehículos autónomos en maniobras dentro de entornos urbanos. In: XVI Congreso Espanol sobre Tecnologías y Lógica Fuzzy (2012)
13. Snider, J.M., et al.: Automatic steering methods for autonomous automobile path tracking. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU RITR 09 08 (2009)
14. Sotelo, M.A.: Lateral control strategy for autonomous steering of ackerman-like vehicles. Robotics and Autonomous Systems 45(3-4), 223–233 (2003)
15. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., et al.: Stanley: The robot that won the darpa grand challenge. Journal of field Robotics 23(9), 661–692 (2006)

ISSN 1870-4069

Estimación automática del gasto energético de la actividad física de niños en videojuegos de ejercicio con el sensor Kinect

Benjamín Gutiérrez-Serafín[1], Humberto Pérez-Espinosa[1,2], Ismael Edrein Espinosa-Curiel[1], Pedro N. Figueroa-García[3], Edgar E. Pozas-Bogarín[1], Juan Martínez-Miranda[1,2]

[1] CICESE-UT3, Tepic, Nayarit, México
[2] Catedrático CONACyT
[3] Universidad Autónoma de Nayarit, Tepic, Nayarit, México
begutierrezse@outlook.com, hperez@cicese.mx, ecuriel@cicese.mx

Resumen. En este artículo se presenta EGEKN, el cual es un método para la estimación automática del gasto energético dirigido a niños de entre 7 y 10 años de edad durante sesiones de juego con videojuegos activos. Este método toma como entrada los datos de seguimiento de las juntas del modelo esquelético capturados con un sensor Kinect para calcular el trabajo mecánico del cuerpo del jugador. Adicionalmente, este método considera información contextual como la edad, el sexo, el peso y el índice de masa corporal de los jugadores, el tipo de actividad realizada, el porcentaje de grasa corporal y la frecuencia cardíaca media. A partir de la representación de esta información en vectores de características se entrenó un modelo de regresión para la estimación del gasto energético. Para desarrollar y validar este método se capturaron datos de 46 niños de entre 7 y 10 años de edad. Los participantes jugaron 12 videojuegos activos con diferentes tipos de movimiento e intensidades de actividad física. El modelo entrenado obtuvo una correlación de 0.85, lo cual es comparable con la precisión de métodos similares encontrados en la literatura, pero dirigidos a adultos.

Palabras clave: aprendizaje máquina, estimación de calorías, caracterización de movimientos.

Automatic Estimation of the Energy Expenditure of Children Physical Activity in Exergames Using the Kinect Sensor

Abstract. In this article, we present a method for automatic estimation of children energy expenditure during playing sessions with exergames,

called EGEKN. This method calculates the mechanical work of the player's body using the tracking data of the joints of the skeletal model captured with a Kinect sensor. It also uses contextual information such as the age, sex, weight, and body mass index of the players, the type of activity performed, the percentage of body fat and the average heart rate. Using feature vectors to represent this information, we trained a regression model to estimate the energy expenditure. We collected data from 46 children aged between 7 and 10 years old to develop and validate this method. The participants played with12 exergames with different types of movement and physical activity intensities. The trained model obtained a correlation of 0.85, which is comparable with the accuracy of similar methods found in the literature but aimed at adults.

Keywords: machine learning, energy expenditure estimation, body movements characterization.

1. Introducción

La obesidad infantil aumentó dramáticamente durante las últimas décadas a nivel mundial, alcanzando niveles epidémicos [4, 23]. La obesidad infantil está asociada a la inactividad física y un estilo de vida sedentario de los niños. Debido a lo anterior, una de las principales estrategias para reducir la obesidad infantil es aumentar la actividad física y reducir las conductas sedentarias de los niños [17]. Gracias al desarrollo de nuevos y novedosos dispositivos de interacción Humano-Computadora (ej. Controles Wii, Kinect, y cámaras de profundidad), los videojuegos activos (exergames) surgieron como una estrategia emergente para promover la actividad física en niños. Este tipo de videojuegos proveen métodos emocionantes, innovadores y atractivos para atraer la atención, educar y promover cambios en las intenciones y los comportamientos humanos [1, 14].

Estudios recientes revelan que la mayoría de los videojuegos activos producen un nivel medio de intensidad de ejercicio; y muy pocos producen un nivel intenso de ejercicio, el cual es requerido para obtener beneficios cardiovasculares considerables [15]. Para tener un mayor impacto en la salud, los videojuegos activos podrían ajustar automáticamente la intensidad de la actividad física de acuerdo al perfil y desempeño del jugador; sin embargo, uno de los retos más importantes para lograr tal fin es medir efectivamente el gasto energético (GE) de los jugadores. El uso de los métodos actuales para medir el GE en los videojuegos activos se ve afectado por su precisión, complejidad de uso, y sus costos. Por ejemplo, la precisión de los métodos que utilizan acelerómetros, unos de los mas ampliamente utilizados, depende del número de dispositivos utilizados, requiriendo en ocasiones hasta 5 dispositivos [12]. Por su parte, los sistemas de análisis metabólico de gases son uno de los mas precisos; sin embargo, son caros y requieren un esfuerzo significativo para ponerlo a punto. Una limitación de la mayoría de los métodos de GE es que son invasivos debido a que requieren que los jugadores porten sensores de medición durante toda la actividad física. Como una alternativa a estos métodos, recientemente se han desarrollado métodos no

invasivos de estimación del GE que utilizan el sensor Microsoft Kinect para rastrear los movimientos corporales sin la necesidad de portar ningún instrumento. Sin embargo, estos métodos han sido desarrollados para adultos y, hasta donde tenemos conocimiento, no existe un método específico para niños en edad escolar. Debido a lo anterior, en este trabajo se presenta EGEKN, el cual es un método no invasivo para la estimación automática del gasto energético dirigido a niños de entre 7 y 10 años de edad durante sesiones de juego con videojuegos activos. EGEKN aprovecha las capacidades del sensor Microsoft Kinect para rastrear de forma precisa los movimientos de las articulaciones esqueléticas del jugador, y con base en esa información estimar el gasto energético. EGEKN puede ser integrado a videojuegos activos para mostrar el gasto energético de los jugadores, o para ajustar la intensidad de actividad física en tiempo real y de forma personalizada.

2. Trabajo previo

A continuación se presentan investigaciones que abordan la estimación del gasto energético empleando el sensor Kinect como medio de captura de movimiento. La herramienta denominada ViziCal desarrollada en [9] es un enfoque basado en regresión que mapea datos cinemáticos recopilados por el sensor Kinect y reporta el gasto de energía en unidades metabólicas de reposo (MET). Este sistema toma en consideración la información de la ubicación y la aceleración de las articulaciones esqueléticas para predecir la actividad física en videojuegos activos. Los datos se recopilaron de un grupo de 9 participantes masculinos entre 18 y 25 años que se ejercitaban regularmente. Para desarrollar el algoritmo de estimación se usaron, máquinas de soporte vectorial para tres conjuntos diferentes de características: aceleración cinemática, posición de articulaciones y la combinación de estos dos. Se empleo un enfoque de validación cruzada dejando uno fuera. ViziCal resultó ser más preciso que las técnicas convencionales basadas en acelerómetros. Otra contribución es la desarrollada en [13] donde se propone un método que calcula el consumo de energía en tiempo real al evaluar la suma de la energía requerida para superar la energía potencial gravitatoria de cada parte del cuerpo del modelo esquelético. En este experimento se utilizó un conjunto de vídeos de baile para obtener y modelar un cuerpo humano, el cual consta de 10 partes formadas por 15 nodos de articulaciones. Al monitorear el cambio de posición de cada parte del cuerpo, se puede conocer la energía empleada para cada uno de ellos y la sumatoria de sus energías da origen al consumo total de energía del cuerpo completo. Una limitación es que el consumo de calorías se estimó en función de un avatar y no directamente del cuerpo de los participantes.

Por otro lado, el estudio [21] propone un método similar para el consumo de calorías en rutinas de aerobics. En esta investigación se usó el sensor Kinect para monitorizar 10 juntas del modelo esquelético, las cuales fueron consideradas como un acelerómetro de 3 ejes. Además, se contempló el ritmo cardíaco para complementar la evaluación de múltiples modelos de regresión. Este método obtuvo mejores resultados en comparación a los obtenidos con un solo acelerómetro y

sin requerir numerosos entrenamientos para crear un modelo estable. En [16] se propuso un modelo más robusto desde un enfoque de aprendizaje estadístico multivariado. Este modelo es capaz de estimar el gasto energético con base en el trabajo mecánico interno y externo realizado en 4 ejercicios individuales diferentes. Con el sensor Kinect, se localizó en un sistema local de coordenadas, 20 juntas del modelo esquelético para rastrear en cada momento del ejercicio los movimientos realizados por los participantes. Posteriormente se calcula el trabajo mecánico realizado en el cuerpo tomando en cuenta los cambios de su energía mecánica y finalmente se estima el gasto energético.

En general, los métodos de estimación del gasto energético que usan el sensor Kinect para medir el movimiento han obtenido buenos resultados y se han adaptado a diferentes actividades físicas. Sin embargo, hasta donde tenemos conocimiento no se ha publicado ningún trabajo que reporte una técnica de estimación de gasto energético en sesiones de juego activo específica para niños como sujeto de prueba. Por tal motivo, la presente investigación se centra en desarrollar una técnica específica para niños que estime el gasto energético durante actividad física con videojuegos activos.

3. Recolección de datos

Para desarrollar el método de estimación se recolectaron datos de 46 estudiantes (23 niños y 23 niñas) de entre 7 y 10 años, inscritos en una escuela primaria pública. El estudio se desarrolló durante tres meses. Previo al inicio de la recolección de los datos, se revisó el protocolo del experimento con un experto en educación física infantil con el objetivo de afinar los ejercicios y su intensidad para evitar poner en riesgo la salud y bienestar de los participantes. Posteriormente, se obtuvo la autorización del director de la escuela y los maestros para realizar el estudio. Finalmente, se obtuvo el consentimiento informado firmado por los padres y el consentimiento verbal de parte de los niños. Las características de los participantes se muestran en la Tabla 1.

Tabla 1. Características de los participantes.

Características	***Promedio***	***Desviación estandar***	***Máximo***	***Mínimo***
Masa (kg)	41.3	9.27	67.4	25.2
Altura (cm)	138	8.51	153	121
Edad	8.9	0.86	10	7
IMC	21.3	2.82	29.5	16.6
BFP	29.1	6.05	46.3	16.2

Los niños participaron en binas en sesiones de juego que duraron en promedio 90 minutos, de los cuales solo 26 minutos fueron de actividad física. Los niños podían terminar su participación en el momento que ellos quisieras. Los niños

pasaban en binas con el objetivo de alternar sus tiempos de ejercicio y descanso además de que reducía el tiempo requerido para dar las instrucciones a los jugadores. A los participantes se les recomendó asistir con ropa cómoda para realizar adecuadamente las actividades. Cada sesión de juego se dividió en las siguientes etapas:

1. Se les colocaba a los participantes un sensor de pulso cardíaco Polar OH1 conectado a la aplicación Polar Beat, con el objetivo de capturar la frecuencia cardíaca y el gasto energético. Además, se calibraba el sensor Microsoft Kinect para capturar adecuadamente el movimiento de sus cuerpos.
2. A los participantes se les explicaba cómo jugar, el objetivo de los videojuegos, y como acumular el mayor puntaje en el tiempo establecido haciendo correctamente lo que los videojuegos requerían.
3. Los participantes realizaban una sesión de calentamiento dinámico de entre 8 a 10 minutos que incluía movimientos articulares: flexiones, extensiones, abducciones y aducciones de hombros caderas, rodillas y tobillos. Para definir las actividades de calentamiento se siguieron las recomendaciones de [5].
4. Los participantes jugaban 3 sesiones de juego de 5 minutos cada una con un videojuego activo de acción. Había cinco posibles videojuegos activos de acción: FoodRateMaster ©, FatCombat ©, DishMaster ©, SuperMarketRace ©, y FoodHitTravel ©. Todos los videojuegos son desarrollos propios y tienen en tramite el registro de propiedad intelectual. Cada videojuego requería un conjunto de movimientos corporales para poder jugar (ej. desplazamientos laterales, saltos, agacharse, tocar objetos con las manos, etc.). Se empezaba con una intensidad baja y en las dos sesiones subsecuentes se le aumentaba la intensidad sin superar las 197 pulsaciones por minuto (PPM) el cual es el limite máximo para niños y adolescentes [7]. Las intensidades son: baja (140 PPM), media (160 PPM) y alta (>160 <180 PPM). Después de jugar cada sesión de juego, el participante tenía 5 minutos de reposo absoluto y el otro miembro de la bina iniciaba a jugar lo mismo que el jugador que estaba descansando.
5. Los participantes jugaban 3 videojuegos activos de reto que requerían realizar ejercicios de alta intensidad (salto de mariposa, skipping y taloneo). La duración de cada videojuego era de 40 segundos y después tenían 40 segundos de recuperación pasiva. Durante el reposo el otro miembro de la bina empezaba a jugar.
6. Los participantes jugaban 6 videojuegos activos de reto que requerían realizar ejercicios de fuerza muscular (bíceps, desplante frontal, desplante lateral, elevación de hombros frontal, elevación de hombros lateral y sentadillas) durante 90 segundos con el mismo tiempo de recuperación pasiva. Durante el reposo el otro miembro de la bina empezaba a jugar.
7. Los participantes realizaban estiramiento para volver a la calma.

Se recopilaron datos del movimiento del cuerpo, frecuencia cardíaca y gasto energético. Se utilizó la aplicación Polar Beat para el registro del gasto energético. Para el caso de los videojuegos de 5 minutos se tomó cada minuto, y en los

videojuegos de ejercicio de alta intensidad y fuerza muscular, una sola vez al finalizar el juego. Durante el estudio se obtuvieron un total de 521 capturas de movimiento.

4. Metodología

Con ayuda del dispositivo Kinect se capturaron las ubicaciones de las juntas del esqueleto de los usuarios en un sistema de referencia local de coordenadas, a fin de localizarlos en cualquier tiempo de la sesión de juego y con ellos determinar variables esenciales para el análisis del movimiento del cuerpo humano. Una vez obtenidos estos parámetros básicos, se implementó el enfoque propuesto por Nathan [16] para el cálculo del trabajo externo e interno, y subsecuentemente, estimar el gasto energético empleando aprendizaje máquina para modelar estas características junto con otras propuestas por los autores del presente trabajo.

4.1. Descripción general del método propuesto

El objetivo principal de este trabajo de investigación es proponer un enfoque más amplio al reportado en [16] al implementar un algoritmo que estime el gasto energético durante actividades físicas realizadas en sesiones de juego serio orientados a disminuir el sobrepeso en niños y adaptar el modelo a esta población. Por este motivo, se propuso evaluar diferentes algoritmos de aprendizaje con 3 conjuntos de características diferentes para examinar y comparar su eficacia cuando esta técnica se aplica a actividades físicas más complejas que las estudiadas por Nathan (sentadilla, salto de mariposa, movimiento de brazos y salto de pie). Las funciones de cada una de las fases principales de nuestro método se describen en las siguientes subsecciones.

4.2. Captura de coordenadas de las juntas

Se utilizó el sensor Microsoft Kinect V2 y su SDK para la obtener las coordenadas de 20 juntas. El modelo del dispositivo fue lanzado en el año 2014 y discontinuado en el 2017. No obstante, Microsoft tiene planes de un nuevo lanzamiento con un módulo de cámara más sofisticado y tecnología de inteligencia artificial basada en la nube de Microsoft [2]. 19 juntas representan una articulación del cuerpo y una representa el centro del modelo esquelético del cuerpo. Está información es representada en un vector XYZ de valores decimales en metros con respecto a la posición del Kinect. La estructura con la que trabajan las juntas sigue un modelo de árbol jerárquico donde se parte de una junta raíz y a partir de ella se van relacionando las demás juntas. Se capturaron fotogramas de las 20 juntas cada 30 milisegundos y se almacenó en un archivo CSV junto con información del participante.

4.3. Cálculo de propiedades de los segmentos del cuerpo

De manera general, la obtención del centro de masa del cuerpo completo, así como otras propiedades inerciales es una tarea indispensable cuando se necesita realizar un análisis cuantitativo del movimiento [20]. Dado que el cuerpo humano es un sistema formado por segmentos conectados entre sí, la distribución de masa del cuerpo varia continuamente dependiendo a la posición de sus segmentos, lo que ocasiona que el centro de gravedad cambie de lugar constantemente. Esta dependencia continua de la postura corporal provoca que el hallazgo del centro de masa del cuerpo completo sea más complejo en comparación con el de simples cuerpos rígidos. Por consiguiente, diversos métodos para encontrar su ubicación se han ideado. La opción más conveniente dadas las características del proyecto, es la determinación del centro de gravedad del cuerpo completo por medio de la suma ponderada de todos los centros de masas individuales de los segmentos. Para esta tarea, se recurrió a los parámetros de segmentos del cuerpo proporcionados por De Leva [3] para primeramente calcular los centros de masa de los segmentos individuales con las siguientes ecuaciones para cada uno de sus componentes cartesianos para todos los fotogramas obtenidos en cada sesión de juego:

$$\begin{aligned} x_{seg} &= x_p + R_p(x_d - x_p), \\ y_{seg} &= y_p + R_p(y_d - y_p), \\ z_{seg} &= z_p + R_p(z_d - z_p). \end{aligned} \tag{1}$$

Es decir, la posición del centro de masa del segmento (x_{seg}, y_{seg}, z_{seg}) se puede determinar a partir de las coordenadas del punto proximal (x_p, y_p, z_p) y distal (x_d, y_d, z_p) de la parte del cuerpo a analizar y el porcentaje de su longitud R_p. Posteriormente, el centro de masa del cuerpo completo (x_{cc}, y_{cc}, z_{cc}) es obtenido empleando la fórmula 2 para ubicar sus coordenadas en el espacio, donde P_i y (x_i, y_i, z_i), son respectivamente, la proporción de masa y cada uno de los componentes rectangulares del centro de masa del segmento i.

$$x_{cc} = \frac{\sum(P_i x_i)}{\sum P_i}, y_{cc} = \frac{\sum(P_i y_i)}{\sum P_i}, z_{cc} = \frac{\sum(P_i z_i)}{\sum P_i}. \tag{2}$$

4.4. Cálculo de la energía y trabajo mecánico

Para llevar a cabo la caracterización del modelo de estimación del gasto energético, se consideró el trabajo mecánico interno y externo realizado en el sistema por variaciones en su energía mecánica. Como se describe en [22], es uno de los métodos más destacados puesto que provee información importante no solo de la mecánica del cuerpo, sino también de su movimiento. Tal como se mencionó con anterioridad, se adoptó el método propuesto por Nathan D. [16] el cual se divide en dos partes. La primera fase se encarga de analizar las interacciones de fuerzas externas capaces de causar cambios de movimiento en el centro de masa de todo el cuerpo relativo al entorno, lo que origina alteraciones en la energía mecánica del sistema, produciendo lo que se conoce como trabajo externo. La

energía en cualquier instante asociada con el trabajo externo del centro de masa del cuerpo completo puede ser obtenida con la siguiente ecuación:

$$E_{COM}(t) = MgH(t) + \frac{1}{2}MV_{cg}(t)^2 \tag{3}$$

donde M es la masa total de todo el cuerpo, g la gravedad, H la altura del centro de masa y V_{cg} la velocidad lineal del centro de masa relativo al entorno. Una vez que calculada la energía en cualquier tiempo t, se prosigue con el cálculo del trabajo externo que se define de la siguiente manera:

$$W_{ext}(t) = E_{COM}(t+1) - E_{COM}(t) \tag{4}$$

En cuanto a la segunda fase, se evalúa los cambios de energía ocasionados por fuerzas internas del sistema y la transferencia de energía interna que existe entre un segmento y segmentos adyacentes dentro de la misma extremidad. Originando de esta manera el trabajo provocado por la diferencia de energía mecánica relativa al centro de masa del cuerpo completo, el cual se denomina trabajo interno. La energía en cualquier momento t asociada con el trabajo interno del segmento i se expresa como:

$$E_{int,i}(t) = \frac{1}{2}m_i V_{r,i}(t)^2 + \frac{1}{2}m_i K_i^2 w_i(t)^2 \tag{5}$$

donde m_i es la masa del segmento, V_{ri} es la velocidad relativa del centro de masa del segmento i con respecto al centro de masa del cuerpo completo, K_i es el radio de giro del centro de masa y w_i es la velocidad angular del segmento i. De manera similar que el paso anterior, el trabajo interno del segmento i está dado por la ecuación:

$$W_{int,t}(t) = E_{int,i}(t+1) - E_{int,i}(t) \tag{6}$$

Teniendo en cuenta que el trabajo realizado por los músculos varía dependiendo del tipo de contracción, también se tomó en consideración diferenciar entre las dos clases que se pueden presentar. Knuttgen y Kraemer [10] exponen que, en caso de que la fuerza producida sea lo suficientemente fuerte para superar la resistencia, el musculo se acorta a medida que se contrae de manera concéntrica (trabajo positivo). En cambio, cuando la contracción es excéntrica (trabajo negativo), la fuerza generada no es suficiente para sobrepasar la carga, lo cual causa que el músculo se alargue mientras se contrae de manera excéntrica. A la luz de este relevante hecho, la suma de todo trabajo positivo efectuado durante la actividad física fue almacenada en las variables de trabajo interno positivo y trabajo externo positivo. De igual forma, la sumatoria de todo trabajo negativo realizado durante la sesión de juego se administró en las variables de trabajo interno negativo y trabajo externo negativo. Cabe señalar que, con respecto a las variables de trabajo interno, se decidió separarlo en extremidades inferiores y superiores. Finalmente, después de caracterizar el movimiento de esta manera se obtuvieron 6 características relevantes (ver Tabla 2).

4.5. Obtención de características contextuales

Tal como se presentó en [8] variables como el ritmo cardíaco, edad, genero, entre otros, contribuyeron de manera significativa para que su modelo de regresión reportara un coeficiente de correlación altamente positivo al momento de estimar el gasto energético durante sus experimentaciones. Por lo tanto, en este trabajo se propone la inclusión de los datos proporcionados por el perfil del jugador que informan sobre las características de los participantes para mejorar la efectividad del modelo. Además de la edad y el género, la funcionalidad especial incluida en nuestra plataforma mantiene registros del tamaño del cuerpo y algunas composiciones corporales de los usuarios que se ingresan al sistema cuando los niños se registran al comienzo de su primera sesión. En adición a estas características personales, también se decidió agregar información sobre la sesión jugada, es decir, descripción acerca del nivel de intensidad, tipo de actividad física o videojuego seleccionado. Por último, el dispositivo empleado para el monitoreo de las calorías quemadas por sesión, Polar OH1, incluye la utilidad de poder llevar un seguimiento de las pulsaciones por minuto presentes durante la sesión de juego. Por consiguiente, se optó por incorporar el ritmo cardíaco como una característica adicional a nuestro modelo de estimación de gasto energético.

Representación del vector características. Esta sección proporciona una perspectiva general de la caracterización empleada para el sistema de estimación de gasto energético. Una vez capturados y calculados los atributos planteados en las secciones anteriores se prosiguió con la formación del vector característica para su posterior análisis.

Las primeras 6 características adquiridas durante el procesamiento de los datos suministrados por el dispositivo Kinect están vinculadas con el trabajo mecánico medido en kilojulios e interpretado como variable cuantitativa el cual es realizado durante la actividad física. El trabajo mecánico se representa como la sumatoria a través del tiempo tanto del trabajo positivo y negativo empleado para mover el centro de masa del cuerpo completo por el entorno, como del trabajo positivo y negativo provocados por cambios en la energía mecánica relativa al centro de masa del cuerpo completo para mover sus elementos superiores e inferiores.

En cuanto a los datos provistos por el sistema, se encuentran asociados en mayor medida a información recabada para el perfil de jugador, el cual consta de la edad del participante, genero, peso, altura, porcentaje de grasa e índice de masa corporal, donde las primeras dos variables fueron contempladas como valores discretos y las cuatro restantes como valores continuos. Asimismo, la plataforma de videojuegos brindó información sobre el tipo de actividad y la sesión seleccionada. Para representar esta información se aplicó la técnica de codificación *one-hot*, que consiste en remplazar la variable categórica por una nueva variable binaria por cada clase contenida originalmente. Finalmente, el dispositivo de recopilación de gasto energético provee adicionalmente la frecuencia cardíaca media como características de tipo continuo. Este último atributo

y las calorías quemadas por sesión se encuentran a nuestra disposición una vez que se haya concluido la actividad física. La Tabla 2 lista las características analizadas en este trabajo.

Tabla 2. Lista de atributos utilizados para el análisis.

Trabajo mecánico
Trabajo externo positivo
Trabajo externo negativo
Trabajo interno positivo superior
Trabajo interno negativo superior
Trabajo interno positivo inferior
Trabajo interno negativo inferior
Datos del participante
Edad
Genero
Peso
Índice de masa corporal
Información de actividad
Tipo de actividad
Sesión seleccionada
Porcentaje de grasa corporal
Dispositivo de medición
Frecuencia cardíaca media

4.6. Estimación del gasto energético

En este estudio se utilizaron diferentes conjuntos de características y se aplicaron varias técnicas de aprendizaje máquina con el objetivo de identificar la combinación más precisa para la estimación del gasto energético. Asimismo, se consideró evaluar cada una de los tipos de sesión tanto de manera independiente como en conjunto. Dicho de otra forma, se agruparon las capturas por tipo de sesión y se realizaron experimentaciones para sesiones de juego, videojuegos de reto y la mezcla de estos tipos. Para evaluar el método propuesto, se decidió entrenar los modelos utilizando solo las 6 características obtenidas por el proceso del cálculo del trabajo mecánico en el primer experimento. En la siguiente etapa de experimentación, se incorporaron las características proporcionadas por el perfil de jugador, pero excluyendo el ritmo cardíaco promedio puesto que esta variable es obtenible una vez que se termina la sesión. Se procuró observar el desempeño del modelo descartando esta variable para visualizar su rendimiento en caso de querer implementar el algoritmo en tiempo real. De igual manera, se examinó el comportamiento de los modelos de regresión al agregar el ritmo cardíaco al vector característica. Como última tarea, se crearon modelos de entrenamiento y prueba independientes del sujeto con el fin de no incluir muestras de un mismo niño en los conjuntos de entrenamiento y de

prueba y así evitar cualquier tipo de dependencia de los modelos entrenados con propiedades de los individuos. En cuanto a los algoritmos de regresión empleados, se propuso explorar el rendimiento de random forest [11], proceso gaussiano [18], perceptrón multicapa [6] y máquinas de soporte vectorial [19] para todas las experimentaciones de este estudio y mostrar sus rendimiento en las tablas 3, 4, 5 y 6 empleando como métricas de desempeño el coeficiente de correlación de Pearson, error medio absoluto y el error cuadrático medio. Además, se validó la estabilidad de estos algoritmos aplicando dos métodos de validación. 1) Validación cruzada de 10 iteraciones: En esta técnica, un 90 % de las muestras del conjunto de datos es entrenado para crear un modelo que es probado con el 10 % restante de los datos. Este proceso es repetido 10 veces usando diferentes conjuntos de entrenamiento/prueba para cada ocasión. 2) Validación 80/20: Para este esquema de validación, se crea un modelo de entrenamiento con el 80 % de las muestras del conjunto total de datos. El 20 % restante es utilizado para probar el modelo.

5. Resultados

Como puede observarse en la Tabla 3, las mejores métricas de regresión tanto en validación cruzada de 10 pliegues como en 80 % entrenamiento / 20 % prueba fueron obtenidos cuando los conjuntos de entrenamiento y prueba contenían muestras de los dos tipos de videojuegos (acción y reto) que se probaron. No obstante, cuando se analizaron por separado los dos tipos de juegos se redujeron de manera significativa los coeficientes de correlación y aumentaron las métricas del error, especialmente en las sesiones de juego de acción.

De acuerdo con las suposiciones establecidas, el comportamiento del modelo mejoró en la segunda experimentación, puesto que las características incluidas para esta etapa lograron que se obtuvieran resultados satisfactorios, no solo para el conjunto mezclado, sino también para el grupo de entrenamiento formado solo por videojuegos de reto. No se observó mejora significativa para el modelo constituido por sesiones de juego de acción, ver Tabla 4. Por otro lado, cuando se realizó experimentación tomando en consideración el ritmo cardíaco, se obtuvieron resultados más precisos que en los dos experimentos anteriores. Vale la pena señalar que, en esta ocasión, el modelo conformado por sesiones de juego de acción presentó un aumentó en los coeficientes de correlación, especialmente cuando se implementó el algoritmo perceptrón multicapa. Los resultados obtenidos de estos dos experimentos se encuentran localizados en la Tabla 5.

Para la última fase de experimentación, se limitó a estudiar solo las características utilizadas en la segunda etapa para visualizar el rendimiento que se originaría en un entorno real de uso. Los resultados obtenidos en este último experimento pueden visualizarse en la Tabla 6. Como se puede apreciar, la eficacia de los modelos se redujo si se comparan con los 2 experimentos previos, especialmente el de sesiones de juego, ya que no se fue capaz de obtener valores favorables para ninguno de los algoritmos de regresión empleados. Sin embargo,

Tabla 3. Resultados de la primera experimentación utilizando solo las 6 características del trabajo mecánico.

Todos								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	**0.78**	0.66	0.73	0.75	**0.79**	0.73	0.72	0.77
MAE	**6.63**	11.44	9.35	7.25	**0.58**	11.91	7.77	6.17
RMSE	**12.18**	15.21	15.17	13.21	**11.12**	14.63	12.70	11.57

Videojuego de reto								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.55	0.41	0.54	0.43	0.46	0.29	0.40	0.35
MAE	2.10	2.53	2.20	2.41	2.03	2.3	2.39	2.16
RMSE	2.89	3.32	2.98	3.13	3.02	3.27	3.29	3.18

Sesiones de juego								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	-0.29	0.25	0.11	0.40	-0.16	0.01	-0.09	0.08
MAE	17.12	9.24	15.18	6.23	13.90	11.44	14.49	10.39
RMSE	25.39	12.28	21.65	9.18	19.49	16.18	19.65	16.80

al utilizar random forest y máquina de soporte vectorial como algoritmos de estimación para actividades de tipo videojuego de reto originaron un coeficiente de correlación superior al 0.70, los cuales son comparables a resultados obtenidos en experimentación orientada a adultos [16]. Así mismo, estos valores son más realistas, ya que se obtienen sin ninguna información previa de los jugadores.

6. Conclusiones y trabajo futuro

En este trabajo se presentó un método denominado EGEKN para la estimación de consumo de energía con base en el movimiento detectado por un sensor Kinect. Se usaron tres tipos de características para entrenar un modelo de regresión. A partir de la experimentación se observó que usando solo las características de movimiento los resultados no son muy buenos, sobre todo cuando se entrenan por separado modelos para videojuegos de reto y de acción. Los resultados mejoran cuando se incluye información sobre características propias del individuo, en particular mejoró en la evaluación de los videojuegos de reto. Cuando se usó la frecuencia cardíaca se obtuvieron mejores resultados. Usando MLP para el proceso de regresión y evaluando con 10FCV se obtuvo una correlación de 0.99 para todos los ejercicios, 0.95 para los videojuegos de reto y 0.92 para videojuegos de acción. Esto nos lleva a la conclusión, en primer lugar, de que las características contextuales, tanto información del usuario como otros sensores, aportan información valiosa para la estimación del consumo

Tabla 4. Resultados de la segunda experimentación utilizando solo características de trabajo mecánico, datos del usuario y actividad física realizada.

Todos								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.87	0.90	**0.92**	0.89	**0.90**	0.90	0.89	0.89
MAE	4.45	4.79	**5.13**	4.50	**3.60**	4.27	4.54	3.62
RMSE	8.84	8.25	**8.07**	9.18	**7.80**	7.97	8.60	8.34

Videojuego de reto								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	**0.78**	0.69	0.72	0.68	**0.78**	0.70	0.65	0.70
MAE	**1.42**	1.58	1.69	1.43	**1.37**	1.48	1.80	1.34
RMSE	**2.18**	2.47	2.43	2.51	**2.16**	2.41	3.01	2.44

Sesiones de juego								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	-0.40	0.11	0.09	0.11	0.38	0.29	0.25	0.07
MAE	12.15	10.73	23.07	10.41	9.99	11.97	16.60	11.33
RMSE	19.43	12.82	29.18	12.19	14.79	15.63	23.71	17.11

de energía. En segundo lugar, nos lleva a la conclusión de que es benéfico incluir en el entrenamiento una mayor variedad de movimientos y de que es más difícil estimar la energía consumida durante la ejecución de sesiones de videojuegos de acción que durante la ejecución de videojuegos de reto. Otra conclusión importante es que cada individuo tiene características propias de movimiento asociadas a su consumo energético, ya que cuando se evaluaron los modelos asegurando que no hubiera muestras de un mismo individuo en los conjuntos de entrenamiento y prueba, los resultados empeoraron, mostrando que los modelos entrenados inicialmente no eran independientes del usuario. Como trabajo futuro será interesante avanzar en la integración del método EGEKN en los videojuegos con el objetivo de mostrar en tiempo real el gasto energético a los jugadores. Además, un aspecto a mejorar será la precisión de los modelos independientes del jugador. Para lograr una mejoría significativa consideramos necesario incrementar el tamaño de la base de datos así como probar otras características de movimiento. Finalmente, se pretende aplicar el método de caracterización de movimientos a otras áreas de aplicación como reconocimiento de actividades.

Agradecimientos. Este trabajo de investigación se ha llevado a cabo en el contexto del programa "Cátedras CONACyT "financiado por el Consejo Nacional de Ciencia y Tecnología (CONACyT).

Tabla 5. Resultados de la tercera experimentación utilizando todas las características propuestas

Todos								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.92	0.92	**0.99**	0.90	0.93	0.92	**0.99**	0.91
MAE	3.81	4.57	**1.52**	3.71	3.08	4.09	**1.42**	2.91
RMSE	7.89	7.40	**2.65**	8.77	6.84	6.83	**2.81**	7.90

Videojuego de reto								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.88	0.83	**0.93**	0.82	0.85	0.84	**0.95**	0.83
MAE	1.24	1.26	**0.92**	0.97	1.18	1.10	**0.68**	0.80
RMSE	1.85	1.92	**1.47**	1.96	1.89	1.82	**1.03**	1.89

Sesiones de juego								
Resumen	80/20				10FCV			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.24	0.53	**0.65**	0.50	0.68	0.68	**0.92**	0.66
MAE	10.00	10.15	**6.14**	7.62	7.37	8.66	**4.22**	7.79
RMSE	15.24	12.38	**9.40**	10.89	11.93	11.70	**6.30**	12.34

Tabla 6. Valores de correlación y errores concluyentes obtenidos al realizar experimentación con modelos entrenamiento y de prueba independientes.

80/20								
Resumen	Solo Videojuego de reto				Solo juegos			
	RF	GP	MLP	SVM	RF	GP	MLP	SVM
Correlación	0.73	0.48	0.60	**0.85**	0.10	0.28	0.35	0.23
MAE	1.58	2.65	5.55	**0.96**	6.04	6.64	10.63	5.82
RMSE	2.42	2.92	6.14	**1.38**	7.99	8.67	16.97	7.22

Referencias

1. Baranowski, T., Buday, R., Thompson, D., Lyons, E.J., Lu, A.S., Baranowski, J.: Developing Games for Health Behavior Change: Getting Started. Games For Health Journal 2(4), 183–190 (Aug 2013)
2. Cranz, A.: Microsoft kinect refuses to die (May 2018), https://gizmodo.com/microsoft-kinect-refuses-to-die-1825847023
3. De Leva, P.: Adjustments to zatsiorsky-seluyanov's segment inertia parameters. Journal of biomechanics 29(9), 1223–1230 (1996)
4. Dehghan, M., Akhtar-Danesh, N., Merchant, A.T.: Childhood obesity, prevalence and prevention. Nutrition Journal 4(1), 24 (Sep 2005)
5. Faigenbaum, A., Jr, J.E.M.: Guidelines for implementing a dynamic warm-up for physical education. Journal of Physical Education, Recreation & Dance 78(3), 25–28 (2007)

6. Gardner, M.W., Dorling, S.: Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment 32(14-15), 2627–2636 (1998)
7. Gelbart, M., Ziv-Baran, T., Williams, C.A., Yarom, Y., Dubnov-Raz, G.: Prediction of Maximal Heart Rate in Children and Adolescents. Clinical Journal of Sport Medicine: Official Journal of the Canadian Academy of Sport Medicine 27(2), 139–144 (2017)
8. Keytel, L., Goedecke, J., Noakes, T., Hiiloskorpi, H., Laukkanen, R., van der Merwe, L., Lambert, E.: Prediction of energy expenditure from heart rate monitoring during submaximal exercise. Journal of sports sciences 23(3), 289–297 (2005)
9. Kim, M., Angermann, J., Bebis, G., Folmer, E.: Vizical: Accurate energy expenditure prediction for playing exergames. In: Proceedings of the 26th annual ACM symposium on User interface software and technology. pp. 397–404. ACM (2013)
10. Knuttgen, H.G., Kraemer, W.J.: Terminology and measurement in exercise performance. The Journal of Strength & Conditioning Research 1(1), 1–10 (1987)
11. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R news 2(3), 18–22 (2002)
12. Liu, S., Gao, R.X., Freedson, P.S.: Computational methods for estimating energy expenditure in human physical activities. Medicine and Science in Sports and Exercise 44(11), 2138–2146 (Nov 2012)
13. Liu, Z., Tang, S., Qin, H., Bu, S.: Evaluating user's energy consumption using kinect based skeleton tracking. In: Proceedings of the 20th ACM international conference on Multimedia. pp. 1373–1374. ACM (2012)
14. Lu, A.S., Kharrazi, H., Gharghabi, F., Thompson, D.: A Systematic Review of Health Video Games on Childhood Obesity Prevention and Intervention. Games for health journal 2(3), 10.1089/g4h.2013.0025 (Jun 2013)
15. Mills, A., Rosenberg, M., Stratton, G., Carter, H.H., Spence, A.L., Pugh, C.J.A., Green, D.J., Naylor, L.H.: The effect of exergaming on vascular function in children. The Journal of Pediatrics 163(3), 806–810 (Sep 2013)
16. Nathan, D., Huynh, D.Q., Rubenson, J., Rosenberg, M.: Estimating physical activity energy expenditure with the kinect sensor in an exergaming environment. PloS one 10(5), e0127113 (2015)
17. Nishtar, S., Gluckman, P., Armstrong, T.: Ending childhood obesity: a time for action. The Lancet 387(10021), 825 – 827 (2016)
18. Rasmussen, C.E.: Gaussian processes in machine learning. In: Summer School on Machine Learning. pp. 63–71. Springer (2003)
19. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing 14(3), 199–222 (2004)
20. Soto, V., Calvo, M.G.: Parámetros inerciales para el modelado biomecánico del cuerpo humano. European Journal of Human Movement (2), 169–189 (1996)
21. Tsou, P.F., Wu, C.C.: Estimation of calories consumption for aerobics using kinect based skeleton tracking. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics. pp. 1221–1226. IEEE (2015)
22. Willems, P., Cavagna, G., Heglund, N.: External, internal and total work in human locomotion. Journal of Experimental Biology 198(2), 379–393 (1995)
23. Williams, E.P., Mesidor, M., Winters, K., Dubbert, P.M., Wyatt, S.B.: Overweight and obesity: Prevalence, consequences, and causes of a growing public health problem. Current Obesity Reports 4(3), 363–370 (Sep 2015)

ISSN 1870-4069

Algoritmo de evolución diferencial con reparador cromosómico aplicado a un problema de secuenciación de vehículos

Elvi Malintzin Sánchez Márquez, Héctor José Puga Soberanes, Luis Ernesto Mancilla Espinoza, Juan Martín Carpio Valadez, Manuel Ornelas Rodríguez, Javier Iván Manzanares Cuadro

Tecnológico Nacional de México / Instituto Tecnológico de León, Departamento de Estudios de Posgrado e Investigación, León, Guanajuato, México
martinezbucita@gmail.com,pugahector@yahoo.com, lmancilla01@hotmail.com,jmcarpio61@hotmail.com, mornelas67@yahoo.com.mx,d2241427@itleon.edu.mx

Resumen. En el artículo se propone un algoritmo de Evolución Diferencial con Reparador Cromosómico (EDRC) aplicado a la secuenciación de vehículos, que consiste en encontrar una secuencia de producción de diferentes modelos de automóviles en una línea de ensamblaje. Este es un problema de satisfacción de restricciones multiobjetivo NP-Duro [1], en el que se busca violar la menor cantidad de restricciones. Para generar la población inicial se utiliza un operador de mutación basado en el cambio. Además, se propone un reparador cromosómico que toma en cuenta las características del problema y asegura la generación de individuos factibles. Las soluciones del EDRC fueron comparadas con los resultados reportados por el algoritmo de recocido simulado usado por la Renault [2] y el equipo que aplicó búsqueda tabú y búsqueda codiciosa greedy [3], mostrando competencia (39% de los casos), mejora (22% de los casos), no logrando competir en el 39% de los casos. En la etapa que dará continuidad al proyecto se analizarán operadores de mutación especializados para mejorar el desempeño de la propuesta.

Palabras clave: problema de secuenciación de vehículos, evolución diferencial, reparador cromosómico

Differential Evolution Algorithm with Chromosome Repair Applied to a Problem of Vehicle Sequencing

Abstract. The article proposes a differential evolution algorithm with chromosome repair (EDRC), applied to sequencing of vehicles, this problem aims to find a production sequence of different car models in an assembly line. This is a NP-Hard multiobjective restrictions satisfaction problem [1], in which it is expected to violate the least amount of restrictions. A change-based mutation operator is used to generate the initial population. It also proposes a chromosomal repairman that takes into account the characteristics of the problem and ensures the generation of feasible individuals. EDRC solutions were compared with the

results reported by the simulated annealing algorithm used by Renault [2] and the team that applied tabu search and greedy search [3], viewing competition (22% of cases), improvement (21% of cases), failing to compete in 37% of cases. The next phase of project will analyze specialized mutation operators to improve the performance of the proposal.

Keywords: vehicle sequencing problem, differential evolution, chromosomal repairman.

1. Introducción

El problema de secuenciación de vehículos fue descrito por Parello y Kabat en 1986 como la programación de vehículos a lo largo de una línea de ensamble, con el fin de instalar distintas opciones por cada vehículo, lo que se conoce como línea de ensamblaje mixto [4]. Este problema se presentó como un problema de satisfacción de restricciones (CSP) por primera vez utilizando una programación de restricción que tiene que ver con el aspecto declarativo de la lógica y la eficiencia con la técnica de manipulación de restricciones denominada CHIP [4].

En el estado del arte hay una amplia cantidad de investigaciones que abordan el problema, algunas referencias de estas investigaciones son [5,6].

En 2005, el problema de secuenciación fue utilizado como punto de referencia para el Desafío Roadef, donde se establecieron restricciones duras y blandas, además de introducir restricciones a los lotes de pintura para minimizar el consumo de solventes en dicho taller. Para resolver el problema, se aplicaron métodos exactos como programación entera y método ad-hoc , además de métodos heurísticos greedy [3], búsqueda local [2] , algoritmos genéticos [7] y búsqueda tabú iterativa [6] por mencionar algunos. El desafío ofreció una visión integral respecto a los métodos propuestos, y el algoritmo ganador fue implementado en Renault y fábricas piloto europeas.

Como resultado de este concurso se publicaron un total de 6 artículos, entre los cuales se encuentra "Búsqueda tabú para el problema de secuenciación de vehículos", el cual hemos tomado como base para nuestra investigación [3].

El problema abordado en este artículo es un problema multiobjetivo con restricciones de suavizado. La función objetivo contiene tres componentes F_1, F_2 , F_3 y es evaluada mediante un enfoque lexicográfico escalonado, respetando $F_1 > F_2 > F_3$; el orden de prioridad de los objetivos viene dado por las instancias utilizadas en el concurso.

En este trabajo se tomaron las instancias del conjunto A que formaban parte de un grupo integrado por el conjunto X y B, también se utilizaron como referente los resultados del concurso Roadef, que son vigentes en la comparación de trabajos recientes como son [3,8].

Los objetivos son minimizar las sobrecargas o violaciones de restricciones de proporción del taller de línea de ensamblaje (suavizado) y minimizar los cambios de color del taller de pintura.

Las restricciones de proporción N_i/P_i están relacionadas con la *i-ésima* estación de trabajo y representan las capacidades de cada estación. La restricción indica que hay

un máximo de N_i vehículos que requieren la opción i, que pueden programarse en una secuencia de vehículos P_i en la estación de trabajo, de lo contrario ocurre una sobrecarga. En este modelo, una solución consiste en una secuencia de los últimos automóviles del día de producción D-1 (anterior, que ya fueron programados) y todos los automóviles del día D (actual).

2. Descripción del problema

El problema consiste en que, dado un conjunto de vehículos se desea encontrar la mejor secuencia *S* de éstos, que satisfaga las restricciones establecidas en cada estación de trabajo de una línea de ensamblaje. En este sentido, la secuencia S se puede identificar como un arreglo lineal de c' vehículos de la producción del día anterior (D-1), los cuales ya fueron programados y no se deben modificar, más c carros del día de producción actual (D) véase ecuación (1) y (2). Se establecen dos tipos de restricciones de proporción: las restricciones H consideradas como de alta prioridad ya que debido a las características del automóvil se requiere una gran carga de trabajo en la línea de ensamblaje y las restricciones L como de baja prioridad ya que las características del automóvil causan pequeños inconvenientes en la línea de ensamblaje. Ambas consideradas como restricciones suaves, ya que no se puede garantizar la satisfacción completa al programar un día de producción véase ecuación (4) y (5).

Se toma en cuenta una restricción dura, que es el tamaño de lote, la cual nos dice el número máximo de carros consecutivos del mismo color tolerable véase ecuación (3).

En este contexto se definen:

$$S_p = l, \tag{1}$$

representa que el vehículo *l* está en la posición *p* de la solución *S*.

$$(l) \in \{D-1, D\}, \tag{2}$$

representa día de producción del automóvil *l*.

$$Color\ (l) \in \{1,\ldots,50\}, \tag{3}$$

$$H_i \in \{0,1\} \tag{4}$$

$$L_i \in \{0,1\}, \tag{5}$$

donde $H_i = 1$ si el automóvil requiere la opción y $H_i = 0$ cuando no, mismo caso para L_i.

Los objetivos que se buscan cumplir son: (1) suavizado de restricciones y (2) disminuir el número de cambios de color en el taller de pintura.

La función fitness se establece de acuerdo a los objetivos, siendo evaluada con un enfoque lexicográfico escalonado que se define en la ecuación (6):

$$F(S) = F_1(S) + F_2(S) + F_3(S), \tag{6}$$

donde F_1, F_2 , $F_3 \in \{FH, FL, FC\}$ de acuerdo a la prioridad de optimización de objetivos de cada instancia. FH es el número de sobrecargas de restricciones de alta prioridad en la solución S, FL es el número de sobrecargas de restricciones de baja

prioridad en la solución *S* (ambos corresponden al objetivo de suavizado) y FC que son los cambios de color en la solución.

Las sobrecargas se calculan mediante las siguientes expresiones ecuación (7) y (8):

$$FH(S) = \sum_{i=1}^{|H|} FH_i(S), \tag{7}$$

$$FL(S) = \sum_{i=1}^{|L|} FL_i(S). \tag{8}$$

En el cálculo de sobrecargas se hace uso de ventanas deslizantes W_k de tamaño *k*. El tamaño de cada ventana, en la *i-ésima* estación de trabajo, es definido mediante las restricciones de proporción N_i/P_i que son únicas para cada estación de trabajo, por ello se realiza un cálculo para la primer ventana y las ventanas finales. Las ventanas intermedias mantienen constante su tamaño $k = P_i$ durante todos sus deslizamientos hasta comenzar las ventanas finales.

La primera ventana de tamaño P_i contemplará vehículos del día $D - 1$ en conjunto con vehículos del día D, para conocer la cantidad de vehículos de cada día se requieren las siguientes expresiones ecuación (9) y (10):

$$\#c' = P_i - 1, \tag{9}$$

$$\#c = P_i - \#c', \tag{10}$$

donde $\#c'$ es la cantidad de vehículos del día $D - 1$ que se deben tomar de la secuencia, y $\#c$ es la cantidad de vehículos del día D. Esto únicamente para la primera ventana.

En el caso de las ventanas finales:

$$NewP_i = P_i - 1, \tag{11}$$

$$K = N_i + 1, \tag{12}$$

donde $NewP_i$ nos indica el tamaño de la primera ventana, en el conjunto de ventanas finales. Este valor va en decremento hasta llegar a K que es el tamaño de la última del conjunto de ventanas finales ecuaciones (11-12).

Este suceso de las últimas ventanas se debe a que la secuencia es infinita y se consideran los carros del día $D - 1$, D y se toma como supuesto que existirán vehículos del día $D + 1$.

Teniendo los tamaños de las ventanas se realiza el proceso de evaluación de la función fitness para cada estación de trabajo mediante la expresión ecuaciones (13):

$$FE_i(S) = \sum_{a}^{b} FE_i(S, W_{Pi}) + \sum_{k}^{NewP_i} FE_i\left(S, W_{NewP_i}\right), \tag{13}$$

donde $FE_i(S)$, representa a la función fitness $FH_i(S)$ para las prioridades *H*, o $FL_i(S)$ para las prioridades *L*. La variable a es la posición donde inicia la primera ventana y b es la posición donde inicia la primera ventana del conjunto de últimas ventanas. Para

obtener los valores de las variables se utilizan las siguientes expresiones ecuaciones (14-15):

$$a = |c'| - \#c', \tag{14}$$

$$b=c-NewPi. \tag{15}$$

El número de sobrecargas en la ventana W_k al momento del deslizamiento de la misma en una secuencia de k carros consecutivos se define como $FE_i(S, W_k)$ y se calcula mediante la expresión ecuación (16-17):

$$FE_i(S, W_k) = \max\left\{\left(\sum_{i=1}^{|E|} E_i(l)\ en\ la\ ventana\ W_k\right) - N_i\right\}, \tag{16}$$

$$\text{Si } FE_i(S, W_k) < 0 \text{ será igual a 0.} \tag{17}$$

3. Algoritmo de evolución diferencial con reparador cromosómico (EDRC)

En esta sección se describe el algoritmo de evolución diferencial con un reparador cromosómico, así como la agrupación de los vehículos en clases de equivalencia usando la distancia de Hamming. Se detalla la estructura del individuo, así como los procesos de cruza y mutación; utilizando para este último el método $(DE/rand/1/bin)$, también se puntualiza el proceso para la generación de la población inicial, implementando un operador de mutación denominado "basado en el cambio" propuesto por Syswerda [9] y se muestra el diagrama general del EDRC.

3.1 Agrupación de vehículos en clases de equivalencia

Como primer paso los automóviles son agrupados en clases de equivalencia definidas por aquellos automóviles que cuentan con los mismos requerimientos u opciones de alta y baja prioridad, H y L respectivamente [7]. Este agrupamiento se realiza mediante el cálculo de la distancia de Hamming entre dos vehículos [3]. El resultado nos indica la similitud o diferencia en cuanto a su configuración. La expresión para este cálculo se muestra en la ecuación (18):

$$\begin{aligned} dist(x,y) = distH(x,y)\left\lfloor\sum_{i=1}^{|H|}\lfloor H_i(x) + H_i(y)\rfloor mod\ 2\right\rfloor \\ +\ distL(x,y)\left\lfloor\sum_{i=1}^{|L|}\lfloor L_i(x) + L_i(y)\rfloor mod\ 2\right\rfloor. \end{aligned} \tag{18}$$

3.2 Individuo

La Tabla 1 muestra la configuración de un individuo (secuenciación de vehículos). El individuo está conformado por dos cromosomas: cromosoma (a) donde cada gen representa una clase y el cromosoma (b) donde cada gen representa un color que le corresponde a la clase del vector (a) [7].

Tabla 1. Ejemplo de configuración de un individuo.

INDIVIDUO															
SECUENCIA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CROMOSOMA (A)	1	1	1	1	1	1	1	2	1	1	1	1	3	1	1
CROMOSOMA (B)	4	4	4	4	4	4	4	5	4	4	4	4	6	5	6

3.3 Generación de población inicial

Los individuos de la población inicial son generados con un operador de mutación denominado "basado en el cambio" a partir del orden de la instancia original, este operador consiste en seleccionar aleatoriamente de la secuencia una subsecuencia de autos, cambiarles el orden y a continuación insertar la subsecuencia en la misma posición que fue extraída [9] .

3.4 Operadores de mutación y cruza

Los operadores de mutación *(DE/rand/1/bin)* y cruza utilizados en esta investigación se muestran en la ecuación (19) y ecuación (20) respectivamente [10]:

Operador de mutación clásico (DE/rand/1/bin). El operador de mutación, definido en la ecuación (19), lleva implícito el proceso de selección, en el cual se eligen de la población, de manera aleatoria 3 individuos $X_{r1}, X_{r2}\ y\ X_{r3}$ diferentes entre sí, y diferentes de X_{ri} que representa al individuo que se busca reemplazar. Observe que el individuo mutado X_m se obtiene realizando operaciones aritméticas, donde F representa al factor de mutación.

$$X_m = X_{r1} + F(X_{r2} - X_{r3}), \tag{19}$$

$$\text{donde: } r1 \neq r2 \neq r3 \neq ri \in [1, N], \qquad 0 < F \leq 2.$$

Operador de cruza. Para el proceso de cruza se utilizan los individuos, X_m y X_{ri}, como padres para generar el individuo de prueba X_t.

El modelo para generar cada X_t se muestra en la Ecuación (20) donde Xm_j representa el j-ésimo gen de X_m , Xri_j representa el j-ésimo gen de X_{ri}.

En este caso, el j-ésimo gen de X_t, se obtiene a partir de generar dos números aleatorios, $rand1$ en un rango de $[0,1]$ y $rand2$ en un rango de $[1, total\ de\ genes]$ y tomando en cuenta la probabilidad de cruza CR, de acuerdo a la regla definida en ecuación (20).

$$X_t = \begin{cases} Xm_j, si \;\; rand1 < CR \text{ ó } rand2 == j, \\ Xri_j, \qquad en\ otros\ casos, \end{cases} \quad donde: j = 1,2,3, \ldots |X_{ri}|. \qquad (20)$$

3.5 Reparador cromosómico

La naturaleza del algoritmo de evolución diferencial (DE) por sus siglas en inglés, requiere de evaluar al individuo para posteriormente, si presenta alguna mejora, elegirlo.

Debido a esta naturaleza al momento de aplicar DE a un problema combinatorio, como es el caso del problema de secuenciación de vehículos, puede darse el caso de que después del proceso evolutivo se generen individuos no factibles, no permitiendo realizar la evaluación de la función objetivo. Por esta situación, es deseable generar un proceso de reparación a dichos individuos convirtiéndolos en factibles, este proceso es conocido como reparador cromosómico.

Se han generado reparadores para otros algoritmos, los cuales permiten reparar fallos en la representación de las soluciones en investigaciones sobre asignación de horarios en una institución educativa y también como operador de selección [11]. Para el problema que abordamos, se desarrollaron dos reparadores cromosómicos, Rclase asociado al objetivo de suavizado y Rcolor asociado al objetivo de cambios de color.

Un cromosoma es un vector cuyos componentes son genes que contienen identificadores de todas las clases. En un cromosoma un mismo identificador puede aparecer varias veces. Un cromosoma es no factible cuando al menos en un gen aparece un identificador de clase que no existe y/o cuando la aparición de un identificador en el cromosoma excede la cardinalidad de la clase.

En este sentido reparar un cromosoma consiste en cambiar aquellos genes que vuelven el cromosoma no factible por genes que contengan identificadores de clase dentro del conjunto de las clases sin que la aparición de los identificadores exceda la cardinalidad de las clases. El proceso para realizar la reparación cromosómica se describe en el Algoritmo 1.

En el Algoritmo 1 se puntualiza el funcionamiento del reparador Rclase, la parte 1 consiste en la reparación del cromosoma (a) del individuo y la parte 2 consiste en la reparación del cromosoma (b), a partir de la secuencia generada por el paso previo. El algoritmo del reparador cromosómico Rcolor (que no se presenta), tiene la misma estructura, pero su ejecución es realizada intercambiando el orden de las partes 1 y 2.

Algoritmo 1: Reparador cromosómico (Rclase)

Parte 1

1. Se toma el individuo $\boldsymbol{Xtclases}$ que se genera después del proceso de mutación y cruza.
2. Se recorre de izquierda a derecha respetando el orden.
3. Para $\boldsymbol{i = 1}$ hasta $|\boldsymbol{Xtclases}|$.
   A. Si $\boldsymbol{Xtclases_i < 0\ or\ Xtclases_i > \#C}$ entonces:
      (1) Eliminar $\boldsymbol{Xtclases_i}$.
   B. Contar cuantas veces existe $\boldsymbol{Xtclases_i}$ en $\boldsymbol{Xtclases}$.
   C. Si $\boldsymbol{\#Xtclases_i > \#c\ C_i}$ entonces:
      (1) Eliminar $\boldsymbol{Xtclases_i}$.
4. Verificar que clases faltan en $\boldsymbol{Xt}$clases.

5. En las posiciones eliminadas, agregar de izquierda a derecha:
6. Las clases faltantes en $\boldsymbol{Xtclases}$.
7. Los carros faltantes para cada clase de acuerdo a $\#\boldsymbol{c}\ \boldsymbol{C_i}$.
8. Regresar el individuo $\boldsymbol{Xtclases}$.

Parte 2

9. Se toma el individuo $\boldsymbol{Xtcolores}$ que se genera después del proceso de mutación y cruza.
10. Se obtiene el $\boldsymbol{color_{max}}$ y el $\boldsymbol{color_{min}}$ que fungirán como el rango.
11. Se recorre de izquierda a derecha respetando el orden.
12. Para $\boldsymbol{i = 1}$ hasta $|\boldsymbol{Xtcolores}|$.
    A. Si $\boldsymbol{Xtcolores_i} \in \boldsymbol{Xtclases_i}$ entonces:
        (1) Contar cuantas veces existe $\boldsymbol{Xtcolores_i}$ en $\boldsymbol{Xtcolores}$.
        (2) Si $\#\boldsymbol{Xtcolores_i} > \#\boldsymbol{ccolor_i}$ entonces:
            I. Eliminar $\boldsymbol{Xtcolores_i}$.
            II. Sino:
                (A) Si $\boldsymbol{Xtcolores_i} < \boldsymbol{color_{min}}\ \boldsymbol{or}\ \boldsymbol{Xtcolores_i} > \boldsymbol{color_{max}}$ entonces:
                    A. Eliminar $\boldsymbol{Xtcolores_i}$.
13. Verificar que colores faltan de acuerdo a su clase en $\boldsymbol{Xt}$colores.
14. En las posiciones eliminadas, agregar de izquierda a derecha:
15. Los colores faltantes en $\boldsymbol{Xtcolores}$.
16. Los carros faltantes para cada color de acuerdo a $\#\boldsymbol{ccolor_i}$.
17. Regresar el individuo completo reparado, tanto el vector de clases como de colores.

3.6 Reemplazo de individuos

El reemplazo utilizado depende de la prioridad de los objetivos, por lo que el proceso de reemplazo se da en tres casos, en cualquiera caso se aplica la proposición que se presenta como Ecuación (21). Si la proposición no se satisface se descarta X_t:

$$Si\ F_b(X_t) \leq F_b(X_{ri})\ entonces\ X_{ri} == X_t \qquad b = 1,2,\ldots|F|. \tag{21}$$

El caso 1 la proposición se ejecuta una sola vez, sin embargo, en el caso 2 y 3, la proposición se ejecuta dos y tres veces respectivamente de manera cíclica, siempre y cuando el paso anterior sea verdadero.

3.7 Aplicación del EDRC

La ejecución del EDRC depende de la forma de evaluar la función objetivo. Para evaluarla se tomó en cuenta la prioridad de los objetivos de manera escalonada, descartando individuos que empeoren el valor fitness del objetivo previo. La población inicial generada en el EDRC es aplicada para F_1 ; la población resultante se toma como inicial para F_2 , y así sucesivamente hasta F_3 .

El diagrama general del algoritmo EDRC se muestra en la fig. 1, al primer criterio de decisión le corresponden 25 generaciones. Al proceso evolutivo (remarcado con rectángulo azul) le corresponden el resto de los criterios de decisión, los cuales se ejecutan durante 100 generaciones cada uno. El proceso evolutivo, sigue el orden: mutación, cruza, reparación cromosómica y reemplazo por cada objetivo.

Se realizaron 10 experimentos por instancia, cada experimento consta de las 25 generaciones del primer criterio de decisión, estas generaciones proporcionan 10

soluciones, de las cuales se elige la de mejor fitness. Al final se obtuvo una solución por cada experimento. En el artículo se reportan el promedio de los 10 experimentos por instancia.

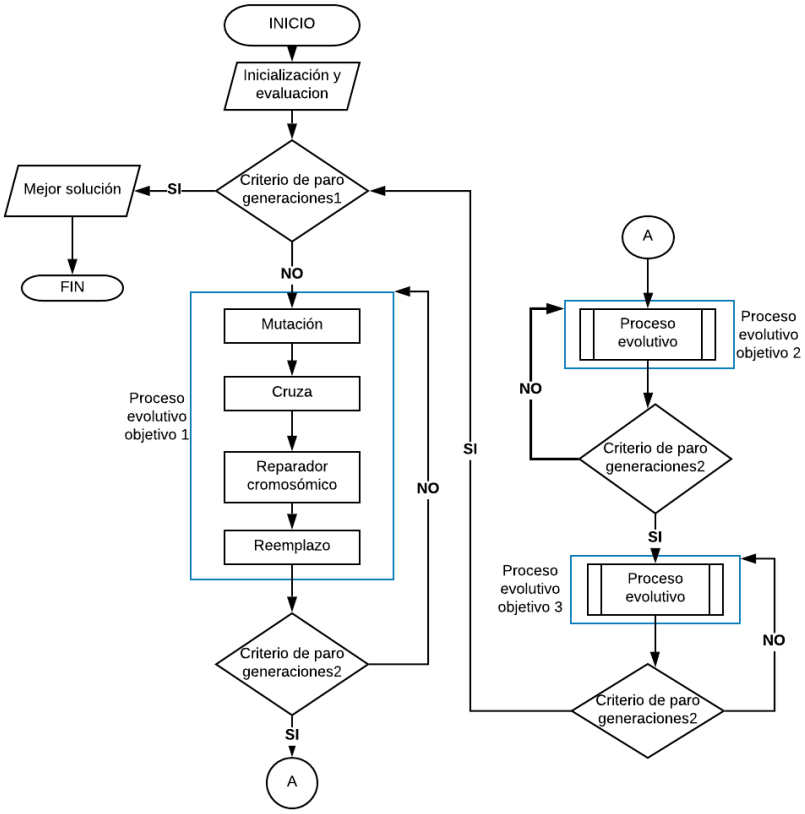


Fig. 1. Diagrama general del Algoritmo EDRC.

4. Diseño de experimentos

El algoritmo EDRC propuesto fue implementado en java y compilado con Intellij IDEA Community Edition 2018. Los experimentos fueron corridos en una computadora Toshiba con un procesador Intel i3-3217u y 6GB de memoria RAM con Windows 10.

Los parámetros utilizados para el algoritmo EDRC, fueron: factor de mutación F=1, este valor fue asignado para evitar la presencia de clases y colores que no existen dentro del cromosoma, factor de cruza CR=0.9 y tamaño de población N=10, ambos parámetros utilizados en el estado del arte.

El conjunto de instancias utilizadas pertenecen al conjunto denominado A de tres, tomadas de la Constraint Satisfaction Problem Lib repository (ver www.csplib.org),

que son un total de 16 instancias. Cada instancia tiene 4 archivos: *vehicles, ratios, paint_batch_limit y optimization_objectives.*

En el archivo *vehicles*, se encuentran los parámetros *c* y *c'* con toda su configuración respecto a los objetivos H, L y color. El archivo de *ratios* muestra las restricciones de proporción para cada opción H y L. El archivo *paint_batch_limit* muestra la restricción dura en cuanto al tamaño de lote de colores consecutivos y *optimization objetives* muestra la prioridad de los objetivos.

5. Resultados

El desempeño del EDRC fue comparado con los resultados reportados por Renault quien utilizo recocido simulado y los resultados reportados en el artículo participantes del concurso Roadef con un algoritmo búsqueda tabú y un greedy para generar la población inicial. Ambos resolviendo el mismo problema.

Los resultados son mostrados en cuatro tablas, una para las instancias de color, otra para las instancias consideradas fáciles para Renault y las dos últimas para las instancias difíciles. Los resultados muestran el desempeño promedio del EDRC propuesto.

Tabla 2. Resultados para las instancias de color.

	PHEL1	**PHEL2**	**PHEL3**	**PHE**
(C´; C)	(99;335)	(14;485)	(29;875)	(27; 954)
(B;|C|)	(15; 12)	(450;12)	(15; 14)	(15;14)
(|H|;|L|)	(4; 2)	(3; 6)	(7; 2)	(5;-)
	µ	µ	µ	µ
ARTICULO FC	27	11	64	68
RENAULT FC	30	11	64	69
PROPUESTA FC	31,2	11	73,88	74,37
ARTICULO FH	367.8	39.4	436	244.6
RENAULT FH	197	48	462	392
PROPUESTA FH	200	49,8	502,55	435,37
ARTICULO FL	101.2	151.4	832.4	NA
RENAULT FL	61	5	883	NA
PROPUESTA FL	63,5	4,7	854,44	NA

En las tablas, los nombres asignados a cada instancia corresponden a sus características en cuanto a nivel de dificultad y prioridad de objetivos. Por ejemplo, PHEL1 significa que FC (representado por P) es más importante que FH (representado por H) y este a su vez mas importante que FL (representado por L), la 'E' significa que H fue considerada como fácil de optimizar para la Renault y cuando aparece 'D' es

considerada difícil de optimizar; si existe más de una instancia de este tipo se le asigna un número entero consecutivo. En el segundo renglón de las tablas, es el par ordenado $(c'; c)$, se indica cuantos carros del día $D-1$ (c') y cuantos del día D (c') tiene esa instancia; en el tercer renglón el par $(B; |C|)$, B indica la restricción dura del objetivo de color y C la cantidad de colores, finalmente en el cuarto renglón el par (|H|;|L|), $|H|$ indica la cantidad de estaciones de trabajo para las restricciones de alta prioridad y $|L|$la cantidad de estaciones de trabajo para las restricciones de baja prioridad. En algunas instancias en el objetivo de FL aparece NA, eso significa que en esa instancia no existen vehículos que requieran pasar por las estaciones de L. Los resultados obtenidos son resaltados con tres distintos colores: azul rey para los resultados donde competimos, verde donde los resultados son mejores que los comparados y rojos donde la propuesta no compite.

Tabla 3. Resultados para las instancias fáciles.

	HEPL1	**HEPL2**	**HEPL3**	**HEPL4**	**HELP**
(C´; C)	(99;335)	(14;485)	(29;875)	(228; 1,004)	(228;1004)
(B;|C|)	(15; 12)	(450;12)	(15; 14)	(10;24)	(10;24)
(|H|;|L|)	(4; 2)	(3; 6)	(7; 2)	(4;18)	(4;18)
	μ	μ	μ	μ	μ
ARTICULO FC	38.8	38.8	137.4	232.8	833.2
RENAULT FC	46	70	195	290	290
PROPUESTA FC	121,77	256,22	195,88	300,44	291,66
ARTICULO FH	0	0	0	0	0
RENAULT FH	28	2	2	0	0
PROPUESTA FH	0	3,22	2,66	26,88	0,1
ARTICULO FL	107.58	49.4	801.4	3705.2	377.2
RENAULT FL	50	2	787	2075	2075
PROPUESTA FL	50,88	14,66	785	2163,44	2041

Tabla 4. Resultados para las instancias difíciles parte 1.

	HDP	**HDPL1**	**HDPL2**	**HDPL3**
(C´; C)	27;954)	(18;600)	(14;1,315)	(14; 1,260)
(B;|C|)	(20; 14)	(10;12)	(10; 13)	(10;13)
(|H|;|L|)	(5; –)	(5; 12)	(5; 8)	(5;8)
	μ	μ	μ	μ
ARTICULO FC	137.2	180.8	303.2	296.2
RENAULT FC	229	182	468	363
PROPUESTA FC	397,33	187,66	537,33	536
ARTICULO FH	13.4	0	4.2	4
RENAULT FH	115	35	98	73
PROPUESTA FH	117,33	61,55	256,44	77,66
ARTICULO FL	NA	1181.4	168.8	293.4
RENAULT FL	NA	861	99	205
PROPUESTA FL	NA	852,55	198,66	218,55

Tabla 5. Resultados para las instancias difíciles parte 2

	HDLP1	**HDLP2**	**HDLP3**
(C´; C)	(18;600)	(14;1,315)	(14;1,260)
(B;|C|)	(10;12)	(10;13)	(10;13)
(|H|;|L|)	(5;12)	(5;8)	(5;8)
	μ	μ	μ
ARTICULO FC	368.2	394	366,8
RENAULT FC	334	392	464
PROPUESTA FC	199,33	464,22	563,88
ARTICULO FH	0.2	4	4
RENAULT FH	42	106	82
PROPUESTA FH	59,11	299,88	89,44
ARTICULO FL	99.4	63.6	29,6
RENAULT FL	98	134	77
PROPUESTA FL	828,44	249,77	132,66

6. Análisis de resultados

Los porcentajes relativos al tipo de instancias fueron los siguientes. Para las instancias de color el 37% (4/11) fue ganado por EDRC, 55% (6/11) compitió y 8% (1/11) obtuvo malos resultados. Para las instancias fáciles el 26% (4/15) fue ganado por EDRC, 53% (8/15) compitió y 21% (3/15) obtuvo malos resultados.

Finalmente, para instancias difíciles el 10% (2/20) fue ganado por EDRC, 20% (4/20) compitió y 70% (14/20) obtuvo malos resultados.

En el total de las 46 comparaciones entre los resultados de la Renault, el artículo seleccionado y EDRC, aproximadamente en el 22% (10/46) fue ganado por EDRC, 39% (18/46) compitió y 39% (18/46) obtuvo malos resultados.

Para determinar la significancia de los resultados obtenidos en los casos donde, de acuerdo a los porcentajes relativos, nuestra propuesta aparece como mejor o compite, se realizó la prueba de normalidad de los resultados con un valor p=0.54, la cual mostro que los datos no se comportan como una distribución normal. Con base en lo anterior se realizaron las siguientes cuatro pruebas no paramétricas: prueba de signos, prueba de rangos con signo de Wilcoxon, análisis de varianza de Friedman y coeficiente de concordancia de Kendall usando un valor de significancia de 0.05.

La hipótesis nula que se planteó para estas pruebas fue que el desempeño de los algoritmos era el mismo. Aunque en la Tabla 2, se observa que EDRC compite, los resultados de las pruebas estadísticas mostraron que la diferencia no es significativa respecto al desempeño de los algoritmos. Lo anterior se concluyó debido a que el valor p que se obtuvo para cada objetivo fue: FC p=0.250, FL p=1 y FH p=0.068; en todos los casos mayores a 0.05.

En el caso de la Tabla 3 donde también se observa que EDRC compite, los valores p de las pruebas estadísticas fueron: para FH p= 0.500 en las pruebas de Wilcoxon y de signos, y p=0.180 para el caso de las pruebas Friedman y Kendal. Para FL p=0.686 para Wilcoxon y prueba de signos, y p=0.655 para Friedman y Kendall. Se observa que en todos los casos los valores p son superiores a 0.05, es decir, en estos casos no existe de diferencia estadística significativa en el desempeño de los algoritmos. En el caso del

objetivo FC, el valor p=0.025, lo que indica que hay diferencia significativa que nos permite concluir que los resultados del estado del arte son mejores a la propuesta.

En el caso de las instancias difíciles (Tabla 4 y 5) donde se aprecia que la propuesta obtuvo malos resultados, las pruebas estadísticas solo confirman que, si existe diferencia significativa en el desempeño de los algoritmos, obteniendo los valores *p* en un rango de $[0.008, 0.046]$.

Algunos aspectos que se observaron durante la experimentación que son importantes de destacar son los siguientes:

Los resultados obtenidos fueron influenciados por la naturaleza contradictoria de los objetivos debido a que, por una parte, el objetivo de suavizado (FH, FL) busca crear agrupaciones pequeñas de vehículos al tratar de respetar las restricciones de proporción, y, por otra parte, el objetivo de color (FC) busca agrupar la mayor cantidad de vehículos del mismo color sin sobrepasar la restricción dura que es el tamaño del lote. Otro aspecto es que la implementación de reparadores cromosómicos en cada objetivo, que se estableció con la finalidad de obtener solo individuos factibles, disminuyo el conflicto de satisfacción de restricciones entre los objetivos, sin embargo, esto fue insuficiente para obtener resultados competitivos en general.

También se hizo evidente que las restricciones de proporción cercanas a la unidad como $2/3$ o $4/5$ en los objetivos de suavizado, provocan una mayor saturación de trabajo en las estaciones y mayor dificultad de crear una secuencia óptima. Los aspectos anteriores deben ser tomados para mejorar el desempeño de la propuesta.

7. Conclusiones

El objetivo de este artículo fue el estudio del desempeño del algoritmo de evolución diferencial en el problema de secuenciación de vehículos (no aplicado en el estado del arte) dotando su estructura clásica con un reparador cromosómico. Se observó que el algoritmo EDRC propuesto mostró, en general, competencia (39% de los casos), mejora (22% de los casos), no logrando competir en el 39% de los casos, particularmente en mayor porcentaje en las instancias difíciles.

El análisis de los resultados mostro la factibilidad del algoritmo EDRC en el problema propuesto y nos permiten establecer, como continuidad de la presente investigación, realizar mayor experimentación e incluir operadores de mutación especializados para cada uno de los objetivos del problema, con la finalidad de buscar mejorar y sustentar el desempeño del EDRC buscando soluciones significativamente mejores a las del estado del arte.

Agradecimientos. Los autores desean agradecer al Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, a través de la beca para estudios de posgrado: 634738 (E. Sánchez) y al Tecnológico Nacional de México/Instituto Tecnológico de León, por el apoyo brindado para la realización de esta investigación.

Referencias

1. Kis, T.: On the complexity of the car sequencing problem. Operations Research Letters 32, pp. 13–17 (2004)

2. Solnon, C.: The car sequencing problem: overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. European Journal of Operational Research 191, pp. 912–927 (2008)
3. Zufferey, N.: Metaheuristics for production systems. Springer, Francia (2016)
4. Dincbas Mehmet, S.H V H P.: Solving the car-sequencing problem in constraint logic programming. In: Proceedings of the European Conference on Artificial Intelligence, pp. 290–295 (1988)
5. Ribeiro, C.C.: A hybrid heuristic for a multi-objetive real-life car sequencing problem with painting and assembly line constraints. European Journal of Operational Research 191, pp. 981–992 (2008)
6. Cordeau, G. J.-F.: Iterated tabu search for the car sequencing problem. European Journal of Operational Research 191, pp. 945–956 (2008)
7. Zinflou, C. a. M.: Design of an Efficient Genetic Algorithm to Solve the Industrial Car Sequencing Problem. Advances in Evolutionary Algorithms (2008)
8. Xiang-yang. Z., and H.Z.-D. Xiang.: A hybrid algorithm based on tabu search and large neigbourdhood search for car sequencing problem. Journal of Central South University 25, pp. 1–16 (2018)
9. Syswerda, G.: Schedule optimization using genetic algorithms. New York (1991)
10. Simon, D.: Evolutionary Optimization Algorithms. Wiley, Estados Unidos de América (2013)
11. Espinal-Dueñas, J.A.A.F.J.: Aplicación de un Reparador Cromosómico al PSO y GA para la solución del problema de las n-Reinas. En: CIINDET (2010)

ISSN 1870-4069

Optimización evolutiva de contextos para la corrección fonética en sistemas de reconocimiento del habla

Rafael Viana Cámara, Diego Campos Sobrino, Mario Campos Soberanis

SoldAI Research, Mérida, México
{rviana,dcampos,mcampos}@soldai.com

Resumen El reconocimiento automático del habla es un área de creciente interés por la demanda de aplicaciones que lo utilizan para brindar una forma de comunicación natural a los usuarios. Es común que los sistemas de reconocimiento presenten fallas en aplicaciones que usan un lenguaje propio de un dominio específico. Para reducir el error se utilizan diferentes estrategias como proporcionar un contexto que modifique el modelo de lenguaje y un pos-procesamiento de corrección. En este artículo se explora el uso de un proceso evolutivo para la generación de un contexto optimizado al dominio de aplicación, así como diferentes técnicas de corrección basadas en métricas de distancia fonética. Los resultados obtenidos muestran la viabilidad de los algoritmos genéticos como una herramienta de optimización de los contextos, lo cual sumado a un pos-procesamiento de corrección basado en representaciones fonéticas es capaz de reducir los errores en el reconocimiento.

Palabras clave: reconocimiento del habla, distancia fonética, algoritmos genéticos.

Evolutionary Optimization of Contexts for Phonetic Correction in Speech Recognition Systems

Abstract. Automatic Speech Recognition (ASR) is an area of growing academic and commercial interest due to the high demand for applications that use it to provide a natural way of communication. It is common for general purpose ASR systems to fail in certain applications that use a domain specific language. Different strategies have been used to reduce the error, such as providing a context that modifies the language model and post-processing correction methods. This article explores the use of an evolutionary process to generate an optimized context for a specific application domain, as well as different correction techniques based on phonetic distance metrics. The results show the viability of a genetic algorithm as a tool for context optimization, which added to a post-processing correction based on phonetic representations is able to reduce the errors on the recognized speech.

Keywords: speech recognition, phonetic distance, genetic algorithms.

1. Introducción

Los sistemas de reconocimiento automático del habla (ASR por sus siglas en inglés) resultan de gran relevancia en entornos académicos y empresariales debido a la facilidad de interacción que ofrecen. Se ha visto un creciente interés en la investigación de dichos sistemas los cuales han migrado de modelos probabilísticos a sistemas de redes neuronales profundas [5] que se han convertido en el estándar para aplicaciones profesionales de transformación de audio a texto. Estos sistemas a menudo utilizan un modelo acústico para realizar el reconocimiento en un primer nivel y posteriormente son pasados a modelos de lenguaje para su corrección [9]. Los servicios comerciales generalmente funcionan como una caja negra, imposibilitando al usuario la modificación de los modelos de lenguaje.

Si bien los ASR tienen un buen rendimiento de manera general, con frecuencia se enfrentan con problemas al utilizarlos para reconocer dominios específicos de lenguaje por lo que las técnicas de pos-procesamiento cobran relevancia [4].

Muchas de las tareas de pos-procesamiento y corrección de estos sistemas utilizan un contexto, entendido como un conjunto de palabras, frases y expresiones relacionadas al dominio particular que se desea reconocer. Algunos de ellos cuentan con mecanismos para proporcionar un contexto con el cual mejoran el reconocimiento de ciertas palabras y frases, sin embargo, en muchas ocasiones no es suficiente para mejorar de manera significativa su desempeño.

Dos temas particularmente interesantes son la generación de contextos y la representación fonética para la corrección. Se ha realizado investigación al respecto, sin embargo, falta experimentación relacionada al funcionamiento conjunto de la generación del contexto y representación fonética óptimas.

En este artículo se presenta un método para la generación de contextos utilizando algoritmos genéticos para corregir la salida del sistema de procesamiento de voz a texto de *Google*. A continuación se realiza la comparación de diferentes estrategias críticas en el proceso de corrección de errores: representación de la frase a corregir, selección de candidatos y métricas de comparación.

El artículo está estructurado de la siguiente manera: en la sección 2 se describen los antecedentes del problema y trabajos relacionados; la sección 3 presenta la metodología empleada para la investigación; en la sección 4 se describe el trabajo experimental realizado cuyos resultados se muestran en la sección 5 y finalmente en la sección 6 se proporcionan las conclusiones junto con algunas ideas para desarrollar como trabajo futuro.

2. Antecedentes

Los algoritmos de corrección de errores en sistemas ASR se han abordado desde diferentes perspectivas, incluyendo las fonéticas. Kondrak [11] plantea un algoritmo para calcular una métrica de similitud fonética entre segmentos utilizando características fonéticas articulatorias multivalor. El algoritmo de Kondrak combina conjuntos de operaciones de edición y modelos de alineación

locales y semiglobales para calcular un conjunto de alineamientos casi óptimos.

Pucher et al. [16] presentan matrices de confusión de palabras mediante el uso de diferentes medidas de distancia fonética. Las métricas presentadas se basan en la distancia de edición mínima entre las transcripciones fonéticas y las distancias entre modelos ocultos de Markov. Su investigación muestra que existe una correlación entre la distancia de edición y la confusión de las palabras en sistemas ASR, por lo que este tipo de correcciones se vuelven útiles para rectificar errores de reconocimiento.

En [2] se resalta la problemática que representa usar la distancia de edición para comparar cadenas en lenguajes como el coreano, donde los caracteres representan sílabas en lugar de letras. Lo anterior se refleja en el hecho de que sustituir una sílaba por otra aporta el mismo valor independientemente de la diferencia entre sus letras. La solución tradicional utiliza métricas híbridas entre caracteres y sílabas, sin embargo, los autores argumentan que este enfoque no resuelve satisfactoriamente el problema por lo que proponen una distancia de edición basada en fonemas como solución.

Droppo y Acero [9] utilizan la distancia de edición fonética para incorporar un tercer elemento de corrección a los sistemas ASR. Los investigadores incorporan esta distancia para aprender la probabilidad relativa de cadenas de reconocimiento fonético, dada una pronunciación esperada. Esta estrategia toma en cuenta el contexto de las transcripciones, cambiando la probabilidad de la corrección dependiendo de las palabras previas y posteriores.

Bassil y Semaan [4] emplean una estrategia de pos-procesamiento para la corrección de errores en sistemas ASR. El método presentado para detectar errores en las palabras utiliza un algoritmo de generación de candidatos y uno de corrección de errores sensible al contexto. Los autores reportan una reducción importante de los errores del sistema.

En [7] se emplean estrategias de corrección fonética para corregir los errores generados por un sistema ASR. En el trabajo citado se pasa la transcripción devuelta por el sistema a una representación en formato de Alfabeto Fonético Internacional (IPA por sus siglas en inglés) y un algoritmo de ventana deslizante para la selección de frases candidatas para su corrección de acuerdo a las palabras provistas en el contexto y la distancia a su representación fonética en formato IPA. Los autores reportan una mejora en el 30 % de las frases reconocidas por el servicio de *Google*.

Un componente importante para el algoritmo de corrección fonética es el contexto utilizado para construir las frases candidatas, por lo que se necesitan soluciones capaces de buscar configuraciones óptimas entre espacios de búsqueda extremadamente grandes.

Los algoritmos genéticos son algoritmos de búsqueda estocástica basados en los principios de evolución biológica y emulan el proceso por medio de operadores genéticos aplicando recombinación, mutación y selección natural en una población [13,18]. Se han aplicado para resolver problemas combinatorios complejos y los resultados muestran que constituyen una estrategia potente y eficiente cuando son utilizados de manera correcta. [18].

Este tipo de algoritmos se han utilizado para analizar una gran cantidad de problemas entre los que se encuentran: knapsack [18], problemas de calendarización de procesos, el vendedor viajero [13], búsqueda de funciones para regresión simbólica [1], funciones de kernel gaussiano para análisis de sentimientos [17], entre otros.

Para utilizar algoritmos genéticos se expresan las posibles soluciones como una cadena de símbolos llamada cromosoma, donde cada símbolo es un gen. A partir de una generación inicial de individuos, se ejecutan iterativamente los procesos de selección, mutación, recombinación y evaluación, combinando los genes de los individuos para producir nuevas variaciones. Cada individuo es evaluado de acuerdo a una función llamada *fitness* que describe que tan bien se desempeña como una solución al problema que se intenta resolver.

Los algoritmos genéticos resultan efectivos para explorar espacios amplios de búsqueda, sin embargo, para ciertas configuraciones y problemas, el tiempo de procesamiento necesario para encontrar una solución óptima puede incrementar considerablemente, por lo que a menudo se utilizan técnicas para variar el tamaño de la población y el rango de mutación [13].

En esta investigación se decidió utilizar algoritmos genéticos para la optimización del contexto debido a la efectividad que ha sido reportada en los diferentes dominios de aplicación en los cuales se han utilizado, así como por su capacidad de explorar una gran parte del espacio de búsqueda.

El presente artículo extiende la investigación previa de [7] tomando en cuenta la generación automática de contextos mediante estrategias evolutivas para mejorar la transcripción y la corrección de errores de sistemas ASR utilizando representaciones fonéticas. Se extiende el algoritmo original para comparar los resultados entre diferentes representaciones fonéticas, estrategias para la selección de candidatos de corrección y métricas de distancia. Se utilizan algoritmos genéticos para optimizar el contexto pasado a los algoritmos de corrección para mejorar el reconocimiento, utilizando una función de *fitness* que evalúa el número de palabras corregidas por un contexto determinado.

3. Metodología

El algoritmo de corrección utiliza los componentes de un contexto (conjunto de palabras y frases propias del dominio de la aplicación) para detectar posibles errores en el reconocimiento y corregir la transcripción de un sistema de reconocimiento automático del habla. Está compuesto por tres elementos principales: representación fonética, generador de frases candidatas para corrección y métrica de distancia de edición. Como medida de evaluación de los resultados se utilizó la métrica WER (*Word Error Rate*), que se define como sigue:

$$WER = \frac{S + D + I}{N}, \tag{1}$$

donde S es el número de sustituciones, D el número de supresiones, I el número de inserciones necesarias para transformar la frase hipotética en la frase real y N el número de palabras en la frase real.

3.1. Transcripción fonética

La transcripción fonética es un sistema de símbolos gráficos que representan los sonidos del habla humana. Es usado como convención para evitar las peculiaridades de cada lengua escrita y representar aquellas lenguas sin tradición escrita [10]. Utilizamos como representaciones fonéticas: texto simple, IPA, Double Metaphone (DM) y una variante del Double Metaphone con vocales (DMV).

El IPA es un sistema de notación fonética basado en el alfabeto latino, utilizado como una representación estandarizada de los sonidos del lenguaje hablado [6,19]. Metaphone es un algoritmo fonético que se encarga de indexar palabras por su pronunciación con origen en el idioma inglés [15]. El algoritmo DM es una versión mejorada del algoritmo Metaphone, la cual devuelve una representación del sonido de las letras en la cadena cuando el texto es pronunciado y omite las vocales. El DM ha sido utilizado a menudo para representación del idioma inglés, sin embargo, los sonidos de las vocales resultan de importancia en el español debido a que sirven al hispanohablante para enlazar palabras que terminan en grupos consonánticos [8], por lo que desarrollamos una variante del DM la cual añade las vocales que son eliminadas en el algoritmo original.

3.2. Algoritmos de generación de frases candidatas

Durante el proceso de corrección fonética la búsqueda de frases candidatas genera segmentos de la cadena de entrada que serán contrastadas mediante una métrica de distancia con las palabras y frases del contexto. Una frase candidata es aquella que tiene similitud con alguna de las frases del contexto y que puede contener un error en la transcripción del sistema ASR. Como algoritmos de generación de frases candidatas se utilizaron el de ventana pivotal y el de comparación incremental basado en el tamaño de la frase en letras o sílabas.

En [7] se presenta la estrategia de ventana deslizante en donde se genera un conjunto S_j con una ventana $v = 1$. La selección de subfrases de realiza utilizando un pivote p_j y el conjunto S_j de frases candidatas se genera por las subfrases $\{p_j,\ p_{j-1}p_j,\ p_jp_{j+1},\ p_{j-1}p_jp_{j+1}\}$.

Para este artículo se implementó un método de búsqueda incremental de subfrases que se describe a continuación:

Sea $C = \{c_1, \ldots, c_n\}$ el conjunto de n frases específicas del contexto, $T = \{t_1, \ldots, t_m\}$ la transcripción original dividida en m palabras, se pretende construir un conjunto $R = \{(s_1, c_1), \ldots, (s_l, c_l)\}$ formado por pares (s_i, c_i) tales que c_i es un elemento del contexto susceptible de sustituir el segmento $s_i = t_j...t_k$ para algún $j \leq k$ en T.

El algoritmo 1 muestra la estrategia utilizada, en la que a partir de cada palabra t_i de la transcripción T se incrementa palabra por palabra la subfrase s a evaluar, hasta que ya no queden elementos del contexto comparables por su tamaño en letras o sílabas de acuerdo al umbral u. Las posibles sustituciones de s por c son agregadas al conjunto R siempre y cuando su distancia sea menor a u. La complejidad del algoritmo es $O(nm^2)$ donde n es el número de elementos del contexto y m el tamaño de la transcripción en palabras.

Algoritmo 1 Algoritmo de búsqueda incremental de candidatos.

Input: El contexto $C = \{c_1, \ldots, c_n\}$, la transcripción $T = \{t_1, \ldots, t_m\}$, un umbral de distancia u, una función de distancia de edición $d(a, b)$.

Output: un conjunto $R = \{(s_1, c_1), \ldots, (s_l, c_l)\}$ de sustituciones candidatas.

```
1: Calcular el tamaño máximo de frase L_M en C.
2: Inicializa el conjunto R = {}
3: for i ← 1...m do
4:     s ← t_i
5:     j ← i
6:     while j ≤ m & length(s) ≤ L_M/(1−u) do
7:         for all c ∈ C | length(s)(1−u) ≤ length(c) ≤ length(s)/(1−u) do
8:             if d(s, c) < u then
9:                 Añade el par (s, c) al conjunto R
10:            end if
11:            s ← s + t_j
12:            j ← j + 1
13:        end for
14:    end while
15: end for
16: return R
```

3.3. Métricas de distancia de cadenas

La distancia de edición es utilizadas para cuantificar la diferencia entre dos cadenas de texto en términos del número de operaciones necesarias para transformar una cadena en la otra. En este artículo se experimenta con las métricas de distancia de Levenshtein, Damerau-Levenshtein y Alineación Óptima de Cadenas (OSA por su siglas en inglés).

La distancia de Levenshtein entre dos cadenas de caracteres es el número de inserciones, eliminaciones y sustituciones necesarias para transformar una cadena de caracteres en otra [12]. La distancia Damerau-Levenshtein se puede definir intuitivamente como una extensión de la distancia de Levenshtein añadiendo como operación válida la transposición de dos caracteres adyacentes [3]. OSA es una variación restrictiva de la distancia *Damerau-Levenshtein*, en donde la operación de transposición solo puede ser efectuada una vez por caracter [14], lo cual la hace menos costosa computacionalmente.

3.4. Optimización evolutiva del contexto

Para la generación de contextos se decidió utilizar un algoritmo genético construido a partir de las transcripciones de las frases a corregir. Para la construcción de los individuos se consideraron todas las palabras individualmente, así como las combinaciones de 2 palabras (bigramas) presentes en las frases objetivo de los audios originales. Los individuos fueron definidos por un cromosoma donde cada gen toma el valor 1 si la palabra o bigrama se encuentra en el contexto y 0 en caso contrario.

Algoritmo 2 Algoritmo de optimización del contexto.

Input: Tamaño de población N, número de generaciones G, tamaño del torneo T_s, probabilidad de cruce C_p y probabilidad de mutación M_p.

Output: La población evolucionada p y el error promedio por cada generación $errores$.

1: $p \leftarrow GeneraPoblacionInicial(N)$, $g \leftarrow 0$
2: **while** $g < N$ **do**
3: $\quad errores[g] \leftarrow Evaluar(p)$
4: $\quad p_t \leftarrow Seleccionar(p, T_s)$
5: $\quad p \leftarrow CruzarMutar(p_t, C_p, M_p)$
6: $\quad g \leftarrow g + 1$
7: **end while**
8: **return** p, $errores$

De esta manera cada individuo representa un contexto que puede ser dado como parámetro al algoritmo de corrección. Para evaluar a los individuos, se ejecutó el algoritmo de corrección con la mejor combinación encontrada en el artículo [7] a cada una de las 451 frases y se devolvió como medida de *fitness* el WER total de cada contexto analizado. Se utilizó un algoritmo genético simple descrito en el algoritmo 2 donde la función $GeneraPoblacionInicial(N)$ produce N individuos de manera aleatoria, $Evaluar(p)$ es una función que calcula el WER de cada individuo, lo asigna como medida de *fitness* y devuelve el WER promedio de la población. La selección se realizó con una estrategia simple de torneo.

La función $CruzarMutar(p_t, C_p, M_p)$ realiza la recombinación genética entre los individuos de la población utilizando la técnica de punto de cruce aleatorio mostrada en el algoritmo 3. Posteriormente se realizó el proceso de mutación individuo a individuo y gen a gen de acuerdo al valor de la probabilidad de mutación, la cual se fue reduciendo cada 10 generaciones para disminuir la fluctuación y estabilizar el error cuando se acerca a un mínimo.

Algoritmo 3 Operación de cruce entre individuos.

Input: Individuos I_1 e I_2 a combinarse, un punto de cruce c_i, y un tamaño de cromosoma c_{size}

Output: Descendencia de los individuos de entrada definida como H_1 y H_2.

1: $H_1 \leftarrow g1_1 g1_2 ... g1_{c_i} g2_{c_i+1} g2_{c_i+2} ... g2_{c_{size}}$
2: $H_2 \leftarrow g2_1 g2_2 ... g2_{c_i} g1_{c_i+1} g1_{c_i+2} ... g1_{c_{size}}$
3: **return** H_1, H_2

4. Experimentación

La experimentación en el presente trabajo se realizó utilizando 451 frases transcritas por el sistema de reconocimiento del habla de *Google*. Este mismo

corpus fue utilizado en [7], donde también se pueden obtener detalles del proceso de recolección y formato de dicho corpus.

4.1. Variantes de configuración del corrector

Con el propósito de comparar las variantes de cada uno de los elementos del algoritmo se realizaron un total de 72 experimentos con todas las combinaciones de los métodos presentados en la tabla 1.

Tabla 1. Variantes de métodos para cada elemento del algoritmo.

Representación	Generación frases	Métrica distancia	Google STT
Texto simple	WIN	Levenshtein	Básico
IPA	LET	OSA	Contextual
DM	SYL	Damerau-Levenshtein	
DMV			

El texto de la frase reconocida por el sistema STT y las frases del contexto fueron representadas de diferente manera para su procesamiento. Para todas las representaciones de texto fue necesario realizar un proceso de normalización con el fin de remover símbolos y caracteres extraños, luego esta versión normalizada fue utilizada directamente en forma de texto simple o transformada a una de las representaciones fonéticas analizadas: Alfabeto Fonético Internacional (IPA), Double Metaphone (DM) o Double Metaphone con vocales (DMV).

Se experimentó con la generación de frases candidatas usando los métodos de ventana pivote (WIN) y la comparación incremental de acuerdo al número de caracteres (LET) o sílabas (SYL).

Como métricas de distancia de edición fueron usadas la distancia de Levenshtein y sus variantes OSA y Damerau-Levenshtein. Cada combinación fue probada usando como datos de entrada las 451 transcripciones obtenidas al usar el método básico de *Google* y posteriormente la transcripción resultante de enviar el contexto reportado en [7] al servicio de *Google*.

Para cada una de las 72 diferentes configuraciones se experimentó con diferentes umbrales de coincidencia de las métricas de edición con incrementos de 0.05 hasta un máximo de 0.6. La evaluación de cada configuración experimental se realizó utilizando la métrica WER acumulada globalmente como resultado de calcular el número de ediciones necesarias para transformar la transcripción hipotética en la frase correcta para cada uno de los 451 ejemplos.

4.2. Optimización del contexto

La experimentación descrita en la sección anterior fue realizada con un contexto generado empíricamente de acuerdo al conocimiento a priori de las frases del dominio en donde se observaban fallos en la transcripción. Con el fin de

optimizar el contexto se ejecutó un algoritmo genético cuyos parámetros fueron calibrados realizando una experimentación de 30 ejecuciones con una versión reducida del problema usando un cromosoma de tamaño 50, arrojando los mejores resultados con una población de 50 individuos, 100 generaciones, 95 % de probabilidad de cruce y 5 % de probabilidad de mutación.

Una vez calibrados los parámetros se optimizó el contexto usando un tamaño de cromosoma de 355 donde cada gen representa una de las palabras o bigramas presentes en las transcripciones de los audios de la experimentación. El factor de mutación se fue reduciendo en un 20 % cada 10 generaciones. Los individuos estuvieron evolucionando durante 100 generaciones.

Como función de *fitness* se empleó el WER total obtenido al ejecutar el algoritmo de corrección para la transcripción simple de *Google* utilizando como contexto al individuo definido por el cromosoma. El corrector fonético se ejecutó utilizando representación IPA, selección de ventana pivote, distancia de Levenshtein y umbral de 0.4, la cual fue la mejor configuración reportada en [7].

Se realizaron 5 procesos evolutivos de 100 generaciones donde se inicializó la población con los 25 mejores individuos de la ronda anterior y 25 generados al azar, para poder explorar diferentes variantes evolutivas. La experimentación se realizó con un procesador intel i7, 8GB de RAM, un sistema operativo Debian GNU/Linux. El algoritmo fue implementado en Python3 y se ejecutó 5 veces durante un total de 70 horas.

4.3. Corrección de errores con el contexto optimizado

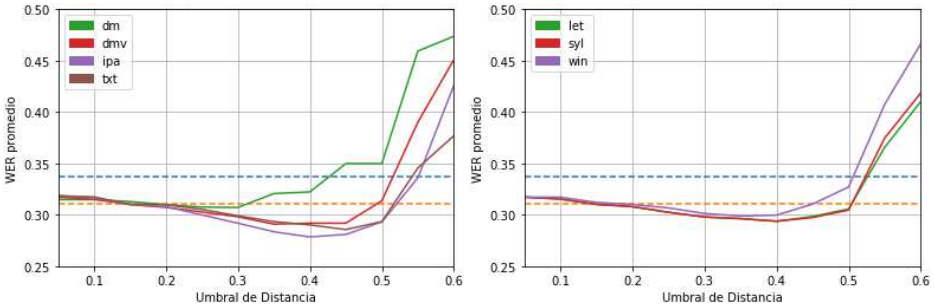
En esta fase de la experimentación se realizaron pruebas para medir los efectos del contexto en el reconocimiento del habla y en el proceso de corrección posterior. Se utilizaron los archivos de audio previamente generados, mismos que fueron enviados de nuevo al sistema de reconocimiento en su modalidad básica y con el contexto genéticamente generado. Esto nos dio una comparación directa sobre el efecto que tiene usar el contexto optimizado con relación a la transcripción obtenida sin enviar contexto, además, nos proporcionó dos lineas de base sobre las cuales aplicar el proceso de corrección a las nuevas transcripciones.

Para comparar los resultados del proceso de corrección, el algoritmo fue aplicado a las nuevas transcripciones producidas por ambas versiones del reconocedor de *Google*. Se usaron las cuatro variantes de configuración que presentaron mejores resultados en la experimentación descrita en la sección 4.1. Tomando como entrada las transcripciones obtenidas por las dos modalidades del servicio de *Google* se ejecutaron dos experimentos para cada una de ellas. En el primero se usó el contexto experimental utilizado en [7], el cual denotaremos como C_m y en el segundo experimento el nuevo contexto generado genéticamente C_g.

5. Resultados

La Fig. 1(a) muestra la variación en el WER promedio obtenido en los experimentos agrupados por el modo de representación con diferentes umbrales

de distancia. Las líneas horizontales corresponden al WER obtenido con la transcripción del STT básico (33.7 %) y el STT contextual (31.1 %). Se observa el efecto de reducción en el WER que se tiene al transformar el texto simple en IPA, sobre todo alrededor del valor 0.4 para el umbral, donde se alcanza un WER promedio mínimo de 27.8 %. Por su parte, la representación en DM produjo resultados similares a los demás métodos para valores pequeños del umbral, sin embargo alrededor de 0.3 empieza a decrecer su capacidad correctiva de manera consistente. La versión de DMV sostiene su rendimiento a la par que el texto simple hasta un umbral de 0.4.



(a) Representación de texto (b) Generación de candidatos

Fig. 1. WER promedio para diferentes representaciones de texto (a) y métodos de generación de candidatos (b).

En la Fig. 1(b) se observa el WER promedio obtenido por las diferentes configuraciones experimentales agrupadas por el algoritmo de generación de frases candidatas. La gráfica muestra un mejor desempeño de las variantes de comparación incremental, ya sea por el tamaño en letras o en sílabas, en relación con la ventana pivote. El WER promedio mínimo (29.4 %) se alcanza con un umbral de 0.4 para el método LET. La versión SYL muestra resultados muy similares, sin embargo el costo computacional de su procesamiento es mayor.

Los mejores resultados se obtuvieron con la configuración de representación IPA y método de selección de candidatos LET. El WER obtenido a partir de la transcripción básica se redujo de 33.7 % a 28.1 % y para la transcripción contextual se redujo de 31.1 % a 27.3 %. Esta configuración presentó una reducción global en el WER relativo de 19.3 %. En relación con las tres métricas de distancia evaluadas la diferencia en los resultados fue prácticamente nula.

Los resultados obtenidos a partir de la experimentación con los algoritmos genéticos para la optimización del contexto utilizando la configuración experimental descrita en la sección 4.2, indican que se obtuvo un error promedio en la quinta ejecución del experimento de 26.5 %, el cual comenzó con un error promedio de 31.2 % y se redujo hasta un promedio de 24.9 % en la generación 100.

En la Fig. 2 se muestran los errores promedio de las 100 generaciones en la quinta ejecución del experimento. El mejor contexto encontrado con esta estrategia quedó conformado por 64 unigramas y 117 bigramas con un WER total de 24.7. %.

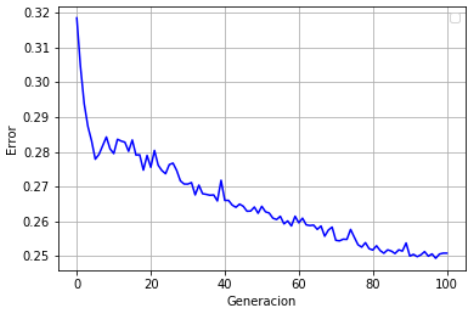
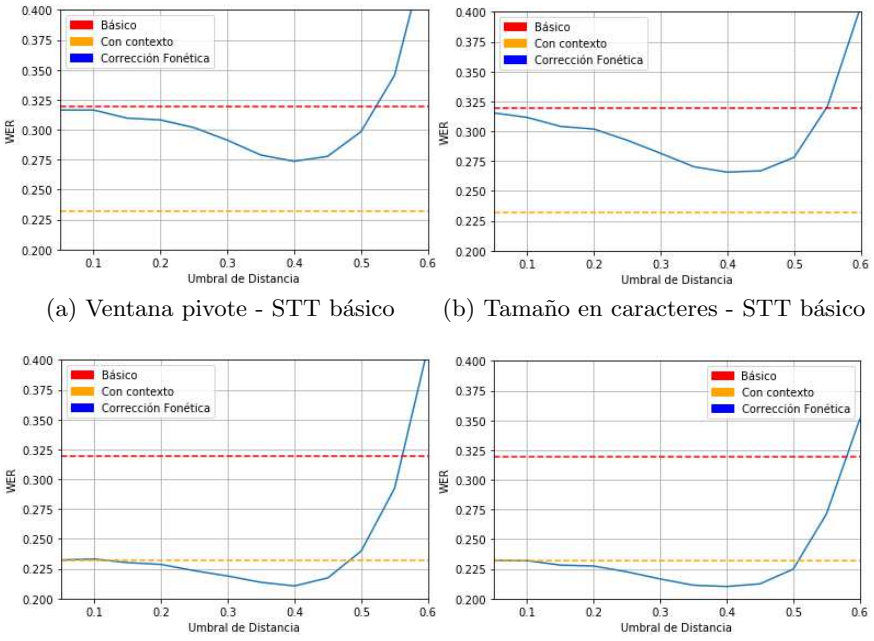


Fig. 2. Gráfica del error promedio por generación del algoritmo genético.



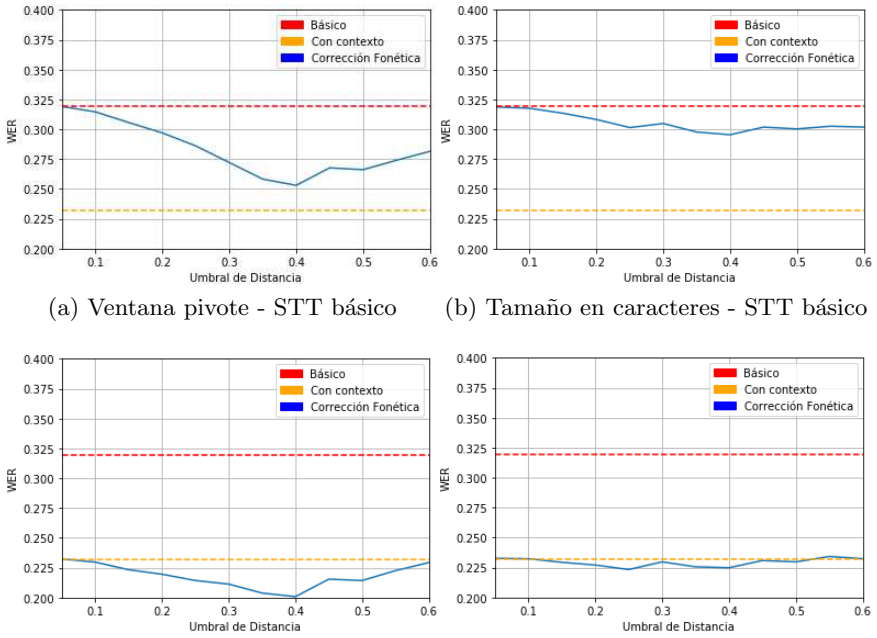
(a) Ventana pivote - STT básico (b) Tamaño en caracteres - STT básico

(c) Ventana pivote - STT contextual (d) Tamaño en caracteres - STT contextual

Fig. 3. Resultados usando C_m como entrada del algoritmo de corrección.

En la fase final de la experimentación descrita en la sección 4.3 se obtienen dos lineas de base a partir de las cuales ejecutar el proceso de corrección. El WER obtenido al comparar las frases reales pronunciadas con la transcripción básica fue de 32.0 %, mientras que al incorporar el contexto generado genéticamente el WER se redujo considerablemente hasta un 23.2 %. Este resultado nos permite ver el impacto que tiene un contexto optimizado en el modelo de lenguaje usado por *Google* al reducir el WER relativo en un 27.3 %.

En la Fig. 3 se ven los resultados del algoritmo de corrección usando la representación IPA con el contexto C_m. Partiendo del STT básico el WER mínimo se obtiene con un umbral de 0.4 y el proceso de selección LET, con esa configuración el WER total se reduce de 32.0 % a 26.6 %, lo que representa una reducción de 16.9 % en el WER relativo. Al iniciar el proceso de corrección a partir de la transcripción del STT contextual el WER se reduce de 23.2 % a 21.0 %, alcanzando una mejora de 9.5 % en el WER relativo.



(a) Ventana pivote - STT básico (b) Tamaño en caracteres - STT básico

(c) Ventana pivote - STT contextual (d) Tamaño en caracteres - STT contextual

Fig. 4. Resultados usando C_g como entrada del algoritmo de corrección.

De manera similar, la Fig. 4 muestra resultados usando como entrada al algoritmo de corrección el contexto generado genéticamente C_g. El uso de este contexto permite reducir el WER mínimo cuando se usa el procedimiento de generación de candidatos WIN, pero parece no generar muy buenos resultados

con el método LET. El mínimo WER absoluto para el STT básico es de 25.3 % con lo que se alcanza una reducción en WER relativo del 21.0 %. Partiendo del STT contextual se alcanza un mínimo de 20.1 % lo que representa una reducción del 13.6 % en WER relativo.

6. Conclusiones y trabajo futuro

A partir de los resultados obtenidos en la experimentación, se muestra la utilidad del algoritmo de corrección fonética para reducir los errores de la transcripción de *Google*, tanto en su versión básica como en la contextual. Se observa que la mejor configuración para el algoritmo se obtiene utilizando IPA como representación fonética y la selección incremental por letras, logrando reducir el WER relativo en un 19.0 %.

De igual manera podemos mencionar que los algoritmos genéticos constituyen una alternativa eficaz para la generación de contextos, puesto que consiguió reducir el WER de la transcripción básica de *Google* de un 32.0 % a 23.2 %. El contexto mostró tener un valor crucial en el desempeño del algoritmo.

Los mejores resultados se obtuvieron de la combinación de la corrección fonética con la optimización evolutiva del contexto logrando una reducción del WER absoluto de 11.9 % al disminuirlo de 32.0 % a 20.1 %, lo que representa una mejora en WER relativo de 37.2 %.

El hecho de que tanto el algoritmo de corrección fonética como la optimización evolutiva del contexto sean independientes del sistema usado para la transcripción y del dominio de aplicación hace que la estrategia presentada pueda ser extendida a diferentes sistemas ASR y dominios de aplicación.

Los algoritmos presentados a lo largo de este artículo pueden tomar ventaja del conocimiento a priori del dominio de aplicación para mitigar el problema del comienzo en frío. Lo anterior se debe a que en caso de que no se cuenten con las transcripciones iniciales, se puede utilizar un contexto generado con el conocimiento humano del dominio, como en [7], el cual puede complementarse con los algoritmos genéticos a medida que se recopila información acerca de las interacciones con usuarios reales del sistema.

Entre las líneas de investigación a futuro se requiere validar los resultados con corpus de diferentes dominios de aplicación, además se prevé la experimentación utilizando costos de edición ponderados tomando en cuenta características fonéticas del español y del audio original tales como ruido, duración, energía de la señal, entre otras. Otra línea de investigación es la comparación con algoritmos de *deep learning*, ya que se puede plantear el problema de la corrección de errores en sistemas ASR como una traducción de transcripciones erróneas a transcripciones correctas, por lo que algoritmos de *Machine translation* pueden resultar de utilidad.

Referencias

1. Anjum, A., Sun, F., Wang, L., Orchard, J.: A novel continuous representation of genetic programmings using recurrent neural networks for symbolic regression. CoRR abs/1904.03368 (2019), http://arxiv.org/abs/1904.03368
2. Bae, B., Kang, S.s., Hwang, B.y.: Edit Distance Calculation by Phonetic Rules and Word-length Normalization 2 Related Works 3 Edit Distance for Korean Words 4 Phoneme-based Edit Distance. Advances in Computer Science (1), 315–319 (2012)
3. Bard, G.V.: Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In: Proceedings of the Fifth Australasian Symposium on ACSW Frontiers - Volume 68. pp. 117–124. ACSW '07, Australian Computer Society, Inc., Darlinghurst, Australia, Australia (2007), http://dl.acm.org/citation.cfm?id=1274531.1274545
4. Bassil, Y., Semaan, P.: ASR context-sensitive error correction based on Microsoft n-gram dataset. CoRR abs/1203.5262 (2012)
5. Becerra, A., de la Rosa, J.I., González, E.: A case study of speech recognition in Spanish: From conventional to deep approach. In: 2016 IEEE ANDESCON. pp. 1–4 (Oct 2016)
6. C., M.M.K.: Phonetic notation. The World's Writing Systems pp. 821–846 (1996)
7. Campos Sobrino, D., Campos Soberanis, M.A., Martínez Chin, I., Uc Cetina, V.: Correccion de errores del reconocedor de voz de google usando métricas de distancia fonética. Research in Computing Science (In Press) (04 2018)
8. Chela-Flores, B.: Consideraciones teórico-metodológias sobre la adqusición de consonantes posnucleares del inglés. RLA. Revista de lingüística teórica y aplicada 44 (12 2006)
9. Droppo, J., Acero, A.: Context dependent phonetic string edit distance for automatic speech recognition. In: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 4358–4361 (March 2010)
10. Hualde, J.: The sounds of Spanish. Cambridge University Press (2005)
11. Kondrak, G.: Phonetic alignment and similarity. Computers and the Humanities 37(3), 273–291 (Aug 2003), https://doi.org/10.1023/A:1025071200644
12. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (feb 1966), doklady Akademii Nauk SSSR, V 163, No 4, pp. 845–848 (1965)
13. Luo, J., Baz, D.E.: A survey on parallel genetic algorithms for shop scheduling problems. CoRR abs/1904.04031 (2019), http://arxiv.org/abs/1904.04031
14. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. 33(1), 31–88 (Mar 2001), http://doi.acm.org/10.1145/375360.375365
15. Philips, L.: Hanging on the metaphone. Computer Language Magazine 7(12), 39–44 (December 1990), accessible at http://www.cuj.com/documents/s=8038/cuj0006philips/
16. Pucher, M., Turk, A., J., A., Fecher, N.: Distance, phonetic and for, measures and recognition, speech and optimization, grammar. 3rd Congress of the Alps Adria Acoustics Association (2007)
17. Roman, I., Mendiburu, A., Santana, R., Lozano, J.A.: Sentiment analysis with genetically evolved Gaussian kernels. CoRR abs/1904.00977 (2019), http://arxiv.org/abs/1904.00977
18. Shah, S.: Genetic algorithm for a class of Knapsack problems. CoRR abs/1903.03494 (2019), http://arxiv.org/abs/1903.03494
19. Wall, J.: International phonetic alphabet for singers: A manual for English and foreign language diction. Dallas, Tex.: Pst (1989)

ISSN 1870-4069

Predicción de trayectorias usando el filtro de Kalman

Astrid C. Estrada-Trejo, Isaac Ramos-Silva, Gabriel Sepúlveda-Cervantes, Edgar A. Portilla-Flores, Eduardo Vega-Alvarado

Instituto Politécnico Nacional,
Centro de Innonvación y Desarrollo Tecnológico en Cómputo,
Ciudad de México, México
email{astrid.adartse.10,isaac92ramos}@gmail.com,
{gsepulvedac,aportilla,evega}@ipn.mx

Resumen. El uso de las tecnologías de cómputo ha tenido un gran auge en los últimos veinte años, aumentando la capacidad de los humanos para interactuar con ambientes que antes parecía imposible recrear. De esta manera, el usuario puede ser capaz de percibir el mundo real a través de la virtualización del mismo, dando pie a que tenga una experiencia sensorial brindada por la computadora. En este trabajo se integra un sistema de captura de movimiento en 3D con un entorno virtual para permitir al usuario interactuar entre el entorno físico y el entorno virtual, aprovechando el medio virtual para desarrollar una interacción atemporal entre el usuario y el entorno real. Por medio de la computadora se presentan los cálculos de predicción de posición dado un objeto a analizar, para que el usuario visualice el comportamiento del objeto a lo largo del tiempo mientras esté inmerso en el entorno virtual, lo que le permite interactuar con el entorno real de diferente manera.

Palabras clave: predicción de trayectoria, filtro de Kalman, realidad virtual.

Trajectory Prediction Using Kalman Filter

Abstract. The use of computer technologies has been booming in the last twenty years, increasing the ability of humans to interact with environments that previously seemed impossible to recreate. In this way, the user can be able to perceive the real world through the virtualization of the same, giving rise to having a sensory experience provided by the computer. This work integrates a 3D motion capture system with a virtual environment to allow the user to interact between the physical environment and the virtual environment, taking advantage of the virtual

environment to develop a timeless interaction between the user and the real environment. The calculations of position prediction given by an object to be analyzed are presented by means of the computer, so that the user can visualize the behavior of the object over time while immersed in the virtual environment, which allows it to interact with the real environment in different ways.

Keywords: Trajectory prediction, Kalman filter, virtual reality.

1. Introducción

La realidad virtual (RV) es una herramienta que ofrece una amplia gama de posibilidades para presentar información, de tal manera que el usuario o receptor la perciba e intérprete en manifestaciones diferentes a las imágenes planas o las nomenclaturas y números, incluyendo además una experiencia sensorial proporcionada por una computadora. En este trabajo se utiliza la RV para la predicción de trayectoria de un objeto rígido con movimiento parabólico, con la intención de que el usuario pueda interactuar con dicho objeto en la posición final calculada.

Predecir se define como *anunciar por revelación, ciencia o conjetura algo que ha de suceder* [1]. Con respecto a las trayectorias, el termino predicción se refiere a la aproximación al estado actual calculada a partir de un conjunto de variables de salida de estados (posiciones) anteriores. Existen diferentes métodos de predicción dependiendo del modelo a predecir: basado en leyes, de caja negra o sin modelo explícito [2]. En este trabajo, con el fin de anticipar la posición de un objeto que cumple la trayectoria de una parábola se aplicó un estimador de estados como método de predicción. El filtro de Kalman es un conjunto de ecuaciones matemáticas que implementan un estimador óptimo del tipo predictor - corrector, que procesa todas las medidas disponibles, sin importar su precisión, para estimar el valor actual de las variables de interés [3].

Para el cálculo de trayectorias se han desarrollado diferentes enfoques. En [4], se propone un método para integrar un algoritmo en un lazo abierto que genera una guía de trayectoria hacía el punto de impacto predicho. Para ello se considera la dinámica de tres cuadrirrotores y la posición y radio de una red, ya que se plantea lanzar y atrapar una pelota con dicha red sostenida por un sistema de tres drones. El algoritmo de predicción se basa en la integración de estimaciones de velocidad y posición actuales mediante un filtro de Kalman. Por su parte, en [5] se presenta un sistema de percepción que rastrea y predice la posición de dos pelotas en movimiento con ayuda de cámaras posicionadas en la cabeza de un robot humanoide, el cual tiene la tarea de atraparlas. El sistema muestra las trayectorias pronosticadas de las dos bolas lanzadas simultáneamente, así como un horizonte artificial que representa la orientación de la cámara. La estimación de la posición y la velocidad de la pelota se realiza mediante filtros de Kalman extendidos (*Extended Kalman Filters*, EKF), o en ciertos casos ajustando una parábola a las mediciones de las cámaras.

Uno de los trabajos más destacados en el área contempla el uso de la realidad virtual para atrapar una pelota real utilizando un sistema de captura para rastrear su movimiento, considerando la ubicación de las manos y la cabeza del receptor [6]. La escena se *renderiza* y se proyecta a través de un casco de realidad virtual. Se desarrollan tres visualizaciones: la primera es la representación de una pelota virtual coincidente con la real, la segunda muestra la trayectoria pronosticada de la pelota y un punto final, y la última únicamente incluye el punto final. Usando filtros de Kalman tipo *Unscented* se generan estimaciones predictivas del movimiento de la pelota a medida que se acerca al receptor.

En el presente trabajo se desarrolla la integración de un sistema de captura de movimiento en 3D con un entorno virtual, que permite al usuario interactuar tanto con el entorno físico como con el virtual, siendo uno reflejo del otro. Así, se aprovecha el entorno virtual para desarrollar una interacción atemporal entre el usuario y el entorno real. Por medio de la computadora se realizan los cálculos de predicción dado un objeto a analizar, en este caso una pelota, la cual será lanzada al usuario para que este la atrape teniendo la oportunidad de visualizar su comportamiento a lo largo del tiempo incluyendo su posición final antes de que ésta ocurra. El algoritmo de predicción es un estimador de estados basado en el filtro de Kalman, que dadas la posición inicial y la velocidad determina la posición final y la trayectoria del objeto analizado.

El trabajo está organizado de la siguiente manera: en la Sección 2 se desarrolla el modelo propuesto para este proyecto, con el filtro de Kalman, junto a su desarrollo matemático. En la Sección 3 se describe la integración del sistema, el algoritmo computacional y su implementación, mientras que en la Sección 4 se presenta la experimentación y sus resultados. Finalmente, las conclusiones y trabajos a futuro se incluyen en la Sección 5.

2. Modelo propuesto

2.1. Tiro parabólico

El tipo de trayectoria predeterminado es el tiro parabólico, por lo que se trabajará su ecuación cinemática mediante los métodos de integración de Euler.[7] Para ello, se proponen los vectores *x, y, z,* dependientes del tiempo, tales que su descomposición (individual) implica una ecuación cuya diferencial se relaciona con la posición respecto a su primer instante en un espacio $(t-1)$. Dichos vectores se indican en las ecuaciones (1) a (3):

$$x_{(t)} = x_{(t-1)} + \dot{x}_{(t)} \Delta_{(t)}, \tag{1}$$

$$y_{(t)} = y_{(t-1)} + \dot{y}_{(t)} \Delta_{(t)}, \tag{2}$$

$$z_{(t)} = z_{(t-1)} + \dot{z}_{(t)} \Delta_{(t)}. \tag{3}$$

Siguiendo el método de integración de Euler, se aplica la segunda derivada para que la diferencia de posición se convierta en una diferencia de velocidad con respecto al tiempo. Esto se muestra en las ecuaciones (4) a (6):

$$\dot{x}_{(t)} = \dot{x}_{(t-1)} + \ddot{x} \Delta_{(t)}, \tag{4}$$

$$\dot{y}_{(t)} = \dot{y}_{(t-1)} + \ddot{y}\Delta_{(t)}, \tag{5}$$

$$\dot{z}_{(t)} = \dot{z}_{(t-1)} + \ddot{Z}\Delta_{(t)}. \tag{6}$$

La variable $\ddot{y}$ es la aceleración debido a la fuerza de gravedad. Dado que su valor es despreciable en las dimensiones x y z, queda expresada solo en y como $\ddot{y} = -9{,}81$ m/s^2. Las ecuaciones (7) a (9) corresponden a los vectores de estado expresados en forma matricial,

$$x_{(t)} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} x + \Delta_{(t)}\dot{x} \\ \dot{x} \end{bmatrix}, \tag{7}$$

$$y_{(t)} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} y + \Delta_{(t)}\dot{y} \\ \dot{y} + \Delta_{(t)}\ddot{y} \end{bmatrix}, \tag{8}$$

$$z_{(t)} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} z + \Delta_{(t)}\dot{z} \\ \dot{z} \end{bmatrix}. \tag{9}$$

2.2. Predicción de la trayectoria

Aplicando el modelo del filtro de Kalman *unscented* considerado para un caso discreto de estimación de estados y realizado para cada vector por separado, se tienen modelos de 1D [3]. La ecuación convencional de la proyección del estado hacia adelante se indica en la expresión (10) [8]:

$$x_{(t)} = Ax_{(t-1)} + Bu_{(t)}. \tag{10}$$

Así, se puede considerar una función de transición g la cual, dadas las variables de posición $x_{(t)}$, aceleración $\ddot{y}$ y un paso de tiempo $\Delta_{(t)}$ regresa una aproximación de la posición siguiente $x_{(t+1)}$. Las ecuaciones (11), (12) y (13) corresponden a tal función en x, y y z, respectivamente, con $x_{(t)}, g(x_{(t)}, \ddot{y}, \Delta_{(t)}) \epsilon \mathbb{R}^2$:

$$g(x_{(t)}, 0, \Delta_{(t)}) = \begin{bmatrix} x + \Delta_{(t)}\dot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \tag{11}$$

$$g(y_{(t)}, \ddot{y}, \Delta_{(t)}) = \begin{bmatrix} y + \Delta_{(t)}\dot{y} \\ \dot{y} + \Delta_{(t)}\ddot{y} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_{(t)} \end{bmatrix} g, \tag{12}$$

$$g(z_{(t)}, 0, \Delta_{(t)}) = \begin{bmatrix} z + \Delta_{(t)}\dot{z} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}. \tag{13}$$

Entonces, $g'(x_{(t)}, \ddot{y}, \Delta_{(t)})$ es una matriz de 2x2 definida como la derivada parcial de g con respecto a $x_{(t)}$ para cada uno de sus componentes. En otras palabras, se considera como la traspuesta de la matriz A de la ecuación (10), que es la covarianza del error de la función de transición $G_{(t)}$. La ecuación (14) corresponde a dicha función:

$$G_{(t)} = g'(x_{(t)}, \ddot{y}, \Delta_{(t)}) = \begin{bmatrix} 1 & 0 \\ \Delta_{(t)} & 1 \end{bmatrix}. \tag{14}$$

Para la distribución del error ($\bar{\Sigma}_{(t)}$) se supone un error inicial de 1 en la distribución del error pasado $\Sigma_{(t-1)}$. De esta manera, la función dispone de un ruido del proceso $Q_{(t)}$, una covarianza del error de la función de transición $G_{(t)}$ y la transpuesta de la covarianza del error de la función de transición $G^T_{(t)}$. Así, se tiene la ecuación (15), sujeta a las observaciones calculadas por la ecuación convencional de observación, expresión (16),

$$\bar{\Sigma}_{(t)} = G_{(t)}\Sigma_{(t-1)}G^T_{(t)} + Q_{(t)}, \tag{15}$$

$$y_{(t)} = Cx_{(t)}. \tag{16}$$

Expresando la ecuación (16) como la función de observación $h(x_{(t)})$ dada en el vector de la posición x, y o z, y entregando la lectura con mayor probabilidad dado el estado actual se obtienen las ecuaciones (17) a (20):

$$h(x_{(t)}) = [H, x_{(t)}], \tag{17}$$

$$h(x_{(t)}) = \begin{bmatrix}1\ 0\end{bmatrix} \begin{bmatrix}x \\ \dot{x}\end{bmatrix}, \tag{18}$$

$$h(y_{(t)}) = \begin{bmatrix}1\ 0\end{bmatrix} \begin{bmatrix}y \\ \dot{y}\end{bmatrix}, \tag{19}$$

$$h(z_{(t)}) = \begin{bmatrix}1\ 0\end{bmatrix} \begin{bmatrix}z \\ \dot{z}\end{bmatrix}. \tag{20}$$

Por las características del Jacobiano, se define la función $h'(x_{(t)})$ de la ecuación (21) como una matriz de 1x2, para calcular la ganancia de Kalman,

$$h'(x_{(t)}) = \left[\Delta_{(t)}, (\Delta_{(t)})'\right]. \tag{21}$$

Puesto que es un sistema lineal, H' es igual a H, donde:

$$h'(x_{(t)}) = \begin{bmatrix}1\ 0\end{bmatrix}. \tag{22}$$

Es necesario calcular los reajustes de la covarianza de la predicción y la suma de las posibles mediciones con respecto al tiempo, utilizando la ecuación 23 de corrección, donde $K_{(t)}$ es la ganancia de Kalman que minimiza los reajustes.

$$K_{(t)} = \bar{\Sigma}_{(t)}H^T_t(H_t\bar{\Sigma}_{(t)}H^T_t + R_{(t)})^-1. \tag{23}$$

A continuación se actualizan el valor estimado de la posición y su covarianza de error siendo ahora $\mu_{(t)}$ la nueva posición con una covarianza reajustada $\Sigma_{(t)}$ que da cálculo al posible error ($\bar{\Sigma}_{(t)}$), empleando las ecuaciones (24) y (25):

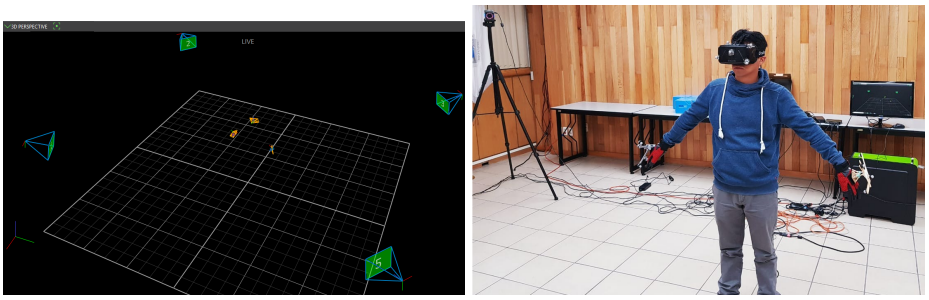
$$\mu_{(t)} = \bar{\mu}_{(t)} + K_{(t)}(z_{(t)} - h(\bar{\mu}_{(t)}), \tag{24}$$

$$\Sigma_{(t)} = (I - K_tH_t)\bar{\Sigma}_{(t)}. \tag{25}$$

3. Integración

3.1. Sistema de captura de movimiento

La captura de movimiento o *motion capture* (mocap) es el proceso de registrar el movimiento de objetos o personas. En este caso se usará la técnica de captura tipo óptico-pasivo, la cual usa marcadores reflectantes que son rastreados por cámaras infrarrojas [9]. Este proceso se muestra en la Figura 1b, con el sistema Vicon utilizado en este desarrollo. Vicon cuenta con el programa *Tracker* (Figura 1a), para visualizar la imagen capturada por las cámaras. Adicionalmente tiene la opción de desplegar imágenes en un plano 3D facilitando la tarea de localización dentro del área. Destaca el software de apoyo para el sistema de captura, ya que permite la interacción con sus elementos (cámaras) y procesa los datos en 2D y 3D.



(a) **(b)**

Fig. 1. Sistema Vicon.

3.2. Implementación en Unity

Unity es un motor de videojuegos que permite desarrollar entornos virtuales y visualizarlos en cascos de realidad virtual, tal como Oculus Rift, para dar al usuario una experiencia inmersiva en un mundo virtual. También permite una comunicación con el sistema de captura de movimiento Vicon, dando los datos de posición de la pelota que son las variables de entrada para el Algoritmo 1: de predicción, en el cual se implementa una estructura de filtro de Kalman Extendido [3]. Se realizan los cálculos respectivos mediante un *script*, que es un programa en lenguaje C# o Java asociado a un *GameObject* (cualquier objeto dentro del entorno de Unity, desde modelos 3D hasta la iluminación, cámara, etc), que le proporciona una funcionalidad dada la programación del script [10]. La función *Update* se encarga de la actualización por marco (*frame*) para el objeto, es decir todo el código que se encuentre dentro de esta función es llamado

de forma automática por el sistema cada marco o fps. Si el proyecto se programa a n fps, *Update* se ejecutará n veces cada segundo [10] permitiendo así que la función de predicción se realice de manera continua y con un tiempo corto en la actualización de datos de posición.

Algoritmo 1: Filtro de Kalman extendido.

Require: Predicción $\{\Sigma_{(t-1)}, \mu_{(t-1)}, x_{(t-1)}, \Delta_{(t)}\}$
$\bar{\mu_{(t)}} = g(x_{(t)}, \mu_{(t-1)})$
$G_{(t)} = g'(x_{(t)}, \dot{\mu}, \Delta_{(t)})$
$\bar{\Sigma}_{(t)} = G_{(t)}\Sigma_{(t-1)}G_{(t)}^T + Q_{(t)}$
return $\bar{\mu}_{(t)}$,$\bar{\Sigma}_{(t)}$
Require: Corrección$\Sigma_{(t)}, \mu_{(t)}, x_{(t)}$
$H_{(t)} = h'(\mu_{(t)})$
$K_{(t)} = \bar{\Sigma}_{(t)}H_t^T(H_t\bar{\Sigma}_{(t)}H_t^T + R_{(t)})^-1$
$\mu_{(t)} = \bar{\mu}_{(t)} + K_{(t)}(z_{(t)} - h(\bar{\mu}_{(t)})$
$\Sigma_{(t)} = (I - K_tH_t)\bar{\Sigma}_{(t)}$
return $\mu_{(t)}$,$\Sigma_{(t)}$
$\bar{\mu}_{(t)}$,$\bar{\Sigma}_{(t)}$= Predicción $\{\Sigma_{(t-1)}, \mu_{(t-1)}, x_{(t-1)}, \Delta_{(t)}\}$
$x_{(t-1)}$=$x_{(t)}$
$\mu_{(t)}$,$\Sigma_{(t)}$=$\Sigma_{(t)}, \mu_{(t)}, x_{(t)}$

4. Experimento y resultados

Para las pruebas del sistema se implementó un área de captura de 7.64 m de largo por 5.96 m de ancho, ubicando al usuario en el centro de este espacio tal como se muestra en la Figura 2A. El sujeto de prueba utiliza el casco de realidad virtual que tiene colocados cinco marcadores reflectantes para registrar la posición y altura del usuario. El objeto de interacción tanto en el entorno virtual como en el real es una pelota con cuatro marcadores reflectantes, como se muestra en la Figura 3. La importancia de llevar acabo su registro de posición es por dos razones; la primera es para emparentarla con el modelo 3D de forma que su posición tanto en el entorno real como en el virtual serán los mismos. La segunda razón es porque la posición del objeto es la analizada para calcular por medio del algoritmo de Kalman su trayectoria futura. Estos dos puntos se ven reflejados en la Figura 2B.

Una vez instalado el usuario dentro del área de captura se coloca frente a él un evaluador a metro y medio de distancia. En la primera etapa de interacción el evaluador sostiene la pelota cuestionando al usuario sobre la posición de está al cambiar la distancia y posición de dicha pelota dentro del área de captura. El usuario inmerso en una escena virtual del área de captura real visualiza una pelota virtual, pero al interactuar con ella lo hace de igual manera con la pelota real (véase la Figura 5).En la segunda etapa, el evaluador permanece a una distancia de metro y medio frente al usuario ahora con la intención de lanzar

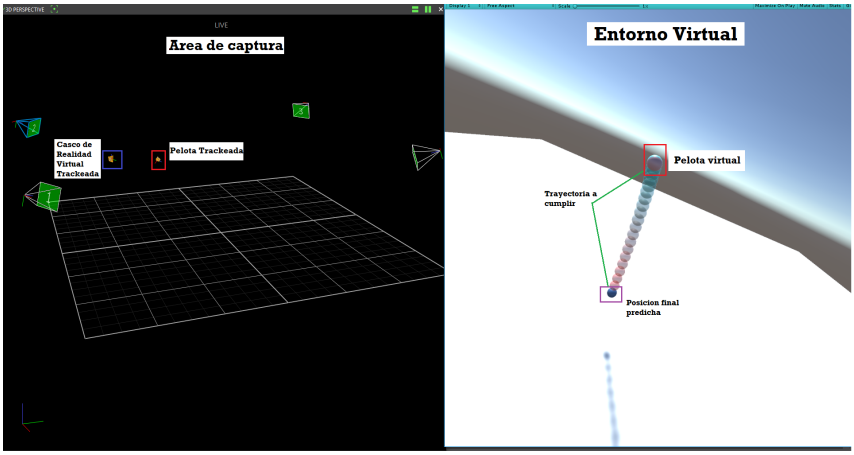


Fig. 2. A) Área de captura donde el usuario interactúa con el objeto real. B) Representación en el entorno virtual, con la trayectoria y la posición final.



Fig. 3. Elementos del sistema de captura de movimiento.

la pelota y que el sujeto de estudio la atrape. En la tercera etapa se activa la función de predicción y se repite la segunda etapa, realizando un intercambio de lanzamientos de la pelota con la ventaja para el usuario al visualizar de manera desplegada la traslación de su posición y su punto final antes de que ocurran.



(a) **(b)**

Fig. 4. A) Interacción con la pelota sin mostrar la predicción. B) Interacción con la pelota visualizando la predicción.

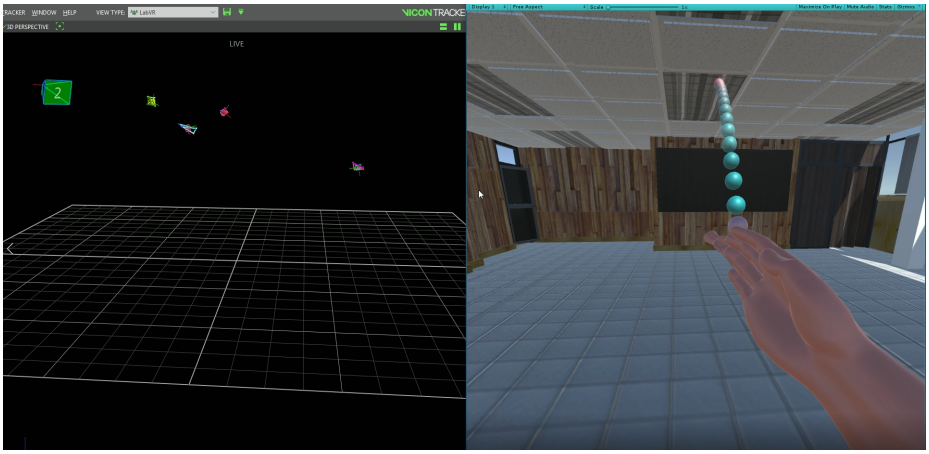


Fig. 5. Entorno virtual del área de captura, donde el usuario tiene además una visualización de una mano 3D que sigue los movimientos de su propia mano.

5. Conclusiones y trabajos a futuro

Es evidente el cambio de actitud en el usuario al visualizar la trayectoria que esta cumpliendo la pelota y su punto final, modificando su postura para atraparla como se observa en la figura 4. La interacción atemporal se da de manera satisfactoria al visualizar la predicción la posición gracias a la integración de los cálculos matemáticos del filtro de Kalman *unscented* en un algoritmo de filtro de Kalman extendido, ya que varias de las matrices se convierten en constantes que no se requiere recalcular.

Una vez completado este proyecto, como trabajo a futuro se tiene planeado desarrollar estudios donde la variable es el entorno de interacción del usuario, para determinar en cual tiene un mayor rendimiento el uso de la predicción, ya sea el entorno real, la realidad virtual, la realidad aumentada o alguna otra

variación entre estas. Para ello se requiere integrar modelos 3D que permitan mayor inmersión en el entorno virtual, por ejemplo manos para el usuario, algún indicador de la posición del evaluador, etc., tal como se muestra en la figura 5.

Agradecimientos. Los autores agradecen al Instituto Politécnico Nacional por el apoyo recibido a través de la Secretaria de Investigación y Posgrado por medio de los proyectos SIP20196288, SIP20190245 y SIP20195772. El primer autor agradece al CONACyT el apoyo otorgado a través de una beca para estudios de maestría.

Referencias

1. Diccionario de la lengua española. 23^a Edición, Real Academia Española (2014)
2. Arahal, M.R., Soria, M.B., Díaz, F.R.: Técnicas de predicción con aplicaciones en Ingeniería. Vol. 15, Universidad de Sevilla (2006)
3. Tellex, S., Brown, A., Lupashin, S.: Estimation for Quadrotors. arXiv preprint arXiv:1809.00037 (2018)
4. Ritz, R., Müller, M.W., Hehn, M., D'Andrea, R.: Cooperative quadrocopter ball throwing and catching. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4972–4978 (2012)
5. Birbach, O., Frese, U., Bäuml, B.: Realtime perception for catching a flying ball with a mobile humanoid. In: 2011 IEEE International Conference on Robotics and Automation, pp. 5955–5962 (2011)
6. Niemeyer, G., Matthew K.X.J.: Pan, Catching a real ball in virtual reality. In: IEEE Virtual Reality (VR), Los Angeles, CA, USA, Marzo 18-22, pp. 269–270 (2017)
7. Hernández, A.G.: Comparación de métodos analíticos y numéricos para la solución del lanzamiento vertical de una bola en el aire. Latin-American Journal of Physics Education, 2(2), 13 (2008)
8. Díaz, J.G., Mejías, A.M., Arteaga, F.: Aplicación de los Filtros de Kalman a Sistemas de Control. Revista INGENIERÍA UC, 8(1) (2001)
9. Vicon Motion Systems Ltd: What is motion capture? Oxford, Reino Unido. Consulta online diciembre 2017. Disponible en https://www.vicon.com/what-is-motion-capture
10. Unity Documentation: https://docs.unity3d.com/es/current/Manual/index.html. Último acceso: 2019/03

ISSN 1870-4069

Modelo para la determinación de la ruta más corta con funciones experimentales para arcos difusos

Eduardo Chandomí-Castellanos, Elías N. Escobar-Gómez, Sabino Velázquez-Trujillo, Héctor R. Hernández De León, Madaín Pérez-Patricio, Carlos Venturino De Coss Pérez

Instituto Tecnológico de Tuxtla Gutiérrez, División de Estudios de Posgrado e Investigación, Tuxtla Gutiérrez, Chiapas, México
{eduardo.chandomi, enescobarg, sabinovelazquez1}@hotmail.com
{hhernandezd, mperez, cdecoss}@ittg.edu.mx

Resumen. En este artículo se propone resolver el problema de la ruta más corta en zonas urbanas utilizando algoritmos de rutas más corta basada en un esquema de lógica difusa, donde los tiempos de recorrido de las calles se consideran como inciertas. Para obtener un resultado de la ruta más corta más cercano a la realidad el modelo evalúa la incertidumbre relacionada con cada una de las calles o aristas de la red con el objeto de ajustar el tiempo de recorrido, es decir, se evalúa la incertidumbre para cada calle de todas las posibles rutas, multiplicando el tiempo de recorrido de la arista por un coeficiente de ajuste. Para cada calle se consideran 3 variables difusas: Estado de las Calles, Zonas de Tráfico e Intensidad de Lluvia, y cada variable está formada con funciones de pertenencia calculadas de manera experimental. Las funciones de pertenencia se determinan por medio de entrevistas, encuestas a expertos en distribución física y datos estadísticos, incluyendo el principio de consistencia propuesto por Zadeh [20]. Para la validación del modelo se comparan los tiempos de recorrido obtenidos con el modelo propuesto con los obtenidos con funciones de pertenencia clásicas (funciones trapezoidales y triangulares) con tiempos de recorrido obtenidos en un muestreo.

Palabras clave: lógica difusa, ruta más corta, tiempo de recorrido incierto, ajuste de tiempo, funciones de pertenencia experimentales.

Model for Determining the Shortest Path with Experimental Functions for Fuzzy Arcs

Abstract. In this paper, we propose to solve the shortest route problem in urban areas using the shortest path algorithms based on a fuzzy logic scheme, where the street's travel times are uncertain. To get a result of the shortest route closer to reality, this model tests the uncertainty level for each one of the edges within the network in order to adjust the travel time that is to say the uncertainty is evaluated for each street of all possible routes, multiplying the edge's travel time by an adjustment coefficient. For each street 3 fuzzy variables are considered: State of the Streets, Traffic zones and, Rain intensity. Each variable based in one membership function estimated experimentally basis and determined through interviews, surveys of experts in physical distribution and statistical data,

including Zade's consistency principle [20]. For model validation, the travel times got with the proposed model are compared with those got with classical membership functions (trapezoidal and triangular functions) and with sampled travel times.

Keywords: fuzzy logic, shortest path, uncertain travel time, time adjustment, experimental membership functions.

1. Introducción

El objetivo principal de toda empresa de distribución es que los productos lleguen al lugar adecuado, en el tiempo determinado, en buenas condiciones y a un costo competitivo [12]. Los costos de un producto generalmente varían dependiendo del tipo de producto que se va a comercializar, el proceso de manufactura empleado y las condiciones del proceso de distribución. En la literatura se reporta que los costos de distribución representan en algunos casos un porcentaje mayor al 30% del valor del producto [2]. El costo de distribución cambia proporcionalmente al precio del combustible, a los costos de almacenaje de la mercancía y de mantenimiento del vehículo.

En el proceso de distribución la flotilla de repartidores debe transitar por diversas partes de la ciudad, en horarios y condiciones ambientales diferentes. En ocasiones esto provoca que los tiempos de reparto, incluso para la misma ruta, cambien considerablemente; ocasionando en muchos casos la demora en la entrega del producto, y mayores gastos de combustible y de mantenimiento [4].

Para la teoría de redes, el problema de la ruta más corta es encontrar el camino con la mínima distancia, tiempo o costo desde un origen a un destino [11]. Actualmente existen métodos que ayudan a determinar la ruta más corta [5], que facilitan la toma de decisión; [7] propone resolver el problema de la ruta más corta dado un mapa de carreteras en el que se encuentre marcado la distancia de cada par de intersecciones de un solo nodo origen a cualquier nodo destino. En [9] se propone un algoritmo más general que el de Dijkstra, debido a que determina la ruta más corta entre todos los pares de nodos o vértices del grafo, es decir de un nodo origen cualesquiera a un nodo destino cualesquiera, en [3], a diferencia de los ya mencionados, se toman en cuentan pesos negativos.

Existen otros métodos para el mismo fin, como, la programación lineal; la cual consiste básicamente en la construcción, solución y análisis del modelo lineal de un problema de ruta más corta [1].

Aplicando alguno de los algoritmos antes mencionados, para la selección de rutas, se observa que los tiempos analizados no son precisos, esto se debe a que las metodologías tradicionales consideran a los datos de manera determinista, siendo que las variables de entrada tienen un comportamiento incierto, provocando que los resultados no se apeguen a la realidad.

En la actualidad existen modelos que procesan información de manera imprecisa, difusa, con variables de entrada que tiene un comportamiento incierto [15], como el presentado por [19], que mejora el algoritmo de Floyd al representar las operaciones de la función canónica con números borrosos triangulares que se basan en el método de representación de la integración gradual media y lo utiliza para obtener un algoritmo

de ruta más corta difusa. En [6] su investigación lo resuelve utilizando el algoritmo de Dijkstra y funciones de pertenencia triangulares y trapezoidales.

2. Método propuesto

Los sistemas de inferencia difusa se basan en los conceptos de la teoría difusa; en la actualidad las aplicaciones varían en áreas como el control automático [10], el diagnóstico de fallas [17], la clasificación de datos, el análisis de decisiones y la robótica [16], entre otros.

En esta investigación, con la implementación de lógica difusa se obtiene un ajuste difuso que adecua el tiempo de recorrido estimado o peso de la arista al tiempo de recorrido real. En [8] presentan un esquema general del concepto de lógica difusa; basados en este esquema se estructura el modelo propuesto, el cual está formado por los cinco elementos mostrados en la Figura 1.

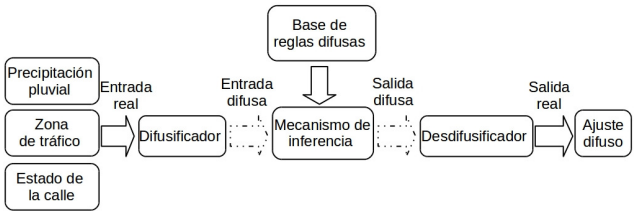


Fig. 1. Elementos de la implementación de lógica difusa.

2.1. Variables lingüísticas

Con base en un estudio realizado a operarios de vehículos de distribución, se detectaron 51 variables que afectan el proceso de distribución. En el análisis se determinan variables cuyo comportamiento es incierto, con mayor frecuencia y con una relación importante en afectación en el proceso de reparto. En los párrafos siguientes se describen las variables difusas y los conjuntos difusos que las forman. Los conjuntos difusos de las tres variables se determinaron de forma experimental, con un coeficiente de correlación, R^2, mayor al 95%, representando el porcentaje de variación de respuesta que explica la relación de los datos reales con las funciones calculadas.

Intensidad de Lluvia (IL). Como variable lingüística, la intensidad de lluvia es la cantidad de precipitación pluvial que provoca una disminución de visibilidad y de velocidad; esta variable está medida en milímetros.

En esta variable, con base en el manual de uso de términos meteorológicos de AEMET (Agencia Estatal de Meteorología de España) [13], se consideran seis casos, los cuales definen los conjuntos difusos.

- Sin lluvia: el nivel de variable se considera conjunto difuso “sin lluvia”, cuando no hay precipitación pluvial.
- Lluvia débil, el nivel de variable se considera conjunto difuso “lluvia débil”, cuando representa una precipitación de 0.1 a 2 ml.

- Lluvia normal: el nivel de variable se considera conjunto difuso "lluvia normal" cuando se tiene una precipitación pluvial de 2.1 a 15 ml
- Lluvia fuerte: el nivel de variable se considera conjunto difuso "lluvia fuerte" cuando se presenta una precipitación de 15.1 a 30 ml.
- Lluvia muy fuerte: el nivel de variable se considera conjunto difuso "lluvia muy fuerte" cuando representa una precipitación de entre 30.1 a 60 ml.
- Lluvia intensa: el nivel de variable se considera conjunto difuso "lluvia intensa" cuando la precipitación pluvial es mayor a los 60 ml.

Para calcular las funciones de pertenencia de cada conjunto difuso se utilizan los datos de precipitación pluvial proporcionada por la Coordinación General del Servicio Meteorológico Nacional (CGSMN) de la Comisión Nacional del Agua (CONAGUA), registrada en el observatorio y en la estación meteorológica automática ESIME (Estación Sinóptica Meteorológica), los datos son recopilados de 1980 a 2010 en la ciudad de Tuxtla Gutiérrez, Chiapas, México; a estos datos se aplica el principio de consistencia propuesto por Zadeh [20]. En la Figura 2 se observan las funciones de pertenencia deducidas a partir de los datos históricos.

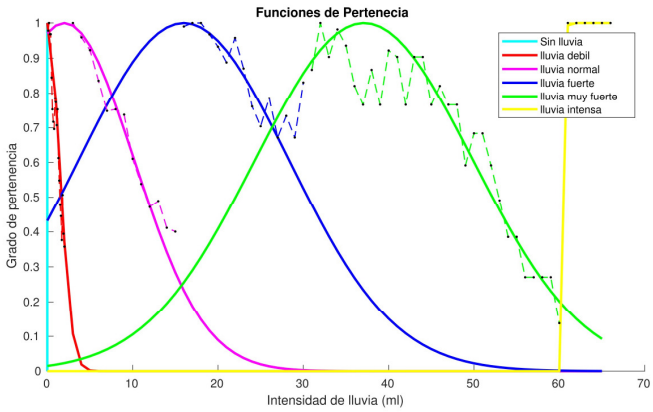


Fig. 2. Funciones de pertenencia de la variable Intensidad de Lluvia obtenidas con datos históricos.

Como se observa en la Figura 2, el primer conjunto difuso (Sin lluvia) tiene un comportamiento representado por un Singleton, ver Ecuación 1. Los siguientes cuatros conjuntos difusos presentan un comportamiento descrito mediante la distribución Gaussiana, ver Ecuación 2, 3, 4 y 5. El último conjunto difuso de la variable lluvia presenta un comportamiento similar al de la distribución Gamma, ver Ecuación 6:

$$\mu_{\{\text{sin } lluvia\}}(x;a) = \begin{cases} 0, & x = a, \\ 1, & x \neq a. \end{cases} \tag{1}$$

$$\mu_{\{débil\}}(x) = \begin{cases} e^{-k(x^2)}, & 0 \leq x \leq b, \\ 0, & b > x. \end{cases} \tag{2}$$

$$\mu_{\{normal\}}(x) = \begin{cases} e^{-k(x-m)^2}, & 0 \leq x \leq c, \\ 0, & c > x. \end{cases} \tag{3}$$

$$\mu_{\{fuerte\}}(x) = \begin{cases} e^{-k(x-n)^2}, & b < x \leq d, \\ 0, & 0 \leq b, d > e. \end{cases} \tag{4}$$

$$\mu_{\{muy\ fuerte\}}(x) = \begin{cases} e^{-k(x-o)^2}, & c \leq x \leq e, \\ 0, & 0 < c. \end{cases} \tag{5}$$

$$\mu_{\{intensa\}}(x) = \begin{cases} 1 - e^{-k(x-p)^2}, & d \leq x \leq e, \\ 0, & 0 \leq d. \end{cases} \tag{6}$$

Estado de las Calles (EC). La variable estado de las calles es definida por la cantidad y tamaño de obstáculos que fijan la velocidad del vehículo, es considerada difusa interpretando un nivel para cada estado de la calle:

- Estado de la calle buena: es definida por la cantidad y tamaño de obstáculos, que permiten conducir a una velocidad constante.
- Estado de la calle regular: representa la cantidad y tamaño de obstáculos que para que el conductor pueda esquivar el obstáculo debe reducir considerablemente la velocidad.
- Estado de la calle mala: es descrito como la cantidad y tamaño de obstáculos, que reducen casi por completo la velocidad pasando obligadamente sobre los obstáculos.

Se realizó un estudio para determinar el tipo de obstáculo y la cantidad que normalmente están presentes en una calle; determinándose que, en las calles de la ciudad de Tuxtla Gutiérrez, Chiapas, México comúnmente se encuentran 1, 2 o 3 topes; 1, 2 o 3 baches, cada uno con el nivel de pequeño, mediano y grande; además de alcantarillado en mal estado, deformaciones de las calles provocadas por hundimientos, reparaciones o mantenimientos, cada uno con un nivel parcial, a mitad y total.

Posteriormente, se encuestaron a 200 operarios de reparto de la ciudad de Tuxtla Gutiérrez, Chiapas, México para determinar el grado de pertenencia de cada uno de los tres conjuntos difusos que se muestra en la Figura 3.

En la Figura 3 se observa que el eje de las abscisas representa el universo en discurso, el cual es determinado en el estudio de obstáculos, alcantarillas y calles realizado; y el eje de las ordenadas representa el grado de pertenencia en el intervalo [0,1] de las funciones de membresía.

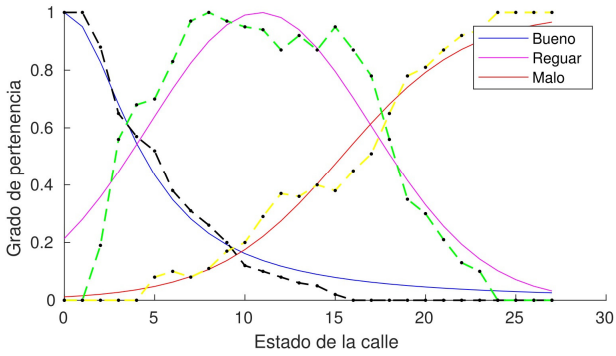


Fig. 3. Funciones de pertenencia de la variable Estado de la Calle obtenidas con datos muestreados.

Para determinar el grado de pertenencia se aplica el método horizontal, el que se basa en las respuestas de un grupo de N "expertos", ver Ecuación 7. El entrevistado responde la pregunta, "¿Puede x ser considerado compatible con el concepto A?", Sólo se acepta un "Sí (1)" o un "No (0)".

$$\mu_x = \frac{Respuestas\ afirmativas}{N}, \tag{7}$$

donde: x= Cada elemento x ∈ X con su grado de pertenencia a μ, N =El número total de encuestados.

Con el análisis de la información recabada en las encuestas realizadas se determinaron el universo en discurso y los grados de pertenencia de cada conjunto difuso. El primer conjunto difuso, "bueno", es representado mediante la distribución Pseudo-exponencial, ver Ecuación 8; el segundo conjunto, "regular", tiene un comportamiento descrito mediante la distribución Gaussiana, ver Ecuación 9; el último conjunto, "malo", se describe mediante la distribución Sigmoidal, ver Ecuación 10.

$$\mu_{\{bueno\}}(x) = \begin{cases} \dfrac{1}{1+kx^2}, & 0 \le x \le a, \\ 0, & x > a. \end{cases} \tag{8}$$

$$\mu_{\{regular\}}(x) = \left\{ e^{-k(x-m)^2}, \quad 0 \le x \le b. \right. \tag{9}$$

$$\mu_{\{malo\}}(x) = \begin{cases} \dfrac{1}{1+e^{-k}(x-n)^2}, & a \le x \le b, \\ 0, & x < b. \end{cases} \tag{10}$$

Zona de Tráfico (ZT). Se define la variable zona de tráfico como el flujo de vehículos que transitan por una calle a cierta hora. Para esta variable se consideran tres casos y cada uno de estos define a un conjunto difuso:

- Zona de tráfico ligero: se considera que la variable difusa zona de tráfico es "ligero" cuando el flujo de vehículos por la calle es rápido.
- Zona de tráfico medio: se considera que la variable difusa zona de tráfico es "medio" cuando el flujo de vehículos es semi-lento.
- Zona de tráfico pesado: se considera que la variable difusa zona de tráfico es "medio" cuando el flujo de vehículos es semi-lento.

Para determinar las zonas que, en la ciudad de Tuxtla Gutiérrez, Chiapas, México, presentan mayor tráfico se identificaron zonas que en cierta hora tienen mayor afluencia de vehículos, como son: escuelas, bancos, supermercados o centros comerciales y cruceros. Para determinar el grado de pertenencia de las zonas con respecto al horario se realizó una encuesta a conductores de transporte de la cuidad. Las encuestas nos proporcionan los valores del grado de pertenencia y el universo en discurso, por consiguiente, obtenemos la función de pertenencia, ver Figura 4.

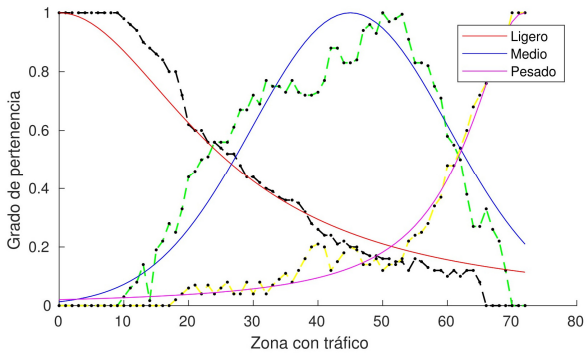


Fig. 4. Funciones de pertenencia de la variable Zona de Tráfico obtenidas con datos muestreados.

Para determinar la función de pertenencia se aplica el método horizontal, ver Ecuación 7. Para el primero y tercero conjunto difuso, "ligero" y "pesado", se determina que los datos se comportan de acuerdo a la distribución Pseudo-exponencial, ver Ecuación 11 y 13; para el segundo conjunto difuso, "medio", se establece que el comportamiento de los datos sigue una distribución Gausiana, ver Ecuación 12.

$$\mu_{\{ligero\}}(x) = \begin{cases} \dfrac{1}{1+kx^2}, & 0 \le x \le a, \\ 0, & x > a. \end{cases} \tag{11}$$

$$\mu_{\{medio\}}(x) = \left\{ e^{-k(x-m)^2}, \quad 0 \le x \le b. \right. \tag{12}$$

$$\mu_{\{pesado\}}(x) = \begin{cases} \dfrac{1}{1+e^{-k(x-n)^2}}, & a \le x \le b, \\ 0, & x < a. \end{cases} \tag{13}$$

2.2. Proceso de difusificación

Es el proceso por el cual a las entradas reales (IL, EC y ZT) se les asigna un valor difuso (valores difusos de entrada), equivalente al grado de pertenencia; para obtener este valor difuso se utilizan las funciones correspondientes a cada conjunto difuso.

2.3. Base de reglas difusas

Quizás la forma más común de representar el conocimiento humano es formar expresiones de lenguaje natural del tipo; "Si" la premisa (antecedente), "Entonces" conclusión (consecuente) [18]. Estas reglas representan el conocimiento del experto en distribución física, el modelo propuesto posee tres variables difusas, dos con tres etiquetas lingüísticas y una con seis, que conforman un total de 54 reglas difusas. En la Tabla 1 se presentan las reglas que se usan para el proceso de inferencia. Ejemplo (ver regla 1): *Si IL es Sin Lluvia, el EC es Bueno y ZT es Ligero entonces el AD es Muy Rápido*.

Tabla 1. Base de reglas difusas.

		Estado de las Calles								
		Bueno			Regular			Malo		
		Zona de Tráfico			Zona de Tráfico			Zona de Tráfico		
		Ligero	Medio	Pesado	Ligero	Medio	Pesado	Ligero	Medio	Pesado
Intensidad de Lluvia	Sin Lluvia	Muy Rápido	Rápido	Lento	Normal	Normal	Lento	Lento	Muy Lento	Muy Lento
	Lluvia Débil	Muy Rápido	Rápido	Lento	Normal	Normal	Lento	Lento	Muy Lento	Muy Lento
	Lluvia Normal	Rápido	Normal	Lento	Normal	Lento	Muy Lento	Muy Lento	Muy Lento	Ext. Lento
	Lluvia Fuerte	Normal	Lento	Muy Lento	Lento	Lento	Muy Lento	Muy Lento	Ext. Lento	Ext. Lento
	Lluvia muy fuerte	Lento	Lento	Muy Lento	Muy Lento	Muy Lento	Ext. Lento	Ext. Lento	Ext. Lento	Ext. Lento
	Lluvia Intensa	Lento	Lento	Muy Lento	Muy Lento	Muy Lento	Ext. Lento	Ext. Lento	Ext. Lento	Ext. Lento

2.4. Motor de inferencia

El motor o mecanismo de inferencia es el proceso que, a partir de los valores difusos de entrada y de la base de reglas difusas [14], nos permite calcular los valores difusos de los conjuntos que forman la variable de salida. En la Figura 5 se presenta la variable de salida (Ajuste Difuso).

Otro concepto importante de lógica difusa es la relación difusa. Una relación difusa representa el grado de presencia o ausencia de asociación, interacción o interconexión entre elementos de dos o más conjuntos difusos. La relación difusa es un conjunto difuso en el espacio producto (llamado producto cartesiano). Si las relaciones difusas pertenecen a diferentes espacios producto, su composición difusa se define en forma análoga a la composición clásica. Entonces, usando la operación de composición máximo–mínimo, la composición difusa se define como una relación difusa cuya función de pertenencia está dada por la Ecuación 14.

$$\mu_{R \circ S}(x, z) = \bigvee_{y \in Y} (\mu_R(x, y) \wedge \mu_S(y, z)), \tag{14}$$

donde: μ representa el grado de pertenencia al conjunto difuso de la variable de salida, $\vee$ representa la unión (operador máximo) y $\wedge$ la intersección (operador mínimo).

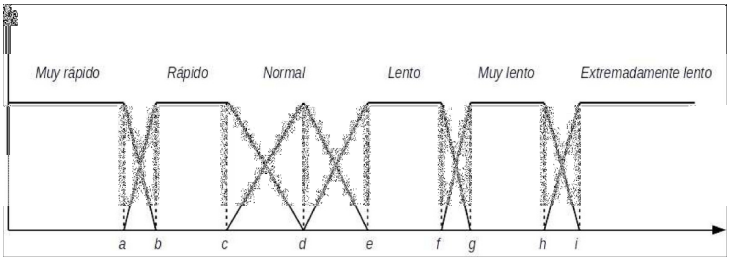


Fig. 5 Conjuntos difusos de la variable de salida: Ajuste Difuso.

2.5. Desdifusificación

A partir del proceso de inferencia se obtienen los grados de pertenencia a los conjuntos difusos de la variable de salida, los que necesitan ser convertidos por un proceso de Desdifusificación, es decir, convertir los valores difusos de la variable difusa de salida a valores concretos que puedan ser procesados. En el modelo propuesto representa el ajuste que se multiplicará por el valor de la arista de la red (tiempo de recorrido), obteniendo un valor más aproximado a lo real. Para convertir las salidas difusas en salidas reales se utiliza el método del centro geométrico o centroide, el que consiste en los siguientes cuatro pasos [8]:

Paso 1. Descomponer en figuras regulares la salida difusa, los grados de pertenencia del conjunto difuso forman el límite de las áreas. Para este paso, se analiza la relación que existe entre dos conjuntos difusos adyacente, con función de pertenencia lineal. De este análisis se define dos casos.

a) El primer caso implica que el grado de pertenencia al primer conjunto difuso es menor o igual al valor de pertenencia al segundo conjunto, ver Ecuación 15. En la Figura 6 se observa el área formada y las figuras regulares en las que se divide cuando se cumple el caso 1

$$\mu_1(AD) \leq \mu_2(AD), \tag{15}$$

AD=Ajuste Difuso.
$\mu_1(AD)$ = Función de pertenencia del primer conjunto difuso.
$\mu_1(AD)$ = Función de pertenencia del segundo conjunto difuso.

b) En el segundo caso, el primer conjunto difuso es mayor que el segundo conjunto, ver Ecuación 16. En la Figura 7 se observa el área formada y las figuras regulares en las que se divide cuando se cumple el caso 2.

$$\mu_1(AD) > \mu_2(AD). \tag{16}$$

En ambos casos se forman cuatro áreas, los cuales se agrupan para después aplicar los pasos 2, 3 y 4.

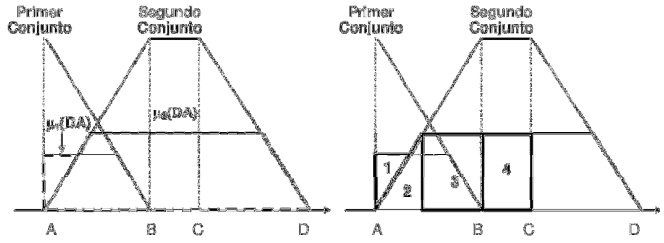


Fig. 6. Área formada y su descomposición en figuras regulares, caso 1.

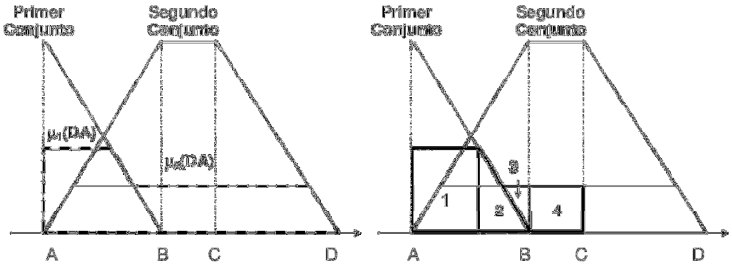


Fig. 7. Área formada y su descomposición en figuras regulares, caso 2.

Paso 2. Calcular la superficie o área de cada figura regular obtenida en el paso 1, aplicando las Ecuaciones 17-20 a los grupos formados.

$$S(A_i) = \begin{cases} \dfrac{[\mu_1(AD)]^2}{2}, & \mu_1(AD) \le \mu_2(AD), \\ \mu_1(AD)[1 - \mu_1(AD)(B - A)], & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{17}$$

$$S(A_i) = \begin{cases} \dfrac{[\mu_2(AD)]^2(B - A)}{2}, & \mu_1(AD) \le \mu_2(AD), \\ \dfrac{[\mu_1(AD)]^2(B - A)}{2}, & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{18}$$

$$S(A_i) = \begin{cases} \mu_2(AD)[1 - \mu_2(AD)(B - A)], & \mu_1(AD) \le \mu_2(AD), \\ \dfrac{[\mu_2(AD)]^2(B - A)}{2}, & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{19}$$

$$S(A_i) = \mu_c(AD)(C - B). \tag{20}$$

Paso 3. Determinar el centroide de cada figura, aplicando las Ecuaciones 21-24.

$$\boldsymbol{C}(A_i) = \begin{cases} A + \dfrac{(B-A)[\mu_1(AD)]}{3}, & \mu_1(AD) \le \mu_2(AD), \\ A + \dfrac{(B-A)[1-\mu_1(AD)]}{2}, & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{21}$$

$$C(A_i) = \begin{cases} A + \dfrac{2(B-A)[\mu_2(AD)]}{3}, & \mu_1(AD) \le \mu_2(AD), \\ B - \dfrac{2(B-A)[\mu_1(AD)]}{3}, & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{22}$$

$$C(A_i) = \begin{cases} B - \dfrac{(B-A)[1-\mu_2(AD)]}{2}, & \mu_1(AD) \le \mu_2(AD), \\ B - \dfrac{(B-A)[\mu_2(AD)]}{3}, & \mu_1(AD) > \mu_2(AD). \end{cases} \tag{23}$$

$$C(A_i) = \frac{B+C}{2}. \tag{24}$$

Paso 4. Calcular el centroide total, ver Ecuación 25 como resultado del proceso de desdifusificación, se obtiene un valor real considerado ajuste difuso el cual se multiplica por el valor de la arista, para obtener un valor aproximado a lo real.

$$Ct = \frac{\sum_{i=1}^{n}\{[S(A_i)][C(A_i)]\}}{\sum_{i=1}^{n} S(A_i)}. \tag{25}$$

2.6. Selección de la ruta más corta

Después de considerar la incertidumbre relacionada con los tiempos de recorrido de cada arista de la red se resuelve el problema de la ruta más corta con resultados más precisos; para el cálculo de la ruta más corta se utiliza el algoritmo de Floyd.

3. Análisis de resultados

3.1. Selección de la ruta más corta

Para el caso de estudio se calcula la ruta más corta de una sección de la ciudad de Tuxtla Gutiérrez, Chiapas, México del primer cuadro de la ciudad que comprende 21 calles y 21 avenidas dando un total de 441 esquinas (nodos) y 884 calles (aristas). Para el tiempo de recorrido se considera las distancias y las velocidades establecidas para cada calle o avenida analizando el estado de las calles, las zonas con tráfico y la precipitación pluvial; los valores para cada variable difusa se presentan en la Tabla 2 (por falta de espacio se muestran los primeros y últimos valores). En el estudio se considera la determinación de la ruta más corta del nodo 1 al nodo 108.

Para el análisis de los tiempos de recorrido se les aplicó un ajuste de tiempos al modelo propuesto y al modelo con funciones de pertenencia triangular y trapezoidal, para las variables: estado de las calles (Figura 8a), zona de tráfico (Figura 8b) e

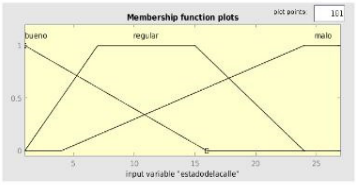
intensidad de lluvia (figura 8c), que suelen ser las funciones de pertenencia usadas comúnmente por su fácil desarrollo y evaluación.

Basado en el método propuesto se desarrolla un programa en MATLAB®. Con este programa se calcula el ajuste de los tiempos de recorrido; al programa se le realizan modificaciones en las funciones de pertenencia para determinar el ajuste considerando las funciones mostradas en la Figura 8.

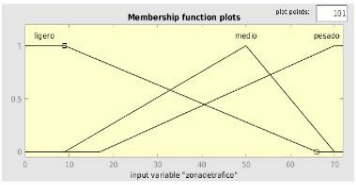
Los resultados de comparación se muestran en la Tabla 3. Los resultados de ambos modelos se comparan con los tiempos obtenidos en el muestreo, con el objetivo de comparar el error porcentual absoluto medio de los modelos con funciones de pertenecía calculadas de manera experimental y el error porcentual absoluto medio de los modelos con funciones clásica, ver Tabla 4.

Tabla 2. Unidad de medida de las variables difusas.

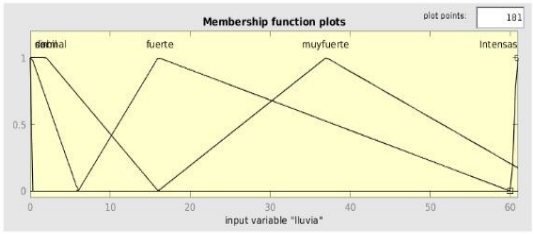
Estado de la calle		Zona de trafico		Intensidad de Lluvia	
Unidad de medida	Descripción	Unidad de medida	Descripción	Unidad de medida	Descripción
1	Un tope pequeño	1	Banco 5 de la mañana	0 ml	Sin lluvia
2	Un bache pequeño	2	Banco 6 de la mañana	1 ml	Cantidad de lluvia en mililitros
3	Dos topes pequeños	3	Banco 7 de la mañana	⋮	
4	Dos baches pequeños	4	Banco 6 de la tarde		
5	Alcantarillado parcialmente en mal estado	5	Banco 7 de la tarde		
⋮	⋮	⋮	⋮		
26	Tres baches grandes	71	2 de la tarde cruceros		
27	Deformación total de la calle	72	2 de la tarde escuelas	60 ml más	



(a) Estado de la calle



(b) Zona de Tráfico



(c) Lluvia

Fig. 8. Funciones de pertenencia trapezoidal y triangular.

Tabla 3. Comparación de funciones experimentales y clásicas.

	Variables difusas			Experimental			Clásicas			Valor Real (min)
	IL	EC	ZT	Ajuste Difuso	Salida (min)	Diferencia	Ajuste Difuso	Salida (min)	Diferencia	
1	0	1	1	0.58	2.9	0.4	0.61	3.05	0.25	3.3
2	0	7	1	0.88	2.552	0.048	1.47	4.263	1.663	2.6
3	0	1	1	0.75	0.2625	0.0125	0.78	0.273	0.023	0.25
4	0	1	21	0.75	0.57	0.01	0.78	0.5928	0.0128	0.58
5	0	1	21	0.75	1.695	0.025	0.78	1.7628	0.0928	1.67
6	0	1	1	0.58	1.1078	0.2622	0.61	1.1651	0.2049	1.37
7	0	1	1	0.58	0.87	0.02	0.61	0.915	0.065	0.85
				Total:	9.95	1.4477	Total:	12.02	3.7115	10.62

Tabla 4. Datos para comparar el error porcentual absoluto medio.

Tiempo de recorrido muestreado	Tiempo con funciones experimentales (FE)	Porcentaje de error absoluto con FE	Tiempo con funciones clásicas (FC)	Porcentaje de error absoluto con FC
3.3	2.9	12%	3.05	8%
2.6	2.552	2%	4.263	64%
0.25	0.2625	5%	0.273	9%
0.58	0.57	2%	0.5928	2%
1.67	1.695	1%	1.7628	6%
1.37	1.1078	19%	1.1651	15%
0.85	0.87	2%	0.915	8%
Media		6%		16%
Varianza		0.47%		4.65%

Se observa en la Tabla 4 que el error porcentual absoluto medio es menor en el modelo con funciones de pertenencia experimentales, también es menor la varianza; por lo que concluimos que el modelo con funciones de pertenencia experimentales predice mejor los tiempos de recorrido que el modelo con funciones de pertenencia clásicas.

4. Conclusión

El problema de la ruta más corta es muy importante para aplicaciones prácticas. Sin embargo, el uso de lógica difusa para considerar los tiempos de recorrido como un valor no determinista mejora los resultados obtenidos.

En este artículo, la implementación de la lógica difusa permite modelar la incertidumbre del problema de la optimización de rutas al entregar productos en la red de carreteras urbanas a través de tres variables lingüísticas: Estado de las Calles, Zonas de Tráfico e Intensidad de Lluvia. Con el cálculo de las funciones de pertenencia se pretende imitar la experiencia de un experto en distribución, tomando en cuenta factores que disminuyen el tiempo de entrega.

Para determinar la ruta más corta dada una serie de nodos o esquina, primero se evalúa la incertidumbre de cada arco de la red, aplicando un sistema de inferencia difuso con funciones de pertenencia calculadas de manera experimental, posteriormente con los algoritmos de ruta más corta se obtiene el recorrido con el menor tiempo. Se

concluye que las funciones de pertenencia calculadas de maneras experimentales y basadas en experiencia del experto, se apegan más a los datos reales en comparación con las funciones de pertenencia clásicas.

Referencias

1. Arreola Risa, J.S., Arreola Risa, A.: Programación lineal: Una introducción a la toma de decisiones cuantitativa. Internacional Thomson Editores (2003)
2. Ballou, R.H.: Logística: Administración de la cadena de suministro. Pearson Educación (2004)
3. Bellman, R.: On a routing problem. Tech. rep., DTIC Document (1956)
4. Bienstock, C.C., Mentzer, J.T., Bird, M.M.: Measuring physical distribution service quality. Journal of the Academy of Marketing Science 25(1), 31–44 (1996)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms, vol. 6. MIT press Cambridge (2001)
6. Deng, Y., Chen, Y., Zhang, Y., Mahadevan, S.: Fuzzy dijkstra algorithm for shortest path problem under uncertain environment. Applied Soft Computing 12(3), 1231–1237 (2012)
7. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik 1(1), 269–271 (1959)
8. Escobar-Gómez, E., Díaz-Núñez, J., Taracena-Sanz, L.: Modelo para el ajuste de pronósticos agregados utilizando lógica difusa. Ingeniería, investigación y tecnología 11(3), 289–302 (2010)
9. Floyd, R.W.: Algorithm 97: shortest path. Communications of the ACM 5(6), 345 (1962)
10. Gutiérrez, T.R., Read, J.S.B., De los Ríos, J.A.S., Gándara, L.C.L., Hernández, J.C.P.: Desarrollo de un controlador difuso y su interfaz de usuario para regulación de potencia en un reactor de investigación (2007)
11. Ji, X.: Models and algorithm for stochastic shortest path problem. Applied Mathematics and Computation 170(1), 503–514 (2005)
12. Kotler, P., Armstrong, G.: Fundamentos de marketing. Pearson Educación (2003)
13. López, M., Viñas Rubio, J.M.: Nuevo manual de uso de términos meteorológicos de aemet (2015)
14. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies 7(1), 1–13 (1975)
15. Martín del Brío, B., Sanz Molina, A.: Redes neuronales y sistemas difusos. Alfaomega Ra-Ma (2002)
16. Orozco-Soto, S.M., Fernández, J.C.R.: Control par calculado difuso basado en pasividad para seguimiento de trayectorias de robots manipuladores. Research in Computing Science 91, pp. 131–141 (2015)
17. Ramos, N.T., Hernández, J.A.R., Cerda, D.A.S., Sánchez, A., López, A.Q.R.: Diagnóstico de fallas en centrales termoeléctricas utilizando modelado neuronal y lógica difusa (2007)
18. Ross, T.J.: Fuzzy logic with engineering applications. John Wiley & Sons (2009)
19. Shukla, K.T.: Fuzzy Floyd's algorithm to find shortest route between nodes under uncertain environment. International Journal of Mathematics and Computer Applications Research (IJMCAR) 1(3), 43–54 (2013)
20. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy sets and systems 1(1), 3–28 (1978)

ISSN 1870-4069

Un nuevo algoritmo de ramificación y acotamiento para el problema de la bisección de vértices

Eduardo Del Ángel-Martínez[1], Héctor Fraire-Huacuja[1], Carlos Soto[1,2], Nelson Rangel[1], Laura Cruz-Reyes[1], Claudia Gómez-Santillán[1]

[1] Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Madero, Tamaulipas, México
[2] Universidad de Cádiz, Escuela Superior de Ingeniería, Cádiz, España
edelangel.mz@gmail.com,automatas2002@yahoo.com.mx,
{nelson.rangel,lauracruzreyes,claudia.gomez}@itcm.edu.mx,
carlos.sotomonterrubio@alum.uca.es

Resumen. En este artículo se aborda el problema de la solución exacta del problema de la bisección de los vértices de un grafo. Este es un problema NP-duro que surge en el contexto de las redes de comunicación. Actualmente, el mejor algoritmo del estado del arte es el denominado BBVBMP, el cual utiliza la metodología de Ramificación y Acotamiento. En este artículo se propone un nuevo algoritmo basado en la misma metodología. El algoritmo propuesto toma ventaja de una heurística constructiva para inicializar la cota superior. Esta heurística considera el grado de los nodos a ser seleccionados como candidatos. Además, el algoritmo propuesto explora el árbol de búsqueda en el espacio combinatorio con una altura máxima de $\lfloor n/2 \rfloor$ nodos, reduciendo de esta forma el tamaño del árbol. Para evaluar el desempeño de las nuevas heurísticas se utilizó un conjunto de instancias estándar con óptimo conocido por construcción. Los resultados experimentales muestran que la heurística incorporada en el algoritmo propuesto supera ligeramente en eficiencia a la utilizada en el BBVBMP. Finalmente se compara el desempeño del algoritmo propuesto contra el BBVBMP, cuando resuelven el conjunto completo de instancias estándar para el problema de la bisección de vértices. Los resultados muestran que el algoritmo propuesto alcanza la misma calidad en las soluciones que BBVBMP, con una reducción del tiempo de ejecución de un 15 %.

Palabras clave: problema de la bisección de vértices, algoritmo de ramificación y poda, heurística constructiva.

A New Branch-And-Bound Algorithm for the Vertex Bisection Problem

Abstract. In this paper we approach the exact solution of the vertex bisection problem. It is a NP-hard problem which ocurrs in the context of

communication networks. Currently, the best state-of-the-arte algorithm is the BBVBMP, it use the branch-and-bound methodology. In this paper, we propose a new algorithm based in the same methodology. Our proposal takes advantage of a constructive heuristic to initiliaize the upper bound. This heuristic considers the degree of the vertex candidates. Also, the proposed algorithm explores the search tree in the combinatorial space with a maximum height of $\lfloor n/2 \rfloor$ nodes, decreasing the size of the tree. The performance of the new heuristics was measure with a benchmark set with known optimals by construction. The experimental results shows that the heuristic in the porposed algorithm outperforms in eficiency the heuristic incorporated in BBVBMP. Finally, the performace of the proposed algorithm is compared against the BBVBMP when solving the vertex bisection problem benchmark. The results show that the proposed algorithm reach the same quality solution as BBVBMP, with a reduction in the execution time of 15 %.

Keywords: vertex bisection problem, branch and bound algorithm, constructive heuristic.

1. Introducción

El problema de la bisección de vértices (VBP) pertenece a la familia de problemas de partición de grafos. Estos problemas consisten en encontrar una partición de sus vértices tal que una función objetivo en particular sea optimizada. El VBP consiste en dividir un grafo en dos subconjuntos B y B' de igual tamaño, excepto cuando el número de vértices es impar ya que el subconjunto B' tendrá un vértice más. Su función objetivo es minimizar el ancho del vértice, definida como el número de vértices en B con al menos un vértice adyacente en B'.

Su formulación formal es la siguiente: Dado un grafo $G(V, E)$ finito, acíclico y no dirigido, el VBP consiste en determinar una partición de V en dos subconjuntos B y B' donde $|B| = \lfloor n/2 \rfloor$ y $|B'| = |V \setminus B|$ tal que el número de vértices en B adyacentes a vértices en B' sea mínimo [4]. Un ordenamiento φ de un grafo no dirigido $G(V, E)$ con $n = |V|$ nodos es una función biyectiva $\varphi : V \rightarrow \{1, 2, ..., n\}$, mientras que $\Phi(G)$ es el conjunto de todos los ordenamientos del grafo G. Un ordenamiento es también llamado un ordenamiento lineal de todos los nodos de G. Dado un ordenamiento φ de un grafo $G = (V, E)$ y un entero i. La separación de vértices en la posición i de φ se define como $\delta(i, \varphi, G) = \left|\{ u \in L(i, \varphi, G) : \exists v \in R(i, \varphi, G), (u, v) \in E \}\right|$, donde $L(i, \varphi, G) = \{ u \in V : \varphi(u) \leq i \}$ y $R(i, \varphi, G) = \{ u \in V : \varphi(u) > i \}$. El costo del ordenamiento es una función F que asocia a cada ordenamiento φ de un grafo G un entero $F(\varphi, G)$. Sea F el costo de ordenamiento; el problema de optimización del ordenamiento asociado con F consiste en determinar algún ordenamiento $\varphi^* \in \Phi(G)$ de un grafo G tal que $F(\varphi^*, G) = \text{mín}_{\varphi \in \Phi(G)} F(\varphi, G)$. Para el problema de la bisección de vértices, $B = L(\lfloor n/2 \rfloor, \varphi, G)$ y $B' = V \setminus B$.

Formalmente, el VBP consiste en determinar el ordenamiento de vértices óptimo $\varphi^* \in \Phi(G)$ tal que minimice la siguiente función objetivo:

$$VB(\varphi^*, G) = \min_{\varphi \in \Phi(G)} VB(\varphi, G). \quad (1)$$

El VBP es un problema combinatorio perteneciente a la clase *NP-Duro* [2]. Este problema se presenta en las redes de telecomunicaciones cuando un mensaje debe ser transmitido a todos los nodos de la red. En esta tarea, en lugar de distribuir el mensaje desde un nodo a todos los nodos; la red de nodos es dividida en dos subconjuntos del mismo tamaño (B y B'). Estos subconjuntos son construidos con la intención de minimizar el número de nodos que tienen una conexión desde B hasta B'. El mensaje es almacenado en B y distribuido hasta B'. En este contexto, el grafo $G = (V, E)$ representa la red, V es el conjunto de nodos de la red y E es el conjunto de canales de comunicación. En este contexto, al VBP se le denomina como el problema de *gossip* [1].

El VBP ha sido resuelto en la literatura por métodos estocásticos y exactos. Los más recientes y relevantes métodos exactos para resolverlo son dos algoritmos de ramificación y acotamiento en [6] y en [8]. En [6], proponen un algoritmo exhaustivo. En su experimentación, usan un benchmark compuesto de 108 instancias: 5 Grids, 84 Small, 15 Trees y 4 Harwell-Boeing. En [8], proponen un algoritmo de ramificación y acotamiento llamado BBVBMP, el cual utiliza una heurística constructiva para generar una cota superior inicial. Además, proponen un nuevo cálculo de la cota inferior. Ya que el BBVBMP explora en el espacio combinatorio, implementa un nuevo procedimiento de ramificación que evita nodos duplicados. Utiliza la estrategia de búsqueda en profundidad para la exploración del árbol de soluciones. Actualmente, el BBVBMP es considerado el algoritmo de ramificación y acotamiento de mejor desempeño en el estado del arte para la solución exacta del VBP.

En este trabajo, se propone un nuevo algoritmo de ramificación y acotamiento que supera al BBVBMP. Este algoritmo es llamado BBVBP y sus principales características son:

1. Una nueva inicialización de la cota superior mediante una heurística constructiva, la cual considera como candidatos a los vecinos de menor grado del nodo actual.
2. Incorpora una estrategia de memorización de cada nivel del árbol de búsqueda. Esta memoria toma ventaja del cálculo previo del grado de los vértices para decidir si un nodo es podado o ramificado.
3. La exploración del árbol de búsqueda se realiza en el espacio combinatorio sin repetición dentro de $C^n_{\lfloor n/2 \rfloor}$.
4. Para optimizar la navegación la implementación de la representación del árbol se basa en estructuras de datos de tipo pila.

Este trabajo está estructurado como sigue: Las dos heurísticas constructivas para inicializar la cota superior se describen en la Sección 2. La estrategia de memorización utilizada para el cálculo de la cota inferior se describe en la

Sección 3. El enfoque de exploración del árbol de búsqueda y el algoritmo de ramificación y acotamiento se presenta en la Sección 4. En la Sección 5 se describen los experimentos computacionales y los resultados obtenidos. Finalmente, las conclusiones y trabajos futuros son presentados en la Sección 6.

2. Definición de la cota superior

En el algoritmo de ramificación y acotamiento, es necesario almacenar y actualizar la cota superior (ub) durante el proceso de exploración del árbol de búsqueda. El valor ub ayuda a decidir si un nodo es ramificado o podado. Si este valor es cercano al óptimo entonces permitirá reducir el tamaño del árbol de búsqueda mediante la poda de más nodos. Este puede ser calculado mediante la generación de una solución inicial aleatoria o por un algoritmo heurístico (como en nuestro enfoque propuesto). Nuestra propuesta es una heurística constructiva voraz para generar un límite superior inicial para el VBP. En esta sección se describen dos heurísticas: la primera heurística (H1) es de [8]; la segunda heurística (H2) genera un conjunto se soluciones que le permiten seleccionar la que presenta un mejor valor objetivo.

2.1. Cota superior con H1

H1 comienza seleccionando el vértice de grado mínimo en G, agregándolo a B junto con el conjunto de sus vecinos, si no es posible agregarlos en su totalidad entonces se agrega un subconjunto del mismo. Mientras el número de elementos en B sea inferior a $\lfloor n/2 \rfloor$, se selecciona un elemento de B en el cual su número de vecinos que no forman parte de B y que no sea 0, es mínimo, agregándolo a B junto con el conjunto de sus vecinos que no forman parte de B, o un subconjunto del mismo si no es posible agregarlos en su totalidad. (Para más detalles vea [8]).

2.2. Cota superior con H2

Sea $N(v) = \{ u \in V \text{ tal que } (v, u) \in E \}$ el conjunto de vecinos del vértice v y φ un conjunto vacío que representa a la mejor solución encontrada, nuestro enfoque propuesto en el Algoritmo 1 tiene el siguiente funcionamiento: inicialmente se crea un conjunto de vértices (A) de grado mínimo de G (línea 1-2). Para cada vértice v de A, se genera una nuevo conjunto B. Se crea un subconjunto C de vértices candidatos que inicialmente solo se encuentra conformado por v (línea 5). En las siguientes $\lfloor n/2 \rfloor$ iteraciones, un vértice v de grado mínimo del conjunto de candidatos que no forme parte del conjunto B, se extrae. De existir más vértices del mismo grado, es elegido el primer vértice de grado mínimo lexicográficamente. Dicho vértice v formará parte del conjunto B (linea 8) y todos sus vecinos que no formen parte de B serán agregados al conjunto de candidatos C (linea 9). Al completar una solución, su valor objetivo es calculado y se actualiza la mejor solución actual φ. Finalmente se regresa la solución de valor mínimo encontrada durante la exploración (φ).

Algoritmo 1: Heurística Constructiva Voraz (H2).

Entrada: Un grafo $G = (V, E)$, donde V es el conjunto de nodos y E es el conjunto de aristas; el número de nodos es $n = |V|$.

Salida: Un ordenamiento φ.

$\varphi \leftarrow \emptyset$
$A \leftarrow \left\{ v \in V \;\middle|\; |N(v)| \leq |N(u)|, \forall u \in V \right\}$
for $a \in A$ **do**
  $B \leftarrow \emptyset$
  $C \leftarrow \{ a \}$
  for 1 to $\lfloor n/2 \rfloor$ **do**
    Seleccionar $v \in C \setminus B$ tal que $|N(v) \setminus B| \leq |N(u) \setminus B|, \forall u \in C$
    $B \leftarrow B \cup \{ v \}$
    $C \leftarrow C \cup N(v) \setminus B$
  end for
end for
if $\varphi = \emptyset$ o $VB(B, G) < VB(\varphi, G)$ **then**
  $\varphi \leftarrow B$
end if
return φ

La figura 1 muestra la construcción de tres soluciones para un problema con 6 vértices. El número a la izquierda de cada vértice indica su grado. Los vértices con un doble círculo son los vértices candidatos. Los vértices en gris forman parte de la solución. Las aristas descartadas en el cálculo de los grados se encuentran marcadas con líneas punteadas.

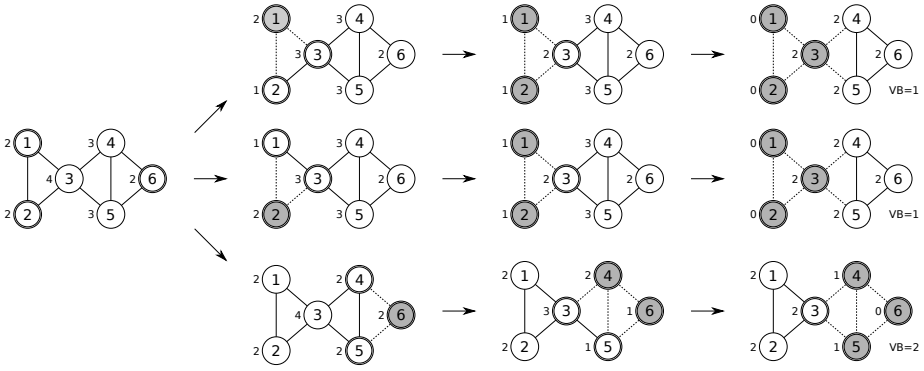


Fig. 1. Heurística Constructiva Voraz (H2).

La heurística comienza identificando el conjunto de los vértices de menor grado (vértices con doble círculo), en este caso, los vértices son $\{ 1, 2, 6 \}$. En los

siguientes pasos, se construye una solución con cada uno de ellos. La primera solución agrega el vértice 1 al conjunto B (marcado en color gris) y los vértices vecinos $\{2, 3\}$ se colocan en el conjunto de candidatos, removiendo las aristas para disminuir el grado. En la siguiente iteración se selecciona el nodo de menor grado disponible en el conjunto de candidatos; siendo el grado el número de vecinos del nodo actual que no forman parte de B, de existir varios de menor grado entonces se selecciona el primero lexicográficamente. En este caso se selecciona al vértice 2 (con grado 1) y son agregados como candidatos aquellos vecinos de 2 que no han sido agregados y que no forman parte del conjunto B. En la siguiente iteración, el vértice de menor grado y único disponible en la posición actual es el vértice 3, el cual es agregado al conjunto B. Al haber alcanzado los elementos requeridos para el conjunto B, los cuales son $\{1, 2, 3\}$, se evalúa la función objetivo con la solución actual, la cual da un valor de $VB = 1$. De igual forma se construyen las dos soluciones restantes. Al terminar las iteraciones, el algoritmo devuelve aquella solución con una función objetivo mínima. En caso de existir varias con el mismo valor objetivo, se devuelve la primera solución de valor mínimo encontrada.

3. Definición de la cota inferior

El cálculo de la cota inferior permite mejorar el desempeño del algoritmo de ramificación y acotamiento. Este enfoque sigue el principio de memorización de la contribución generada por las soluciones parciales en cada nivel del árbol de búsqueda.

La cota inferior (lb) es la contribución de la solución parcial en el nivel actual (n) que es dada por un vector Cbn de tamaño $\lfloor n/2 \rfloor$ que almacena cada una de las contribuciones por nivel del árbol siguiendo:

$$lb = Cbn_i = \begin{cases} l@ \quad l@@l@ \quad l@ \begin{matrix} Cbn_{i-1} + 1, \\ Cbn_{i-1} \end{matrix} & \begin{matrix} |N(nodo) \setminus B| > (\lfloor n/2 \rfloor - |B|), \\ \text{en otro caso.} \end{matrix} \end{cases} \tag{2}$$

Si el $lb \geq ub$, entonces el nodo actual es podado ya que este garantiza que no existe un nodo que al ser ramificado de él mejore el límite superior.

4. Algoritmo de ramificación y acotamiento

Un Algoritmo de Ramificación y Acotamiento es una estrategia para encontrar la solución óptima de problemas de optimización, explorando exhaustivamente todo el espacio de soluciones. Este algoritmo estructura el espacio de soluciones como un árbol de búsqueda y su objetivo es explorar todo el árbol completo para garantizar que encuentra la solución óptima. En el árbol de búsqueda, cada nodo interior representa una solución parcial del problema, mientras que los nodos en las hojas representan soluciones completas. El algoritmo

requiere el cálculo de una cota superior (ub) y una inferior (lb) para decidir si una solución parcial es ramificada o podada. Usualmente, una heurística constructiva voraz o una estrategia aleatoria inicializan el ub y este es actualizado cuando se encuentra un mejor valor en el cálculo de la función objetivo de las hojas del árbol de búsqueda. Mientras se exploran las soluciones parciales, lb es calculado y comparado contra ub para decidir si se poda o ramifica la solución parcial actual. En los problemas de minimización, si lb es inferior a ub; entonces el nodo actual es ramificado; en otro caso, el nodo actual es podado. Este procedimiento permite reducir el tamaño del árbol de búsqueda. Cuando una hoja es alcanzada, el valor objetivo de la solución completa es calculado. Si este es mejor que ub; entonces ub es actualizado. Cada vez que esto ocurre, la nueva solución completa actualiza a la mejor solución actual. Ya que se explora el árbol de búsqueda completo, se puede garantizar que ub es el valor objetivo óptimo y la última solución almacenada es la solución óptima.

En la literatura, existen diversas estrategias para la exploración del árbol de búsqueda, por ejemplo, mediante el uso de listas enlazadas [9,10], pilas [3,7] y una nueva estructura de datos como lo es la IVM [11]. En esta sección, proponemos un nuevo Algoritmo de Ramificación y Acotamiento basado en la representación con pilas. Las pilas controlan el nivel correspondiente de cada nodo desde el árbol de búsqueda mientras se realiza una exploración en profundidad. En el algoritmo 2, presentamos el Algoritmo de Ramificación y Acotamiento propuesto basado en pilas como una estrategia de navegación. El número de niveles o profundidad del árbol de búsqueda para este problema es de $\lfloor n/2 \rfloor$ como puede ser observado en la figura 2. Los niveles internos representan a las soluciones parciales en las cuales el número de nivel corresponde al número de nodos en B.

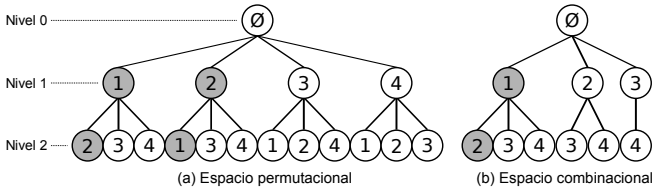


Fig. 2. Árbol de búsqueda en las permutaciones y en las combinaciones.

En la figura 2 se puede observar el árbol de búsqueda en el espacio de las permutaciones (figura 2a) y en el de las combinaciones (figura 2b) para un problema con 4 vértices. Este árbol de búsqueda tiene una profundidad de 2 niveles. En la figura 2a, se puede observar que la solución $\{1, 2\}$ tiene el mismo valor objetivo que $\{2, 1\}$. En la figura 2b, esta solución sólo aparece una vez. También, esto reduce el número de hijos en los nodos que se encuentran más a la derecha, teniendo con esto una reducción del árbol de búsqueda. La última rama en este árbol es $\{3, 4\}$.

Cuando se explora el árbol de búsqueda en el espacio de las permutaciones, algunas soluciones pueden contener nodos duplicados. En el VBP, las soluciones con nodos repetidos son soluciones infactibles. Así, soluciones con los mismos nodos en diferentes posiciones tendrán el mismo valor objetivo ya que esto garantiza tener el mismo número de vértices adyacentes entre B y B'. Estas soluciones se evitan mediante la exploración del espacio de búsqueda combinatorio sin repeticiones, el cual reduce el tamaño del árbol de búsqueda significativamente. En nuestro enfoque, exploramos el espacio de búsqueda combinatorio sin repeticiones con únicamente soluciones factibles. Para lograrlo, en cada nivel se usa la ecuación 3 para realizar el proceso de ramificación y agregar los nodos al árbol en orden lexicográfico. Esta estrategia permite reducir la exploración del árbol de búsqueda como se ve en la figura 2b.

$$hijos(v, nivel) \leftarrow \{\, u \in V : v < u \leq (n - \lfloor n/2 \rfloor + nivel + 1) \,\}. \tag{3}$$

4.1. Algoritmo de ramificación y acotamiento basado en pilas

En el algoritmo 2, B es un vector que almacena los nodos de la solución actual. Cbn es un vector el cual trabaja como memoria de la contribución (lb) de cada elemento en su nivel correspondiente. Su tamaño es equivalente al número de nodos en B. Este vector ayuda a acelerar la decisión de ramificar o podar un nodo evitando calcular la contribución de todos los nodos de la solución parcial. La pila *Elementos* almacena los nodos pendientes a ser explorados en cada nivel, mientras que la pila $Niveles$ almacena el nivel correspondiente de cada elemento almacenado. El conjunto de los nodos adyacentes o vecinos de un nodo v es denotado por $N(v) = \{\, u \in V : (v, u) \in E \,\}$.

El algoritmo 2 recibe el grafo G y el número de nodos n en G. De la línea 1 a la línea 5, se inicializan las variables necesarias. De la línea 6 a la línea 9 se inicializan las pilas *elementos* y *niveles* con todos los nodos posibles. En la línea 10, comienza el proceso principal; se realizan estos pasos mientras la pila *elementos* tenga elementos pendientes. En la línea 11 y 12 se recibe un nodo de la pila *elementos* y el nivel correspondiente de la pila *niveles*. La contribución de la solución parcial es actualizada de la línea 13 a la línea 16. En la línea 17 se decide si la solución actual será podada o será evaluada para ramificar. Si se cumple la condición de la línea 17, se agrega v al vector B en la posición *nivel* en la línea 18. La línea 19 verifica si una hoja es alcanzada. De la línea 20 a la línea 23, si una hoja es alcanzada se evalúa la función objetivo de la solución actual y si la nueva solución mejora al ub, la mejor solución global es actualizada. De la línea 25 a la línea 28, si una hoja no es alcanzada, se calcula el valor del *limite* y son agregados los nuevos elementos a la pila con su respectivo nivel. Finalmente, se regresa la solución óptima en la línea 29.

La figura 3 muestra el desarrollo del algoritmo de ramificación y acotamiento con la exploración del árbol de soluciones mediante pilas para una instancia de 6 nodos. El grafo representativo del problema es el observado en la figura 1. En el paso (a), la pila de elementos E es rellenada con los primeros nodos que conforman el problema desde 1 hasta el *limite* calculado desde el nivel 0 y la pila

Algoritmo 2: Algoritmo de Ramificación y Acotamiento exploración del árbol de búsqueda con pilas.

Entrada: Un grafo $G = (V, E)$, el número de nodos $n = |V|$.
Salida: El ordenamiento óptimo φ^* para G.

$\varphi^* \leftarrow \emptyset$
$ub \leftarrow$ inicializacion-ub(G, n)
$B \leftarrow \emptyset$
$nivel \leftarrow 0$
$Cbn_0 \leftarrow 0$
$limite \leftarrow (n - \lfloor n/2 \rfloor + nivel + 1)$
for $i \leftarrow 1$ **to** $limite$ **do**
  $Elementos.push(i)$
  $Niveles.push(0)$
end for
while $Elementos \neq \emptyset$ **do**
  $v \leftarrow Elemento.pop()$
  $nivel \leftarrow Niveles.pop()$
  if $\left|N(v) \setminus B\right| > (\lfloor n/2 \rfloor - |B|)$ **then**
    $Cbn_{nivel} \leftarrow Cbn_{nivel-1} + 1$
  else
    $Cbn_{nivel} \leftarrow Cbn_{nivel-1}$
  end if
  if $Cbn_{nivel} < ub$ **then**
    $B_{nivel} \leftarrow v$
    if $|B| = \lfloor n/2 \rfloor$ **then**
      $lb \leftarrow BV(B)$
      if $lb < ub$ **then**
        $\varphi^* = B$
        $ub \leftarrow lb$
      end if
    else
      $limite \leftarrow (n - \lfloor n/2 \rfloor + nivel + 1)$
      for $i \leftarrow (v + 1)$ **to** $limite$ **do**
        $Elementos.push(i)$
        $Niveles.push(nivel + 1)$
      end for
    end if
  end if
end while
return φ^*

N se rellena con el nivel correspondiente a cada uno de los nodos. El valor del ub se establece con el valor de 6 equivalente al número de nodos pertenecientes al problema al no provenir de una solución inicial. El vector B inicialmente se encuentra vacío. En el paso (b), se extrae el elemento 1 de la pila y se coloca en la variable e así como el nivel 1 y se coloca en la variable n. La actualización de B

se realiza ingresando en la posición n del vector B el elemento e, representando la solución parcial actual. El cálculo de la contribución se realiza mediante la ecuación 2 y se coloca en el vector Cbn en el nivel actual (Cbn_n). En el paso (c), ya que el nivel de la contribución actual es inferior al ub y no se ha alcanzado una hoja, se procede a ramificar, por lo cual, se agrega el conjunto de hijos a la pila E obtenido mediante la ecuación 3 y su nivel correspondiente ($nivel + 1$) en la pila N. En el paso (d), se realiza la extracción de los elementos siguientes de las pilas y se realiza la actualización tanto del vector B como de la contribución. Como la contribución es inferior al ub y aún no se alcanza una hoja, se realiza una nueva ramificación. En el paso (e), se realiza una nueva extracción, actualización de B y cálculo de la contribución. En este paso la contribución es inferior que el ub y es alcanzada una hoja. En el paso (f) se realiza el cálculo de la función objetivo y se compara con el ub. Ya que el valor de la función objetivo resultó tener un valor inferior al ub, se realiza la actualización del ub con el nuevo valor obtenido y se procede a guardar la solución actual. En el paso (g), se realiza la extracción de los siguientes elementos de las pilas, la actualización del vector B y el cálculo de la contribución del nivel actual. En este paso, ya que la contribución tiene el mismo valor que el ub, se procede a podar el nodo actual, omitiendo la ramificación. En los pasos siguientes, se puede observar que los cálculos de las contribuciones siempre igualan (o superan) el valor del ub, por lo que se procede a podar cada nodo siguiente. Este proceso se realiza hasta que la pila E se queda vacía.

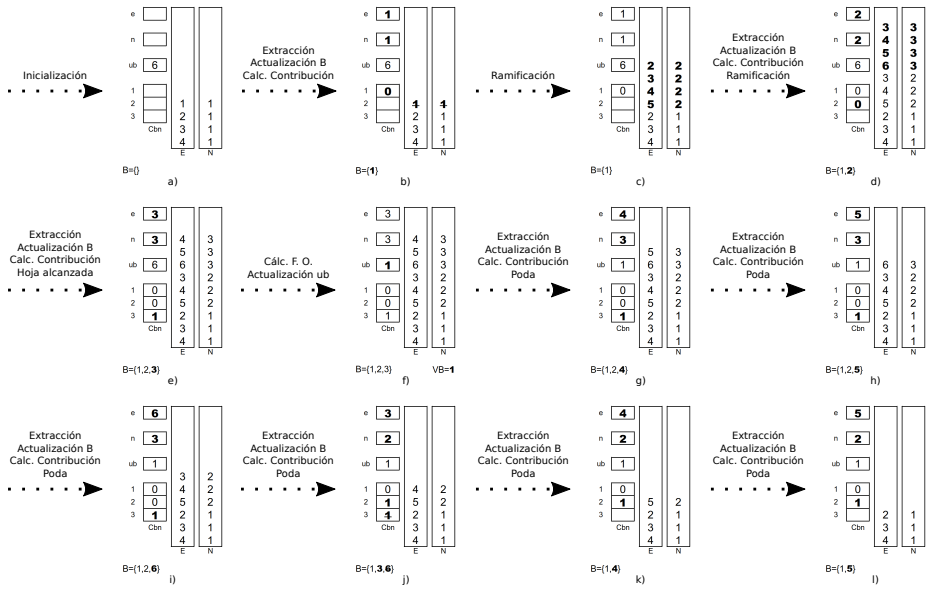


Fig. 3. Algoritmo de Ramificación y Acotamiento (BBVBP).

5. Experimentos computacionales

Los experimentos computacionales consisten en dos pruebas. En la primera, la calidad de las heurísticas (H1, H2) es medida en base al porcentaje del número de óptimos logrados y el porcentaje de proximidad a la optimalidad utilizando instancias con óptimo conocido por construcción. En la segunda prueba, se compara la eficiencia del algoritmo propuesto (BBVBP) contra el algoritmo de ramificación y acotamiento perteneciente al estado del arte (BBVBMP) mediante el tiempo de procesamiento acumulado, utilizando el benchmark utilizado en el estado del arte. Nuestra propuesta utiliza la mejor heurística de la experimentación anterior. La segunda prueba es una réplica de la experimentación reportada en [8].

La experimentación se realizó sobre un equipo con Procesador Intel Core i5-7200U a 2.50GHz Dual Core y 7.7GB de RAM. Los algoritmos fueron implementados en C++ y compilados con G++ 7.3.0. Se utilizó un total de 108 instancias del benchmark VSPBLIB [5]. De este benchmark se seleccionaron algunas instancias de cada conjunto: 1) Small con 84 instancias; 2) Grid con 5 instancias; 3) Tree con 15 instancias; 4) Harwell-Boeing (HB) con 4 instancias.

5.1. Calidad de las heurísticas

La calidad de las heurísticas (H1, H2) es medida en base al porcentaje del número de óptimos logrados y el porcentaje de proximidad al óptimo conocido por construcción. La tabla 1 muestra el porcentaje de valores óptimos logrados (columnas 2 y 3) y el porcentaje de proximidad al óptimo o porcentaje de *gap* (columnas 4 y 5) de cada heurística por cada tipo de instancia. La primera columna indica el tipo de instancia; entre paréntesis se indica el número de instancias. La última fila contiene los porcentajes resultantes. Los mejores resultados se encuentran con fuente negrita.

De la tabla 1 podemos observar que H2 logra un porcentaje de 80.95 % de valores óptimos encontrados y un menor porcentaje de proximidad al óptimo o porcentaje de *gap* de 10.78 %. Podemos observar que las dos heurísticas logran todos los valores óptimos en las instancias de tipo Grid y Tree. Como H2 logró mejores resultados, es seleccionada para inicializar el *ub* del BBVBP.

5.2. Eficiencia de los algoritmos de ramificación y acotamiento

La experimentación reportada en [8] establece un tiempo límite de 5 minutos por instancia de tipo Small, Grid y Tree, así como un tiempo de 1 hora por instancia de tipo Harwell-Boeing (HB). La tabla 2 muestra los resultados de la experimentación. Se midió la calidad de los algoritmos de ramificación y acotamiento en términos del porcentaje del número de los valores óptimos logrados y la eficiencia mediante el tiempo requerido para alcanzarlos.

Los resultados de la tabla 2 muestra la eficiencia en la búsqueda de las soluciones óptimas en el tiempo límite dado. Ambos algoritmos encontraron el mismo número de soluciones óptimas, pero el BBVBP logró una reducción del tiempo de cálculo del 84.35 % con respecto al BBVBMP.

Tabla 1. Porcentaje promedio de valores óptimos alcanzados y distancia hacia el óptimo (*gap*).

	% Optimal		% Gap	
Tipo de instancia	H1	H2	H1	H2
Grid (5)	100	100	0	0
Small (84)	25	42.86	49.23	32.34
Tree (15)	100	100	0	0
Promedio	75.00	**80.95**	16.41	**10.78**

Tabla 2. Porcentaje de valores óptimos y tiempo de los algoritmos de ramificación y acotamiento.

	% Optimal		Tiempo (sec)	
Tipo de instancia	BBVBMP	BBVBP	BBVBMP	BBVBP
Grid (5)	60	60	737.9	602.1
Small (84)	100	100	1680.5	16.2
Tree (15)	100	100	313.2	3.72
Harwell-Boeing (4)	25	25	10810.4	10801.0
Promedio	71.2	71.2	3385.5	**2855.7**

6. Conclusiones y trabajo futuro

En este trabajo se abordó el Problema de la Bisección de Vértices. Se propone un nuevo algoritmo de ramificación y acotamiento (BBVBP) y una nueva heurística constructiva voraz (H2) para la inicialización del límite superior (ub). Los resultados muestran que el algoritmo propuesto supera en eficiencia al algoritmo de ramificación y acotamiento del estado del arte (BBVBMP), reduciendo el tiempo de ejecución en un 18 %, 99 %, 98 %, 0.08 % sobre los conjuntos de instancias Grid, Small, Tree y Harwell-Boeing respectivamente. La reducción global que se logra sobre todos los conjuntos de instancias es de un 15 %. Como ambos algoritmos utilizan un proceso de exploración de complejidad similar, la diferencia en eficiencia se deriva de la complejidad del cálculo de la cota inferior. Para el algoritmo del estado del arte, dicho cálculo es de complejidad temporal $O(n^2)$ y para el algoritmo propuesto es $O(n)$. La heurística propuesta supera ligeramente en calidad a la heurística utilizada en el estado del arte mientras que la eficiencia es similar al tener ambas estrategias una complejidad $O(n^2)$. A fin de resolver el VBP, se propone una nueva estrategia de exploración del árbol de búsqueda. Esta estrategia únicamente explora soluciones factibles en el espacio combinatorio sin repeticiones, reduciendo así el número de nodos explorados significativamente. El algoritmo propuesto incluye un nuevo cálculo de la cota inferior para las soluciones parciales.

Como trabajo futuro se considera diseñar un nuevo mecanismo de cálculo de la cota inferior para mejorar la eficiencia del algoritmo de ramificación y

acotamiento mediante la reducción de los nodos ramificados durante el proceso de exploración del árbol de soluciones. Adicionalmente se implementará la exploración del árbol de soluciones utilizando una estructura de datos diferente a la utilizada en este trabajo.

Agradecimientos. Los autores manifiestan su agradecimiento a Jain Pallavi por haber apoyado con el código fuente de su algoritmo. Al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Tecnológico Nacional de México por el apoyo financiero brindado para el desarrollo de este proyecto. Al Laboratorio Nacional de Tecnologías de la Información (LANTI) por el acceso a los recursos de cómputo de alto desempeño bajo su resguardo.

Referencias

1. Böckenhauer, H.J.: Two open problems in communication in edge-disjoint paths modes. Acta Mathematica et Informatica Universitatis Ostraviensis 7(1), 109–117 (1999)
2. Brandes, U., Fleischer, D.: Vertex Bisection is Hard, too. Journal of Graph Algorithms and Applications 13(2), 119–131 (2009)
3. Delling, D., Fleischman, D., Goldberg, A.V., Razenshteyn, I., Werneck, R.F.: An exact combinatorial algorithm for minimum graph bisection. Mathematical Programming 153(2), 417–458 (nov 2015)
4. Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Computing Surveys (CSUR) 34(3), 313–356 (2002)
5. Duarte, A., Escudero, F., Martí, R., Mladenović, N., Pantrigo, J.J., Sánchez-Oro, J.: Vertex separation problem (2012), `http://www.optsicom.es/vsp/#instances`
6. Fraire, H., Terán-Villanueva, J.D., García, N.C., Barbosa, J.J.G., del Angel, E.R., Rojas, Y.G.: Exact methods for the vertex bisection problem. In: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, pp. 567–577. Springer (2014)
7. Huacuja, H.F., Castillo-García, N., Rangel, R.A.P., Flores, J.A.M., Barbosa, J.J.G., Valadez, J.M.C.: Two New Exact Methods for the Vertex Separation Problem. International Journal of Combinatorial Optimization Problems and Informatics 6(1), 31 (2015)
8. Jain, P., Saran, G., Srivastava, K.: Branch and bound algorithm for vertex bisection minimization problem. In: Advanced Computing and Communication Technologies, pp. 17–23. Springer (2016)
9. Kleniati, P.M., Adjiman, C.S.: A generalization of the Branch-and-Sandwich algorithm: From continuous to mixed-integer nonlinear bilevel problems. Computers and Chemical Engineering 72, 373–386 (2015)
10. Lalami, M.E., El-Baz, D.: GPU Implementation of the Branch and Bound Method for Knapsack Problems. In: 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum. pp. 1769–1777. IEEE (may 2012)
11. Leroy, R.: Parallel Branch-and-Bound revisited for solving permutation combinatorial optimization problems on multi-core processors and coprocessors. Ph.D. thesis, Université Lille 1 (2015)

ISSN 1870-4069

Navegación en interiores utilizando la tecnología beacons

Alicia Martínez-Rebollar[1], Luis Antonio López-García[1], Hugo Estrada-Esquivel[2], Manuel Mejía Lavalle[1], Javier Ortiz Hernández[1]

[1] Tecnológico Nacional de México/CENIDET, Cuernavaca, Morelos, México
[2] Consejo Nacional de Ciencia y Tecnología, México
{luis.lopez17ce@cenidet.edu.mx, amartinez, mlavalle,ortiz}@cenidet.edu.mx, hugo.estrada@conacyt.mx

Resumen. El internet de las cosas (IoT por sus siglas en inglés) está impulsando la generación de edificios inteligentes. Estos edificios tienen como requisito principal la navegación de interiores. Sin embargo, la tecnología GPS, la cual es utilizada para llevar a cabo el posicionamiento, no es posible utilizarla dentro de un edificio debido a que las señales satelitales no viajan bien a través de materiales sólidos. De esta manera, se han propuesto sensores, dispositivos Bluetooth, o RFID que se utilizan en los edificios inteligentes. En este artículo presentamos una propuesta de navegación de interiores, la cual utiliza la tecnología beacons y los teléfonos inteligentes. Nuestra aplicación de software obtiene información del contexto y genera la mejor ruta para llegar al destino dentro de un edificio inteligente. Se utilizó el algoritmo Dijkstra para procesar toda la información. De ahí que nuestra propuesta tiene como objetivo combinar diferentes tecnologías, y adaptar algoritmos desarrollados a la navegación de interiores.

Palabras clave: navegación de interiores, tecnología beacons, edificio inteligente, internet de las cosas.

Indoor Navigation using Beacon Technology

Abstract. The Internet of Things (IoT) is promoting the generation of smart buildings. These buildings have as main requirement the navigation of interiors. However, GPS technology, which is used to carry out positioning, it cannot be used within a building because satellite signals do not travel well through solid materials. In this way, sensors, Bluetooth devices, or RFID that are used in smart buildings have been proposed. In this paper, we present an interior navigation proposal, which uses beacons technology and smartphones. Our software application obtains information from the context and generates the best route to reach the destination within an intelligent building. The Dijkstra algorithm was used to process all the information. Hence, our proposal aims to combine different technologies, and adapt developed algorithms to indoor navigation.

Keywords: navigation indoor, beacons technology, smart building, internet of things.

1. Introducción

En las últimas décadas, la disponibilidad de dispositivos móviles con el Sistema de Posicionamiento Global (GPS) ha impulsado el desarrollo de servicios y aplicaciones basados en la ubicación. Entre los más populares están las aplicaciones de navegación que proporcionan contextos y son instrucciones para guiar a sus usuarios. Sin embargo, dado que el GPS requiere una línea de visión sin obstrucciones entre el receptor de GPS y múltiples satélites, su disponibilidad en ambientes interiores típicos está severamente limitada. Como resultado, la mayoría de las aplicaciones de navegación de amplio espectro se centran en escenarios al aire libre o proporcionan solo soporte para entornos interiores [1].

Para proporcionar una alternativa al GPS, los investigadores y profesionales han desarrollado diversos enfoques para la localización en interiores que aprovechan la visión, el sonido, la tecnología de radiación infrarroja o radiofrecuencia, por nombrar solo algunos. Al usar estos sistemas, muestran claramente que es posible proporcionar información precisa sobre la ubicación a un costo razonable. Sin embargo, la mayoría de las aplicaciones en este campo se enfocan en escenarios interiores aislados. Como consecuencia, sólo brindan soporte en un ambiente interior particular, como un centro comercial o un edificio universitario, por lo tanto, no pueden proporcionar una verdadera experiencia de navegación de extremo a extremo [2].

Otro aspecto el cual se debe de tomar en cuenta es el tiempo, es decir vivimos constantemente en un proceso de cambio. Según la ONU el 54 por ciento de la población mundial actual reside en áreas urbanas y se prevé que para 2050 llegará al 66 por ciento [3]. Actualmente se está viviendo la transformación de las ciudades en ciudades inteligentes. Para preparar un proyecto de ciudades inteligentes se requiere de diferentes etapas, retos, un diagnóstico previo de los problemas de la administración pública y de la ciudad y una evaluación de las oportunidades. El diseño de una solución inteligente requiere identificar los recursos tecnológicos que se necesitan para el desarrollo de proyectos.

Una ciudad se vuelve más eficiente en la medida en que es capaz de obtener datos generados en el ambiente, en las infraestructuras instaladas por prestadores de servicios y ciudadanos en los edificios y en las calles para luego procesar esos datos y transformarlos en información que facilite la toma de decisiones capaces de mitigar, organizar, anticipar o prever innumerables retos urbanos. Para captar esos datos, en algunos casos es necesario instalar sensores y cámaras de video en la infraestructura física de la ciudad, conectarlos entre sí en una red de comunicación de datos, para luego usar esos datos enviados en tiempo real y apoyar de esa manera a la toma de decisiones [3].

Cisco estima que el universo de “Internet de las Cosas” (IoT) en 2020 contará con 50.000 millones de dispositivos [4].

La navegación de interiores es una gran tendencia que se encuentra ya vigente en las grandes ciudades. Se ha dado un gran salto de la navegación en exteriores a la navegación en interiores, teniendo la necesidad de localizar lugares y objetos en espacios cerrados.

De esta manera, nuestra propuesta tiene como objetivo combinar diferentes tecnologías como: beacons y teléfonos inteligentes para la navegación de interiores, extendiendo el algoritmo Dijkstra para llevar a cabo la navegación.

Esté artículo se encuentra organizado de la siguiente manera: en la Sección 2 se presentan los fundamentos teóricos y trabajos relacionados. En la sección 3 se detalla nuestra propuesta de navegación en interiores y exteriores. La sección 4 muestra la manera en cómo estamos probando nuestra propuesta de navegación. Finalmente, la sección 5 se presentan las conclusiones y trabajos futuros de nuestra investigación.

2. Fundamentos teóricos y trabajos relacionados

En esta sección se detallan los fundamentos teóricos y trabajos relacionados en navegación de interiores y exteriores, utilizando diferentes tecnologías.

2.1. Marco teórico

Algunos conceptos relevantes en este trabajo de investigación se presentan a continuación.

Sistema de Posicionamiento Global. El Sistema de Posicionamiento Global o GPS es un sistema global de navegación por satélite que permite determinar en todo el mundo la posición de un objeto, una persona, un vehículo o una nave. Se puede alcanzar una precisión hasta de centímetros, usando el GPS diferencial, pero lo habitual son unos pocos metros.

El GPS funciona mediante una red de 27 satélites (24 operativos y 3 de respaldo) en órbita a 20.200 km sobre el globo terráqueo, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar una posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del Sistema de Posicionamiento y calcula el retraso de las señales; es decir, la distancia al satélite. Por "triangulación" los tres satélites calculan la posición en que el GPS se encuentra. La triangulación en el caso del GPS se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición [5].

Navegación en interiores. La navegación en interior como la palabra lo indica navegar dentro de los edificios. Debido a que la recepción del GPS normalmente no existe dentro de los edificios, aquí se usan otras tecnologías de posicionamiento cuando se desea el posicionamiento automático. A menudo se utilizan Wi-Fi o beacons (Bluetooth Low Energy, BLE) para crear un llamado “GPS interior”. La mayoría de las aplicaciones requieren una funcionalidad de "enrutamiento interior" que guía a las personas de forma precisa a través de un edificio utilizando una aplicación de navegación interior y de esta manera determina automáticamente su posición, muy similar a los sistemas de navegación que usamos en nuestros automóviles [6].

Beacons. En esta propuesta de investigación se utiliza la tecnología de beacons, como medio de localización de puntos de interés en la navegación de interiores. Los beacons son transmisores diseñados para usar señales de transmisión de bluetooth de baja energía para atraer la atención a una ubicación específica. Estos transmisores son muy pequeños y se pueden colocar en paredes o pilares de lugares como tiendas, aeropuertos, restaurantes, apartamentos, oficinas, etc.

La tecnología de beacons o baliza se usa principalmente para detectar la posición de su identificador en espacios interiores donde es imposible ubicar los dispositivos con la ayuda de señales de GPS [7].

Los siguientes tres conceptos (Identificador único universal, major, minor) son atributos que tienen todos los beacons que utilizan el protocolo ibeacon, estos atributos fueron utilizados para hacer una primera ubicación del usuario sin la necesidad de utilizar las credenciales que la plataforma Estimote genera para el uso de los beacons.

Identificador único universal. Un UUID es un número de 128 bits utilizado para identificar de manera única algún objeto o entidad en Internet. Dependiendo de los mecanismos específicos utilizados, se garantiza que un UUID es diferente o, al menos, es muy probable que sea diferente de cualquier otro UUID [8].

Major. Los valores principales son valores enteros sin signo entre 1 y 65535 los cuales están destinados a identificar y distinguir un grupo; por ejemplo, a todos los beacons de un determinado piso o habitación de su sede se les podría asignar un valor principal único [9].

Minor. Los valores principales son valores enteros sin signo entre 1 y 65535 los cuales están destinados a identificar y distinguir a un individuo, por ejemplo, distinguir beacons específicas dentro de un grupo de beacons asignadas a un valor Mayor [9].

2.2. Trabajos relacionados

Esté trabajo de investigación [10] describe un método para determinar la ubicación de un dispositivo móvil, como una computadora portátil o un teléfono móvil, en un ambiente interior usando beacons bluetooth. Una ventaja importante de este método es que permite que el dispositivo móvil determine su ubicación mientras permanece de manera anónima. En un despliegue de beacons en un edificio, una de las tareas importantes es el diseño de la colocación de beacons.

La propagación de señales en entornos interiores son complejos, ya que son afectadas por factores como la estructura del edificio, conductos, propiedades de transmisión y reflexión de los materiales de construcción, muebles y la interferencia de otros dispositivos. Por lo tanto, el área desde donde se ve una beacon es muy irregular y no muy aproximado por modelos simples como los elipsoides.

Otro trabajo de investigación es [11], el cual presenta un marco de combinación de sensores de teléfonos inteligentes e iBeacons para una localización interior precisa. El Pedestrian Dead Reckoning (PDR) se puede aplicar para la localización mediante sensores de teléfonos inteligentes, pero desafortunadamente podría ir a la deriva por la poca distancia que abarca. Por lo tanto, iBeacons aprovecha para corregir esta desventaja del enfoque PDR. Los autores realizaron un filtro de señales para la

corrección de la poca distancia que abarca y los experimentos que son presentados en este documento muestran una mejora significativa de la precisión de localización con la dispersión del despliegue de iBeacon en un área de 47.3m x 15.9m. Además, dado que la precisión de la estimación del punto inicial es vital para el enfoque de PDR, evaluaron el desempeño de su enfoque propuesto con respecto al error de estimación del punto inicial. Los resultados subrayan la solidez de su enfoque propuesto para el error de estimación del punto inicial**.**

El siguiente trabajo de investigación presenta el desarrollo de un Sistema de Navegación en Interiores usando la tecnología NFC [12]. NFC es una tecnología de comunicación inalámbrica bidireccional de corto alcance. La comunicación se produce entre dos dispositivos cercanos a pocos centímetros. NFC en interiores es muy simple y fácil de usar, el usuario sólo necesita llevar y usar un dispositivo móvil habilitado para NFC. La aplicación de navegación interior debe ser OTA (Over the Air) precargada en la tarjeta inteligente. De lo que el usuario necesita simplemente toque la etiqueta URL (etiqueta NFC), que contiene la URL de la información del mapa interior justo antes de ingresar al edificio o al área. El mapa en el sitio web es OTA descargado al dispositivo móvil del usuario desde MapServer (es decir, un servidor web que contiene la información del mapa). La aplicación de navegación en el interior del teléfono móvil se inicia automáticamente y luego utiliza está información del mapa. A medida que el usuario selecciona el punto de destino de su viaje dentro del edificio, la aplicación de navegación interior proporciona la ruta óptima hacia su destino.

Tabla 1. Comparativa de Google maps indoor y nuestra propuesta: NavIndoor.

	Google maps indoor	NavIndoor
Tecnología	GPS	Bluetooth
Visualización	Mapas	Indicaciones
Intermediarios	Si	No
Disponibilidad	Sólo está disponible para algunas ciudades	Para todas las ciudades
Tipo de público	Centros comerciales y socios existentes	Todo el público

El siguiente trabajo de investigación presenta el desarrollo de un sistema de navegación en interiores usando la tecnología GPS [13], el cual fue desarrollado por la empresa Google, este sistema de navegación está incorporado a la aplicación móvil de Google maps que ya es actualmente utilizada por la mayoría de la gente para buscar lugares o nos ayude a llegar a un destino. Esta aplicación funciona con solo acercarse al edificio, una vez dado el primer paso se muestra un mapa del edificio donde se puede observar aquellos lugares de interés, navegar a través del edificio. Sin embargo, la aplicación solo se encuentra enfocada a una específica parte de la población como socios y centros comerciales. En la tabla 1 se muestran las características de la aplicación de google maps indoor y la propuesta en este trabajo de investigación.

El siguiente trabajo de investigación presenta el desarrollo de un sistema de navegación en interiores usando la tecnología GPS [14], este sistema de navegación en interiores es llamado FSKTM 2D basado en Android usando el algoritmo A*, es una aplicación para Android para los estudiantes y visitantes de la Universidad Tun Hussein

Onn Malasia. Esta aplicación de navegación se utiliza para encontrar la ubicación del aula por parte de los estudiantes. Permite a los estudiantes elegir su destino (el aula) y proporcionar el mapa de ruta a los estudiantes. Calcula el camino más corto para los estudiantes y los lleva a su destino.

3. Propuesta de navegación de interiores: navIndoor

La navegación en interiores es el siguiente paso en la localización después del GPS. Este tipo de navegación se ocupa de encontrar lugares y objetos en espacios cerrados como edificios, en donde las señales satelitales GPS no llegan con precisión.

Se utilizan en este tipo de navegación tecnologías como Wi-Fi o Balizas (Bluetooth Low Energy, BLE) para crear el llamado "GPS de interior". Este tipo de navegación interior permite guiar a las personas a navegar en edificios y determinar automáticamente su posición, muy similar a los sistemas de navegación que se utiliza en los automóviles.

En este artículo se presenta nuestra propuesta de investigación sobre navegación de interiores, en la cual se presenta nuestra metodología de solución que está formada de cuatro fases: definición de la estructura, desarrollo del algoritmo, desarrollo de los módulos de navegación y pruebas (Figura 2), la información del contexto que va recolectándose en cada momento es almacenada en la base de datos.

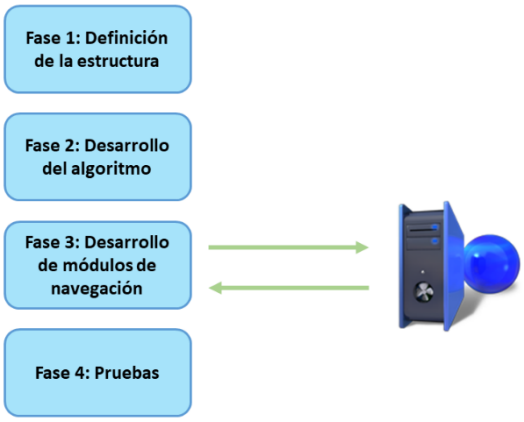


Fig. 1. Metodología de solución propuesta.

3.1. Definición de estructura

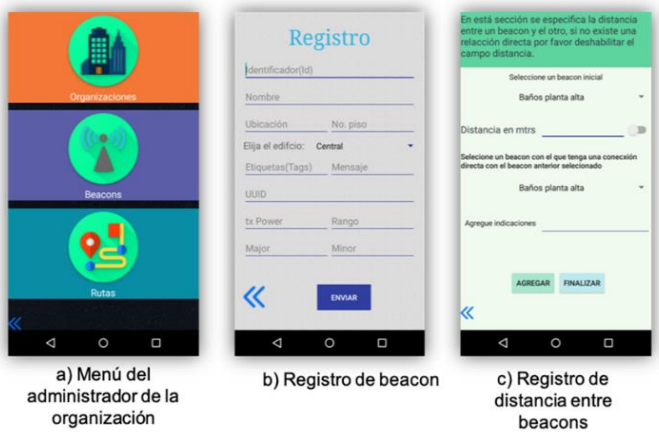
La primera fase de nuestra metodología de solución es la definición de la estructura, la cual tiene como objetivo registrar la organización en donde el sistema de navegación va a implementarse, así como los edificios de esta organización, el registro de los beacons que serán colocados en la organización y las rutas de navegación en cada uno de los edificios de la organización.

Para llevar a cabo la definición de la estructura fue necesario realizar un prototipo de software el cual se le denominó navIndoor, la cual es una aplicación móvil de Internet

de las cosas para dispositivos móviles Android. La aplicación móvil tiene como objetivo gestionar los recursos que se necesitan para hacer una navegación en interiores.

La aplicación está destinada para ser usada por un usuario de la organización el cual le llamamos administrador de la organización, este usuario debe de tener los conocimientos del tipo de edificación en donde se hará la navegación de interiores, además de la información de los beacons que se utilizarán dentro de la organización.

La figura 3 muestra el menú de la aplicación navIndoor, la cual está formada de tres secciones: a)organizaciones, esta sección tiene como objetivo dar de alta, modificar o eliminar una organización, o un edificio dentro de la organización; b) gestión de beacons, esta sección permite al administrador de la organización capturar la información de cada uno de los beacons, estos son: valores uuid, major, minor, además de las credenciales que la compañía de los beacons brinda para poder ser utilizados; c) Rutas, en esta sección se registra las distancias existentes entre un beacon y otro, con esta información el algoritmo Dijkstra puede hacer el cálculo de la ruta óptima.



a) Menú del administrador de la organización

b) Registro de beacon

c) Registro de distancia entre beacons

Fig. 2 Aplicación Móvil de IoT: navIndoor.

3.2. Desarrollo de algoritmos de navegación

La navegación de interiores que se propone en este artículo utiliza un grafo, el cual es una pareja de conjuntos $G = (V, A)$, donde G es el grafo, V es el conjunto de vértices o nodos, y A es el conjunto de aristas que relacionan a los nodos, este último es un conjunto de pares de la forma (u, v) tal que $u, v \varepsilon V$[5]. El algoritmo se genera por medio de una matriz de nxn en la cual se establecen los valores de las distancias reales entre los beacons a usar, si es que existe un enlace directo, es decir que no se necesita pasar por un beacon intermedio, si no existe dicho enlace se establece un valor por default, esta matriz es generada por el administrador de la organización en la aplicación navIndoor.

Los algoritmos desarrollados para llevar a cabo la nevegación fueron un algoritmo de posicionamiento y un algoritmo de generación de rutas. Estos algoritmos juegan un rol muy importante en nuestra propuesta ya que mediante ellos se puedieron hacer las actividades mas importantes:

– Establecer la ubicación de un usuario,

– Generar una ruta a partir de un punto inicial a un punto final.

Algoritmo de posicionamiento. Esté algoritmo tiene como objetivo obtener la ubicación de un usuario. Este algoritmo es utilizado para identificar el lugar donde se encuentra el usuario, es decir la ubicación del usuario aproximadamente, la cual nos ayudara saber cuál es su punto de partida, cual ha sido los lugares que ha estado, cuándo ha llegado a su destino.

Esta tarea se realiza con uso del SDK de los beacons Estimote, el cual nos proporciona una función de monitoreo que constante lanza un evento cuando algún usuario con su celular y la aplicación en ejecución entra dentro de la región de un beacon así podemos ejecutar cualquier actividad, como mostrar un mensaje, mostrar alguna imagen, hacer una consulta a la base de datos, etc.

Algoritmo de generación de rutas. esté algoritmo nos ayuda a poder formar las rutas a partir de la información generada por el administrador de la organización. Para poder generar una ruta es necesario tener un punto de inicio, un punto final y que exista al menos una distancia entre un beacon, es decir no debe de existir algún beacon que se encuentre aislado.

Existe una variedad de algoritmos que trabajan con rutas como puede ser: Kruskaly[15], Breadth first search[16], Bellman-Ford[17] los cuales trabajan de maneras diferentes por ejemplo Kruskaly trabaja con árboles de tipo de expansión mínima, Breadth first search y Bellman-Ford trabajan con grafos, pero las distancias (pesos en aristas) en caso de Breadth first search son distancias entre 0 y 1 y Bellman-Ford puede trabajar con cualquier tipo de distancias incluyendo negativas, sin embargo lo que necesitamos es que la información brindada por el administrador de la organización pase casi directamente y no hay un proceso de tratamiento de datos es por ello que se decidió utilizar el algoritmo Dijkstra además de trabajar con la ruta óptima. El algoritmo Dijkstra tuvo que ser modificado para poder hacer uso de él, de los cambios que se realizaron se agregó el parámetro del destino, se adaptaron los identificadores de los beacons como los nodos del grafo, se hizo con un número de nodos dinámico, se obtiene solo la ruta a la cual lleva al destino y finalmente se modificó para que trabajara con números de tipo decimal.

3.3. Desarrollo de módulos de navegación

El prototipo desarrollado en la tercera fase de la metodología propuesta se refiere al prototipo de navegación por parte de los usuarios, el cual está formado por cinco módulos los cuales fueron implementados en una aplicación móvil del sistema operativo Android. Este prototipo ha sido denominado como navIndoor:

– Registro de usuario,
– Inicio de sesión,
– Búsqueda de lugares dentro del edificio,
– Generación de ruta,
– Detección de ubicación.

navIndoor es una aplicación móvil de IoT para dispositivos móviles Android. Los usuarios deben registrarse para poder hacer uso de la aplicación desarrollada y poder hacer inicio de sesión. Una vez que el usuario se encuentra registrado e inicia sesión,

entonces puede hacer búsquedas de los lugares dentro de un edificio inteligente, mostrándose la ubicación del usuario para posteriormente generar la ruta hacia del lugar al que el usuario desea ir.

Además, la aplicación navIndoor tiene la capacidad de poder detectar cuando no se ha seguido las instrucciones correctamente. La aplicación manda un mensaje con la descripción *"Te has desviado de la ruta"*, posteriormente la aplicación automáticamente recalcula la ruta a partir de la posición actual del usuario. Otra funcionalidad que se incorporó a la aplicación es detectar si el usuario ha cambiado de edificio habiendo iniciado una ruta en un edificio diferente.

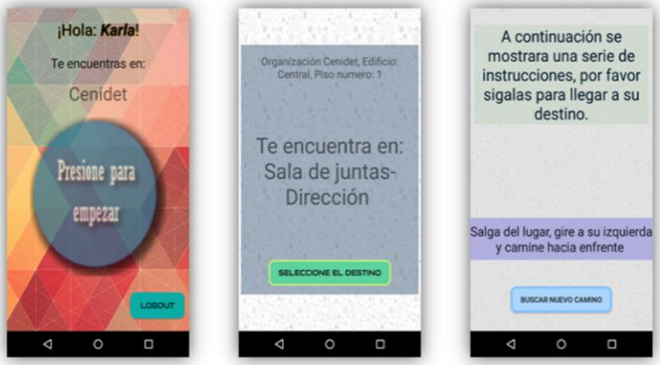
La Fig. 4 muestra una vista de la aplicación. La aplicación móvil tiene como objetivo guiar al usuario mediante una serie de instrucciones que la aplicación muestra y reproduce para llegar a un destino de interés en interiores.

En esta aplicación se hace uso del algoritmo de ubicación y del algoritmo Dijkstra, la aplicación detecta automáticamente en la organización que se encuentra, para posteriormente dar una ubicación más precisa mostrando en que edificio, en qué piso y en qué oficina, salón o baño se encuentra el usuario.

Posteriormente se busca el lugar donde se requiera ir, con esto se obtiene el punto inicial y punto final para poder ser enviados al algoritmo Dijkstra y obtener la ruta óptima, esta ruta se encuentra formada por al menos dos nodos los cuales tienen instrucciones que fueron dadas de altas por el administrador de la organización, finalmente se recorren todos estos nodos hasta llegar al final y haber concluido la navegación.

3.4. Pruebas

En esta se fase se enfocó en hacer pruebas al algoritmo Dijkstra para comprobar si se podía acoplar a las necesidades que teníamos. Se hicieron pruebas con la aplicación navIndoor con algoritmo Dijkstra y la interacción con los beacons. Se establecieron 3 pruebas con 3 diferentes grafos: 15 nodos; 20 nodos, y 30 nodos.



a) Detección automática de la organización. b) Detección automática detallada. c) Recomendaciones a seguir

Fig. 3 Aplicación Móvil de IoT: navIndoor.

En el grafo de 15 nodos se hicieron 8 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

En el grafo de 20 nodos se hicieron 10 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

En el grafo de 30 nodos se hicieron 15 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

En total se hicieron 33 recorridos desde los más sencillos recorriendo un solo nodo, hasta aquellos en donde se tuvieron que evaluar las distancias de los múltiples recorridos que existen para llegar al mismo destino.

En cuanto las pruebas de la aplicación navIndoor se hicieron tres diferentes pruebas con tres grafos elaborados a partir de un edifico de 2 pisos los grafos resultantes fueron los siguientes: 6 nodos; 4 nodos, y 10 nodos

En el grafo de 6 nodos se hicieron 5 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

En el grafo de 4 nodos se hicieron 3 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

En el grafo de 10 nodos se hicieron 2 recorridos teniendo una efectividad del 100% dando la ruta óptima en todos los recorridos.

Con los tres diferentes grafos la aplicación fue capaz de dar asesoramiento para llegar al destino seleccionado, además que siempre sugirió la ruta óptima.

4. Conclusiones

En este trabajo de investigación se presentó un sistema de navegación en interiores el cual permite dar una serie de indicaciones que el usuario tiene que seguir para llegar a su destino, las instrucciones que se dan al usuario permite recorrer la ruta óptima a partir de un punto inicial (ubicación del usuario) a un punto destino. Se demostró que la tecnológica beacon y el algoritmo Dijkstra es una opción viable para hacer navegación en interiores por los buenos resultados que se dieron. Se desarrolló una metodología de solución genérica, es decir que cualquiera que siga dicha metodología podrá hacer una navegación en interiores básica. La combinación de tecnologías como beacons, GPS y wifi podrían producir una herramienta muy poderosa de navegación en interiores haciendo más precisa la navegación, además de expandir dicha navegación a lugares abiertos dentro de las organizaciones sin tener que hacer un gasto fuerte en compra de beacons o de repetidores de señal.

Como trabajos futuros, puede explorarse la opción de mandar notificaciones a los usuarios cuando se haga una navegación en interiores en una organización nueva, Otra función que le daría un gran impulso sería que se hiciera una cartografía y geolocalización de la organización con sus edificios para poder ser mostrada la ruta y la ubicación del usuario gráficamente. Por otra parte, sería útil generar una versión web de la aplicación navIndoor para el administrador de la organización con el fin de brindarle un mejor espacio de trabajo.

Agradecimientos. Esta investigación ha sido financiada parcialmente por el proyecto SmartSDK, Num Ref. 272727, así como el proyecto TecNM/CENIDET, Num Ref. 5138.19-P.

Referencias

1. Wagner, S., Fet, N., Handte, M., Marrón, P.J. An Approach for Hybrid Indoor/Outdoor Navigation. En: International Conference on Intelligent Environments (IE), pp.36–43 (2017)
2. Bouskela, M., Casseb, M., Bassi, S., De Luca, C., Facchina, M. La ruta hacia las Smart Cities. Migrando de una gestión tradicional a la ciudad inteligente. Biblioteca Felipe Herrera Del Banco Interamericano de Desarrollo, pp. 454 (2016)
3. Cisco, https://newsroom.cisco.com/feature-content?articleId=1208342, último acceso: 02/05/2019
4. Giménez Rodríguez T.: Sistema de Posicionamiento Global (GPS): Sociedad Latinoamericana en Percepción Remota y Sistemas de Información Espacial., pp. 94 (2008)
5. infSOfT, https://www.infsoft.com/solutions/indoor-navigation, último acceso: 01/05/2019
6. Shende, P., Mehendarge, S., Chougule, S., Kulkarni, P., Hatwar, U.: Innovative ideas to improve shopping mall experience over E-commerce websites using beacon technology and data mining algorithms. En: International Conference on Circuit, Power and Computing Technologies, pp. 1–5 (2017)
7. What is UUID: https://searchmicroservices.techtarget.com/definition/UUID-Universal-Unique-Identifier/, último acceso: 20/12/2018.
8. Beacon Strategy Guide – UUID, Major, Minor: https://kontakt.io/blog/beacon-id-strategy-guide-quick-deployment/, último acceso: 20/12/2018.
9. Chawathe, S.: Beacon Placement for Indoor Localization using Bluetooth. En: 11th International IEEE Conference on Intelligent Transportation Systems, pp. 980–985 (2008)
10. Chen, Z., Zhu, Q., Jiang, H., Soh, Y.C.: Indoor localization using smartphone sensors and iBeacons. En: IEEE 10th Conference on Industrial Electronics and Applications, pp. 1723–1728 (2015)
11. Ozdenizci, B., Ok, K., Coskun, V., Aydin, M.N.: Development of an Indoor Navigation System Using NFC Technology. En: Fourth International Conference on Information and Computing, pp. 11–14 (2011)
12. Ingresa a los edificios con Mapas de interiores: https://www.google.com/intl/es-419_mx/maps/about/partners/indoormaps/, último acceso: 01/05/2019
13. Kasim, S.: Indoor Navigation Using A* Algorithm. En: Herawan T., Ghazali R., Nawi N., Deris M. (eds) Recent Advances on Soft Computing and Data Mining. SCDM 2016. Advances in Intelligent Systems and Computing, vol 549 (2017)
14. Kruskal's Minimum Spanning Tree Algorithm | Greedy Algo-2. https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/, último acceso: 20/12/2018
15. The breadth-first search algorithm: https://www.khanacademy.org/computing/computer-science/algorithms/breadth-first-search/a/the-breadth-first-search-algorithm/, último acceso: 20/12/2018
16. Goldberg, A., Radzik, T.A.: Heuristic improvement of the Bellman-Ford algorithm. Stanford Univ CA Dept. of Computer Science (1993)

ISSN 1870-4069

Adquisición de señales digitales utilizando teléfonos inteligentes y FIWARE

Alicia Martínez Rebollar[1], Leon Alberne Torres Restrepo[1], Fabio Alberto Vargas Agudelo[2], Javier Ortiz-Hernández[1]

[1] Tecnológico Nacional de México/CENIDET, Cuernavaca, México
[2] Tecnológico de Antioquia, Medellín, Colombia
{amartinez, leontorres16c,ortiz}@cenidet.edu.mx,
fvargas@tdea.edu.co

Resumen. El procesamiento de señales que provienen de diferentes fuentes físicas, generalmente son adquiridas a través de dispositivos y sensores especializados. Sin embargo, los teléfonos inteligentes son utilizados con más frecuencia para la adquisición de diferentes señales. En este trabajo se presenta una aplicación que permite la adquisición de las señales provenientes del acelerómetro, el giroscopio, el magnetómetro y el GPS de un teléfono inteligente con sistema operativo Android. La aplicación propuesta utiliza algunos componentes de la plataforma FIWARE para cubrir la etapa de monitoreo y visualización de los datos. Las pruebas realizadas mostraron que es factible utilizar nuestra aplicación como una herramienta para la adquisición y la visualización de señales provenientes de los sensores de un teléfono inteligente.

Palabras clave: adquisición de señales, FIWARE, teléfonos inteligentes.

Acquisition of Digital Signals using Smartphones and FIWARE

Abstract. The processing of signals that come from different physical sources, are usually acquired through specialized devices and sensors. However, smartphones are used more frequently for the acquisition of different signals. In this paper an application that allows the acquisition of signals from the accelerometer, the gyroscope, the magnetometer and the GPS of a smartphone with Android operating system is presented. The proposed application uses some components of the FIWARE platform to cover the monitoring and visualization stage of the data. The tests carried out showed that it is feasible to use our application as a tool for the acquisition and visualization of signals coming from the sensors of a smartphone.

Keywords: acquisition of signals, FIWARE, smartphones.

1. Introducción

El tratamiento de señales digitales y la manipulación de las señales provenientes del entorno no es un tema nuevo [1]. Sin embargo, los avances tecnológicos en capacidad de cómputo, conectividad, portabilidad y procesamiento de datos de manera casi

instantánea, permiten la implementación de soluciones antes inimaginables. Es así como en la actualidad contamos con aplicaciones que nos informan sobre el tráfico o la calidad del aire.

Vemos por un lado que el desarrollo de estas aplicaciones nos pone en un escenario de soluciones multidisciplinarias las cuales requieren conocimientos cada vez más profundos y de más de un área del conocimiento y por otro lado nos requiere del auxilio de técnicas cada vez más sofisticadas de inteligencia artificial y de aprendizaje de máquina.

En este trabajo, se presenta una solución de monitoreo y visualización de los datos que basa la adquisición de datos en un teléfono inteligente y el análisis y procesamiento en la plataforma FIWARE, la cual consiste en un conjunto de componentes basadas en estándares abiertos que facilitan el desarrollo de soluciones inteligentes en múltiples sectores verticales [2].

El teléfono inteligente se propone como dispositivo de toma de señal, dadas las características que posee como lo son: sus sensores, su portabilidad y su versatilidad. Estas características lo hacen idóneo para ser utilizados en diversas aplicaciones y servicios. Así mismo, se propone la plataforma FIWARE porque es de código abierto y provee una serie de componentes de software que facilitan la implementación de soluciones en el contexto del Interne de las cosas (IoT por sus siglas en inglés) [3].

Este documento está estructurado de la siguiente manera, en la sección dos se presentan los antecedentes. En la sección tres se presentan trabajos relacionados. En la sección cuatro se describe la aplicación propuesta. En la sección cinco se describen algunos escenarios en los cuales se evaluó la aplicación, y por último se presentan las conclusiones y trabajos futuros.

2. Antecedentes

El procesamiento de señales es la aplicación de una serie de operaciones lógicas y matemáticas a un conjunto de datos provenientes de una señal que proviene de una determinada fuente que puede ser analógica o digital. Las señales provenientes del mundo real son adquiridas por diferentes tipos de dispositivos como los sensores.

Por su parte, el IoT se refiere a la capacidad potencial de un objeto para integrarse e interactuar de manera independiente con cualquier otra entidad, ya sea una máquina o un ser humano. Para operar, el IoT ha dado paso a la existencia de objetos equipados con una serie de sensores y algoritmos capaces de detectar automáticamente las características del entorno en que se encuentran [4].

Los avances tecnológicos y en comunicaciones han dado paso a la aparición de los teléfonos inteligentes. Un teléfono inteligente es un dispositivo portátil que tiene gran capacidad de comunicación, procesamiento y almacenamiento, además del manejo de contenidos multimedia y facilidad para integrarse a redes inalámbricas [5]. Los teléfonos inteligentes cuentan con un conjunto de sensores para la detección de señales físicas como: aceleración, rotación, sonido, luminosidad, proximidad, campo magnético, entre otros.

En la actualidad existen diversas plataformas orientadas a soportar los conceptos de IoT, una de ellas es FIWARE, la cual se trata de una plataforma de código abierto para el desarrollo de aplicaciones en el entorno del IoT [6]. Esta plataforma proporciona un

conjunto de herramientas y librerías denominadas "*componentes genéricos*" que facilitan tareas como: la integración de los dispositivos IoT, el análisis y procesamiento de datos a mediana y gran escala, y la incorporación de interfaces avanzadas para la interacción con los usuarios [3]. Dentro de estos *componentes genéricos* se encuentra el Orion Context Broker (OCB) que es un intermediario que recibe la información de contexto generada por los productores de información (dispositivos, sensores, aplicaciones, etc.) y la distribuye a los consumidores (actuadores, otros dispositivos, aplicaciones, etc.), los cuales toman decisiones a partir de la información recibida. Para el intercambio de la información, el OCB se apoya en la especificación NGSI (Next Generation Service Interface), la cual se debe seguir al momento de empaquetar la información que será distribuida a través de la plataforma FIWARE[7].

3. Trabajos relacionados

En esta sección se presentan los trabajos relacionados en la adquisición de datos. Los teléfonos inteligentes son utilizados en diferentes áreas como dispositivos para la adquisición de datos [8-11]. Esto se debe a factores como su conectividad, los múltiples mecanismos de adquisición de datos que van desde la cámara y el micrófono, hasta sensores como el acelerómetro, el GPS o el giroscopio. Dentro de las ventajas más relevantes de los teléfonos inteligentes se encuentran el costo, son ampliamente utilizados por la población y en la actualidad son prácticamente un objeto personal de uso diario.

Los diferentes mecanismos que poseen los teléfonos inteligentes para la adquisición de señales del mundo real son utilizados por ejemplo en el área de la salud para detectar síntomas como la depresión, la ansiedad o el aislamiento social, para monitorear señales de electrocardiografía y frecuencia cardiaca, o para el monitoreo de la actividad física de una persona [9,10]. Asimismo, en el área de transporte se utiliza para la planificación de rutas, reportes de accidentes, detección automática de anomalías en la vía pública, conductas agresivas de parte de los conductores, monitoreo de mercancías, trazado de rutas, etc. [11,12]. Además de las aplicaciones y servicios enfocados en tareas específicas como el trazado de rutas o el análisis de movimiento, los teléfonos inteligentes también se utilizan en la recolección de datos [13-15].

Por su parte, la Plataforma FIWARE también es utilizada en diversas actividades que involucran el procesamiento de señales y el monitoreo de señales. Algunos de los trabajos están relacionados con el cuidado y monitoreo de pacientes, logística y transporte, agricultura, entre otros [16-17].

Como se puede observar, la adquisición de señales digitales por medio de teléfonos inteligentes es un tema que ha cobrado fuerza e importancia [18]. Su importancia radica en el incremento de servicios y aplicaciones que delegan en el teléfono inteligente la responsabilidad de obtener las señales del entorno. Además, varios trabajos hacen uso de las capacidades de localización y de comunicación del teléfono para brindar información en la que además de contar con la información del contexto, también nos permite obtener su ubicación geoespacial.

4. Aplicación para la adquisición de señales digitales

En esta sección se describe la aplicación propuesta para la adquisición de señales digitales utilizando teléfonos inteligentes y FIWARE, así como su arquitectura.

4.1. Arquitectura

La arquitectura de la aplicación para la adquisición de señales digitales utilizando teléfonos inteligentes y la plataforma FIWARE está compuesta por dos módulos. El módulo de adquisición de datos del teléfono está enfocado en la adquisición de la señal proveniente del teléfono inteligente y el otro módulo se encarga del monitoreo y la visualización de la señal adquirida, para lo cual se utiliza la plataforma FIWARE (Ver Fig. 1).

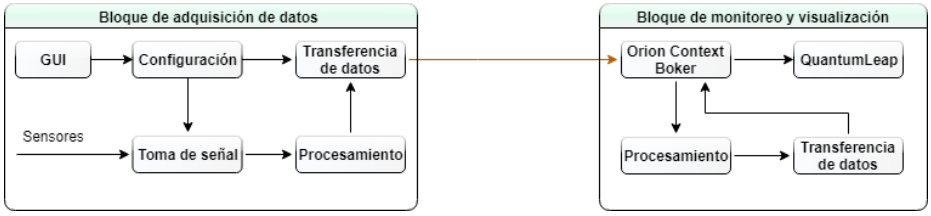


Fig. 1. Arquitectura general para la adquisición de señales digitales utilizando teléfonos inteligentes y la plataforma FIWARE.

4.2. Módulo para adquisición de señales

El módulo para la adquisición de señales permite obtener las señales provenientes de los sensores que se encuentran en el teléfono y realizar operaciones básicas sobre las mismas. Este módulo está compuesto por cinco componentes: GUI, configuración, toma de señal, procesamiento y transferencia de datos. El módulo está desarrollado en la versión 4.0 de Android, con un patrón de diseño modelo vista controlador. Todas las funcionalidades están desarrolladas en Java nativo e incluidas en una librería única, a excepción de la interfaz gráfica. Esto permite que la librería pueda ser utilizada en otros entornos diferentes a Android. Los sensores incluidos en esta versión son el acelerómetro, el giroscopio o sensor de orientación, el magnetómetro y el GPS. A continuación, se describen los componentes que conforman este módulo.

Interfaz Gráfica de Usuario (GUI). El objetivo principal de este componente es permitir que el usuario interactúe con la aplicación. Algunos ejemplos de la GUI se muestran en la Fig. 2.

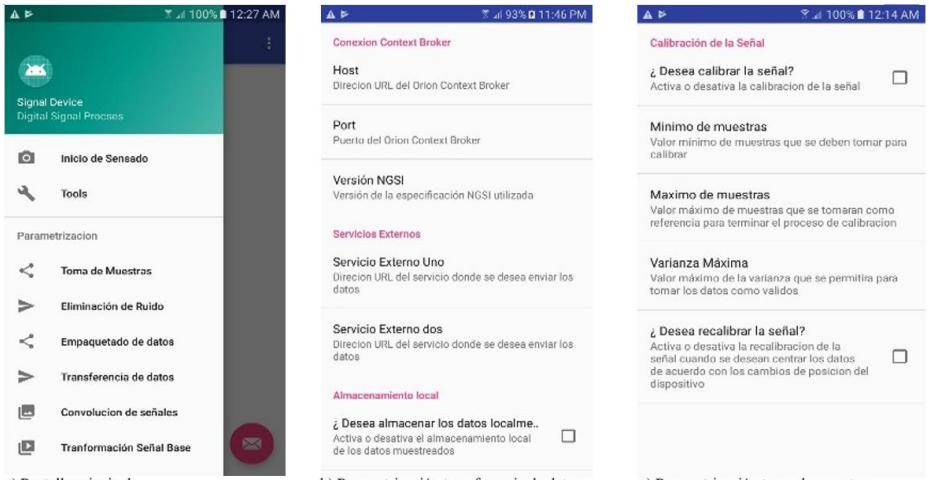
Componente de configuración. Este módulo es el encargado de gestionar los parámetros de configuración establecidos por el usuario en las vistas de parametrización.

Componente de toma de señal. Este componente es el encargado de obtener todas las señales provenientes de los sensores del teléfono. En este componente se realizan dos procesos importantes: obtención de la señal y calibración. El proceso de toma de señal se inicializa una vez que el usuario lo indica y que el proceso de calibración indica que

el teléfono se encuentra en una posición válida para empezar la toma de las señales. Una posición será válida cuando la varianza máxima del tipo de señal a obtener se encuentro dentro de los parámetros establecidos en el componente de configuración. La calibración puede además ser omitida, si el usuario lo considera pertinente, ver la figura Fig. 2-c

Componente de procesamiento. Este componente es el más importante del componente de adquisición de la señal, porque en él se ejecutan todas las operaciones que permiten manipular o modificar una señal. Las tareas realizadas por este componente son:

Centrado de la señal. Esta función se encarga de tomar el dato proveniente del componente de toma de señales y centrarla en un valor de referencia, el cual puede ser proporcionado por el usuario teniendo en cuenta la media de los datos calculada en el proceso de calibración.



a) Pantalla principal b) Parametrización transferencia de datos c) Parametrización toma de muestras

Fig. 2. Ejemplos de algunas vistas de GUI de la aplicación de adquisición de datos.

Centrado de la señal. Esta función se encarga de tomar el dato proveniente del componente de toma de señales y centrarla en un valor de referencia, el cual puede ser proporcionado por el usuario teniendo en cuenta la media de los datos calculada en el proceso de calibración.

Reducción de ruido. Esta funcionalidad permite reducir el ruido de la señal, el método utilizado para la reducción del ruido es el de medias móviles.

Translación de una señal en el tiempo. Esta función permite realizar una representación de la señal aplicando un determinado parámetro de corrimiento de la señal.



```
{
  "type": "SignalDevice",
  "id": "SD_E",
  "refDevice":  {
    "value": "Device_Prueba_1",
    "type": "Text"
  },
  "SD-Acel_Y": {
    "value": 9.8,
    "metadata": {
      "description": {
        "value": "Base signal"
      }
    }
  },
  "SD-Acel_X": {
    "value": 0.8,
    "metadata": {
      "description": {
        "value": "Base signal"
      }
    }
  },
  "SD-Acel_Z": {
    "value": 0.35,
    "metadata": {
      "description": {
        "value": "Base signal"
      }
    }
  },
  "timestamp": {
    "value": 1523804024221,
    "type": "Integer"
  },
  "dateModified": {
    "value": "2018-04-15T07:21:24.285Z",
    "type": "DateTime"
  }
}
```

```
{
  "type": "SignalDevice",
  "id": "SD_E",
  "refDevice":  {
    "value": "Device_Prueba_1",
    "type": "Text"
  },
  "SD1": {
    "value": 9.8,
    "metadata": {
      "description": {
        "value": "Base signal"
      }
    }
  },
  "SD2": {
    "value": 0.8,
    "metadata": {
      "description": {
        "value": "Centered signal"
      }
    }
  },
  "timestamp": {
    "value": 1523804024221,
    "type": "Integer"
  },
  "dateCreated": {
    "value": "2018-04-15T07:21:24.285Z",
    "type": "DateTime"
  },
  "dateModified": {
    "value": "2018-04-15T07:21:24.285Z",
    "type": "DateTime"
  }
}
```


Fig. 3. Ejemplo instancia del modelo de datos SignalDevice.

Escalado de la señal. Esta función permite aplicar un parámetro de escala a la señal para contraer o dilatar la señal base.

Diezmado e interpolación de la señal. Esta función permite aplicar diferentes valores de diezmado e interpolación a una señal base, para realizar este proceso el usuario debe establecer en las opciones de parametrización el número de señales derivadas que desea obtener y los valores de interpolación o diezmado para cada señal.

Empaquetado de la señal. Esta funcionalidad permite que los datos sean enviados de uno en uno, o que el usuario seleccione el número de datos que tendrá una muestra, en el caso de que el usuario quiera usar las opciones de monitoreo y visualización, el sistema crea una copia exacta de la señal original y a partir de esta realiza el empaquetado de acuerdo con los parámetros establecidos por el usuario.

Ventanas de tiempo. Esta opción permite manejar diferentes ventanas de tiempo ya sea para el cálculo de medias móviles, para el empaquetado de datos o para el uso en funciones que dependen de valores anteriores o futuros para el cálculo del valor actual.

Componente de transferencia de datos. Este componente permite la comunicación entre el módulo de adquisición de datos y el módulo de monitoreo y visualización. La función principal de este componente es la gestión del almacenamiento y distribución de los datos. El componente empaqueta los datos en objetos (modelos de datos) de tipo JSON basados en la especificación NGSI de FIWARE. En este trabajo se propone un nuevo modelo de datos bajo la especificación NGSI, al cual hemos llamado *SignalDevice.* Este es un modelo de datos flexible que permite el envío de una o varias señales. En la Fig. 3 se muestra la estructura del objeto JSON del modelo de datos propuesto. Este modelo de datos se caracteriza por tener una estructura básica que identifica de manera única al tipo de señal que se está procesando. Los campos id, type, refDevice, timestamp, dateModifieds y SD1 son de tipo obligatorio. La estructura SD representa la señal que se está procesando, siempre se debe identificar si se trata de la señal original o se trata de una señal derivada.

4.3. Módulo de monitoreo y visualización

Este módulo permite visualizar en la plataforma FIWARE los datos que han sido adquiridos en el módulo de adquisición de la señal. El módulo está compuesto por cuatro componentes: Orion Context Broker, QuantumLeap, procesamiento y transferencia de datos. Para la visualización de los datos se utiliza Grafana que es una aplicación WEB incluida dentro del componente QuantumLeap.

Componente de procesamiento y componente de transferencia de datos. Estos componentes tienen un comportamiento similar a los descritos en el componente de adquisición de datos, su diferencia principal es que fueron desarrollados para ejecutarse en la plataforma FIWARE.

Componente Orion Context Broker. Este componente es un componente genérico de la plataforma FIWARE. El objetivo de este componente es recibir la información de contexto generada por el módulo de adquisición de señales del teléfono y enviarla al componente de procesamiento y al componente de QuantumLeap. Para la transferencia de información se utilizan los modelos de datos *Device* [19], *Alert* [20] y *SignalDevice*. Los dos primeros modelos de datos hacen parte de la especificación NGSI y el último es el nuevo modelo de datos que nosotros proponemos.

Componente QuantumLeap. Este componente es un componente genérico de *FIWARE* y es utilizado como la interfaz gráfica del módulo de monitoreo y visualización. El objetivo principal de *QuantumLeap* es manejar series de tiempo de diferentes señales y permitir su visualización en una interfaz *WEB*. El modo de funcionamiento es el siguiente: el componente de QuantumLeap recibe información del OCB y la almacena para que pueda ser visualizada por el componente Grafana. Un aspecto importante a resaltar es que el componente Grafana no forma parte de los componentes genéricos de FIWARE, pero está integrado el componente QuantumLeap para la visualización de series de tiempo.

5. Evaluación de la aplicación de adquisición de señales digitales

La evaluación de nuestra aplicación para la adquisición de señales digitales utilizando un teléfono inteligente y la plataforma FIWARE se centró en explorar tres de sus características principales: adquisición de datos, etiquetado de datos y visualización de señales. Para explorar estas características se plantearon los siguientes escenarios: adquisición de datos para identificación de irregularidades en la vía pública, cambios de velocidad de un vehículo basado en los datos del GPS y visualización de una señal externa utilizando el módulo de visualización y monitoreo.

Los recursos utilizados para la evaluación de nuestra aplicación fueron: tres teléfonos inteligentes (ZTE Blade V6, un Samsung Galaxy J4 y un Motorola Moto G5) y dos máquinas virtuales alojadas en la plataforma FIWARE Lab nodo México. Además, en el caso de los dos primeros escenarios se utilizaron diferentes medios de transporte público y particular.

La fase de evaluación de nuestra aplicación se realizó por un periodo de 14 meses. Durante este periodo se realizaron más de 100 recorridos en zonas rurales y urbanas; y se recolectaron más de 50 millones de registros provenientes del acelerómetro, el GPS, el magnetómetro y, el giroscopio o el sensor de orientación. En la tabla 1, se muestra el resumen de todos los datos recolectados en la fase de evaluación.

Tabla 1. Registros recolectados en la fase de pruebas

Sensor	Núm. registros
Acelerómetro	54´452,442
GPS	391,388
Magnetómetro	5´006,596
Giroscopio / Orientación	7´115,138
Total	**66´965,564**

En el caso de los archivos planos, el formato utilizado para almacenar los datos de cada registro fue el siguiente: Identificador del registro, aceleración en el eje *x*, aceleración en el eje *y*, aceleración en el eje *z*, giro sobre el eje *x*, giro sobre el eje *y*, giro sobre el eje *z*, magnetómetro en el eje *x*, magnetómetro en el eje *y*, magnetómetro en el eje *z*, latitud, longitud, altitud, velocidad registrada por el GPS, precisión horizontal del GPS, marca de tiempo en milisegundos, tipo de evento, tipo de desplazamiento, área del evento, tipo de espacio en el que se genera el desplazamiento, fecha y hora en la que se toma el registro.

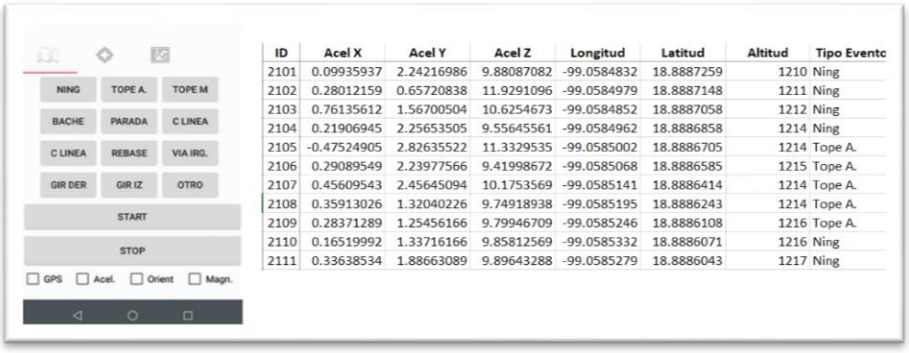
5.1. Escenario 1: adquisición de datos para identificación de irregularidades en la vía pública

El escenario enfocado en la adquisición de datos para la identificación de irregularidades en la vía se centra en la característica de recolección y etiquetado de datos de nuestra aplicación. Para evaluar esta característica, se propone utilizar nuestra aplicación para recolectar un conjunto de datos en tiempo real, el cual pueda ser utilizado posteriormente para analizar este tipo de irregularidades.

Para el etiquetado de datos se seleccionaron cuatro categorías: tipo de evento, medio de desplazamiento, zona de desplazamiento y tipo de vía o camino. En la tabla 2 se muestran las categorías y las opciones seleccionadas para cada categoría.

Tabla 2. Categorías y eventos de etiquetado por cada subcategoría.

Categoría	Opciones de etiquetado por categoría
Tipo de evento	Tope de metal, tope de asfalto, bache, parada, cambio de carril a la derecha, cambio de línea a la izquierda, giro a la izquierda, giro a la derecha, camino irregular
Medio de desplazamiento	Reposo, caminando, auto, ruta, autobús, moto, bicicleta, corriendo
Zona de desplazamiento	Urbana, rural, ciudad, pueblo, municipio, residencial, comercial, escolar
Tipo de vía	Pista, estatal, inter estatal, municipal, escaleras, circuito cerrado, anden



ID	Acel X	Acel Y	Acel Z	Longitud	Latitud	Altitud	Tipo Eventc
2101	0.09935937	2.24216986	9.88087082	-99.0584832	18.8887259	1210	Ning
2102	0.28012159	0.65720838	11.9291096	-99.0584979	18.8887148	1211	Ning
2103	0.76135612	1.56700504	10.6254673	-99.0584852	18.8887058	1212	Ning
2104	0.21906945	2.25653505	9.55645561	-99.0584962	18.8886858	1214	Ning
2105	-0.47524905	2.82635522	11.3329535	-99.0585002	18.8886705	1214	Tope A.
2106	0.29089549	2.23977566	9.41998672	-99.0585068	18.8886585	1215	Tope A.
2107	0.45609543	2.45645094	10.1753569	-99.0585141	18.8886414	1214	Tope A.
2108	0.35913026	1.32040226	9.74918938	-99.0585195	18.8886243	1214	Tope A.
2109	0.28371289	1.25456166	9.79946709	-99.0585246	18.8886108	1216	Tope A.
2110	0.16519992	1.33716166	9.85812569	-99.0585332	18.8886071	1216	Ning
2111	0.33638534	1.88663089	9.89643288	-99.0585279	18.8886043	1217	Ning

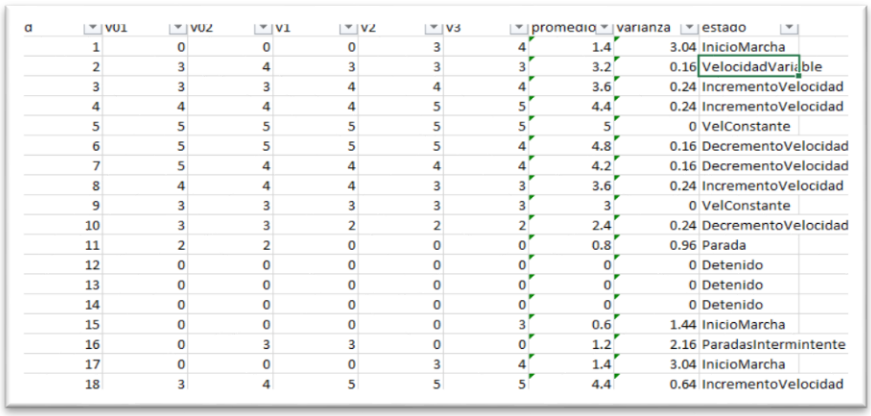
Fig. 4. Ejemplo de etiquetado y datos recolectados con el módulo de adquisición de señales.

Para la adquisición de los datos se colocó el teléfono en una posición fija dentro del vehículo y se seleccionaron los sensores. Los sensores fueron el acelerómetro para la detección de los eventos y el GPS para geolocalizar los registros. Una vez que el vehículo iniciaba la marcha era presionado el botón *"start"*. En la Fig. 4 se pueden observar las opciones de etiquetado para la categoría tipo de evento y un extracto de uno de los archivos que contiene un evento etiquetado.

5.2. Escenario 2: cambios de velocidad de un vehículo basado en los datos del GPS

Este escenario es enfocado en la adquisición de datos para entrenar un modelo para monitorear los cambios de velocidad durante el desplazamiento de un automóvil. Este escenario se centra en la evaluación de la característica de adquisición de datos de nuestra aplicación. Para evaluar esta característica, se propone utilizar nuestra aplicación como una herramienta de adquisición y empaquetado de datos.

En este caso se utilizó la aplicación para adquirir los datos de geolocalización en el desplazamiento de un automóvil. Los datos obtenidos del GPS de un teléfono inteligente colocado al interior del automóvil. En este caso se filtró de la señal, luego se discretizó y posteriormente se crearon paquetes con siete datos por registro, de los cuales cinco correspondían a muestras de la señal con una ventana de dos, uno al promedio y el otro a la varianza. Los datos fueron almacenados en un archivo plano y posteriormente los datos fueron etiquetados manualmente, en la Fig. 5 se presenta un ejemplo.



a	v01	v02	v1	v2	v3	promedio	varianza	estado
1	0	0	0	3	4	1.4	3.04	InicioMarcha
2	3	4	3	3	3	3.2	0.16	VelocidadVariable
3	3	3	4	4	4	3.6	0.24	IncrementoVelocidad
4	4	4	4	5	5	4.4	0.24	IncrementoVelocidad
5	5	5	5	5	5	5	0	VelConstante
6	5	5	5	5	4	4.8	0.16	DecrementoVelocidad
7	5	4	4	4	4	4.2	0.16	DecrementoVelocidad
8	4	4	4	4	3	3.6	0.24	IncrementoVelocidad
9	3	3	3	3	3	3	0	VelConstante
10	3	3	2	2	2	2.4	0.24	DecrementoVelocidad
11	2	2	0	0	0	0.8	0.96	Parada
12	0	0	0	0	0	0	0	Detenido
13	0	0	0	0	0	0	0	Detenido
14	0	0	0	0	0	0	0	Detenido
15	0	0	0	0	3	0.6	1.44	InicioMarcha
16	0	3	3	0	0	1.2	2.16	ParadasIntermintente
17	0	0	0	3	4	1.4	3.04	InicioMarcha
18	3	4	5	5	5	4.4	0.64	IncrementoVelocidad

Fig. 5. Resultados obtenidos después de pasar los paquetes por un clasificador.

Después de realizar el proceso de adquisición y el etiquetado de los datos, los datos fueron pasados por algunos clasificadores. Los algoritmos utilizados en el proceso de clasificación fueron *Naive Bayes*, *Random Fores* y J48, para lo cual se utilizó la herramienta *R Studio*. En la tabla 3 se pueden observar los resultados obtenidos después de pasar los datos por cada clasificador para entrenar el modelo para el monitoreo de los cambios de velocidad durante el desplazamiento de un automóvil

Tabla 3. Resumen de resultados obtenidos al pasar los datos por los clasificadores.

Técnica	**Clasificados Correctamente**		**Clasificados Incorrectamente**	
	Eventos	**%**	**Eventos**	**%**
Naive Vayes	281	71.867 %	110	28.133 %
RandomForest	368	94.1176 %	23	5.882 %
J48	367	93.8619 %	24	6.138 %

5.3. Escenario 3: monitoreo de señal externa utilizando el módulo de visualización y monitoreo

Este escenario es enfocado en la evaluación de la característica de visualización y monitoreo de nuestra aplicación. Para evaluar esta característica, se creó un script en Python para simular una señal senoidal base. A partir de la señal senoidal base se aplicaron algunas operaciones básicas de procesamiento de señales, obteniendo así 8

señales derivadas. En la Fig. 6 se describe en seis paneles, en donde el primer panel corresponde a la visualización de la señal original y las 8 señales derivadas. Un aspecto importante es que se utiliza el modelo de datos *SignalDevice* para la transferencia de daos entre el script y el módulo de monitoreo y visualización.



Fig. 6. Visualización de una señal senoidal utilizando el componente de QuantumLeap de FIWARE.

En el panel 2 se muestra la señal original, además, de las señales equivalentes al escalar la señal original en un factor de ½ y un factor de 2. En el panel 3, se muestra la señal original y sus equivalentes al desplazarla ±5 unidades con respecto al eje horizontal. En el panel 4 se muestra la señal original y su equivalente al multiplicar la señal original por un factor de 1/2 y 1/3, o lo que es igual al muestrear la señal 2 y 3 veces más rápido respectivamente. En el panel 5, se muestra la señal original y su equivalente al desfasar la señal en 1/4 y por último en el panel 6, se muestra un diezmado por 2 de la señal original.

6. Conclusiones y trabajos futuros

En este trabajo de investigación se muestra que es factible la adquisición de señales digitales utilizando teléfonos inteligentes y la plataforma FIWARE. Sin embargo, es importante tener en cuenta que la calidad y precisión de los datos obtenidos dependen de la calidad de la señal, los parámetros de configuración y las características técnicas del teléfono inteligente.

Las características del módulo de adquisición de datos y del módulo de visualización permiten que estos puedan ser utilizados en diversos dominios. Sus características de etiquetado, recolección de datos y manipulación de la señal pueden ser de utilidad en la fase de adquisición de datos y de prueba para el desarrollo de diversas aplicaciones y servicios.

Las pruebas realizadas mostraron que el componente de adquisición de datos y el componente de visualización pueden ser utilizados de manera independiente. Además, en el caso de estudio relacionado con los cambios de velocidad, se pudo observar que los resultados de clasificación son aceptables utilizando los sensores de un teléfono inteligente.

En este trabajo también se presenta el modelo de datos SignalDevice el cual puede ser utilizado en el área de procesamiento de señales para el intercambio de datos en la plataforma FIWARE. Un aspecto importante de este modelo es que permite el envío de señales base y sus derivadas.

Durante el desarrollo del trabajo se observó que la utilización de la plataforma FIWARE, en conjunto con la especificación NGSI, facilitan la implementación de soluciones orientadas a las ciudades inteligentes y el IoT.

Como trabajos futuros se plantean: la evaluación de nuestra aplicación en otros escenarios como pueden ser el trazado de rutas, monitoreo de actividad física, análisis de vibraciones, entre otras; La evaluación del módulo de monitoreo con señales provenientes de fuentes diferentes a teléfonos inteligentes; La adición de funcionalidades que permitan adquirir la señal proveniente de otros sensores que vienen incluidos en el teléfono inteligente y que no son implementados en este trabajo; Finalmente, también se plantea como trabajo futuro el desarrollo de una aplicación similar para otros tipos de dispositivos como lo es el caso de los dispositivos IOS.

Agreadecimientos. Esta investigación ha sido financiada parcialmente por el proyecto SmartSDK, Num Ref. 272727, así como el proyecto TecNM/CENIDET, Num Ref. 5138.19-P.

Referencias

1. Oppenheim, A.V., Willsky, A.S., Nawab, S.H.: Señales y sistemas. Pearson Educación (1998)
2. El ecosistema FIWARE, http://www.cudi.edu.mx/cudi2017/presentaciones/1.%20Ecosistema_FIWARE.PDF, último acceso: 2019/05/10
3. Fiware: Una plataforma abierta y estándar para Ciudades Inteligentes. https://www.esmartcity.es/comunicaciones/i-congreso-ciudades-inteligentes-fiware, último acceso: 2019/05/10
4. Sosa, E.O., Godoy, D.A., Neis, R., Motta, G., Luft, R., Sosa, D., Quiñones, P.: Internet del futuro y ciudades inteligentes. In: XV Workshop de Investigadores en Ciencias de la Computación. pp. 21–27, Argentina (2013)
5. Organista-Sandoval, J., McAnally-Salas, L., Lavigne, G.: El teléfono inteligente (smartphone) como herramienta pedagógica. Apertura 5(1), 6–19 (2013)
6. Las plataformas de código abierto en las Smart Cities. http://www.fractaliasystems.com/las-plataformas-de-codigo-abierto-en-las-smart-cities/, último acceso: 2019/05/10.
7. FIWARE-NGSI v2 Specification. http://fiware.github.io/specifications/ngsiv2/stable/, último acceso: 2019/05/10
8. Pielot, M., Church, K., De Oliveira, R.: An in-situ study of mobile phone notifications. In: Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services, pp. 233–242. Mobile HCI, New York (2014)
9. Case, M.A., Burwick, H.A., Volpp, K.G., Patel, M.S.: Accuracy of Smartphone Applications and Wearable Devices for Tracking Physical Activity Data. JAMA 313(6), 625–626 (2015)

10. Fazio, M., Celesti, A., Marquez, F.G., Glikson, A., Villari, M.: Exploiting the FIWARE cloud platform to develop a remote patient monitoring system. In: 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 264–270. IEEE, Cyprus (2015)
11. Ali, H.M., Alwan, Z.S.: Car Accident Detection and Notification System Using Smartphone. IJCSMC, pp. 620–635 (2017)
12. Gonçalves, J., Gonçalves, J.S., Rossetti, R.J., Olaverri-Monreal, C.: Smartphone sensor platform to study traffic conditions and assess driving performance. In: 17th IEEE International Conference on Intelligent Transportation Systems, pp. 2596–2601, IEEE, China (2014)
13. Martinez, A., Ramirez, F., Estrada, H., Torres, L.A.: Generic Module for Collecting Data in Smart Cities on Remote Sensing & Spatial Information Sciences, pp. 65–72, México (2017)
14. Bersch, S.D., Azzi, D., Khusainov, R., Achumba, I.E., Ries, J.: Sensor data acquisition and processing parameters for human activity classification. Sensors 14(3), 4239–4270 (2014)
15. Rife, T.W., Poland, J.A.: Field Book: An Open-Source Application for Field Data Collection on Android. Crop Science 54(4), 1624–1627 (2014)
16. Fernández, P., Santana, J., Ortega, S., Trujillo, A., Suárez, J., Domínguez, C., Sánchez, A. (et. al.): SmartPort: A Platform for Sensor Data Monitoring in a Seaport Based on FIWARE. Sensors 16(3), 417 (2016)
17. López-Riquelme, J.A., Pavón-Pulido, N., Navarro-Hellín, H., Soto-Valles, F., Torres-Sánchez, R.: A software architecture based on FIWARE cloud for Precision Agriculture. Agricultural Water Management 183, 123–135 (2017)
18. Wahlstrom, J., Skog, I., Handel, P.: Smartphone-Based Vehicle Telematics: A Ten-Year Anniversary. IEEE Transactions on Intelligent Transportation Systems (2017)
19. Device Fiware DataModels , https://fiware-datamodels.readthedocs.io/en/latest/Device/Device/doc/spec/index.html, último acceso: 2019/05/10
20. Alert Fiware-DataModels, https://fiware-datamodels.readthedocs.io/en/latest/Alert/doc/spec/index.html, último acceso: 2019/05/10

ISSN 1870-4069

Detección efectiva de rostros en imágenes utilizando descriptores basados en HOG

Cinthia Viviana Alcántara-Montiel, Jesús Carlos Pedraza-Ortega,
Juan Manuel Ramos-Arreguín, Efrén Gorrostieta-Hurtado, Saúl Tovar-Arriaga,
José Emilio Vargas-Soto

Universidad Autónoma de Querétaro, Facultad de Ingeniería
Santiago de Querétaro, Querétaro, México

cinthia.vamontiel@hotmail.com, caryoko@yahoo.com,
jsistdig@yahoo.com.mx, efrengorrostieta@gmail.com,
saul.tovar@uaq.mx, emilio@mecatronica.net

Resumen. El propósito de esta investigación es desarrollar un detector de rostros eficiente basado en descriptores HOG (Histogram of Gradient) y clasificadores tipo cascada; mediante la combinación de estos descriptores y clasificadores es posible obtener un detector de rostros robusto, el cual pueda detector rostros en posiciones diferentes en diversos escenarios, los cuales pueden tener una variación en el nivel de iluminación. La propuesta presentada mejora el rango de detección en comparación con otros detectores de rostro existentes. El entrenamiento de los detectores se llevó a cabo utilizando las bases de datos de rostros de libre acceso.

Palabras clave: detector de rostros, descriptores, clasificadores tipo cascada, histogram of gradient, entrenamiento.

Effective Face Detection on Images by Using Descriptors Based on HOG

Abstract. The aim of this work is to develop an efficient face detector based on HOG descriptors and cascade classifiers; by combine such descriptors and classifiers it is possible get a robust face detector which can detect faces on different positions into sceneries which can vary on lighting. This proposal improves the range of detection in comparison with another literature face detectors. The training of the detectors was made with the available face databases.

Keywords: face detector, descriptors, cascade classifiers, histogram of gradient, training.

1. Introducción y estado del arte

La detección de rostro hace referencia a los mecanismos computacionales mediante los cuales es posible determinar si existe o no un rostro en una imagen digital dada, y en su caso determinar su posición en la misma, así como su tamaño [1].

La detección de objetos es una tarea estudiada y aplicada en el ramo de la visión por computadora. Para detectar un objeto específico, lo que se hace es buscar en la imagen aquellas características visuales que son inherentes a él, no obstante, cuando se trata de localizar o detectar un rostro el grado de dificultad se eleva dado que los rostros pueden encontrarse en un amplio rango de poses, con una infinidad de gestos y detalles visuales que modifican su apariencia; todo aquello que visualmente ayuda a nuestro cerebro a distinguir entre una persona y otra, resulta en un grado extra de complejidad en el caso de la localización facial mediante el procesamiento de imágenes [2]. En su trabajo, Ming-Hsuan Yang *et al.* [3] enlista los factores que se relacionan directamente con los retos de la detección de rostros: a) La variabilidad de componentes estructurales: La diferencias entre rostros incluye características intrínsecas del rostro, como la forma, color y tamaño, así como variaciones en la presencia o ausencia de características más específicas, como por ejemplo la barba, el bigote y los lentes. b) Pose: La posición en la que sea captado un rostro mediante la cámara influirá de forma directa en la apariencia de esta, la pose podría afectar la visibilidad de algunas partes esenciales en la descripción de un rostro. c) Expresión Facial: Esta afecta de forma inmediata y directa la apariencia del rostro. Oclusión: Hace referencia a la posibilidad de encontrar, en la imagen de entrada, rostros parcialmente ocultos; ya sea por otra clase de objetos o por otros rostros. d) Orientación de la imagen: La posición de la cámara respecto al rostro que desea capturarse afectará la forma en que visualmente se aprecia el rostro. e) Condiciones de la imagen: Engloba todos los factores que pueden afectar el proceso de adquisición de imagen, tales como las características de la cámara, la iluminación, el movimiento, etc.

1.1. Técnicas de detección

Hjelmas y Low [4] proponen encontrar una clasificación de las técnicas empleadas para la detección de rostros, tal clasificación hace distinción del enfoque que se toma para llevar a cabo la tarea de detección asignada, Fig. 1. De acuerdo a la literatura existen dos tipos de detectores, según su enfoque, para realizar la detección de rostro; los basados en las características faciales y los que se basan en la imagen. En los primeros la detección de lleva a cabo mediante el análisis de ángulos, longitudes y áreas de las características visuales en búsqueda de determinados elementos que componen un rostro, por otra parte, los segundos se basan en comparar las imágenes de prueba con otras llamadas de entrenamiento.

Las técnicas basadas en características se subdividen en tres; a) Análisis de bajo nivel, las cuales utilizan bordes, información de la intensidad de los pixeles, información de color, movimiento de los pixeles o bien el uso de algunas medidas generalizadas. b) Los modelos de silueta activa utilizan información de los rasgos locales de la imagen, bordes y brillo, y se deforma gradualmente de tal manera que vaya tomando la forma del rasgo, teniendo tres tipos de modelos de silueta activa como son serpientes (snakes), plantillas deformables y modelos de distribución de punto (PDMs).

c) Análisis de rasgos: Este tipo de técnicas se consideran más certeras que aquellas llamadas de bajo nivel, dado que estas últimas pueden fácilmente generar resultados ambiguos, mientras que en el análisis de rasgos se aplica el conocimiento de la geometría facial de tal manera que las características o rasgos sean detectados y verificados. Dentro de las técnicas que se incluyen en este grupo tenemos a la búsqueda de rasgos, el análisis de constelaciones.

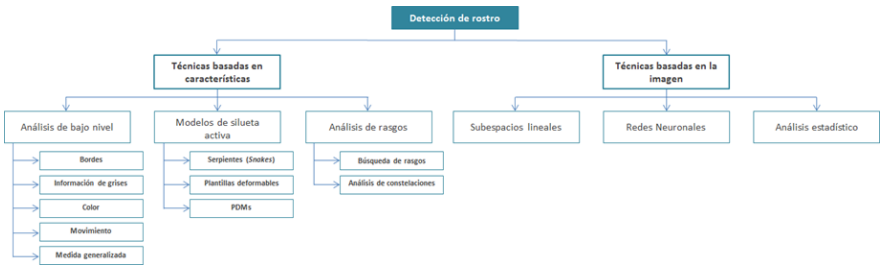


Fig. 1. Técnicas para la detección de rostros, clasificadas según su enfoque.

Las técnicas basadas en la imagen se subdividen en sub-espacios lineales, redes neuronales y análisis estadístico.

2. Fundamentos teóricos

2.1. Características

Las características (features, en inglés), en términos de procesamiento de imágenes y visión por computadora, son considerados como atributos individuales, relevantes, que describen a un objeto y que son de gran utilidad para su reconocimiento y diferenciación de este entre otros [5].

Es común que, para el proceso de detección de objetos, se elija trabajar con características y no con valores de pixeles, el principal motivo de ello es que las primeras pueden codificar información de alto nivel, mientras que los segundos son más lentos. En general, existen dos formas de clasificar las características; acuerdo con su complejidad se clasifican como características de bajo o alto nivel [6]. Las características de alto nivel requieren de un método de procesamiento previo, generalmente se basan en las características de bajo nivel. La extracción de estos ocurre mediante la búsqueda de formas y objetos dentro de una imagen digital [7]. Y las características de bajo nivel son aquellas que pueden extraerse, de la imagen, de forma automática, sin ninguna información de forma.

Por otro lado, de acuerdo con la información que proporcionan, se dividen en características locales y globales. Las globales describen a la imagen como un todo, pueden considerarse como un atributo particular de la imagen, mismo que involucra a todos sus pixeles, mientras que las locales se centran en puntos clave y en la descripción de las áreas que rodean tales puntos, dicho de otra forma, representan una pequeña área del total de la imagen.

2.2. Detector de características

Estos detectores, pueden ser clasificados en tres categorías; detectores a una escala, detectores multi-escala y detectores afines invariantes. Los primeros son detectores que son invariantes a transformaciones de la imagen, es decir, pueden detectar las características del objeto ante cambios tales como desplazamiento, rotación y cambios de iluminación, sin embargo, no funcionan ante cambios de escala. Los detectores multi-escala tienen la capacidad de lidiar con los problemas de escalamiento, a diferencia de los primeros. Por último, los detectores afines invariantes pueden trabajar incluso con imágenes tomadas desde diferentes perspectivas visuales.

Es importante conocer las propiedades básicas que un detector de características debe poseer para cumplir con su objetivo; proveer una adecuada representación de un objeto determinado, de tal manera que este pueda ser fácilmente identificable de entre otros en las imágenes dadas. Tales características son: a) Exactitud: La localización de características debe llevarse a cabo de forma precisa, principalmente en tareas de emparejamiento, b) Repetibilidad: Debe ser capaz de repetir las detecciones de características de los mismos objetos o escenas bajo diferentes condiciones de visualización. c) Robustez: Debe detectar las mismas características sin importar si estas se han desplazado, si presentan rotación o cambios de escalamiento. d) Generalidad: Debe tener la capacidad de detectar características que puedan utilizarse en diferentes aplicaciones. e) Eficiencia: Debe realizar las detecciones de forma rápida, de tal manera que pueda ser empleado en aplicaciones en tiempo real, f) Alta densidad: La cantidad o densidad de las características detectadas debe corresponder a la información contenida en la imagen. Es decir, que el detector deberá detectar la mayoría de las características existentes en la imagen de entrada.

2.3. Descriptores de características

Una vez que el conjunto de características ha sido localizado en la imagen, su contenido debe ser codificado en un descriptor, por lo que los descriptores habrán de entenderse como representaciones compactas del contenido de una región de la imagen [8]. Algunos de ellos se listan a continuación:

- SIFT (Scale Invariant Feature Transform): Algoritmo presentado por Lowe [9], es considerado como un descriptor basado en distribución, se encarga de la obtención de características en imágenes en escala de grises. Las características de este algoritmo son invariantes ante cambios de rotación, escala, corrimiento, y parcialmente invariantes a cambios de iluminación y perspectiva 3D.

- Histograma de Gradientes Orientados HOG (Histogram of Oriented Gradients): Este descriptor, abordado por Dalal y Triggs [2], forma parte de los descriptores basados en distribución, de acuerdo con Nixon y Aguado [7]. Este descriptor trabaja en la construcción de una caracterización más detallada de la estructura espacial con una pequeña ventana de imagen. Se basa en la afirmación de que la apariencia local de un objeto puede caracterizarse bastante bien mediante la distribución de los gradientes locales de intensidad o las direcciones de los bordes en la imagen.

- SURF (Speeded-up Robust Features): según lo escrito por Awad y Hassaballah [10], este detector-descriptor fue desarrollado por Bay *et al.* [11] y se considera como una alternativa al SIFT, siendo el SURF mucho más eficiente y robusto, aun cuando se

basa en propiedades similares a las del SIFT. El primer paso de este algoritmo consiste en establecer una orientación reproducible, basada en la información de una región circular que rodee al punto de interés. Lo siguiente se trata de construir un área cuadrada, alineada a la orientación seleccionada, para finalmente, extraer el descriptor.

- LBP (Local Binary Pattern): De acuerdo con Ruiz Sancho [5] este es el principal descriptor, basado en textura, empleado en la detección de peatones. El operador describe cada pixel respecto a los niveles de grises relativos de los pixeles contiguos; si el nivel de gris del pixel vecino es menor que el del pixel de referencia, el valor se establece en 0, mientras que si es mayor o igual, el valor será de 1.
- Haar-like features: Fueron desarrollados por Viola y Jones sobre las bases planteadas por Papageorgiou *et al.* [12].

2.4. Clasificadores

El proceso de clasificación es aquel mediante el que elementos individuales son agrupados de acuerdo con la similitud entre sus características y la descripción de grupos preestablecidos, de tal manera que a los elementos de entrada se les asignará una etiqueta correspondiente a diferentes clases o categorías predefinidas con características comunes. Mientras que Tan *et al.* [13] dice de la clasificación que esta es la tarea de aprender una función objetivo, o modelo de clasificación, la cual mapea cada uno de los atributos de las instancias de entrada en una de las clases predefinidas.

Un clasificador, también conocido como técnica de clasificación, es un enfoque sistemático para construcción de funciones objetivo a partir de un conjunto de datos de entrada. Cada uno de ellos necesita de una técnica o algoritmo de aprendizaje para identificar la función que mejor describa la relación entre el atributo buscado y la clase del objeto de entrada. De hecho, se considera que la clasificación es un proceso que consta de dos pasos:

- Etapa de aprendizaje o entrenamiento y construcción de modelo (o función objetivo).
- Fase de prueba o aplicación del modelo.

3. Método propuesto

La metodología propuesta se representa de forma gráfica en la Fig. 2, es importante destacar que las etapas propuestas para la implementación del proyecto son consideradas como lineamientos generales del proceso a seguir, será en las secciones siguientes en las que se describirá con mayor profundidad el proceso de ejecución de cada una de las mencionadas etapas.

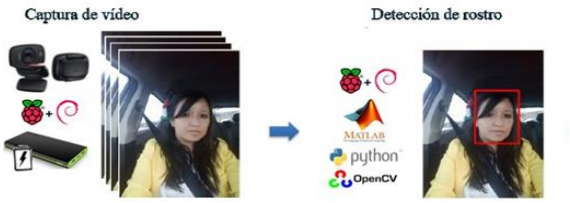


Fig. 2. Metodología general propuesta.

3.1. Captura de imágenes y/o vídeo

Para llevar a cabo la primera fase de la metodología propuesta se realizaron pruebas con cámaras de vídeo de 3MP, 5MPy 8MP de resolución, tras las pruebas se determinó que una cámara de 3MP proporcionaba una imagen poco clara, por ello se decidió utilizar una cámara de al menos 5MP de resolución. Adicionalmente, es indispensable que ésta se comunique por medio de un cable USB, además, es recomendable que el ángulo de enfoque sea fácilmente modificable. Por todo lo anterior se seleccionó la cámara Logitech c525, que cuente con una calidad de fotos de hasta 8 MP, resolución de video en alta definición, tiene un enfoque automático a 7 cm o más, cuenta con corrección de iluminación automática, conexión USB 2.0 y longitud de cable de 1.8 m.

3.2. Detección de rostro

Pruebas preliminares

En el amplio campo de aplicación de la detección de rostro existen de detectores pre-entrenados, basados en lenguajes tales como MATLAB y la librería de procesamiento de imágenes llamada OpenCV. Por esta razón, se tomó la decisión de realizar pruebas preliminares con uno de tales detectores; el detector de rostros frontales de Python. El detector de rostros seleccionado se basa en el trabajo de Viola y Jones [14], y fue entrenado para detectar rostros frontales [15].

De acuerdo con las pruebas realizadas se pudo establecer que tras configurar adecuadamente sus parámetros, dicho detector tiene un muy bajo porcentaje de errores de clasificación, sin embargo, el cuadro de detección que delimita la zona que forma parte del rostro encontrado no es consistente, mientras que, como parte de dichas inconsistencias, no siempre se extrae el rostro hallado desde la parte superior de la frente hasta justo debajo de la barbilla, es decir, se excluían segmentos del rostro.

Algunas muestras de los resultados de tales pruebas preliminares se muestran en la Fig. 3. Cabe destacar que tales pruebas se llevaron a cabo sin un pre-procesamiento riguroso de las imágenes, dado que se consideraron únicamente como punto de partida para la primera etapa de la metodología propuesta, de tal manera que se tuvieran claros los factores a mejorar en cuanto a la detección de rostro.



Fig. 3. Muestras de los resultados obtenidos de las pruebas preliminares, desarrolladas con la aplicación del detector de rosto frontal de OpenCV.

Descriptores HOG en detección de rostro

De acuerdo con Dalal y Triggs [2] el primer paso para la localización de rostro es definir un conjunto robusto de características que permitan discernir si dentro de una imagen hay un rostro o no, aún si el fondo de la imagen es desordenado o si esta fue tomada en un entorno con mala iluminación, a tales conjuntos se les denomina descriptores.

Usando descriptores locales normalizados basados en el método de (HOG) se puede realizar la detección con un excelente resultado en comparación con cualquier descriptor desarrollado, incluyendo Wavelet. Asimismo, aun cuando los descriptores propuestos se procesan sobre una red densa de celdas uniformemente espaciadas y usan la superposición de la normalización del contraste local como un esfuerzo para mejorar su rendimiento, lo que los diferencia de algunos otros como los SIFT.

Un método de detección de rostro basado en los descriptores HOG puede resumirse en el diagrama de la Fig. 4.

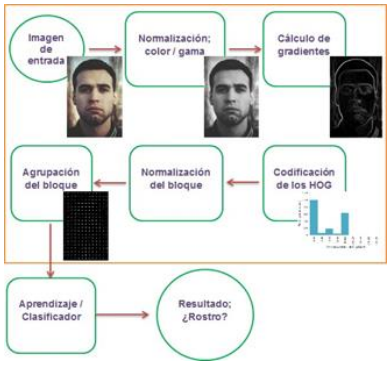


Fig. 4. Resumen gráfico de la detección de rostros basada en un descriptor HOG.

Las etapas que integran al algoritmo de aplicación del descriptor HOG, ilustradas en la Figura 4, son:

- **Normalización color/gamma**: También llamada pre-procesamiento. En este punto, se ecualiza la gamma o el color de la imagen como medida para reducir la influencia de la iluminación [16]. En algunas fuentes se sugiere que este es un paso opcional [17] o bien, que la normalización más útil es la aplicada al tamaño de la imagen [18], esto debido a que la normalización de gamma o color no representa una mejora significativa en los resultados.
- **Cálculo de gradientes**: El método más común para el cálculo de los gradientes es aplicar una máscara, derivada discreta 1D de punto centrado, tanto vertical como horizontal, filtrando así la imagen. El proceso se describe en las ecuaciones (1) – (6):

$$D_x = \{-1,0,1\}, \tag{1}$$

$$D_y = \begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix}, \tag{2}$$

$$I_x = I * D_x, \tag{3}$$

$$I_y = I * D_y, \tag{4}$$

$$|G| = \sqrt{{I_x}^2 + {I_y}^2}, \tag{5}$$

$$\theta = \arctan(I_x/I_y). \tag{6}$$

- **Codificación de los histogramas locales de orientación**: En este paso, la representación gráfica de $|G|$ se divide en pequeñas regiones espaciales rectangulares denominadas celdas, las cuales se componen de una cantidad definida de pixeles adyacentes. Las celdas de 64 pixeles (la agrupación de [8 x 8] pixeles) son las que mejores resultados ofrecen [2].

Después, para cada celda se genera un histograma basado en la orientación de los pixeles; dentro de la celda, cada pixel aporta un valor ponderado para un canal del histograma de acuerdo con los valores obtenidos durante el cálculo del gradiente. Los canales del histograma pueden definirse dentro de los intervalos [0°- 360°] o [0°-180°], dependiendo de si se considera su signo o no. El valor de θ está en radianes y de ser necesario se convierte a grados. Estos histogramas son la representación básica del descriptor HOG. Es aquí donde se encuentra la no-linealidad del descriptor, pues aun cuando la codificación es sensible al contenido de la imagen, tolera leves cambios de pose o apariencia.

Para la obtención de los histogramas, Dalal [16] sugiere el empleo de gradientes sin signo, es decir en el intervalo de [0°-180°], en conjunto con histogramas de 9 canales para un mejor desempeño del descriptor. El peso ponderado, asignado a cada pixel contenido en las celdas, puede ser igual a la magnitud del gradiente, la raíz cuadrada de este o bien su elevación al cuadrado.

- Normalización del bloque: Para reducir los problemas debidos a las diferencias de iluminación y contraste, los gradientes habrán de normalizarse; para ello las celdas se agruparán en grandes bloques espacialmente conectados. Estos bloques, generalmente se traslapan de tal manera que cada celda contribuya en más de una vez en el descriptor final, lo que mejorará el rendimiento del sistema. En el trabajo de Dalal [16] se presentan tres diferentes métodos de normalización, ecuaciones (7) - (9), de los cuales puede seleccionarse uno:

$$L2 - norm: f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}, \tag{7}$$

$$L1 - norm: f = \frac{v}{\|v\|_1 + e}, \tag{8}$$

$$L1 - sqrt: f = \sqrt{\frac{v}{\|v\|_1 + e}}. \tag{9}$$

Dentro de las ecuaciones anteriores, v describe al vector no normalizado que contiene a todos los histogramas que conforman un bloque, mientras que $||v_k||$ es su el

vector *k*-normalizado de los histogramas por bloque, con $k = 1,2$. Finalmente, *e* es una constante cuyo valor es muy cercano a cero.

Para un mejor resultado, se generan bloques de cuatro celdas, es decir un arreglo de [2x2] celdas, estando estas conformadas por 64 pixeles.

- **Agrupación de los descriptores**: Se concatenan los descriptores de los bloques creados en el paso anterior, y de esta forma el descriptor final poseerá una descripción completa de la ventana de detección, que se emplea en un clasificador.

Clasificadores en cascada

Método propuesto por Viola y Jones [19], el cual, es descrito como una combinación sucesiva de clasificadores en una estructura en forma de cascada en la que la dificultad de los clasificadores incrementa etapa tras etapa.

De acuerdo con [20], el objeto buscado, en este caso el rostro, se localiza tras utilizar una sub-ventana que recorrerá la imagen de entrada (I), determinando si esta es positiva o negativa. Será considerada como positiva si, de acuerdo con los clasificadores, forma parte del objeto de interés, en ese caso la sub-ventana será evaluada por clasificadores cada vez más complejos, si llega hasta la última etapa entonces se entenderá que objeto ha sido localizado dentro de la imagen; mientras que será declarada negativa si en cualquiera de las etapas se determina que dicha sub-ventana no forma parte del objeto buscado. En la Fig. 5 se muestra el funcionamiento de dichos clasificadores.

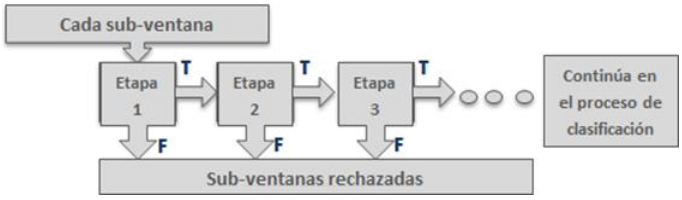


Fig. 5. Representación gráfica del sistema de clasificación por cascada.

El uso de los clasificadores en cascada incrementa considerablemente la velocidad de respuesta del clasificador pues centra su atención en las regiones de la imagen que resultan prometedoras para la localización del rostro, lo que desecha rápidamente las regiones que conforman el fondo de la imagen de entrada [14].

Entrenamiento del detector de rostro

Posterior a los resultados de las pruebas preliminares, se decidió entrenar un nuevo detector, con lo que se esperaba mejorar la capacidad de detección en rostros con ángulos mayores de inclinación y giro, es decir, reducir las restricciones de posición en el rostro para que este pueda ser detectado. También se espera que el recuadro de detección del nuevo detector sea consistente; seleccionando los rostros desde la parte superior de la frente hasta el límite inferior de la barbilla.

Selección y pre-procesamiento de imágenes de entrenamiento. Para llevar a cabo este proceso es necesario contar con imágenes positivas y negativas. Se le da el nombre

de imágenes positivas a aquellas que conforman el grupo de imágenes de entrenamiento que contienen el objeto que se desea detectar, en este caso el rostro.

Existen un gran número de bases de datos públicas con imágenes faciales [21], el seleccionar unas u otras depende del uso que se les quiera dar. Para aumentar la robustez del detector propuesto para este proyecto se decidió usar un alto número de imágenes positivas, por lo que fue necesario emplear más de una base de datos, para la selección de las mismas se consideraron las siguientes condiciones:

- Los rostros en las imágenes deberían ser de tamaño y resolución muy similar.
- Las imágenes deberían presentar diferencias de iluminación y fondo.
- Las imágenes debían mostrar variaciones de posición en el rostro contenido. Considerando el propósito para el que es creado tal detector, se establece, además, que las posiciones de los rostros en las imágenes deben permitir la localización de ambos ojos, nariz, boca y frente.
- Se emplearon únicamente bases de datos con imágenes de rostros reales, descartando aquellas que se conforman por imágenes de rostros creados con animación, tanto 2D como 3D.

Tras atender los puntos mencionados, se eligió un conjunto de poco más de 8 000 imágenes para fungir como muestras positivas para el entrenamiento. Dichas imágenes fueron tomadas de: The ORL Database of Faces [22], JAFFE Database [23], FEI Face Database Thomaz y Giraldi [24], Samaria y Harter [25] y Bainbridge *et al.* [26].

Tras seleccionar las imágenes, estas fueron pre-procesadas, mediante un recorte de tal manera que se retirara el fondo innecesario, dejando únicamente el área de interés (la que contiene al rostro). En este paso es importante que el área de interés recortada sea consistente con los requisitos de detección establecidos, de tal manera que los rostros en las imágenes seleccionadas luzcan exactamente como se espera que sean detectados los rostros una vez finaliza- do el detector. Se estandarizó el tamaño y color de las imágenes, todas ellas quedaron de (576 x 768) pixeles y convertidas a escala de grises.

Las imágenes resultantes se guardan en un directorio específico, y sus nombres se listan en un archivo *.txt, que será necesario en el proceso de entrenamiento, en este archivo se anotan también las coordenadas de localización del rostro en la imagen, así como el tamaño del rostro.

Las imágenes negativas para entrenamiento son aquellas que carecen del objeto buscado, que en este caso son rostros. Para la selección de imágenes se buscó que mostraran escenas en las que es común encontrar rostros, tales como oficinas e interiores de autos, por mencionar algunos. Además, se incluyeron imágenes que mostraran componentes de rostros que por sí mismos no pudieran ser considerados como un rostro en sí, como ojos, nariz y labios; con ello se buscó reducir los falsos positivos en los que se localizaran solo pequeñas partes del rostro, pues dada la aplicación a la que se enfoca este proyecto es importante que se realice una apropiada detección de rostros de tal manera que en ellos sean visibles ambos ojos, la frente y la boca.

Es importante que la cantidad de muestras negativas sea considerablemente superior a la de imágenes de entrenamiento positivas, por tal motivo se seleccionaron alrededor de 24 000 imágenes para este conjunto, estas fueron tomadas de páginas de Internet de

acceso público en su mayoría, aunque se incluyen también algunas imágenes tomadas de los repositorios online de la Universidad de Oxford, dichas imágenes fueron utilizadas únicamente para la fase de entrenamiento. Las imágenes seleccionadas se estandarizaron en color y, aun cuando no se estandarizaron en tamaño, se buscó que este fuera cerca de 4 veces mayor al de las muestras positivas. Este conjunto de imágenes se almacenó en un directorio dedicado, en el que se guardó también un registro de los nombres de cada una de las imágenes que conforman este grupo, lo cual será necesario durante el proceso de entrenamiento.

Una vez que las imágenes de entrenamiento, tanto positivas como negativas, se encontraron listas era factible el entrenamiento del clasificador. El clasificador fue entrenado mediante la función trainCascadeObjectDetector() de MATLAB, la cual soporta la identificación de objetos con ayuda de tres diferentes descriptores; Haar, LBP y HOG. Un esquema del entrenamiento se muestra en la Fig. 6.

De entre dichos descriptores se decidió utilizar los HOG, esto considerando el desempeño efectivo y eficiente de dicho descriptor. Al configurar la función para utilizar tales descriptores esta trabajará con los parámetros de bloques de [2 x 2] celdas, las cuales se conforman de [8 x 8] pixeles.

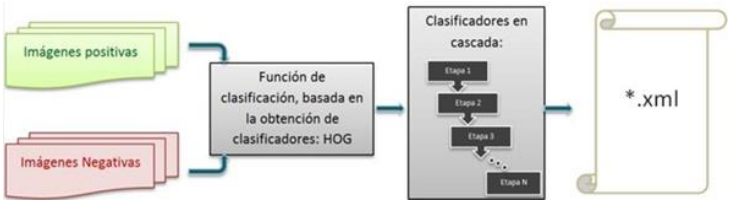
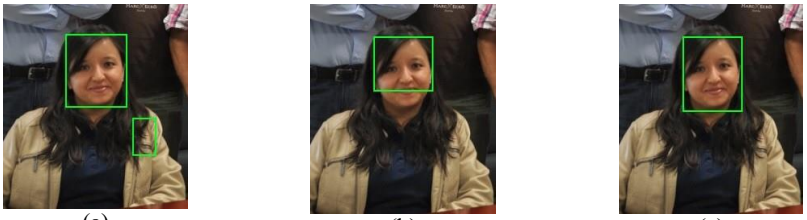


Fig. 6. Ilustración del principio básico de funcionamiento de la función de MATLAB trainCascadeObjectDetector().

Localización de rostro

Se probó el detector con las 2 511 imágenes de la base de datos citada por Gourier et al. (2004), de tal manera que se pudiera determinar el alcance del detector de rostros respecto a la posición de los mismos en la imagen.



(a) (b) (c)

Fig. 7. Muestras de la localización del rostro utilizando los detectores entrenados; (a) Ejemplo de detección incorrecta, (b) imagen ilustrativa de localización imprecisa, (c) muestra de detección correcta utilizando el detector HOG.

Después de comparar la eficiencia relativa de cada detector, se combinan los tres con mejor desempeño de tal manera la etapa de detección de rostro será más robusta, cumpliendo con el objetivo general de este proyecto, como se muestra en la Fig. 7.

Una vez que se implementó el detector combinado, fue sometido a las mismas pruebas a las que se habían sometido los detectores de forma individual. En la Tabla1 se muestran los porcentajes de detección que se obtuvieron con el detector combinado.

Tabla 1. Eficiencia del detector facial combinado, en relación con la posición facial del usuario.

Eficiencia del detector combinado			
Relación posición facial / eficiencia		**Relación posición facial / eficiencia**	
Posición facial	***Porcentaje de detección***	***Posición facial***	***Porcentaje de detección***
-15° / -15°	92.6	+15° / -15°	96.3
-15° / 0°	100	+15° / 0°	96.3
-15° / +15°	92.6	+15° / +15°	96.3
0° / -15°	96.3	+15° / +30°	96.3
0° / 0°	100	+30° / -15°	96.3
0° / +15°	100	+30° / 0°	96.3
0° / +30°	96.3	+30° / +15°	85.2

4. Resultados y discusión

Como producto de haber llevado a cabo las pruebas de detección, se determinó que podía mantener constantes sus resultados únicamente en cinco de las posiciones seleccionadas para realizar las pruebas, ello se refleja en un porcentaje de 78 % de detección imprecisa en las imágenes seleccionadas. Esto se observa en la Fig. 8.

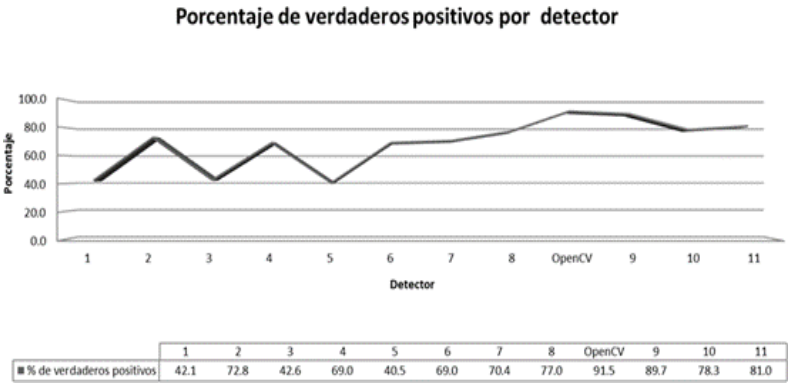
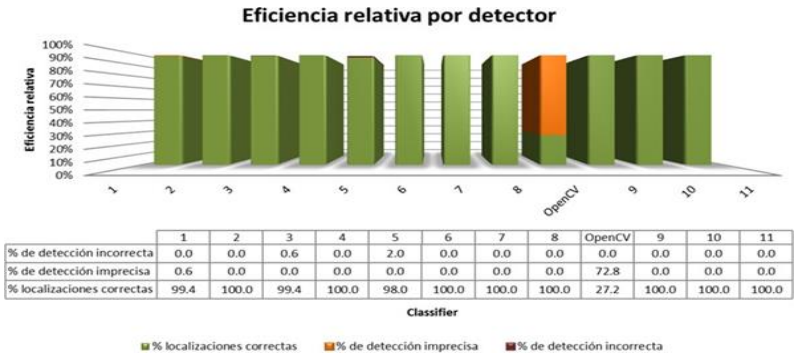


Fig. 8. Comparativo de la cantidad de detecciones hechas por cada detector, sin importar la precisión de la localización, solo se considera la apropiada clasificación de la imagen.

En la Fig. 9 se puede observar el porcentaje de eficiencia relativa para cada uno de los detectores entrenados y el detector de OpenCV, para ello se consideran únicamente

las detecciones realizadas, clasificándolas en detecciones imprecisas, erróneas y acertadas.



	1	2	3	4	5	6	7	8	OpenCV	9	10	11
% de detección incorrecta	0.0	0.0	0.6	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
% de detección imprecisa	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	72.8	0.0	0.0	0.0
% localizaciones correctas	99.4	100.0	99.4	100.0	98.0	100.0	100.0	100.0	27.2	100.0	100.0	100.0

Fig. 9. Gráfico de la eficiencia de detección para cada detector entrenado y el pre-entrenado de OpenCV; considerando el porcentaje de detecciones imprecisas, erróneas y acertadas.

5. Conclusiones y trabajo a futuro

En este trabajo se presentó un sistema eficiente para detección de rostros mediante la implementación de algoritmos descriptores de imágenes basadas en Histograma de Gradiente HOG. El método utilizó un entrenamiento de clasificadores el cual utilizó dos conjuntos de imágenes (positivas y negativas) con una relación de 1 a 3. Para validar el método, otro conjunto de imágenes se probó, resultando un porcentaje elevado de arriba del 85% de detección de rostro para posiciones horizontales, verticales, superiores e inferiores de más de 30 grados de variación con respecto a la vista frontal. Este resultado mejora el porcentaje que se utiliza con clasificadores basados en características como Haar cascades, los cuales tienen una variación de únicamente 15 grados. Este método se puede utilizar en sistemas de autenticación y/o reconocimiento facial, así como en sistemas de detección de emociones y/o detección de somnolencia, fatiga o aburrimiento de personas mediante la captura y procesamiento de expresiones faciales. Como trabajo futuro se puede llevar a cabo un estudio comparativo del método propuesto con otras técnicas de detección facial publicadas recientemente [27-29].

Agradecimientos. El desarrollo de este trabajo fue posible gracias al financiamiento del Fondo para el Fortalecimiento de la Investigación de la Universidad Autónoma de Querétaro (FOFIUAQ 2018).

Referencias

1. Modi, M., Macwan, F.: Face Detection Approaches: A Survey. International Journal of Innovative Research in science, engineering and technology 3(4), 11107–11116 (2014)

2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 1, pp. 886–893 vol. 1 (2005)
3. Yang, M.-H., Kriegman, D.J., Ahuja, N.: Detecting faces in images: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
4. Hjelmas, E., Low, B.K.: Face Detection: A Survey. Computer vision and image understanding 83(3), 236–274 (2001)
5. Ruiz Sancho, C.: Pedestrian Detection using a boosted cascade of Histogram of Oriented Gradients (2014)
6. Negri, P., Clady, X., Prevost, L.: Benchmarking haar and histograms of oriented gradients features applied to vehicle detection. In: ICINCO-RA (1). pp: 359–364 (2007)
7. Nixon, M.S., Aguado, A.S.: Feature extraction & image processing for computer vision. 3rd ed. Academic Press (2012)
8. Prince, S.J.D.: Computer vision: models, learning, and inference. Cambridge University Press (2012)
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. 2, pp: 1150–1157 (1999)
10. Awad, A.I., Hassaballah, M.: Image Feature Detectors and Descriptors. 630 of Studies in Computational Intelligence Springer International Publishing Cham (2016)
11. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features (2006)
12. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: IEEE 6h International Conference on Computer Vision, pp. 555–562 (1998)
13. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to data mining. The MathWorks Inc. 2017. Train a Cascade Object Detector (2006)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR, pp. 511–518 (2001)
15. Mordvintsev, A., Abid, K.: Face Detection using Haar Cascades (2013)
16. Dalal, N.: Finding People in Images and Videos Theses Institut National Polytechnique de Grenoble - INPG (2006)
17. Del Abril De Mur, A.: Detección de personas en secuencias de imágenes en entornos interiores. PhD thesis Universidad de Alcalá (2017)
18. Mallick, S.: Histogram of Oriented Gradients. Learn OpenCV (2017)
19. Viola, P., Jones, M.: Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96 3 (14), 2 (2003)
20. Soo, S.: Object detection using Haar-cascade Classifier. Institute of Computer Science, University of Tartu (2014)
21. Grgic, M., Delac, K.: Face Recognition Homepage: Databases, http://www.face-rec.org/databases/ (2017)
22. AT&T Laboratories Cambridge. The ORL Database of Faces. Audi AG. 2011a. Adaptive light. https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (2002)
23. Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J.: Coding facial expressions with Gabor wavelets. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200–205 (1998)
24. FEI Face Database Thomaz y Giraldi, https://fei.edu.br/~cet/facedatabase.html (2010)
25. Samaria, F.S., Harter, A.C.: Parameterization of a stochastic model for human face identification. In: Proceedings of 1994 IEEE Workshop on Applications of Computer Vision, pp. 138–142 (1994)
26. Bainbridge, W.A., Isola, P., Oliva, A.: The Instrict Memorability of Face Photograps. Journal of Experimental Psychology 142(4), 1323–1334 (2013)

27. Sharma, R., Ashwin, T.S., Guddeti, R.M.R.: A Novel Real-Time Face Detection System Using Modified Affine Transformation and Haar Cascades. In: Recent Findings in Intelligent Computing Techniques, pp. 193–204. Springer, Singapore. (2019)
28. Rehman, B., Ong, W.H., Tan, A.C.H., Ngo, T.D.: Face detection and tracking using hybrid margin-based ROI techniques. The Visual Computer, pp. 1–15 (2019)
29. Huang, J., Shang, Y., Chen, H.: Improved Viola-Jones face detection algorithm based on HoloLens. EURASIP Journal on Image and Video Processing (1), 41 (2019)

ISSN 1870-4069

Desarrollo de un guía virtual conversacional para museos en un entorno inmersivo de realidad virtual y aumentada

Isidro Hernán Martínez Juárez[1], María Eugenia Puga Nathal[1], Juan Martínez-Miranda[2]

[1] Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Guzmán, Ciudad Guzmán, Jalisco, México
[2] CONACyT – Centro de Investigación Científica y Educación Superior de Ensenada, Unidad de Transferencia Tecnológica Tepic (CICESE-UT3), Tepic, Nayarit, México
{isidroM18290017, mepuga}@itcg.edu.mx
jmiranda@cicese.mx

Resumen. El uso de agentes virtuales conversacionales, como un tipo de interfaces inteligentes de interacción, son cada vez más utilizados en diferentes áreas de aplicación. Una de estas áreas es el sector turístico, en el cual estos agentes actúan como guías turísticos para proveer información más detallada y personalizada acorde a los intereses de los turistas. En este artículo se describen las etapas iniciales del desarrollo de un agente virtual conversacional el cual actuará como guía de visitantes a un museo. En concreto, se presenta el diseño del mecanismo para generar el comportamiento emocional en el agente a partir de la interacción con el usuario. Una diferencia con trabajo similares es que el guía virtual propuesto estará ubicado en un entorno inmersivo de realidad virtual y aumentada. Esto permitirá evaluar si la interacción en este tipo de entornos mejora la experiencia de interacción con el usuario respecto a agentes virtuales ubicados en interfaces de usuario tradicionales.

Palabras clave: agente virtual, realidad virtual y aumentada, modelo cognitivo-emocional, sistemas conversacionales.

Development of a Conversational Virtual Guide for Museums in an Immersive Environment of Virtual and Augmented Reality

Abstract. The use of conversational virtual agents, as a type of intelligent interaction interfaces, is increasingly in different areas of application. One of these areas is the tourism sector, where these agents act as touristic guides able to provide more detailed and personalized information according to the interests of the tourists. This article describes the initial stages in the development of a conversational virtual agent that will act as a guide for the visitors in a museum. In particular, the design of the mechanism to generate the emotional behavior in the virtual agent during the interaction with the user is presented. A difference with similar works is that the proposed virtual guide will be located in an immersive environment of virtual and augmented reality. This will allow to evaluate if the interaction in this type of environments improves the user's

interaction experience regarding existent virtual agents located in traditional user interfaces.

Keywords: virtual agents, virtual and augmented reality, cognitive emotional model, conversational systems.

1. Introducción

El sector turístico es una de las áreas en las que el uso de las TIC está cambiando la manera de promocionar, ofrecer y proveer servicios por parte de todos los actores relevantes dentro del sector a nivel mundial. Además de las aplicaciones *clásicas* tales como la gestión de reservas en hoteles, vuelos y restaurantes a través de Internet y el uso de las guías de viaje electrónicas, se han desarrollado sistemas inteligentes e interactivos, así como aplicaciones móviles que han ayudado a aumentar la calidad de los servicios turísticos y mejorar las experiencias de los viajeros. El objetivo principal de estas aplicaciones es el atraer y retener a los visitantes de los diferentes destinos que se promocionan en cualquier parte del mundo, así como mejorar la competitividad de las empresas públicas y privadas proveedoras de servicios turísticos.

Una de las tecnologías que desde ya hace algunos años ha comenzado a aplicarse en el sector turístico es la realidad virtual y aumentada la cual ofrece el potencial de mejorar la experiencia de los turistas mediante el modelado de los alrededores del usuario en un espacio tridimensional y proveer información complementaria sobre un determinado contexto (ej. un punto de interés turístico) con apariencia de coexistir con el mundo real del turista. Además de la información que se pueda modelar como parte del ambiente en estos entornos inmersivos, un componente capaz de mantener una interacción personalizada y proporcionar información específica a petición expresa del usuario es el conocido como Agente Virtual Conversacional. Estos agentes virtuales toman la apariencia de una persona y son desarrollados como una herramienta avanzada de interacción hombre-máquina. Para comunicarse con el usuario estos agentes implementan un mecanismo para la generación de diálogos basados en lenguaje natural combinados con expresiones faciales y movimientos corporales para simular una conversación persona a persona [1].

En sus inicios esta tecnología fue principalmente utilizada con objetivos de aprendizaje y/o entrenamiento en actividades concretas [2, 3], así como en el área de videojuegos actuando como personajes autónomos complementarios al jugador ("*non-player characters*") [4, 5]. Sin embargo, en los últimos años estos agentes han mostrado su utilidad y beneficios en una gran área de aplicaciones, tales como sistemas interactivos dentro del sector salud [6–9], turístico [10, 11], e incluso como asistentes personales o acompañantes para adultos mayores [12]. Cuando el comportamiento y la apariencia de estos agentes son generados de la manera correcta, el agente virtual puede establecer un buen nivel de confianza con el usuario maximizando su utilidad y consecución de los objetivos para los cuales fueron creados [13].

Diferentes líneas de investigación son necesarias para el desarrollo de estos agentes virtuales para lograr el objetivo de generar una apariencia y comportamientos coherentes y acordes a los objetivos de la aplicación. Estas áreas de investigación incluyen el modelado y animación por computadora para hacer que el personaje gráfico se vea y mueva de una manera realista; técnicas de inteligencia artificial y computación

afectiva para crear modelos artificiales capaces de generar un comportamiento cognitivo y emocional que ayude a generar una mayor aceptabilidad por parte de los usuarios; o las áreas de psicología, sociología y ciencias cognitivas como bases teóricas para la construcción de estos modelos artificiales [14, 15].

El trabajo presentado en este artículo describe las primeras etapas en el diseño y desarrollo de un agente virtual conversacional con el objetivo de que proporcione información a los visitantes de un museo. Este agente virtual estará ubicado en un entorno virtual e inmersivo y adoptará el rol de un guía capaz de interactuar con los visitantes del museo y ofrecer información relevante sobre el material de exhibición de la sala en la que se encuentre visitando el turista. En particular, este trabajo presenta el diseño interno de este guía virtual mediante el cual se generarán los diálogos y el comportamiento emocional durante la interacción con el usuario. También se presenta el mecanismo de extracción de información para proporcionar datos relevantes al usuario sobre lo que se encuentra visitando dentro del museo.

El principal objetivo de este trabajo es valorar el impacto de esta tecnología en la experiencia turística de los visitantes a un museo. Uno de los objetivos particulares es medir el nivel de aceptabilidad y adherencia de los usuarios hacia un guía virtual inmerso en un entorno de realidad virtual y aumentada respecto a trabajos similares en los que el agente virtual se encuentra ubicado en una interfaz estática. El resto de las secciones del artículo presentan lo siguiente: la sección 2 describe algunos de los trabajos relacionados. La sección 3 presenta el trabajo inicial desarrollado para la construcción del modelo interno del agente, poniendo énfasis en el modelo cognitivo-emocional y en el mecanismo de generación de diálogos incluida la extracción de información sobre el área del museo que el turista se encuentra visitando y proporcionársela al usuario. Finalmente, la sección 4 presenta algunas conclusiones y el trabajo futuro para completar el desarrollo de este trabajo.

2. Trabajo relacionado

En años recientes, algunos museos han comenzado a utilizar agentes virtuales conversacionales como guías de los visitantes a los museos. Un ejemplo es el caso del Museum of Science, Boston (MoS) que en colaboración con el USC Institute for Creative Technologies han desarrollado *Ada* y *Grace*, un par de agentes mellizos que interactúan con los visitantes del MoS a través del procesamiento de lenguaje natural y usando una apariencia realística para atraer e incrementar el interés del público, principalmente estudiantes de secundaria, en áreas de ciencia y tecnología [16]. Una evaluación hecha a niños de edad entre 7 y 14 años demostró el impacto logrado por Ada y Grace en 14 de 15 indicadores en distintas áreas, entre las que se encontró un incremento en la atracción e interés hacia la informática y la tecnología; una actitud positiva hacia estas dos áreas, así como un aumento de su conciencia y conocimientos sobre dichas áreas. Las condiciones que permitieron esta evaluación fueron dos: interacción directa con el visitante y una combinación de la interacción con el personal del museo y el visitante. Los instrumentos de evaluación incluyeron la observación directa, entrevistas a los usuarios y la aplicación de cuestionarios de seguimiento [10].

Otro ejemplo de guía virtual es Max, quien interactúa con visitantes del Heinz Nixdorf Museums Forum. Max se encuentra dentro de una pantalla con una cámara que

detecta el paso de los visitantes de tal forma que él puede darse cuenta de la presencia de las personas y actuar como su guía con voz sintética en alemán, gestos, movimientos y expresiones faciales lo más naturales posibles. El agente está orientado principalmente a involucrar a los usuarios en conversaciones en las que él les proporciona información relevante del museo, así como de las exhibiciones dentro de él y otros temas de interés, de tal manera que sea comprensible e interesante. La interacción se da a través del procesamiento de lenguaje natural ya sea por voz o bien usando un teclado conectado al sistema [17].

Un tercer ejemplo de guía virtual es MINERVA el cual, a diferencia de los dos anteriores, es empleado en casos donde el usuario no está presente físicamente. MINERVA actúa como un agente "telepresente" que guía a los usuarios a través de diferentes espacios del Smithsonian's National Museum of American History vía acceso remoto a través de Internet [11].

De manera complementaria, durante los últimos años múltiples proyectos de investigación han comenzado a crear aplicaciones de realidad aumentada (AR por sus siglas en inglés) que utilizan agentes de realidad aumentada, o AuRA (Augmented Reality Agents), como su principal paradigma de interacción y desarrollo [18]. La mayoría de las aplicaciones de realidad aumentada están relacionadas principalmente con permitir que un usuario explore un mundo virtual en 3D mezclado con el mundo real. Las interfaces AR más avanzadas permiten al usuario interactuar con el entorno mixto, pero la parte virtual suele ser bastante finita y determinista. Este tipo de aplicaciones pueden hacer uso de "agentes de realidad aumentada" los cuales son un caso particular de agentes virtuales. Su punto fuerte es que pueden apelar a múltiples sentidos e involucrar a un usuario en una conversación aún más realista y similar a la que se da de persona a persona [19].

3. Diseño del mecanismo interno del agente virtual

3.1. Generación del comportamiento

El trabajo llevado a cabo para el desarrollo del guía virtual conversacional se ha concentrado durante las primeras fases en obtener la arquitectura interna que generará el comportamiento del agente durante la interacción con el usuario. Basado en trabajos previos en los cuales se ha demostrado que un comportamiento consistente, coherente y empático en el agente virtual ayuda a promover una mayor credibilidad, aceptabilidad y nivel de confianza en los usuarios [14, 20], se ha trabajado en el mecanismo interno para generar un comportamiento de este estilo.

Actualmente existen diferentes arquitecturas computacionales que pueden ser integradas como el mecanismo interno para la generación de un comportamiento cognitivo-emocional tanto en agentes virtuales como en robots sociales [15]. Una de las teorías en la que la mayoría de estas arquitecturas se basan para implementar el mecanismo de generación de emociones es la conocida como teoría de valoración cognitiva [21, 22]. De manera general, esta teoría postula que los eventos que se producen en el entorno de una persona son constantemente (consciente o inconscientemente) evaluados acorde a las metas, creencias y preferencias del individuo. Esta evaluación *cognitiva* genera una respuesta emocional (de acuerdo con

la relevancia que el individuo asigna al evento ocurrido), lo cual a su vez genera un comportamiento específico para hacer frente a dicho evento.

Basado es esa teoría, el modelo interno de nuestro guía virtual (ver Fig. 1) implementa un mecanismo de valoración de los eventos ocurridos en su entorno. En nuestro caso, estos eventos son las acciones y diálogos generados por el usuario durante su recorrido por el museo y la interacción que mantiene con el agente. La valoración sobre el evento detectado puede producir una emoción (positiva o negativa) con una intensidad determinada acorde a la importancia que le asigna el agente al evento ocurrido. El conjunto de emociones que se generan en el agente está basado en el modelo OCC [23], el cual describe una organización jerárquica de 22 emociones a partir del proceso de valoración cognitiva respecto a las metas internas del agente virtual.

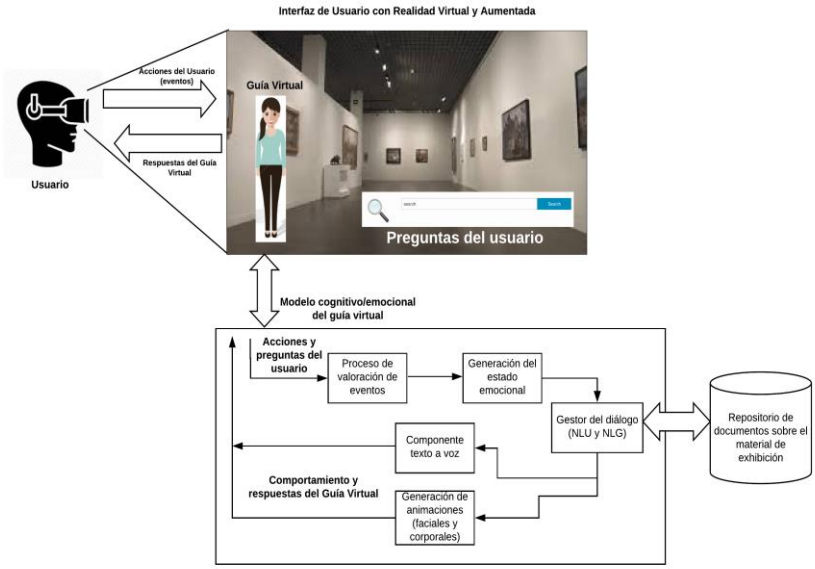


Fig. 1. Mecanismo Interno del Agente Virtual.

El estado emocional producido en el agente es utilizado por el componente gestor del diálogo para construir las oraciones –basadas en lenguaje natural– que se utilizarán como respuesta a las acciones del usuario. Una vez construido el diálogo, esta información y el estado emocional generado en el agente son procesados por el componente de texto a voz, y por el componente que genera las animaciones de expresiones faciales y movimientos corporales para reproducir las respuestas verbales y no verbales del agente hacia el usuario.

Para ejemplificar el mecanismo completo, considérese el siguiente escenario: el usuario visitante al museo recorre las piezas exhibidas en una sala y a través del componente de realidad aumentada se detecta que el usuario está observando un objeto determinado. Esta acción (evento para el agente que puede ser etiquetado como *objeto_atencion_usuario*) es identificada, la cual es evaluada por el componente de

valoración de eventos. Una de las metas internas pre-definidas del agente puede ser *ofrecer_informacion_objeto*. En consecuencia, la ocurrencia de dicho evento puede ser valorada positivamente generando una emoción positiva y activando dicha meta. La manera de llevar a cabo esta meta es a través de una determinada acción, por ejemplo, *confirmar_interes_objeto*.

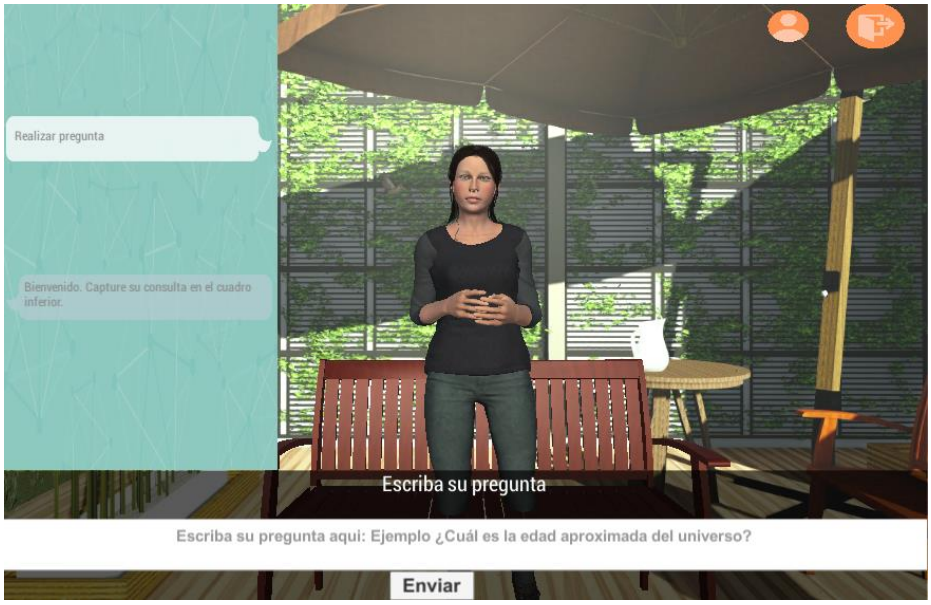


Fig. 2. Ejemplo de guía virtual dando la bienvenida al usuario.

Esta acción se puede considerar un acto de diálogo, el cual es transformado por el gestor del diálogo en una frase concreta construida a partir de la emoción generada en el agente, y un conjunto de frases pre-definidas para implementar los diferentes actos de diálogos que pueden ocurrir durante la interacción.

En este caso, el acto de diálogo *confirmar_interes_objeto* junto a la emoción positiva del agente puede generar tanto el comportamiento verbal (ej. "Me alegra que te interese ese objeto! –*parte emocional*– ¿Quieres saber más sobre él? –acción para confirmar el interés del usuario en el objeto–"); y no verbal (ej. el agente muestra una sonrisa) para dirigirse al usuario. La respuesta del usuario a la pregunta del agente se evaluará como un nuevo evento, iniciando el siguiente ciclo de interacción.

3.2. Extracción de información

Un componente importante adicional al mecanismo de generación del comportamiento del agente virtual es la manera de extraer y proveer la información sobre los objetos exhibidos en el museo. En nuestro caso, el guía virtual puede ofrecer información de un determinado objeto de dos formas: *i)* al detectar que el usuario está observando el objeto; o *ii)* a pregunta expresa del usuario. En cualquiera de las dos formas es necesario implementar el mecanismo de extracción de información sobre

documentos que contengan información sobre el dominio del objeto o pregunta hecha por el usuario. Estos documentos pueden ser los provistos por el propio museo, o incluso se puede extraer información de repositorios abiertos tales como Wikipedia.

Si la consulta de información es generada a partir de una pregunta realizada por el usuario, el sistema gestor de diálogo del agente tendrá integrado un componente de análisis y reconocimiento de lenguaje natural similar al que implementan los sistemas Question and Answering (QA) [24]. Este componente, el cual integrará algoritmos basados en ranking permitirá ir refinando y mejorando las respuestas ofrecidas a los usuarios a partir de recolectar la retroalimentación de los diferentes visitantes sobre la relevancia de la información proporcionada por el agente.

El desarrollo de este guía virtual está en sus primeras fases y de momento se tiene implementado únicamente el mecanismo interno de generación del comportamiento. Este mecanismo genera un conjunto de respuestas básicas para interactuar con el usuario (ver Fig. 2). El diseño y desarrollo del sistema QA para proveer información sobre un objeto y dominio determinado es el siguiente componente que se desarrollará y permitirá hacer una evaluación inicial completa del agente virtual antes de ubicarlo en el entorno inmersivo de realidad virtual y aumentada.

4. Conclusiones y trabajo futuro

El trabajo presentado en este documento describe las etapas iniciales en el desarrollo de un agente virtual que sirva como guía de los visitantes a un museo. Se espera que a través de este agente, los usuarios tengan acceso a una información más personalizada y contextualizada sobre las salas de exhibición que se encuentran visitando. La hipótesis de investigación de este trabajo es que este tipo de interfaces mejorarán la experiencia del usuario, ayudando a incrementar el interés y conocimiento sobre las temáticas expuestas en el museo. De manera concreta, se presentó el diseño y adaptación del componente cognitivo-emocional como el mecanismo interno de generación del comportamiento del agente virtual durante la interacción con el usuario.

El trabajo por desarrollar en las siguientes etapas contempla la implementación del sistema QA para procesar las preguntas concretas del usuario, la extracción de información de documentos sobre un dominio determinado, así como el desarrollo de la interfaz de realidad virtual y aumentada en la cual estará ubicado el guía virtual. De manera paralela se identificarán las métricas de evaluación del sistema, el cual incluirá información objetiva (ej. logs del sistema durante la interacción del usuario con el sistema) y subjetiva (ej. cuestionarios, entrevistas) que permita establecer el nivel de adherencia, usabilidad y aceptabilidad por parte de los usuarios. Finalmente, una vez terminado el desarrollo, el sistema se validará en un museo con un conjunto de visitantes a los cuales se invitará a probar el sistema.

Se prevé que la versión con las que se llevarán a cabo las pruebas de evaluación sea a través de un dispositivo móvil y un equipo de realidad virtual. El equipo de realidad virtual se le proporcionará al usuario durante su recorrido por la(s) sala(s) del museo y consistirá en unos lentes mediante los cuales el usuario experimentará el entorno inmersivo en el cual se encuentre ubicado el guía virtual. Para la evaluación se establecerá un protocolo que permita identificar el tipo de interacción que se llevará a cabo con el usuario a partir de características tales como edad, sexo, preferencias sobre

la apariencia del agente virtual, etc., con el fin de ofrecer sesiones interactivas personalizadas.

A partir de este piloto de evaluación, se podrá valorar el impacto de esta tecnología sobre los usuarios objetivo y se identificarán las ventajas y desventajas de estos guías virtuales en el contexto propuesto y reconocer para qué tipo de usuarios, salas o exhibiciones sería más adecuado su uso.

Referencias

1. Cassell, J., Sullivan, J., Prevost, S., Churchill, E. (eds.): Embodied conversational agents. Cambridge, MA: MIT Press (2000)
2. Core, M., Traum, D., Lane, H.C., Swartout, W., Gratch, J., van Lent, M., Marsella, S.: Teaching Negotiation Skills through Practice and Reflection with Virtual Humans. Simulation 82(11), 685–701 (2006)
3. Martínez-Miranda, J., Jung, B., Payr, S., Petta, P.: The Intermediary Agent's Brain: Supporting Learning to Collaborate at the Inter-Personal Level. In: Proceedings of the 7th Conference on Autonomous Agents and Multiagent Systems (AAMAS'08) 3: pp. 1277–1280 (2008)
4. Gebhard, P., Schröder, M., Charfuelan, M., Endres, C., Kipp, M., Pammi, S., Rumpler, M., Türk, O.: IDEAS4Games: Building Expressive Virtual Characters for Computer Games. In: Proceedings of the 8th International Conference on Intelligent Virtual Agents (IVA 2008), pp. 426–440 (2008)
5. Mateas, M., Stern, A.: Façade: An experiment in building a fully-realized interactive drama. In: Game Developers Conference (GDC'03) (2003)
6. Zhang, Z., Bickmore, T.W., Mainello, K., Mueller, M., Foley, M., Jenkins, L., Edwards, R.A.: Maintaining Continuity in Longitudinal, Multi-method Health Interventions Using Virtual Agents: The Case of Breastfeeding Promotion. In: Proceedings of Intelligent Virtual Agents (IVA 2014), pp. 504–513 (2014)
7. Micoulaud-Franchi, J.A., Sagaspe, P., de Sevin, E., Bioulac, S., Sauteraud, A., Philip, P.: Acceptability of Embodied Conversational Agent in a Health Care Context. In: Proceedings of Intelligent Virtual Agents (IVA 2016), pp. 416–419 (2016)
8. Bickmore, T.W., Utami, D., Matsuyama, R., Paasche-Orlow, M.: Improving Access to Online Health Information with Conversational Agents: A Randomized Controlled Experiment. Journal of Medical Internet Research. 18(1) (2016)
9. Martínez-Miranda, J.: Embodied Conversational Agents for the Detection and Prevention of Suicidal Behaviour: Current Applications and Open Challenges. Journal of Medical Systems, 41(9), 135 (2017)
10. Traum, D., Aggarwal, P., Artstein, R., Foutz, S., Gerten, J., Katsamanis, A., Leuski, A., Noren, D., Swartout, W.: Ada and Grace: Direct Interaction with Museum Visitors. Intelligent Virtual Agents, pp. 245–251 (2012)
11. Richards, D., Agent-based museum and tour guides: applying the state of the art. In: Proceedings of the 8th Australasian Conference on Interactive Entertainment: Playing the System (IE '12). ACM, New York, NY, USA, Article 15, 9 pages (2012)
12. Shaked N.A.: Avatars and virtual agents – relationship interfaces for the elderly. Healthcare Technology Letters 4(3), 83–87 (2017)
13. Lucas, G.M., Gratch, J., King, A., Morency, L.P.: It's only a computer: Virtual humans increase willingness to disclose. Computers in Human Behavior 37, pp. 94–100 (2014)
14. Scherer, K.R.: Emotion and emotional competence: conceptual and theoretical issues for modelling agents. In: Scherer, K.R., Bänziger, T., Roesch, E.B. (eds.): Blueprint for Affective Computing: A Sourcebook; Chap. 1.1. Oxford University Press; pp. 3–20 (2010)

15. Marsella, S., Gratch, J., Petta, P. Computational Models of Emotion. In: Scherer, K.R., Bänziger, T., Roesch, E.B. (eds.): Blueprint for Affective Computing: A Sourcebook. Chapter 1.2. pp. 21–46. Oxford University Press (2010)
16. Swartout, W., Traum, D., Artstein, R., Noren, D., Debevec, P., Bronnenkant, K., Williams, J., Leuski, A., Narayanan, S., Piepol, D., Lane, C., Morie, J., Aggarwal, P., Liewer, M., Chiang, J., Gerten, J., Chu, S., White, K.: Ada and Grace: Toward Realistic and Engaging Virtual Museum Guides. In: International Conference on Intelligent Virtual Agents. Intelligent Virtual Agents, pp. 286–300 (2010)
17. Kopp, S., Gesellensetter, L., Krämer, N., Wachsmuth, I.: A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In: International Workshop on Intelligent Virtual Agents. Intelligent Virtual Agents, pp. 329–343 (2005)
18. Campbell, A.G., Stafford, J.W., Holz, T., O'Hare, G.M.P.: Why, when and how to use augmented reality agents (AuRAs). Virtual Reality 18, pp. 139–159 (2014)
19. Barakonyi, I., Schmalstieg, D.: Augmented reality agents for user interface adaptation. Computer Animation and Virtual Worlds 19, pp. 23–35. Published online in Wiley InterScience (www.interscience.wiley.com) (2008)
20. Ortony, A.: On making believable emotional agents believable. Emotions in Humans and Artifacts, Cambridge, MA: MIT Press, pp. 189– 211 (2001)
21. Scherer, K., Schorr, A., Johnstone, T.: Appraisal Processes in Emotion: Theory, Methods, Research. Oxford University Press (2001)
22. Lazarus, R.: Emotion and Adaptation. Oxford University Press (1991)
23. Ortony, A., Clore, G., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press (1998)
24. Ito, T., Iwama, Y.: Implementation of Question Answering System Based on Reference-Based Ranking Algorithm. In: Ali, M., Pan, J.S., Chen, S.M., Horng, M.F. (eds): Modern Advances in Applied Intelligence. IEA/AIE 2014. Lecture Notes in Computer Science, vol 8481. Springer, Cham (2014)

ISSN 1870-4069

Topología de una RNA para calcular la cinemática inversa del Puma 560 para aplicaciones en tiempo real

David Bedolla-Martínez[1], Felipe de Jesús Trujillo-Romero[2], Esther Lugo-González[3]

[1] Universidad de la Sierra Juárez, Instituto de Estudios Ambientales, Ixtlán de Juárez, Oaxaca, México
[2] Universidad de Guanajuato, Departamento de Ingeniería Electrónica, División de Ingenierías Campus Irapuato-Salamanca, Guanajuato, México
[3] Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica, Huajuapan de León, Oaxaca, México
bedolla@unsij.edu.mx, elugog@mixteco.utm.mx, fdj.trujillo@ugto.mx

Resumen. Se presenta una red neuronal artificial que resuelve la cinemática inversa de un robot manipulador Puma 560 de 6 grados de libertad, utilizando el menor número de parámetros en su topología que aseguren un buen desempeño. La topología de la red neuronal artificial se obtuvo usando un algoritmo genético para determinar el número de neuronas en las capas ocultas. Como resultado se presenta una red neuronal para eficientar la respuesta de los resultados para su implementación en tiempo real. Se muestran gráficas del error cuadrático medio, así como la comparación del método analítico y la red neuronal para los 6 grados de libertad.

Palabras clave: cinemática inversa, Puma 560, red neuronal artificial, algoritmo genético.

Topology of an RNA to Calculate the Inverse Kinematics of the Puma 560 for Real-Time Applications

Abstract We present an artificial neural network that solves the inverse kinematics of a Puma 560 manipulator robot with 6 degrees of freedom, using the least number of parameters in its topology that ensure good performance. The topology of the artificial neural network was obtained using a genetic algorithm to determine the number of neurons in the hidden layers. As a result, a neural network is presented to improve the response of the results for its implementation in real time. Graphs of the mean square error are shown, as well as the comparison of the analytical method and the neural network for the 6 degrees of freedom.

Keywords: inverse kinematics, Puma 560, artificial neural network, genetic algorithm.

1. Introducción

Los robots manipuladores son de gran importancia debido a las diferentes aplicaciones donde se usan, desde la industria manufacturera hasta servicios de robótica

de rehabilitación. Para desarrollar estas tareas se debe conocer la posición y orientación del efector final. La acción individual de cada junta debe ser controlada de tal forma que se ejecute la trayectoria de movimiento deseado. Específicamente, la localización espacial del efector final se determina con el desplazamiento traslacional y rotacional de cada junta con respecto a una referencia dada [1].

La cinemática inversa de un robot manipulador tradicionalmente se obtiene de forma analítica donde se usan funciones trigonométricas que necesitan varias operaciones para obtener un valor aproximado. En el área de sistemas embebidos que operan en tiempo real, los ciclos de reloj que toma cada operación son críticos, al igual que el espacio en memoria. Por esta razón se utilizarán como método alternativo las RNA's (redes neuronales artificiales).

Las RNA's se han utilizado para el cálculo de la cinemática inversa de robots manipuladores de 3 grados de libertad [2] y robots bípedos [3]. Una alternativa es encontrar la mejor topología de una RNA para minimizar el error en el cálculo de la cinemática inversa, variando los parámetros de aprendizaje y los pesos iniciales [4].

Durante la optimización, es necesario modificar los parámetros de la topología. Este proceso se realiza comúnmente a prueba y error [5] y [6], necesitando una inversión considerable de tiempo y experiencia por tanto se usa un Algoritmo Genético (AG).

Los AG's han sido usados para optimización o entrenamiento de RNA's [7]. En [8] se probó la técnica diseño robusto de RNA's para obtener los parámetros que consigan el mejor diseño de una RNA para resolver un problema específico. En [9] se usa un AG para entrenar una RNA de tipo Feedforward. En [10], [11] y [12] se realiza una búsqueda de la mejor topología de una RNA para conseguir un mejor desempeño usando entrenamiento estándar.

En [13] una RNA se usa en combinación de un algoritmo genético para el entrenamiento. Aunque hay algoritmos para entrenar RNA's, es necesario presentar una topología predeterminada, comúnmente se han usado AG's para obtener dicha topología.

En este trabajo se presenta la topología de una RNA que minimiza el error de entrenamiento usando el algoritmo de Levenberg-Marquardt, dicha topología se encontró usando un algoritmo genético.

1.1. Cinemática inversa del Puma 560

Las siguientes ecuaciones se toman de [14], El Unimation Puma 560 es un robot con seis grados de libertad con juntas rotativas.

De la tabla 1 se tiene $\propto_i$, a_i, d_i y θ_i que son los parámetros Denavit-Hartenberg del robot de 6 grados de libertad.

Tabla 1. Parámetros Denavit-Hartenberg.

i	$\propto_i - 1$	$a_i - 1$	d_i	θi
1	0	0	0	θ_1
2	-90°	0	0	θ_2
3	0	a_2	d_3	θ_3
4	-90°	a_3	d_4	θ_4
5	90°	0	0	θ_5
6	-90°	0	0	θ_6

La matriz de transformación homogénea se muestra en la ecuación (1), donde $p_{x,y,z}$ representa la posición del efector final. Los valores de r_{ij} representan la matriz de orientación.

$$ {}^{0}_{6}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1} $$

La solución de la cinemática inversa está dada por las ecuaciones (2-8). Los valores de s_i y c_i, corresponden a los valores de senos y cosenos de θ_i respectivamente y los valores de a_i y d_i, corresponden a los parámetros Denavit-Hartenberg.

Para θ_1 se escribe como:

$$ \theta_1 = atan2(p_x, p_y) - atan2\left(d_3, \pm\sqrt{p_x^2 + p_y^2 - d_3^2}\right), \tag{2} $$

donde

$$ d_3 = -s_1 p_x + c_1. \tag{3} $$

Para θ_3

$$ \theta_3 = atan2(a_3, d_4) - atan2\left(\mathrm{K}, \pm\sqrt{a_3^2 + d_4^2 - K^2}\right), \tag{4} $$

donde

$$ K = \frac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}. \tag{5} $$

Para θ_2

$$ \begin{aligned} \theta_2 = atan2\Big((-a_3 - a_2c_3)p_z - \big(c_1p_x + s_1p_y\big)(d_4 - a_2s_3), (a_2s_3 - d_4)p_z \\ - (a_3 + a_2c_3)\big(c_1p_x + s_1p_y\big)\Big). \end{aligned} \tag{6} $$

Para θ_4

$$ \theta_4 = atan2(-r_{13}s_1 + r_{23}c_1, -r_{13}c_1c_{23} - r_{23}s_1c_{23} + r_{33}s_{23}). \tag{7} $$

Para θ_5

$$ \begin{aligned} \theta_5 = atan2(&-r_{11}(c_1c_{23}s_4 - s_1c_4) - r_{21}(s_1c_{23}s_4 + c_1c_4) \\ &+ r_{31}(s_{23}s_4), r_{11}[(c_1c_{23}c_4 + s_1s_4)c_5 - c_1s_{23}s_5] \\ &+ r_{21}[(s_1c_{23}c_4 - c_1s_4)c_5 - s_1s_{23} \\ &- r_{31}(s_{23}c_4c_5 + c_{23}s_5)]). \end{aligned} \tag{8} $$

El ángulo θ_6 modifica la orientación a lo largo del eje z del efector final.

2. Metodología

Los datos de entrenamiento, validación y prueba, se generaron de forma analítica usando las ecuaciones para el robot Puma 560 (ecuaciones (2)-(8)). El toolbox de P. Corke, "Robotic Toolbox" [15] se usó para generar los ejemplos de entrenamiento, los cuales se encuentran dentro de su espacio de trabajo (ver figura 1). Cada ejemplo consiste en 6 valores para definir la posición y orientación del efector final y 6 valores para definir la solución de la cinemática inversa.

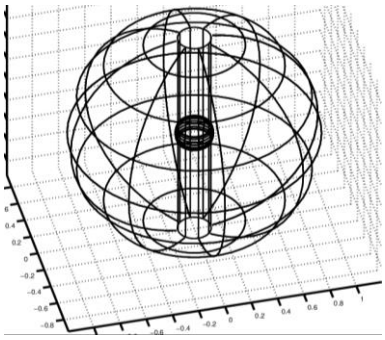


Fig. 1. Espacio de trabajo del Robot Puma 560 (Tomado de [16]).

Cada red es entrenada con diferentes pesos iniciales aleatorios para evitar caer en mínimos locales, posteriormente es evaluada usando el error cuadrático medio. El objetivo es encontrar una RNA que presente el mejor desempeño mientras se varía el número de neuronas de dos capas ocultas, con el fin de obtener el mínimo número de parámetros.

Los parámetros para la implementación del algoritmo genético son:

- Los alelos del gen son de tipo entero.
- Los genes de un individuo tienen la forma de $[n_1\ n_2]$.
- El tamaño de la población es de 10 individuos.
- La selección se hizo por torneos.
- El máximo número de generaciones es de 1000.
- El rango de búsqueda en el número de neuronas es de 6 a 30.
- La población inicial se genera aleatoriamente.
- La probabilidad de cruce y mutación es de 0.8 y 0.1 respectivamente.

La figura 2 muestra el diagrama de flujo del AG, que procesa cada RNA. Como primer paso una RNA es generada con n_1 y n_2 para la primer y segunda capa oculta posteriormente la RNA se entrena, valida y evalúa obteniendo el error cuadrático medio.

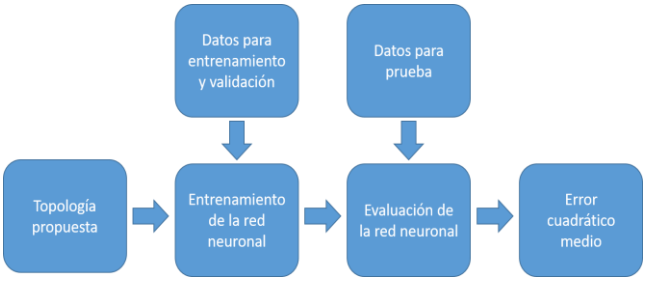


Fig. 2. Diagrama de flujo de la función fitness.

En la tabla 2 se muestran las consideraciones para el entrenamiento de las RNA's:

Tabla 2. Parámetros de entrenamiento de la RNA.

Parámetro	Valor
Método de entrenamiento	Levenberg-Marquardt
Número máximo de épocas	20,000
Función de activación en las capas ocultas	Sigmoidal
Número de neuronas en cada capa	6 - 30
Pesos iniciales	Aleatorio
Mínimo gradiente	$1x10^{-7}$
Función de desempeño	Error cuadrático medio
División de datos	70% Entrenamiento 20% Validación 10% Evaluación final

3. Resultados

Al término de la ejecución del AG se tienen 10 RNA's con el mejor desempeño (ver Fig. 3). La red con el error cuadrático medio menor es elegida.

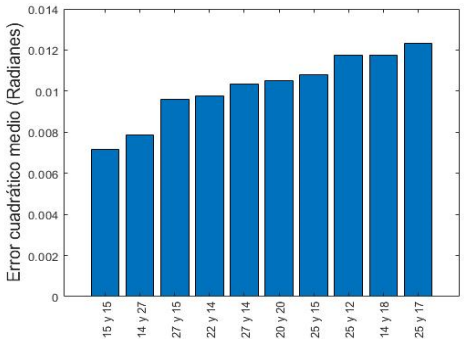


Fig. 3. Desempeño de las redes neuronales finales.

Como resultado se obtuvo una RNA con un error cuadrático medio de 0.00060752 metros con 15 neuronas en la primer y segunda capa oculta. En la figura 4 se muestra el error entre la trayectoria deseada y la obtenida con la RNA.

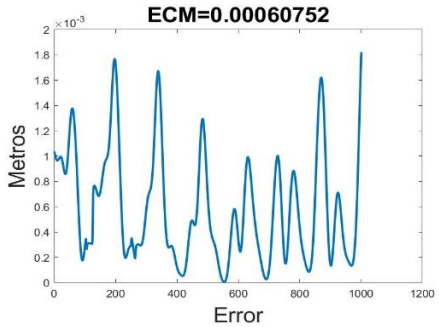


Fig. 4. Error entre la trayectoria deseada y la salida de la RNA.

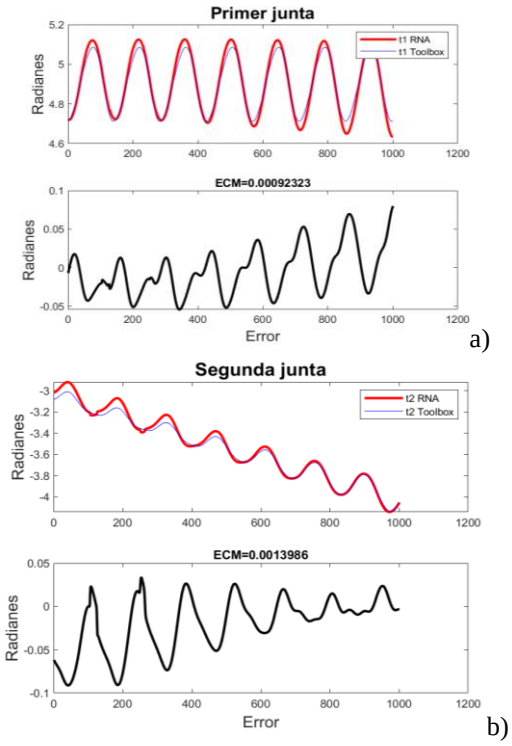


Fig. 5. Resultado para la primer y segunda junta.

En la tabla 3 se muestra el error cuadrático medio en radianes para cada junta.

Tabla 3. Máximo error para cada junta.

Junta	Error cuadrático medio (rad)
1°	0.00092323
2°	0.0013986
3°	0.0030617
4°	0.037452
5°	0.00089664
6°	0.11495

En la figura 5, 6 y 7 se muestra la comparación entre los datos de evaluación (en azul) y la salida de la RNA (en rojo) y su gráfica de error (en negro) para cada junta. Los datos de evaluación se tomaron del 10% de un total de 286746 ejemplos analíticos.

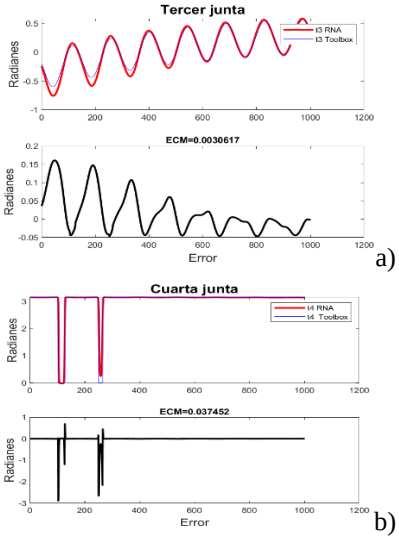


Fig. 6. Resultado para la tercer y cuarta junta.

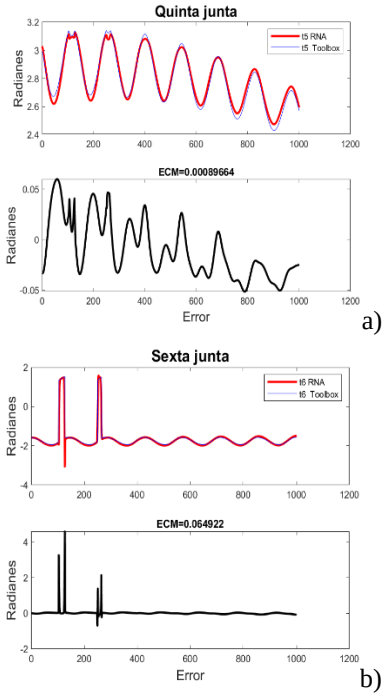


Fig. 7. Resultados para la quinta y sexta junta.

Los resultados obtenidos demuestran que la RNA obtenida es competitiva con el método analítico (Ec. 2-8) para su aplicación en tiempo real.

4. Conclusiones

La principal ventaja de la RNA es la velocidad con la que resuelve la cinemática inversa a comparación de los métodos analíticos que requieren la aproximación de funciones trigonométricas. El error de salida es relativamente pequeño en la mayoría de los casos. La desventaja es que el proceso de optimización toma tiempo considerable debido a que cada individuo (RNA) debe ser entrenado y evaluado. Como trabajo a futuro se busca reducir el error cuadrático medio usando diferentes algoritmos de entrenamiento y la implementación de la RNA obtenida en un sistema embebido para aplicaciones de tiempo real.

Referencias

1. Ramos, G.: Cinemática inversa de Robots industriales. Tecnura 10, Con Ciencias, Colombia pp. 54–58 (2002)
2. Hossian, A., Sierra, E., Fernández, E., Britos, P., García, R.: El problema cinemático en manipuladores robóticos industriales, un abordaje de solución mediante redes neuronales artificiales. En: VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del conocimiento, Lima, Perú, pp. 427–434 (2017)
3. Vele, O.: Cinemática inversa de un robot bípedo. Investigación personal, Researchgate.net, pp. 1–20 (2006)
4. Castillo, P.A., García, F., Merelo, J., Prieto, M.: Diseño de redes neuronales artificiales mediante algoritmos evolutivos. Revista Iberoamericana de Inteligencia Artificial 5(14), 2–32 (2001)
5. Giraldo, L., Delgado, E., Castellanos, G.: Cinemática inversa de un brazo robot utilizando algoritmos genéticos. Avances en sistemas e informática, Medellín, 3(1), 29–34 (2006)
6. Kardan, M.R., Setayeshi, S., Koohi-Fayegh, R., Ghiassi-Nejad, M.: Neutron spectra unfoldin in bonner spheres spectrometry using neural networks. Radiation protection dosimetry 104(1), 27–30 (2003)
7. Vaca-González, J.J., Peña-Caro, C.A., Vaca- González, H.: Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS. Tecnura 19(44), 33–46 (2015)
8. Reyes-Alfaro, A., Ortiz-Rodríguez, J.M., Reyes-Haro, A., Castañeda-Miranda, R., Solís-Sánchez, L.O., Vega-Carrillo, H.R.: Optimización de redes neuronales artificiales para la reconstrucción del espectro de neutrones y sus dosis equivalentes. ISSSD, Cusco, Perú, pp. 4–5 (2014)
9. Montana, D.J., Davis, L.: Training feedforward neural network using genetic algorithms. In: IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence, pp. 762-767 (1989)
10. Miller, G., Todd, P., Hedge, S.: Designing neural networks using genetic algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 2–3 (1989)
11. Kitano, H.: Deigning Neural Networks using genetic algorithms with graph Generation System. Complex System, vol. 4, pp. 461–476 (1990)
12. Delgado, J. A.: Redes neuronales y algoritmos genéticos. En: Jornadas de Ingeniería Eléctrica y Electrónica (FIEE), vol. 13, pp. 65–68 (1992)
13. Tellez, P.: Training feedforward neural networks using neural networks and genetic algorithms. In: International conference on computing, communications and control technologies, pp. 3–4 (2006)

14. Craig, J.J.: Introduction to robotics: mechanics and control. Pearson Education, México, (2005)
15. Corke, P.: Robotic Toolbox. www.petercorke.com. Último accesso: 03/05/2019
16. Ghosal, A.: Manipulator Kinematics. Handbook of Manufacturing Engineering and Technology, pp. 1777–1808 (2015)

ISSN 1870-4069

Aplicación para el monitoreo colaborativo de la calidad de cuerpos de agua usando bioindicadores

Irving Galindo-Márquez, Ángel López-Sánchez, Luis Hernández-Mercado, Carlos Sosa-Paz, Juan-Pablo Posadas-Durán

Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco (ESIME Zacatenco), México
jdposadas@esimez.mx, omar.syn10@gmail.com, enriquehm97@gmail.com, csosapaz@gmail.com

Resumen. El monitoreo colaborativo móvil (*Mobile Crowd Sensing*) es una técnica de monitoreo que se basa en la participación de personas que usan algún dispositivo móvil (teléfonos inteligentes, tabletas, etc.) para colectar datos de diversos tipos del entorno en que se encuentran y los transmiten hacia un nodo central para medir, georreferenciar, analizar y estimar procesos de interés. En este trabajo se propone un prototipo de plataforma móvil para la evaluación y monitoreo de la calidad de cuerpos de agua, mediante el uso de teléfonos inteligentes, que permita evaluar el cambio temporal de la calidad de cuerpos de agua por medio de bioindicadores utilizando el índice BMWP. El prototipo, además de utilizar estrategias de monitoreo colaborativo, realiza un análisis de los datos recolectados para determinar el estado del cuerpo de agua e indicar el nivel de participación de los colaboradores.

Palabras clave: crowdsensing, dispositivos móviles, internet de las cosas, ciudades inteligentes.

Crowdsensing App for Monitoring Quality of Water Bodies Using Bioindicators

Abstract. Mobile Crowd Sensing is a monitoring technique that relies on the participation of people who use a mobile device (smartphones, tablets, etc.) to collect data from various types of the environment and transmit them to a central node to measure, georeference, analyze and estimate processes of interest. This work proposes a prototype of a mobile platform for the evaluation and monitoring of the quality of water bodies, using bioindicators described in the BMWP index. The prototype also performs an analysis of the data collected to determine the status of the body of water and indicates the level of participation of the collaborators.

Keywords: crowdsensing, mobile devices, IoT, smart cities.

1. Introducción

México se encuentra en transición hacia una sociedad de la información, en la que las Tecnologías de la Información y Comunicación (TIC), son utilizadas con frecuencia por la sociedad para realizar la mayor parte de sus actividades. Esta integración permite acceder a información o servicios ofrecidos local o internacionalmente, mediante el uso de diversos dispositivos además de las computadoras. El nivel de integración de las TIC en una sociedad es un indicador del desarrollo de un país, además le permite participar en políticas o programas a nivel global.

En México, empresas y diversos sectores del gobierno como el educativo, medio ambiente, salud y seguridad, por mencionar algunos, requieren de acciones que les permitan iniciar o aumentar su nivel de integración de las TIC, preferentemente hasta un nivel como el que se tiene en países desarrollados. Las TIC son un conjunto de herramientas tecnológicas utilizadas para procesar y administrar información de manera eficiente; abarcan a las telecomunicaciones y diversas áreas de la computación como son tecnologías de la informática, tecnologías de software, programación avanzada, inteligencia artificial, interfaces hombre-máquina, ciberseguridad, bases de datos, entre otras.

En el presente documento, se describe una propuesta de infraestructura que tiene como objetivo desarrollar un sistema informático, que permita recopilar información sobre la calidad de cuerpos de agua en México, mediante el uso de teléfonos inteligentes con la participación de los pobladores de las comunidades cercanas a los cuerpos de agua. La evaluación del cuerpo de agua se realizará a través de una encuesta que implemente el método BMWP para conocer el índice que calidad de cuerpos de agua. La encuesta se implementará como una aplicación para teléfonos inteligentes. Una vez recopilada la información mediante la aplicación por parte de los participantes se almacenará en un servidor central. El sistema informático permitirá a un experto acceder a la información recopilada para su valoración y posterior integración en una base de datos. Existirá una retroalimentación e información mostrada en páginas Web dinámicas, a los pobladores y autoridades locales y federales, sobre el estado que guarda cada cuerpo de agua georreferenciado mediante páginas Web dinámicas que mostrarán indicadores sobre la calidad de los cuerpos de agua.

El resto del artículo se organiza de la siguiente forma: en la Sección 1 se muestra una breve descripción de las características sobre el monitoreo colaborativo móvil, en la Sección 2 se describe brevemente el índice de calidad del agua mediante el paradigma de BMWP que se basa en la identificación de organismos macroinvertebrados en los cuerpos; en la Sección 3 se describe la propuesta para el desarrollo de la aplicación y en la Sección 4 se describen las conclusiones.

2. Monitoreo colaborativo móvil

El Monitoreo Colaborativo Móvil (en inglés, *Mobile Crowd Sensing*, MCS), hace referencia a un paradigma en el que se utilizan dispositivos móviles para recolectar información sobre un fenómeno de interés y compartir esta información

con sistemas de cómputo a través del Internet. En el MCS intervienen un conjunto de participantes distribuidos geográficamente con la finalidad de realizar una tarea de recolección de información, explotando la capacidad que tienen los dispositivos móviles para obtener datos a través del uso de los sensores con los que cuentan [1]. También es posible recopilar metadatos sobre las mediciones que se realizan como son la ubicación, opiniones y experiencias de los usuarios [16]. La información recolectada mediante el MCS es la base para el desarrollo de servicios y aplicaciones en la novedosa área del Internet de las Cosas (en inglés, *Internet of Things*, IoT).

Los dispositivos móviles y su capacidad para obtener información del medio, así como su capacidad para compartirla, son elementos clave en el paradigma MCS. Además de los dispositivos móviles diseñados para uso industrial o específico, son de especial interés aquellos que se utilizan a un nivel personal como son los teléfonos inteligentes, tabletas, relojes inteligentes, sistemas embebidos de videojuegos (Wii, Xbox Kinect) y altavoces inteligentes (Google Home, Amazon Echo), debido a las capacidades de procesamiento que poseen y a su bajo costo.

Los teléfonos inteligentes han mostrado una evolución constante desde el año 2010, integrando mejoras en su sistema operativo y aumentando los sensores que incluyen en su hardware. El equipamiento de sensores depende de la marca y modelo del teléfono inteligente, sin embargo, algunos de los sensores que podemos encontrar en los celulares de gama media de modelos lanzados en el año 2018 son: cámara, micrófono, barómetro, podómetro, GPS, acelerómetro, brújula, giroscopio, intensidad de luz ambiental y sensores biométricos (lector de huella digital). Adicionalmente, los teléfonos inteligentes cuentan con diversos mecanismos para poder compartir las mediciones realizadas además de la red celular, como son el WiFi, Bluetooht, NFC y cable USB. Las características que poseen los teléfonos inteligentes como su nivel de equipamiento de sensores, facilidades que poseen para comunicar datos y su uso generalizado entre las personas, los convierten en la elección frecuente para aplicaciones de MCS donde se requiere recopilación de datos masiva a un bajo costo [9].

Existen diversas áreas donde se han desarrollado aplicaciones que utilizan el paradigma de MCS utilizando solamente teléfonos inteligentes como medios para la recopilación de datos. En el ámbito ambiental, por ejemplo, se ha utilizado el MCS en el desarrollo de sistemas para la medición de contaminantes en el aire [8,12], de niveles de ruido [5] o condiciones climatológicas[13]. El paradigma MCS ha sido utilizado para desarrollar aplicaciones orientadas hacia ciudades inteligentes como lo es el monitoreo del tráfico para tener transportes eficientes [17] y monitoreo de vialidades con dañadas [10]. Otra área de aplicación es la salud, donde se han desarrollado sistemas para el monitoreo de la actividad física de las personas [2], de hábitos alimenticios [4] y monitoreo de datos relacionados con diversas enfermedades [6,11]. En áreas como el comercio [15] y turismo [18,14] también se han desarrollado aplicaciones que utilizan teléfonos inteligentes bajo un esquema de MCS.

Las aplicaciones desarrolladas bajo el paradigma de MCS se pueden catalogar dentro de dos categorías, en términos de la forma en que los participantes realizan

la recopilación de los datos [7]: 1) monitoreo participativo (en inglés, *Participatory Sensing*) y 2) monitoreo oportunista (en inglés, *Opportunistic Sensing*).

El monitoreo participativo se caracteriza porque el participante actúa con iniciativa propia en la recopilación de los datos, asumiendo la responsabilidad de la información que se recopila y la forma que se lleva a cabo esta acción. El monitoreo participativo, requiere que el participante asuma una postura activa dentro del proceso de recopilación de los datos y no requiere de una infraestructura adicional para realizar este proceso [3,7].

El monitoreo oportunista se caracteriza porque requiere de un conjunto de aplicaciones preinstaladas en el dispositivo móvil que realicen una serie de acciones, que se ejecutan en segundo plano, para la recopilación de datos. En el monitoreo oportunista los datos se recopilan de manera automática y no requiere de una intervención activa del participante [1].

El factor humano es un elemento importante en el paradigma MCS que requiere de la definición de criterios para su selección y regular su participación el monitoreo de datos. En el trabajo [1] se presentan los siguientes aspectos a considerar para una adecuada elección de participantes: a) se deben considerar un número suficiente de participantes por locación; b) se requiere definir incentivos que motiven la participación del factor humano en el proceso de recopilación de datos y garantice la calidad del proceso; c) se requiere de mecanismos de seguridad que protejan la ubicación de los participantes; d) se requiere que los participantes conozcan los mecanismos que proveen los teléfonos inteligentes para compartir datos (WiFi, Bluetooth, datos celulares, entre otros).

3. Monitoreo biológico de cuerpos de agua

Las fuentes de agua superficiales, representan el pilar de desarrollo de las diferentes actividades socioeconómicas, realizadas en las diferentes poblaciones alrededor del planeta, sin embargo, muchas de estas actividades provocan alteraciones e inclusive deterioro de la calidad de las fuentes de agua. En general, los cuerpos de agua superficiales están sometidos a dos tipos de contaminación: la primera, contaminación natural (provocada por el arrastre de materia particulada y disuelta en presencia de materia orgánica natural) y la segunda, de origen antrópico (descarga de aguas residuales domésticas, actividades agrícolas, actividades industriales, etc.). Diversos problemas se originan debido al mal estado de cuerpos de agua, por ejemplo, el mayor impacto sobre salud pública, tiene su origen a través de los sistemas de abastecimiento de agua; el cual es definido, como riesgo de transportar agentes contaminantes que puedan provocar enfermedades.

En la mayoría de países en vías de desarrollo (incluyendo México), el riesgo microbiológico es bastante elevado, aunado a una mala sanidad, dato que se ratificó en la Agenda 21 de la Conferencia de Naciones Unidas sobre Medio Ambiente y Desarrollo, en la cual afirma que aproximadamente 80 % de todas las enfermedades, y más de la tercera parte de los decesos en estos países (en vías de desarrollo), se tienen a causa del consumo de agua contaminada. El notable

deterioro de cuerpos de agua, provocó que se torne prioritario el monitoreo de la calidad de aguas superficiales, con el fin de tomar medidas de control y reducción del nivel de riesgo, factores que serán determinantes en la complejidad y costos de los tratamientos o medidas a tomar.

Para la determinación de la calidad de cuerpos de agua, se determinaron diversos factores, tales como los usos del suelo, la producción industrial y agrícola en áreas aledañas, e incluso, la misma cantidad de agua en ríos, lagos y lagunas. Con la finalidad de evaluar la calidad o también llamado grado de contaminación del agua, se han desarrollado índices de calidad, tanto generales, como específicos. En México se emplea un método llamado, Índice de Calidad del Agua (ICA), que agrupa algunos parámetros del deterioro de la calidad del líquido [1]. Para la evaluación, el índice toma valores en una escala de 0 a 100 %, en donde mientras mayor sea el valor, la calidad aumenta. El ICA se calcula a partir de la ponderación de 18 parámetros fisicoquímicos, entre los cuales resaltan la demanda bioquímica de oxígeno (DBO), oxígeno disuelto, coliformes, fosfatos, pH, sólidos suspendidos. Una de las desventajas del ICA es el costo que conlleva su implementación en los cuerpos de agua, que requiere de reactivos, instrumentos especializados y técnicos para su manipulación.

Una alternativa es el para el monitoreo biológico de los cuerpos de agua es el *Biological Monitoring Working Party* (BMWP por sus siglas en inglés). El índice BMWP utiliza principalmente la presencia de macroinvertebrados (organismos invertebrados de tamaño suficientemente grande para ser vistos sin el uso de microscopio) para determinar la calidad de un cuerpo de agua, debido a que los macroinvertebrados tienen diferentes grados de tolerancia a distinto tipos de contaminación. Los macroinvertebrados son generalmente sedentarios, tienen un ciclo de vida largo y son sensibles a las sustancias presentes en el agua. Las características de los macroinvertebrados permiten que el índice BMWP sea una alternativa confiable y económica para el monitoreo de cuerpos de agua al no requerir de equipo especial. El índice debe ser calibrado y validado para cada región ecológica en virtud de las importantes diferencias en la composición taxonómica, clima, y a los cambios en el nivel de sensibilidad de los organismos indicadores (macroinvertebrados acuáticos).

Por otro lado, a la fecha, en México, la normativa para la evaluación de la calidad el agua no cuenta con el uso de bioindicadores, los cuales son ampliamente usados en países desarrollados como el caso de Estados Unidos y la Unión Europea, así como otros países latinoamericanos, Costa Rica y Panamá. De igual forma, otros países han establecido el monitoreo participativo con apoyo de las poblaciones locales, las cuales realizan de forma voluntaria la vigilancia de sus recursos. Entre los programas internacionales se encuentran: El Programa de monitoreo voluntario *Adopta tu cuenca* de la Agencia de Protección Ambiental de Estados Unidos de América (EUA)[2]. El Programa voluntario de Alabama Water

[1] `http://dgeiawf.semarnat.gob.mx:8080/ibi_apps/WFServlet?IBIF_ex=D3_R_AGUA05_01&IBIC_user=dgeia_mce&IBIC_pass=dgeia_mce`

[2] `http://water.epa.gov/type/rsl/monitoring/`

Watch[3] en USA, que forma parte de la amplia red de Global Water Watch de la Universidad de Auburn, Alabama[4]. Monitoreo comunitario participativo en Canadá que tiene apoyo gubernamental (`http://www.enr.gov.nt.ca/programs/water-resources/community-based-monitoring`). Programa de monitoreo comunitario de la calidad del agua en Australia[5]. A su vez, en América Latina algunos países como Brasil, Colombia y Costa Rica, cuentan con redes de monitoreo comunitario. Estos ejemplos de monitoreo participativo ponen de manifiesto la importancia de llevar a cabo el monitoreo comunitario que coadyuvan a la comprensión de la condición ambiental de las corrientes hidrográficas, y además permiten que las comunidades se apropien de las técnicas de monitoreo y utilicen la información generada (a la par con el conocimiento local existente) para sus necesidades. En México, donde se presencian múltiples problemáticas vinculadas con el recurso hídrico asociado a los cambios en el uso del suelo, que implican sobre todo la contaminación y abatimiento de las fuentes de abastecimiento de agua potable, hace falta la implementación del monitoreo participativo con un protocolo que permita abordar de forma sistemática la evaluación de la calidad del agua. Las Áreas Naturales Protegidas (ANPs) constituyen un modelo de estudio adecuado para poner a prueba el uso de indicadores biológicos a través del índice BMWP y el uso de aplicaciones móviles para el monitoreo participativo, ya que, además de contar con altos niveles de diversidad que pueden aportan a una calibración del índice BMWP con gran diversidad de familias, también cuentan con Planes de Manejo y de Programas de Vigilancia a través de la participación de la Comisión Nacional de Áreas Naturales protegidas. Por lo que en estas ANPs es posible trabajar en coordinación con la dirección de las ANPs para llevar a cabo el monitoreo participativo y generación de ciencia ciudadana. En este proyecto se seleccionarán como zonas de estudio a las ANPs: reserva de la biosfera Tehuacán Cuicatlán, Reserva de la biosfera Sierra Gorda y ANP Camécuaro y el río Duero.

4. Propuesta de aplicación para el monitoreo de cuerpos de agua

En este módulo, se describen los elementos que conforman el prototipo para el monitoreo de los cuerpos de agua. La propuesta se enfoca en el monitoreo colaborativo, por lo que, se requiere de un mecanismo que permita a colaboradores participar en la recolección de datos sobre el fenómeno de interés relacionado con el estado de los cuerpos de agua. En el prototipo se asume que los colaboradores no poseen conocimiento especializado en las TIC, sin embargo, se consideran nativos digitales con conocimientos básicos sobre el uso de teléfonos inteligentes (uso de interfaz gráfica y uso de cámara) que coincide con un escenario real.

El prototipo utiliza la estrategia del índice BMWP para determinar la calidad de un cuerpo de agua. El índice BMWP considera aspectos como tipo de

[3] `http://www.alabamawaterwatch.org`
[4] `http://www.globalwaterwatch.org/`
[5] `http://www.seqcatchments.com.au/programs`

macroinvertebrados presentes, tipo de sedimentación, variables climatológicas, tipo de cauce, entre otros, para evaluar la calidad de un cuerpo de agua.

La infraestructura propuesta se muestra en la figura 1. Para la plataforma móvil se propone desarrollar dos aplicaciones orientadas a teléfonos inteligentes. La primera aplicación, implementará el protocolo para la bioevaluación de la calidad de agua, mediante una serie de vistas que incluyen preguntas acerca de las características del cuerpo de agua y la identificación de organismo que se encuentran en el cuerpo de agua. La segunda aplicación, integrará la información recolectada por los colaboradores mediante el uso de la primera aplicación y del protocolo de Bluetooth, para realizar la transferencia de datos hacia la segunda aplicación. Será la segunda aplicación la responsable de enviar la información recolectada hacia un servidor central, debido a la falta de conectividad en la zona donde se encuentran los cuerpos de agua. Las aplicaciones serán utilizadas por colaboradores que radiquen cerca de los cuerpos de agua que se quieren analizar.

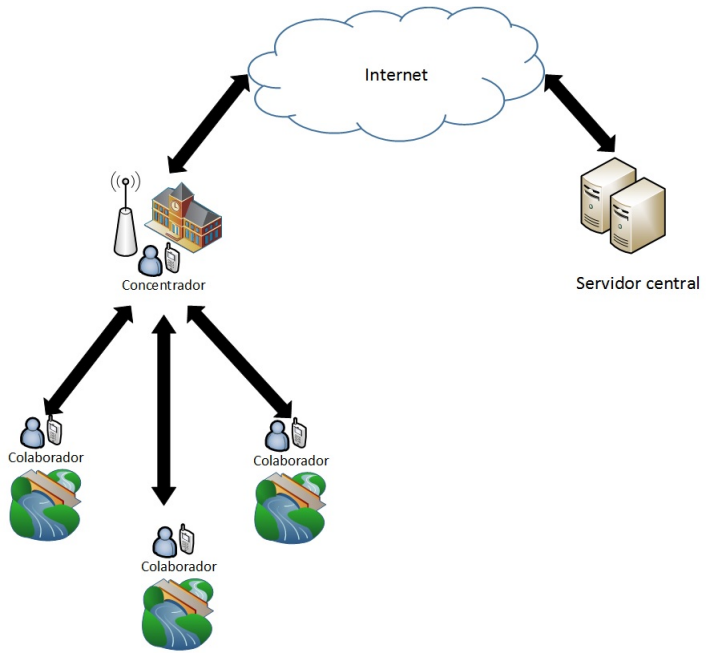


Fig. 1. Esquema general de la propuesta.

5. Funcionamiento general de la aplicación móvil

En la figura 2 se ejemplifica el funcionamiento general de la aplicación. Está dividida estratégicamente en etapas que nos permiten avanzar de manera sencilla

y dinámica en la evaluación de los cuerpos de agua. En todas las etapas se proporcionará ayuda e información para facilitar la toma de decisiones, así como también evitar errores. En la primera etapa, de registro, al encontrarnos en un lugar sin acceso a internet se omitirá la fase de registro, rescatando únicamente los datos necesarios para la evaluación:

- Nombre de usuario,
- Nombre del río,
- Identificador,
- Municipio,
- Estado,
- Cuenca,
- Orden del río.



Fig. 2. Diagrama general del funcionamiento de la aplicación.

Los datos anteriores son los básicos necesarios para comenzar la evaluación. En la siguiente etapa, encontraremos las medidas de seguridad para realizar la evaluación, de igual manera, se presenta un tutorial, en el cual se describirá de manera sencilla el funcionamiento de la aplicación y la forma correcta de recopilar la información en los test que se encuentran en las siguientes etapas. La tercera etapa, presenta una pequeña encuesta, auxiliar en la determinación de la hidrología del río. Está información sirve como preámbulo y se complementará con la información recabada por los test. En la aplicación, inicialmente se desplegarán tres opciones con imágenes representativas (facilitadores para la toma de decisión), de las cuales, únicamente se podrá seleccionar una; dichas opciones abordan el tema de la fase acuática del río, las opciones son:

- El agua corre,
- El río presenta pozas,
- El río es seco.

Una vez seleccionada una opción, el siguiente paso, es determinar un régimen hidrológico de acuerdo a la época del año; para este proceso se desplegarán las

cuatro estaciones del año consecutivas, de las cuales es necesario seleccionar al menos dos estaciones y como varió, si es que existió un cambio, la hidrología a través de las mismas, por ejemplo, en primavera el río se mantuvo con pozas, mientras que en verano el río se secó. Finalmente se procederá a determinar la tipología del río, de igual manera que en procesos anteriores, se desplegarán alrededor de 7 imágenes, cada una correspondiente a una tipología de un río.

Además de las imágenes, la aplicación presentará botones de información, con datos de relevancia, que sirvan como herramienta auxiliar para seleccionar la respuesta más adecuada, y por lo tanto, la evaluación sea precisa. La cuarta etapa, etapa de test, se podría contemplar como la más importante, ya que en esta se lleva a cabo el proceso de recopilación de información. Como primer paso, es necesario tomar una fotografía del río, de vital información para corroborar la información recaudada por los test. A continuación de la fotografía se presentan 3 test, los cuales son:

- Test de Estado Hidrológico,
- Test Hidroformológico,
- Test Biológico.

Cada test recauda información pertinente a su área, pero en conjunto nos ayudan a realizar un análisis de la calidad de cuerpos de agua, todo esto con la validación de un experto. En el test hidrológico, se busca obtener información acerca de:

- Infraestructura cercana,
- Extracción humana de agua,
- Cambios históricos del caudal del río.

Todo con la finalidad de determinar si estamos tratando un río que ha sido alterado por la actividad humana, o nos encontramos en un río que se ha mantenido intacto al contacto de las poblaciones. El segundo test, tiene como propósito conocer información acerca de:

- Naturaleza del río,
- Continuidad de la naturaleza,
- Paisaje,
- Canales fluviales,
- Contaminación.

Y finalmente, el tercer test, en el cual es necesario remover algunas rocas, recolectar alrededor de 50 macro invertebrados, los cuales, son determinantes para el análisis final. En este apartado, se desplegará una lista con las imágenes de cada invertebrado que se pudiera encontrar en el río, sin importar su calidad (buena, mala o regular). Es necesario seleccionar el invertebrado encontrado, así como tomar una fotografía del mismo, para proveer al experto, las herramientas necesarias para que el lleve a cabo se labor.

6. Características de la encuesta para la aplicación móvil

La técnica de biomonitoreo de la calidad del agua, empleando macroinvertebrados acuáticos como bioindicadores, requiere de una recopilación primaria de características físicas y químicas del sitio de muestreo. Esta información le permitirá al grupo de expertos biólogos, interpretar de manera correcta y confiable, los resultados que emitirá la aplicación. Esta caracterización se basada en los Protocolos de Bioensayos Rápidos de la Agencia de Protección Ambiental de Estados Unidos (United States Enviroment Protection Agency, USEPA).

El proceso de caracterización comienza por delimitar un tramo del río a muestrear tomando una distancia de 100m, que cubra las características más representativas de cada hábitat y la variedad de condiciones físicas de la estación. Dicha caracterización consta de 4 pasos, los cuales se muestran al iniciar la aplicación móvil.

- Datos generales: Corresponde a la información general sobre el sitio de muestreo. Recopila información como: nombre del río, identificador ID del sitio, cuenca, orden del río, localidad más cercana, municipio, fecha del muestreo, hora del muestreo, altitud y coordenadas geográficas.
- Número de control: Corresponde a una numeración única que se asigna a la estación de muestreo.
- Condiciones climáticas: Se evalúa eventos como tormentas, nubosidad, lluvias intermitentes, temperatura ambiental y velocidad del viento (de poder contar con el dato), que pudieran afectar a la muestra.
- Características del cuerpo de agua: Dado que existe una estratificación natural de ensamble de macroinvertebrados, es importante tener en claro si la muestra es llevada a cabo en la parte alta (Crenon), media (Ritron) o baja de la cuenca (Potamon), esto a su vez corroborado por una fotografía tomada al momento y guardada en el formulario, que incluya las características del sitio en el tramo de 100 metros.

La segunda parte del formulario de la aplicación móvil, consiste en información más puntual acerca del estado del cuerpo de agua que está siendo muestreado. Apoyado a su vez por fotografías del sitio y de los macroinvertebrados encontrados. Los datos obtenidos en esta parte de la aplicación móvil tendrán una ponderación, la cual permitirá obtener un puntaje al final de la sesión, que servirá a los expertos como un referente, antes de que den el visto bueno y califiquen el estado de la calidad del agua en dicho sitio. La información (proporcionada por el área de biología) para tomar en cuenta en el formulario es la siguiente:

- Fauna: Sustrato/cobertura disponible. Se refiere a la diversidad de habitas dependientes del sustrato y la variedad de ellos presentes en el lecho del río. Los óptimos se caracterizan por ser altamente heterogéneos, mientras que los pobres contienen un solo tipo de hábitat como arenas.
  - Óptimo: más del 70 % del sustrato favorable para la colonización fauna acuática.

  - Subóptimo: 40-70 % de hábitat estable; adecuado para el desarrollo de potencial colonización.
  - Marginal: 20-40 % de hábitat estable; se encuentra reducido por perturbación o remoción.
  - Pobre: menos de un 20 % de hábitats estables, poca disponibilidad de colonización.
- Sustrato/tipo: En esta sección, se requiere definir la proporción de tamaño de partículas disponible en el sustrato y la variedad en la proporción alcanzará las máximas calificaciones, por el contrario, la proporción predominante de arena alcanzará calificaciones más bajas.
  - Óptimo: 0-25 % de gravas, cantos y partículas está rodeada de sedimento fino.
  - Subóptimo: 25-50 de la superficie de rocas, cantos y grava rodeadas de sedimento fino.
  - Marginal: 50-75 % de la superficie de rocas, cantos y grava rodeadas de sedimentos fino. Sin vegetación sumergida.
  - Pobre: más del 75 % de la superficie de rocas, cantos y grava rodeadas de sedimento fino. Sin vegetación o raíces.
- Relación profundidad y velocidad: Hace referencia a los cuatro patrones de velocidad y profundidad que se encuentran en corrientes de alto gradiente. Las pautas generales son: 0.5 m de profundidad para separar superficial de profundo y 0.3 m/s para separar rápido de lento.
  - Óptimo: Cuatro combinaciones: (1) lento-profundo, (2) lento-somero, (3) rápido-profundo y (4) rápido-somero.
  - Subóptimo: Tres combinaciones. La ausencia de rápido/bajo determina menor puntaje
  - Marginal: Dos combinaciones. La ausencia de rápido/bajo y lento/bajo determina menor puntaje.
  - Pobre: Una sola combinación presente. Usualmente está presente lento/profundo.
- Depósito de sedimento: Evalúa la cantidad de sedimento que se ha acumulado, y los cambios que se han producido en el fondo de la corriente como resultado de la alteración de las riveras.
  - Óptimo: Poco depósito sin islas ni barras, menos del 5 % del fondo afectado por la deposición de sedimentos.
  - Subóptimo: Formación de barras de grava, arena o sedimento, 5 a 30 % del fondo afectado por la deposición de sedimentos; ligera deposición en los pozos.
  - Marginal: Presencia de barras y 30 a 50 % del fondo afectado. Sedimento sobre obstrucciones, constricciones y recodos. Moderada deposición en pozos.
  - Pobre: Alto depósito de material y sin pozas. Mas del 50 % del fondo cambia con frecuencia.
- Estado del canal: Evalúa el grado en que el canal está lleno de agua. El estado del flujo cambiará a medida que el canal se amplíe o cuando el flujo disminuya, como resultado de las presas y otras obstrucciones, desviaciones

para el riego o sequía. Cuando el agua no cubre gran parte del lecho del río, la cantidad de sustrato adecuado para los organismos acuáticos es limitada.
  - Óptimo: El nivel de agua alcanza la base de los bancos y el sustrato se encuentra infra expuesto.
  - Subóptimo: El agua cubre más del 75 % del cauce o menos del 25 % del sustrato de fondo que expuesto.
  - Marginal: El nivel del agua cubre entre el 25 al 75 % del cauce y queda expuesta la mayor parte del sustrato de los rápidos.
  - Pobre: Muy poca agua sobre el cauce, casi como pozas.
- Alteración del cauce: Es una medida de los cambios a gran escala en la forma de la corriente del canal. Muchos arroyos en áreas urbanas y agrícolas han sido enderezados, profundizados o desviados a canales concretos, a menudo para control de inundaciones o para fines de riego. Tales arroyos tienen mucho menos hábitats naturales disponibles para ser colonizados.
  - Óptimo: Sin canalización o evidencia de ello, sustrato favorable para la colonización.
  - Subóptimo: Alguna alteración o evidencia de ello, sin alguna canalización reciente.
  - Marginal: Canalización extensiva con bancos o playas y de 40 a 80 % de canalizado. Se puede considerar la presencia de barreras o diques.
  - Pobre: Más del 80 % del trecho del río canalizado y alterado. Márgenes protegidas con gaviones o cemento.
- Obtención de muestras de macroinvertebrados: En esta parte de la encuesta, se presentan imágenes alusivas a los macroinvertebrados más comunes que habitan en cuerpos de agua y cada uno cuenta a su vez con una ponderación definida, de acuerdo a su persistencia en los diferentes tipos de calidad de agua, con el propósito de que el usuario seleccione la opción que sea o se parezca al animal muestreado. Para efectos prácticos, se propone tomar una fotografía del animal para su verificación en el centro de control por un experto.

Como se comentó previamente, el ponderado final les dará a los expertos un parámetro con el cual dar un veredicto acerca de la calidad del cuerpo de agua.

7. Uso de los datos recopilados

Los datos obtenidos a partir de la encuesta, principalmente, serán usados para dar un panorama de la calidad del agua en una zona, pero, pueden ser usados para cubrir otras necesidades, o incluso, pueden proporcionar información adicional importante para los expertos.

Individualmente, cada encuesta servirá para determinar la calidad del agua en un determinado, río, lago, o algún otro tipo de cuerpo de agua, esto se hará, proporcionando un puntaje a los resultados obtenidos, pero este puntaje, tendrá como objetivo, asignar un color que indique que tan buena o mala es la calidad del cuerpo de agua, teniendo el rojo como la peor calidad y el azul como excelente calidad.

La base de datos, recopilará información importante de cada río o lago de la zona, entre esta información, se encuentra la ubicación y fecha. Una vez que se haya recopilado la información de cada zona, se podrá procesar la información, y a partir de los datos obtenidos, se determinará en que zona se encuentra mayormente contaminada el agua, además, en qué fechas se cumple esto, ya que, las condiciones del medio ambiente, cambian a lo largo del año. Cabe destacar que, la encuesta y el puntaje asignado a cada organismo, puede cambiar a lo largo del año.

Adicionalmente, se podrá procesar la información acerca de los macroinvertebrados acuáticos presentes en el lugar, y analizar si alguno de ellos aparece demasiado seguido, o, por el contrario, si aparece muy pocas veces, o incluso nunca. Esta información es adicional, pero puede ayudar a los expertos, a conocer, o alertar si hay alguna anomalía en cuanto a la presencia de algún organismo en específico. Expresando lo anterior en otras palabras, se podrá tener estadísticas de la presencia de cada organismo, para poder determinar si su ciclo de vida se está efectuando de manera correcta.

Dejando de lado los datos, se tienen las fotografías, las cuáles son parte esencial de cada encuesta, ya que, si el usuario no está seguro de si un organismo es o no el mostrado en la aplicación, tendrá total libertad de tomar una foto al organismo, y esta podrá ser avalada por un experto. Teniendo clara la importancia de las fotografías, se llega a una parte importante en cuanto al procesamiento de la información, ya que, el usuario puede tomar una fotografía en la cual, en el fondo, haya objetos irrelevantes para la fotografía, o incluso, puede que el organismo se vea muy pequeño en comparación al tamaño de la toma, es por esto que, se realizará un proceso que pueda recortar las fotografías, de forma que, se reduzca en medida de lo posible, la información irrelevante que haya en ésta, y así, lo que prevalezca sea la imagen del organismo.

Resumiendo, los datos obtenidos, serán procesados para cubrir las necesidades primarias del proyecto, pero, proporcionarán información adicional a los expertos, que puede resultar muy útil para ellos. En cuanto las imágenes, serán procesadas para que, en cada fotografía, se aprecie mayormente el organismo, y no otra información irrelevante, además de que, también serán procesadas para reducir la cantidad de memoria requerida para ser almacenadas.

8. Conclusiones y trabajo futuro

La propuesta busca desarrollar una aplicación que permite realizar el monitoreo de cuerpos de agua mediante el uso del paradigma de monitoreo colaborativo móvil (MCS). El enfoque de MCS ofrece una alternativa viable para el monitoreo de las características de los cuerpos de agua de una región a través de la participación ciudadana, presentando las siguientes ventajas: 1) permite implementar el índice BMWP para determinar la calidad de los cuerpos de agua en términos de bioindicadores, 2) la propuesta promueve la participación ciudadana haciendo que éstas sean responsables de sus propios recursos, 3) permite integrar la información sobre los diversos cuerpos de agua en una sola red a nivel nacional

y 4) permite identificar los sitios que requieren de atención para implementar estrategias de conservación, restauración o recuperación.

Como trabajo futuro se propone desarrollar módulos que permitan implementar técnicas de inteligencia artificial para obtener tendencias sobre el estado de los cuerpos de agua y técnicas de para el reconocimiento semiautomático de los macroinvertebrados recolectados.

Agradecimientos. Este trabajo ha sido realizado gracias al apoyo del Instituto Politécnico Nacional (Proyecto Multired 2045 Módulo IV con registro 20196153 de la SIIP-IPN).

Referencias

1. Abualsaud, K., Elfouly, T.M., Khattab, T., Yaacoub, E., Ismail, L.S., Ahmed, M.H., Guizani, M.: A survey on mobile crowd-sensing and its applications in the iot era. IEEE Access 7, 3855–3881 (2019)
2. Denning, T., Andrew, A., Chaudhri, R., Hartung, C., Lester, J., Borriello, G., Duncan, G.: Balance: towards a usable pervasive wellness application with accurate activity inference. In: Proceedings of the 10th workshop on Mobile Computing Systems and Applications. p. 5. ACM (2009)
3. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine 49(11), 32–39 (2011)
4. Gao, C., Kong, F., Tan, J.: Healthaware: Tackling obesity with health aware smart phone systems. In: 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO). pp. 1549–1554. Ieee (2009)
5. Hachem, S., Mallet, V., Ventura, R., Pathak, A., Issarny, V., Raverdy, P.G., Bhatia, R.: Monitoring noise pollution using the urban civics middleware. In: 2015 IEEE First International Conference on Big Data Computing Service and Applications. pp. 52–61. IEEE (2015)
6. Jin, Z., Oresko, J., Huang, S., Cheng, A.C.: Hearttogo: a personalized medicine technology for cardiovascular disease prevention and detection. In: 2009 IEEE/NIH Life Science Systems and Applications Workshop. pp. 80–83. IEEE (2009)
7. Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. IEEE Communications magazine 48(9), 140–150 (2010)
8. Leonardi, C., Cappellotto, A., Caraviello, M., Lepri, B., Antonelli, F.: Secondnose: an air quality mobile crowdsensing system. In: Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational. pp. 1051–1054. ACM (2014)
9. Liu, J., Shen, H., Narman, H.S., Chung, W., Lin, Z.: A survey of mobile crowdsensing techniques: A critical component for the internet of things. ACM Transactions on Cyber-Physical Systems 2(3), 18 (2018)
10. Mohan, P., Padmanabhan, V.N., Ramjee, R.: Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: Proceedings of the 6th ACM conference on Embedded network sensor systems. pp. 323–336. ACM (2008)
11. Oliver, N., Flores-Mangas, F.: Healthgear: Automatic sleep apnea detection and monitoring with a mobile phone. JCM 2(2), 1–9 (2007)

12. Pan, Z., Yu, H., Miao, C., Leung, C.: Crowdsensing air quality with camera-enabled mobile devices. In: Twenty-Ninth IAAI Conference (2017)
13. Pankratius, V., Lind, F., Coster, A., Erickson, P., Semeter, J.: Mobile crowd sensing in space weather monitoring: the mahali project. IEEE Communications Magazine 52(8), 22–28 (2014)
14. Pereira, F.C., Vaccari, A., Giardin, F., Chiu, C., Ratti, C.: 19 crowdsensing in the web: Analyzing the citizen experience in the urban space. From Social Butterfly to Engaged Citizen: Urban Informatics, Social Media, Ubiquitous Computing, and Mobile Technology to Support Citizen Engagement p. 353 (2011)
15. Sehgal, S., Kanhere, S.S., Chou, C.T.: Mobishop: Using mobile phones for sharing consumer pricing information. In: Demo Session of the Intl. Conference on Distributed Computing in Sensor Systems. vol. 13 (2008)
16. Sherchan, W., Jayaraman, P.P., Krishnaswamy, S., Zaslavsky, A., Loke, S., Sinha, A.: Using on-the-move mining for mobile crowdsensing. In: 2012 IEEE 13th International Conference on Mobile Data Management. pp. 115–124. IEEE (2012)
17. White, J., Thompson, C., Turner, H., Dougherty, B., Schmidt, D.C.: Wreckwatch: Automatic traffic accident detection and notification with smartphones. Mobile Networks and Applications 16(3), 285–303 (2011)
18. Wise, N., Heidari, H.: Developing smart tourism destinations with the internet of things. In: Big Data and Innovation in Tourism, Travel, and Hospitality, pp. 21–29. Springer (2019)

ISSN 1870-4069

Sintonización óptima de un controlador PD utilizando evolución diferencial con un modelo dinámico virtualizado en Unity 3D

Rosaura A. Suárez-Santillán, Gabriel Sepúlveda-Cervantes, Edgar A. Portilla-Flores, Maria Bárbara Calva-Yáñez, Cuauhtémoc Morales-Cruz, Noemi Hernández-Oliva

Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Grupo de Investigación e Innovación en Mecatrónica (GIIM), Ciudad de México, México
{rsantillans1800}@alumno.ipn.mx,
{gsepulvedac,aportilla,nhernandezo}@ipn.mx,
josechh@yahoo.com, {cumoralescruz}@gmail.com,{b_calva}@hotmail.com

Resumen. La creciente demanda de desarrollo científico tecnológico ha permitido la conjunción de diversos campos de la ciencia con el fin de resolver problematicas actuales .Permitiendo de esta manera cruzar los límites tradicionales entre varias disciplinas académicas para el surgimiento de nuevos conocimientos con base en las necesidades que anteriormente eran nulas. La creciente demanda de virtualizar los sistemas presentes en el mundo real ha obligado a desarrollar diversas herramientas que permiten recrear el mundo físico en una computadora. Actualmente, los sistemas físicos tienden a ser más complejos implicando que el modelo matemático tambien es complejo. En este trabajo se presenta la solución de un problema de optimización dinámico haciendo uso del motor de física NVIDIA PhysX para realidad virtual con el fin de reproducir el comportamiento de un sistema no lineal como caso de estudio demostrando la manipulación y recreación del comportamiento del sistema sin la necesidad de tener el modelo dinámico del mismo.

Palabras clave: sintonización óptima, modelado de sistemas, virtualización de sistemas, optimización, evolución diferencial, control PD.

Optimal Tuning of a PID Controller Using Differential Evolution with a Virtualized Dynamic Model in Unity 3D

Abstract. The growing demand for scientific and technological development has allowed the combination of diverse fields of science in order to solve current problems. This way, it is possible to cross the traditional limits between several academic disciplines for the emergence of new knowledge based on the needs that previously they were nil. The growing

demand to virtualize present systems in the real world has forced to develop several tools that allow recreating the physical world in a computer. Currently, physical systems tend to be more complex implying that the mathematical model is also complicated. In this paper, we present the solution of a dynamic optimization problem using the physics engine NVIDIA PhysX for virtual reality in order to reproduce the behavior of a nonlinear system as a case study demonstrating the manipulation and recreation of the behavior of the system without the need to have the dynamic model of it.

Keywords: differential evolution, Unity 3D, optimization, PD control.

1. Introducción

En la actualidad el uso de nuevas herramientas tecnológicas de virtualización para el manejo de sistemas ha permitido el avance y crecimiento en diferentes áreas del conocimiento. El uso de simuladores ha facilitado el desarrollo de nuevos productos a un menor costo, reduciendo así su tiempo de producción. Permitiendo que el desarrollo de los productos tenga un mejor enfoque antes de ser elaborado y su vida útil se prolongue. La tecnología computacional ha crecido a pasos agigantados brindado diversos enfoques para un mismo objetivo. La Realidad Virtual representa escenas o imágenes de objetos producidas por un sistema informático que permite crear la sensación de su existencia real para un usuario. Para lograr una buena aproximación de los fenómenos físicos que se representan se han desarrollado múltiples y muy diversos motores de física para calcular el comportamiento de los sistemas de interés [2,3,18].

La creación del modelo de manera virtual permite al usuario tener una mejor visualización del comportamiento del sistema sin tener el desgaste del mismo. Esta es una ventaja que brinda virtualizar el sistema. Unity 3D es una herramienta de desarrollo de juegos en 3D integral multiplataforma desarrollada por Unity Technologies [5]. Unity3D es más prominente en las capacidades de representación tridimensional, el detalle y el rendimiento sólido, admite 3DMAX, MAYA, Sketcheup, CAD y otros modelos tridimensionales [17]. La tecnología de recorrido virtual es una herramienta que se utiliza en el diseño y la presentación de arquitecturas. Integra gráficos de computadora, multimedia, inteligencia artificial, tecnología de sensores múltiples, redes, etc., nos brinda un sólido soporte para crear un mundo virtual [8]. Así mismo, al utilizar la computadora para generar un entorno de simulación, se puede recrear el mundo físico de una manera sencilla y darse cuenta de la interacción natural que se presenta con el entorno sin dañar el mismo. El modelar un sistema comienza por identificar sus elementos y las relaciones entre ellos. El modelar es multifacético o por perspectiva porque el modelo que se construya de un sistema real depende del objetivo del usuario que lo modela [19].

Un sistema físico debe ser representado en un modelo matemático para poder simular su comportamiento. Sin embargo, entre más complejo sea el sistema el modelo matemático también incrementa su complejidad. Es por ello que diversos

sistemas existentes en el mundo físico aun no tienen un modelo que lo describa. El proceso de modelado analítico se divide en dos grandes etapas importantes para cualquier sistema. La primera es delimitar el modelo en función de los fenómenos físicos relevantes a la problemática a resolver. En esta etapa, la sistematización del sistema no es fácilmente aplicable y requiere de una vasta experiencia en relación con el sistema a modelar. La siguiente etapa consiste en formalizar las relaciones constitutivas y estructurales asociadas respectivamente a los fenómenos considerados y a la forma en que estos se disponen dentro del sistema. En los sistemas físicos, estas relaciones constitutivas y estructurales encuentran su expresión matemática en las leyes fundamentales de los dominios de la física asociados a los fenómenos mencionados. Los modelos matemáticos describen las relaciones existentes entre las magnitudes caracterizantes del sistema, vinculando de esta manera variables matemáticas representativas de las señales en el sistema [6,7,9].

De manera general todos los trabajos necesitan la planta o un modelo para realizar las pruebas necesarias de las teorías que se estén desarrollando. En el caso de sistemas de control es una práctica habitual modelar el sistema de forma matemática y realizar el análisis a lazo abierto y a lazo cerrado en conjunto con el controlador que se esta evaluando [10,11,13].

Actualmente se utilizan herramientas computacionales que permiten utilizar el sistema físico en un ambiente virtual sin comprometer el desempeño ni la vida útil del mismo. El ambiente virtual NVIDIA PhysX es una tecnología especialmente diseñada para su aceleración por hardware a través de procesadores de alta capacidad dotados de cientos de núcleos. Los cuales proporcionan un crecimiento exponencial de la potencia de cálculo de la física que da lugar a entornos de juego increíblemente ricos y realistas gracias a mejoras como personajes de geometrías complejas y perfectamente articuladas para proporcionar interacciones y movimientos más naturales. Una de las herramientas utilizadas es Unity ya que tiene el mejor motor de física, lo que implica que al ser aplicada a sistemas permite un realismo impresionante tanto en la visualización como en el comportamiento del mismo.

De esta manera, Unity brinda la ventaja más importante del desarrollo: no se requiere tener el modelo matemático del sistema para poder recrearlo, incluso algún modelo complejo que aún no tenga su representación matemática puede ser implementado en esta herramienta computacional. Con ello permite configurar las dimensiones de los elementos del sistema de manera precisa y fácil, y reconfigurar el sistema cuando se desee. Modificar no solo sus dimensiones sino los parámetros que puede tener cualquier sistema físico. Unity brinda al usuario la facilidad de realizar cálculos de colisiones, fuerzas ejercidas, torques y fricciones del sistema. Al trabajar con este tipo de herramientras se brinda libertad para trabajar en el control de la posición y velocidad, ya que sus características son ajenas a cualquier ley de control aplicada al sistema.

Otra de las mayores ventajas que presenta el uso de herramientas como NVIDIA PhysX es el tener instancias múltiples del sistema con el que se está trabajando. Es decir, se emula cierta cantidad de sistemas a la par con las mismas

características pero aplicandole diferente control.

Por otro lado, la optimización es un proceso que pretende encontrar una solución o conjunto de soluciones que maximice o minimice lo que se conoce como función objetivo, es decir, una o varias funciones que en su conjunto permitan obtener el resultado deseado. En el área de optimización en ingeniería, ésta se define como el acto de obtener los mejores resultados bajo ciertas circunstancias [12]. Algunos otros autores definen la optimización como el proceso de buscar la mejor solución posible a un problema, bajo ciertas circunstancias [16]. Los métodos de optimización son conocidos como técnicas de programación matemática, las cuales son aplicables a problemas de toma de decisiones, así como al establecimiento de las mejores soluciones posibles.

En este trabajo se presenta la solución de un problema de optimización dinámico haciendo uso de motor de física para realidad virtual con el fin de reproducir el comportamiento del sistema de estudio sin la necesidad de tener el modelo dinámico del sistema. El problema de optimización planteado representa el problema de sintonización óptima de un controlador PD. La solución se obtiene al aplicar una técnica metaheurśtica para obtener las ganancias apropiadas para conseguir que el controlador lleve al sistema a la posición deseada. Adicionalmente, se presenta el modelo matemático del sistema y la solución del mismo problema haciendo uso del modelo. La información mostrada en este artículo se estructura de la siguiente forma: en la sección II se presenta el modelo dinámico del sistema no lineal del péndulo simple con el cual se va a trabajar, en la sección III se presenta la virtualización del sistema en el entorno de Unity3D, la sección IV muestra el problema de sintonización como un problema de optimización, la sección V expone el algoritmo implementado y finalmente en la sección V y VI se presentan los resultados y las conclusiones.

2. Modelo dinámico del péndulo simple

Sea el sistema mecánico denominado péndulo simple como se muestra en la Fig. 1, donde m denota el elemento mecánico situado en el extremo de una barra de masa despreciable con una longitud l. Por otro lado τ define el par aplicado y el coeficiente de fricción viscosa es b, el cual se debe a la fricción en los sistemas mecánicos de rodamiento.

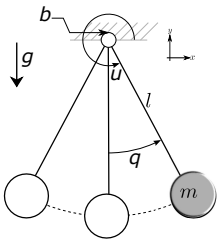


Fig. 1. Péndulo simple.

De acuerdo con el Método de Euler-Lagrange, las ecuaciones dinámicas para un sistema de n coordenadas generalizadas (q), están dadas por la ecuación (1).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) + \frac{\partial D}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau, \tag{1}$$

donde:
L = Lagrangiano del sistema,
τ = par,
D = función de disipación de Rayleigh para sistemas amortiguados.

El Lagrangiano del sistema y la función de disipación de Rayleigh, están dados por la ecuación (3):

$$L = K - U, \tag{2}$$

$$D = \frac{1}{2} b\dot{q}^2, \tag{3}$$

donde:
$U = -mglcos(q)$, energía potencial,
$K = \frac{1}{2} m\dot{q}^2 l^2$, energía cinética.

Debido a que el movimiento de la masa está limitada a permanecer en un círculo de radio igual a la longitud de la barra. El ángulo θ se considera como la variable de salida, y se designa a q como la coordenada generalizada, por lo tanto $q = \theta$, la cual indica la posicíon de la masa del péndulo. Por lo que al sustituir U y K, en el Lagrangiano se obtiene la ecuación de movimiento que describe al sistema del péndulo simple, la cual se presenta en la ecuación (4):

$$ml^2\ddot{q} + b\dot{q} + mglsin(q) = \tau. \tag{4}$$

A partir de la ecuación de movimiento descrita en la ecuación (4), el vector de variables de estado es el siguiente $x = [x_1, x_2]^T = [q, \dot{q}]^T$ y la entrada del sistema es τ, la dinámica en el espacio de estados se presenta en la ecuación (5):

$$\begin{aligned} \dot{x} &= f\left(\boldsymbol{x}, \tau(t), t\right), \\ \dot{x} &= \begin{bmatrix} x_2 \\ \dfrac{\tau}{ml^2} - A\dot{x_2} - Csin(x_1) \end{bmatrix}, \end{aligned} \tag{5}$$

donde:

$$A = \frac{b}{ml^2},$$
$$C = \frac{g}{l}.$$

En [15] se realizó la construcción del sistema real de péndulo simple para comparar el comportamiento del sistema virtualizado con el físico. En este sistema el único parámetro medido fué el peso del péndulo y se hizo por medio de una báscula para después dividir ese valor entre la gravedad (g) y finalmente obtener la masa del sistema. Por otro lado, el resto de los párametros del sistema no se asumen o se miden de forma directa debido a que no se consideró que la distribución de la masa fuera homogénea y la construcción del péndulo fuera completamente simétrica. Por lo tanto, se implementó un estimador de parámetros para calcular la posición del centro de masa (l) y la constante de fricción viscosa (b) como se muestra en la Fig. 2.

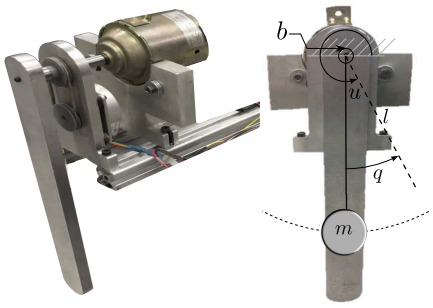


Fig. 2. Sistema: péndulo Simple.

Finalmente todos los parámetros descritos en la Fig. 2 del sistema se muestran en la Tabla 1.

Tabla 1. Parámetros.

Parámetros	Valor
l	0.2326 m
b	0.0015 Nm/s
m	0.04617 kg
g	9.81 m/s

3. Sistema del péndulo en Unity3D

El sistema de péndulo simple fue configurado en una escena de Unity3D. El primer paso fue realizar una base para sostener el péndulo la cual se diseñó de dos partes: una base con forma cuadrada y un eslabón fijo perpendicular a la base con la altura necesaria para ver el comportamiento del sistema. Se fijó el punto de unión entre la base y el péndulo. Se añadió el péndulo como se visualiza en la Fig. 2, al cual se le proporcionó las mismas caracteristicas dandole el valor de la fricción, la masa presente en el péndulo y la cinemática del sistema. Así

mismo se le agregó la opción de uso de gravedad para tener un sistema idéntico al físico pero de manera virtual, como se muestra en la Fig. 3.

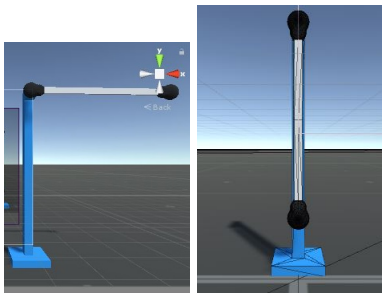


Fig. 3. Péndulo simple virtualizado.

4. Enfoque de optimización

Este enfoque permite el cálculo de cada una de las constantes del controlador para que el sistema de péndulo simple realice la acción de control de posición.

4.1. Función objetivo

En la ecuación (7) se expresa la función objetivo la cual evalua el rendimiento del sistema es la sumatoria del error entre el comportamiento deseado y la respuesta de la planta y una función de sobretiro. Esta función evalúa la integral del error cuadrático como resultado de un conjunto dado de parámetros y un tiempo de simulación de t_0 a t_f. En este caso, el vector $\boldsymbol{p}$ contiene las variables de diseño y n indica la dimensionalidad del problema.

4.2. Variables de diseño

La solución del problema de optimización debe obtener el conjunto de parámetros para satisfacer los requisitos de diseño. Por lo tanto, el vector de diseño en (6) contiene el conjunto de parámetros mencionado. Los componentes de este vector son las ganancias del controlador PD: la constante del controlador de proporcionalidad (K_p) y la constante del controlador derivativo (K_d). El problema se debe acotar con los valores límite para cada una de las variables en (9)-(10). Estas cotas representan la energia del sistema definida por el usuario:

$$\begin{aligned} &\boldsymbol{p} \in \mathbb{R}^2, \\ &\boldsymbol{p} = [K_p, K_d]^T. \end{aligned} \tag{6}$$

4.3. Problema de optimización

El problema descrito anteriormente se resume en las ecuaciones (7)-(10), el cual se trata de un problema de optimización dinámico mono-objetivo sin restricciones y las variables de diseño se acotan a valores mínimos y máximos.

$$\underset{\boldsymbol{p}\in\mathbb{R}^2}{min} \quad f_1(\boldsymbol{p}) = \int_{t_0}^{t_f} e^2 dt + F_{sobretiro}, \tag{7}$$

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}(u,\boldsymbol{p},t), u(\boldsymbol{p},t), t), \tag{8}$$

$$considerando \quad las \quad siguientes \quad cotas:$$

$$0 < K_p \leq 40, \tag{9}$$

$$0 < K_d \leq 40, \tag{10}$$

$$\tag{11}$$

$$F_{sobretiro} = \begin{cases} 0 & \text{si } sobretiro < 0 \\ sobretiro & \text{si } sobretiro > 0 \end{cases} \tag{12}$$

$$sobretiro = q_{max} - q_d(t_{qmax}), \tag{13}$$

siendo t_{qmax} el instante donde la existe el sobretiro máximo.

5. Evolución diferencial

Este estudio usa el Algoritmo de Evolución Diferencial[14] para resolver el problema de optimización planteado. Esta herramienta ha sido usada ampliamente y se ha posicionado como una buena herramienta para la resolución de problemas de ingeniería [4]. Este algoritmo basa su comportamiento en la reproducción de individuos en una población y el reemplazo de individuos por sus hijos se da únicamente si el hijo es mejor que el padre respecto de la función objetivo. Para la solución de los problemas planteados en este trabajo se hizo uso de la versión de Evolución Diferencial: rand/1/bin el cual fue programado en C# haciendo uso del pseudocódigo mostrado en Fig. 4.

Los parámetros a obtener son los del controlador PD. Este controlador es utilizado para llevar al sistema de péndulo simple a la posición deseada y cumplir con la tarea de seguimiento es un PD (siglas de Proporcional y Derivativo). Este tipo de controlador se encarga de aplicar una acción correctora para que el error entre la señal de referencia y la señal de salida de la planta, se reduzca. Teniendo como ventajas la estabilidad general del sistema, es capaz de manejar procesos con retraso de tiempo y reduce el tiempo de asentamiento al mejorar la amortiguación y reducir el sobreimpulso. El PD convencional tiene solo dos parámetros para ajuste [1] como se describe en la ecuación (14):

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt}, \tag{14}$$

Algorithm 1: DE/rand/1/bin

```
create a random population ^0x_i i = 1, ...NI
evaluate initial fitness population f_0 = f(^0x_i)
for g := 1 to G_max do
    for i := 1 to NI do
        select randomly {r_1 ≠ r_2 ≠ r_3}
        for j := 1 to NV do
            if U(0, 1) < CR then
            v_ij = r_1j + F(r_2j − r_3j)
            else
            v_ij =^{g−1} x_ij
            end
        end
        if ^g v_i <<^{g−1} x_i then
        ^g x_ij =^g v_i
        else
        ^g x_ij =^{g−1} x_i
        end
    end
end
```

Fig. 4. Algoritmo ED/rand/1/bin

definiendo:

$$e(t) = q_d(t) - q(t), \tag{15}$$

donde:
$e(t)$ es el error existente entre la posición deseada y la posición actual.
q_d es la posición deseada a la cual debe de llegar el péndulo.
$q(t)$ es la posición actual en la que se encuentra el sistema.

6. Resultados

El algoritmo se ejecutó treinta veces con el fin de resolver el problema de optimización planteado en la ecuación (7). La inicialización de parámetros del algoritmo se estableció como se muestra en la Tabla 2.

Tabla 2. Parámetros de inicialización.

Descripción	Valor
Individuos	6
Generaciones (G_{max})	50
Factor cruza	0.8
Factor de mutación	0.5

En la Fig. 5 se muestra como el algoritmo de Evolución Diferencial en conjunto con Unity 3D corren los 6 individuos de forma paralela teniendo cada uno su gráfica del comportamiento en cada generación y los valores obtenidos al finalizar dicha generación. Así mismo, se observa como funciona cada uno de los individuos en cada evaluación que se realiza. El mostrar el valor obtenido de cada péndulo al paso de las generaciones permite al usuario observar las soluciones brindadas por el algoritmo de Evolución Diferencial y el comportamiento del péndulo en cada solución.

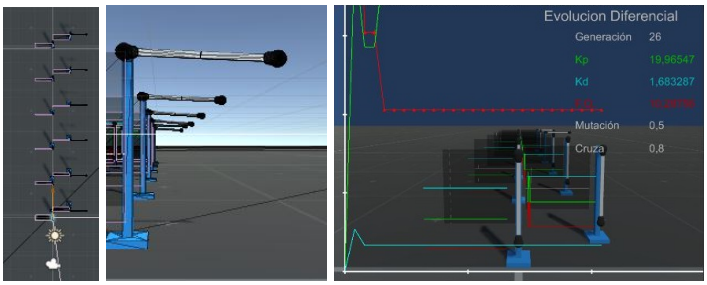


Fig. 5. Péndulos trabajando en paralelo.

En la Tabla 3 se muestran los resultados obtenidos al finalizar las treinta corridas del algoritmo de optimizacion. Se puede observar que los resultados obtenidos de despues de la ejecución del algoritmo de optimización treinta veces. Se muestra el mejor valor objetivo el cual es el que tiene menor valor, es decir el que implica que el error es casi cero y la función sobretiro es cero.

Tabla 3. Estadística.

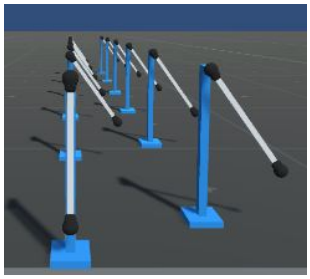
	Valor de la Función objetivo
Mejor	$3{,}8925x10^{-5}$ $[rad]$
Mediana	$4{,}0875x10^{-5}$ $[rad]$
Peor	$4{,}9456x10^{-5}$ $[rad]$
Desviación Estandar	$5{,}6026x10^{-6}$ $[rad^2]$

$$\boldsymbol{p}^* = [40{,}0000 \quad 23{,}439]^T, \tag{16}$$

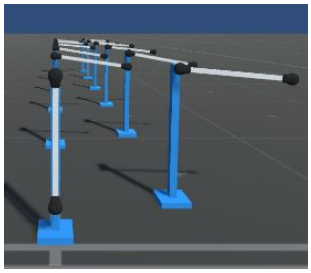
$$f(\boldsymbol{p}^*) = 3{,}8925x10^{-5}. \tag{17}$$

En la Fig. 6.(a) se muestra la inicializacion de los péndulos y la Fig. 6.(b) muestra que se están evaluando todos los hijos y todos los padres en el ambiente

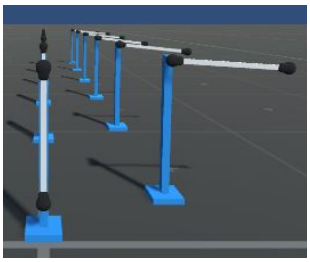
virtual. Por otro lado, en la Fig. 6.(c) se observa que a partir de la segunda generación únicamente los hijos son evaluados. En estas imagenes resulta sencillo observar las propiedades de paralelización con las que cuenta el uso del ambiente virtual de Unity y de esta manera se evaluan en un mismo *mundo* todos los individuos de la población.



(a) Inicio



(b) Padres



(c) Hijos

Fig. 6. Péndulos al paso de las generaciones.

7. Conclusiones

El presente trabajo muestra la posibilidad de hacer experimentación científica sobre fenómenos descritos por las leyes de Newton sin la necesidad de tener el modelo matemático del mismo. Esto se logra haciendo uso de los motores de física que hoy en dia se desarrollan para múltiples actividades los cuales resultan muy útiles para virtualizar el modelo de un sistema y tomar un camino alterno a la hora de correr simulaciones utilizando equipo de cómputo convencional. Este enfoque se probó al aplicarlo al virtualizado del sistema de un péndulo simple el cual sirvío para solucionar el problema de sintonización de un controlador PD para llevar al sistema a una posición deseada. La solución a este problema se obtuvó aplicando el algoritmo de Evolución Diferencial al problema de optimización resultante. De este modo se comprobó la viabilidad y aplicación de virtualizar sistemas mecánicos para la aplicación de técnicas de estudio. Esta herramienta representa un camino alterno para sistemas complejos

de modelar pero relativamente sencillos de virtualizar. Los resultados obtenidos fueron satisfactorios tanto para reproducir el comportamiento del sistema como para lograr la sintonización óptima de las ganancias de un controlador PD para llevar al péndulo a la posición deseada.

Agradecimientos. El primer, quinto y sexto autor agradecen al CONACyT el apoyo otorgado a través de la beca para estudios de doctorado. Todos los autores agradecen al Instituto Politécnico Nacional por el apoyo otorgado a través de la Secretaria de Investigación y Posgrado por medio de los proyectos SIP20196288 y SIP20190245.

Referencias

1. Asl, R.M., Pourabdollah, E., Salmani, M. (eds.): Optimal fractional order PID for a robotic manipulator using colliding bodies design. Soft Comput (2017)
2. Bose, M., Rajagopala, V.: Physics engine on reconfigurable processor — low power optimized solution empowering next-generation graphics on embedded platforms. In: 2012 17th International Conference on Computer Games (CGAMES). pp. 138–142 (July 2012)
3. Bzhikhatlov, I., Perepelkina, S.: Research of robot model behaviour depending on model parameters using physic engines bullet physics and ode. In: 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). pp. 1–4 (May 2017)
4. Cabrera, J., Simon, A., Prado, M.: Optimal synthesis of mechanisms with genetic algorithms. Mechanism and Machine Theory 37(10), 1165–1177 (2002), http://www.sciencedirect.com/science/article/pii/S0094114X02000514
5. He, Z., Shi, M., Li, C.: Research and application of path-finding algorithm based on unity 3d. In: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS). pp. 1–4 (June 2016)
6. Ito, Y., Fujimoto, K., Tadokoro, Y., Yoshimura, T.: Risk-sensitive linear control for systems with stochastic parameters. IEEE Transactions on Automatic Control 64(4), 1328–1343 (April 2019)
7. Jadbabaie, A., , Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on Automatic Control 48(6), 988–1001 (June 2003)
8. Jie, J., Yang, K., Haihui, S.: Study on the virtual natural landscape walkthrough by using unity 3d. In: 2013 International Conference on Computational and Information Sciences. pp. 1–4 (June 2013)
9. Lamperski, A., Ghusinga, K.R., Singh, A.: Analysis and control of stochastic systems using semidefinite programming over moments. IEEE Transactions on Automatic Control 64(4), 1726–1731 (April 2019)
10. Mohapatra, T.K., Sahu, B.K.: Design and implementation of ssa based fractional order pid controller for automatic generation control of a multi-area, multi-source interconnected power system. In: 2018 Technologies for Smart-City Energy Security and Power (ICSESP). pp. 1–6 (March 2018)
11. Nayak, P.C., Prusty, U.C., Prusty, R.C., Barisal, A.K.: Application of sos in fuzzy based pid controller for agc of multi-area power system. In: 2018 Technologies for Smart-City Energy Security and Power (ICSESP). pp. 1–6 (March 2018)

12. Rao, S.S.: Engineering optimization: Theory and practice. John Wiley and Sons, Inc. (New York, USA 1994)
13. Shutnan, W.A., Abdalla, T.Y.: Artificial immune system based optimal fractional order pid control scheme for path tracking of robot manipulator. In: 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA). pp. 19–24 (March 2018)
14. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (Dec 1997), https://doi.org/10.1023/A:1008202821328
15. Suárez-Santillán, R.A., Hernández-Oliva, N., Morales-Cruz, C., Calva-Yánez, M.B., Sepúlveda-Cervantes, G., Portilla-Flores, E.A.: Implementación en hardware-in-the-loop de un esquema completo de control para un sistema de péndulo simple utilizando un controlador pid.
16. Velázquez Reyes, J.: Evolución diferencial para problemas de optimización de espacios restringidos. Master's thesis, Sección de Computación, CINVESTAV-IPN (México 2006)
17. Xiaofeng, L., Tianhuang, C., Tingchao, Q.: Making a virtual sand table based on unity 3d technique. In: 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. pp. 278–281 (Nov 2014)
18. Yang, J., Hu, G.: A physics simulation engine for modeling autonomous motion through reverse engineering of sodaconstructor. In: 2009 IEEE/ACS International Conference on Computer Systems and Applications. pp. 287–292 (May 2009)
19. Zeigler, B.P., Muzy, A.: From discrete event simulation to discrete event specified systems (devs). IFAC-PapersOnLine 50(1), 3039–3044 (2017), http://www.sciencedirect.com/science/article/pii/S2405896317310601, 20th IFAC World Congress

ISSN 1870-4069

Integración de aprendizaje por refuerzo a simulador físico y análisis de funciones de recompensa para robots humanoides

Alfonso Brown González, Fernando Cossío Ramírez, Edgar Omar López Caudana

Instituto Tecnológico y de Estudios Superiores de Monterrey,
Ciudad de México, México
{abrown.glez,oissoc.ref}@gmail.com, edlopez@tec.mx

Resumen. Se presenta la integración de un entorno de simulación del robot NAO en Webots junto con el entorno de programación para aprendizaje por refuerzo de OpenAI: Gym. Se utilizan funciones de recompensa para medir el desempeño de distintos entornos de aprendizaje y así definir cuáles son las características, variables y parámetros con las que esta función debe contar para llegar a un resultado observable. El proyecto se encuentra en desarrollo por lo que aún no existen resultados definitivos. El presente artículo concluye que hace falta trabajo para alcanzar la meta principal, pero que los avances actuales son observables y medibles. El proyecto final busca crear un sistema de control jerárquico basado en inteligencia artificial y compararlo con los sistemas de control que existen actualmente para el NAO. Por tanto, una vez terminada la primera etapa, se utilizarán dichas funciones de recompensa en un controlador de bajo nivel y se desarrollará el entorno de entrenamiento por RL para tareas de alto nivel.

Palabras clave: aprendizaje de máquina, aprendizaje por refuerzo, NAO.

Reinforcement Learning Framework Integration with Physics Simulator and Reward Function Analysis in Humanoid Robot

Abstract. We present the integration of a NAO robot in Webots with the Reinforcement Learning framework from OpenAI: Gym. We then try different reward functions to measure the ability of a single algorithm to learn, in order to determine the key characteristics, variables, and parameters that this function should have in order to reach observable results. This is a working proyect so the results are not final. This article concludes that there is still work to be done in order to reach the objective, but that actual results are already measurable. On the long term, we intend to design a hierarchical control system based on Reinforcement Learning, and compare it to other control systems currently being used

with the NAO robot. Therefore, once we complete this first stage, the reward functions defined here will be used for a Low Level Controller, and a training environment for a High Level Controller will be designed.

Keywords: machine learning, reinforcement learning, NAO.

1. Introducción

1.1. Problemática y justificación

Poco a poco las redes neuronales y las tendencias de inteligencia artificial se han ido apoderando de áreas en donde la programación tradicional solía reinar. Uno de estos campos es la robótica, donde muchos problemas de control y realización de tareas han empezado a migrar a soluciones más óptimas basadas en IA. Sin embargo, la implementación de estas nuevas tecnologías no siempre es directa ni sencilla. Este trabajo busca presentar una alternativa amigable de integración y evaluación de algoritmos de Aprendizaje por Refuerzo (RL por sus siglas en inglés) en un simulador físico, para luego llevar el aprendizaje y tareas realizadas al robot real. Buscamos desarrollar inicialmente un estándar de programación a seguir para poder integrar *toolkits* y APIs de alto nivel (como Gym, Baselines, Keras, entre otros) que simplifiquen el proceso de diseñar y evaluar algoritmos de aprendizaje en un simulador físico como Webots.

1.2. Objetivos

El objetivo general del proyecto es: "Desarrollar un estándar de programación y entorno de simulación de robots basado en RL que permite entrenar tareas definidas por una función de recompensa y evaluar el desempeño del aprendizaje y los algoritmos utilizados."

Específicamente, se plantean los siguientes objetivos:

- Desarrollar un ambiente de simulación del robot humanoide NAO, utilizando el framework para Aprendizaje por Refuerzo de OpenAI, Gym.
- Diseñar y probar funciones de recompensa para lograr tareas de bajo nivel[1] y medir las variables de estado claves en conseguir que el robot aprenda a resolver la tarea de manera eficiente.
- Evaluar diferentes algoritmos de RL, e hiperparámetros de los mismos, según su eficiencia y capacidad para converger a modelos que solucionan las tareas propuestas.

2. Marco teórico

2.1. Aprendizaje por refuerzo

El término de Aprendizaje por Refuerzo, RL, es una de las tres grandes áreas del Aprendizaje de Máquina (*Machine Learning*) junto con el Aprendizaje

[1] Como mantener el balance y dar pasos para caminar

supervisado y el Aprendizaje no supervisado. Surgió como una solución a los procesos de decisiones de Markov (MDP por sus siglas en inglés). Gráficamente un MDP se puede representar como una cadena de Markov, como se muestra en la figura 1.

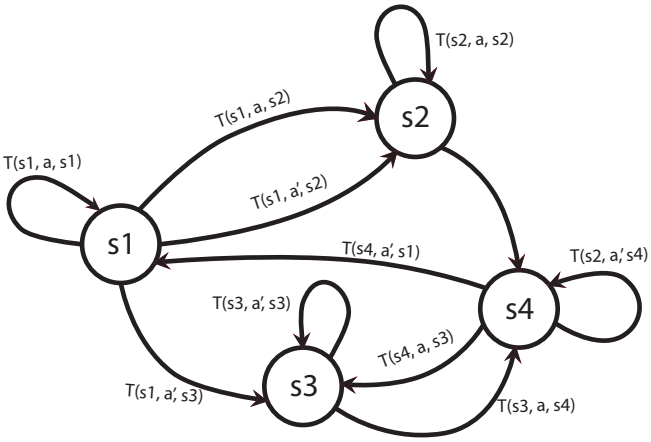


Fig. 1. Ejemplo de la representación gráfica de un proceso de Markov sencillo con 4 estados (s1, s2, s3, s4) y dos acciones (a, a'). La función T representa la probabilidad de transición de un estado a otro dada una acción. A cada flecha de transición de estado también le corresponde una recompensa.

Estos procesos cuentan con un espacio de estados S y un espacio de acciones A y cumplen con algunas características, como que existe una probabilidad $T(s, \pi(s), s')$ de transicionar del estado $s \in S$ al estado s' dada una acción $a \in A$ determinada por la política $\pi(s)$. Entonces, para cada s está definida una política $\pi(s)$ que determina la acción a tomar. Además, en cada transición existe una función de recompensa $R(s, \pi(s))$, que se busca maximizar. De todo esto surgen los valores V, definidos por la ecuación de Bellman como:

$$V^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V^{\pi}(s'), \tag{1}$$

donde γ es el factor de descuento, que sirve para dar preferencia a las recompensas obtenidas en la menor cantidad de transiciones (o en el menor tiempo) posibles.

Sin embargo, resolver un MDP por valores de estado V se vuelve en extremo tedioso y complicado, sobre todo porque en aplicaciones reales la primera vez que un agente es sometido a un entorno, no se conoce por completo el espacio de estados, ni mucho menos la probabilidad de transición entre ellos. De esto surgió el algoritmo de Q-Learning [12], el cual modifica la ecuación de Bellman para encontrar valores Q de cada par acción-estado sin necesidad de conocer la

función T. Estos valores Q se calculan por el siguiente método iterativo:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(r_t + \gamma \max_a(Q(s_{t+1}, a)), \tag{2}$$

donde α es el factor de aprendizaje, y r_t es la recompensa de la transición en el instante t.

Este algoritmo, junto con sus variantes de aprendizaje profundo que incorporan redes neuronales para calcular los valores Q, es bueno para entornos relativamente simples, y preferiblemente con espacios de estado y acción discretos. Para entornos más sofisticados y continuos, como lo es la simulación de un robot, es necesario recurrir a algoritmos que actúan directamente sobre la política π [5,6,9].

Proximal Policy Optimization (PPO) [10]

En 2017 investigadores de OpenAI desarrollaron un algoritmo que buscaba ser escalable, eficiente y robusto, de tal forma que sin modificar los hiperparámetros se pudieran obtener buenos resultados en distintos entornos, discretos o continuos. Con esta intención surgió PPO que, como todos los algoritmos de gradiente de política, calcula una estimación del gradiente basado en las recompensas y realiza un ascenso estocástico. La ventaja es que además incorpora regiones de confianza (basado en algoritmos de TRPO) [8], que evitan que el ascenso de gradiente se aleje demasiado de la máxima recompensa hasta el momento, para no causar actualizaciones destructivas a la política. Se calcula entonces una pérdida L^{CPI}, donde CPI es la abreviación de "Conservative Policy Iteration" (Actualización de política conservadora), como:

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t[r_t(\theta)\hat{A}_t], \tag{3}$$

donde $r_t(\theta)$ es la razón de probabilidad $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$, $\hat{\mathbb{E}}_t$ es la expectativa y $\hat{A}_t$ es una estimación de la ventaja en el tiempo t, basada en las recompensas. Se define además un hiperparámetro ϵ pensado para que $r_t(\theta)$ no se devíe del intervalo $[1-\epsilon, 1+\epsilon]$. La estimación de pérdida con región de confianza utilizando un factor de ajuste (*clip factor*) se calcula como:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]. \tag{4}$$

Se pueden lograr resultados similares utilizando otros métodos de ajuste para la región de confianza, como penalizaciones según la divergencia KL como describe [10].

Para incorporar además los valores de estado $V(s)$ y asegurar que en la implementación exista suficiente exploración, se introduce un bonus por entropía $S(\pi)$ y una pérdida por los valores de estado, y se obtiene la ecuación completa de pérdida:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]. \tag{5}$$

Finalmente, el algoritmo queda como:

Algoritmo 1 Proximal Policy Optimization.

para iteración=1,2,... **hacer**
  para actor=1,2,... **hacer**
    Correr la política $\pi_{\theta_{\text{old}}}$ en el entorno por T pasos
    Calcular estimados de ventaja $\hat{A}_1, ..., \hat{A}_T$
  fin para
  Optimizar L con respecto a θ, con K epocas y minibatch tamaño $M <= NT$
  $\theta_{\text{old}} \leftarrow \theta$
fin para

2.2. Esquema de trabajo Gym

Gym[2] es un *framework* open-source de entrenamiento por RL creado por OpenAI. Permite desarrollar y definir entornos por completo, partiendo de los agentes, los estados, las transiciones y sobre todo las funciones de recompensa. Los agentes además son fácilmente integrados con otras herramientas computacionales como Baselines, Keras o TensorFlow para realizar su entrenamiento. Gym establece la estructura general que debe seguir cualquier entorno en cuanto al orden de los directorios, las clases y métodos de Python que el agente utiliza para aprender. Existen más de 2000 entornos de entrenamiento públicos en internet y muchos proveen ejemplos de cómo integrar el entorno a otro software.

Para nuestro proyecto esperamos crear diferentes ambientes de aprendizaje, definiendo funciones de recompensa según la acción a realizar e integrando Gym a un simulador físico en donde podamos observar los resultados. Esta integración hace que no se necesite trabajo ni conocimientos en implementar todos los factores físicos involucrados en la respuesta del entorno y en las recompensas (como la gravedad, fuerzas, torque, etc.).

En alguna etapa posterior, ya que esperamos que este proyecto se vuelva accesible a cualquier persona, proponemos realizar el proceso de aprendizaje en entornos jerárquicos, como se observa en el proyecto DeepLoco[7]. De alguna forma se puede entender que el propósito es usar Gym como backend de nuestro software para un controlador a bajo nivel (LLC), mientras que una interfaz amigable define el controlador de alto nivel (HLC). El LLC se encarga de entrenar y premiar al robot por tareas sencillas, como mantener el equilibrio o dar un paso, mientras que el HLC se encarga de tareas complejas, como resolver un laberinto o patear un objeto.[2]

Los autores en [3] también proponen un sistema jerárquico en cuatro niveles para el robot NAO, basando las recompensas de los controladores a bajo nivel en

[2] Definimos tareas sencillas como problemas de control a bajo nivel, cuya solución consiste en llegar a un estado estable. Tareas complejas se refiere a tareas que dependen de variables no observables por el controlador de bajo nivel y que a menudo involucran la participación de un supervisor externo. En un caso de programación tradicional, el LLC viene implementado en las funciones ya definidas y sistema operativo interno del robot, mientras que el HLC sería el usuario que programa el robot con una finalidad en específico.

el Punto de Momento Cero. Nosotros buscaremos explorar diferentes recompensas, desde alternativas más generales (como el tiempo que se mantiene de pie), hasta otras más específicas basadas en el Centro de Masa o en la desviación de una pose ya definida.

2.3. Simulador Webots y robot NAO

El NAO[11] es un robot diseñado por Aldebaraan Robotics en el 2007, creado como una poderosa herramienta de programación y educación en robótica para niños, adolescentes y adultos jóvenes. Posteriormente se empezó a usar en competencias como el RoboCup y en diferentes aplicaciones académicas, de investigación y de cuidado de salud. Aldebaraan fue comprado en 2013 por una compañía japonesa llamada Softbank Robotics que hasta la fecha son quienes mantienen al robot NAO.

El robot actualmente se encuentra en su sexta iteración. Cuenta con 25 grados de libertad si se toman en cuenta los dedos y 21 si no. Opera con una versión de Linux llamada NAOqi como sistema operativo. Tiene además dos cámaras de alta definición, cuatro micrófonos, dos transmisores/receptores de infrarojo, un sonar, siete sensores de tacto, ocho sensores de presión y sensores de posicionamiento como acelerómetro y giroscopio.

Para trasladar el entrenamiento a un entorno físico y medible donde se puedan observar los resultados, integraremos nuestro entorno de aprendizaje de Gym con el simulador Webots [13]. Escogimos este simulador ya que cuenta con modelos hechos del robot NAO, con todos sus grados de libertad y con los sensores internos ubicados y calibrados de manera que simulan a los del robot real. Webots permite establecer si cada objeto en el entorno es un supervisor o funciona con un controlador. Un supervisor tiene más facultades sobre el entorno que un controlador, lo que nos posibilita reiniciar la posición del robot al final de cada episodio, para que el controlador únicamente se dedique a explorar y aprender a maximizar sus recompensas. En la figura 2 se muestra el robot nao en el simulador Webots.

3. Resultados

3.1. Integración de Webots, Gym y el modelo del robot NAO

Se hicieron pruebas con diferentenes simuladores físicos como PyBullet y MuJoCo. Sin embargo, no existen modelos libres del NAO ni la integración del robot con sus sensores internos. Al utilizar Webots, tenemos el modelo del robot integrado y en los programas de ejemplo se cuentan con ejercicios de calibración de los sensores para que operen de manera similar a los del robot real. Estos valores son los que utilizamos para generar el estado y las observaciones en los que se encuentra el NAO.

Lo primero que se logró es integrar los controles del robot virtual para que se puedan leer los sensores de la tabla 2. También logró leer las posiciones en

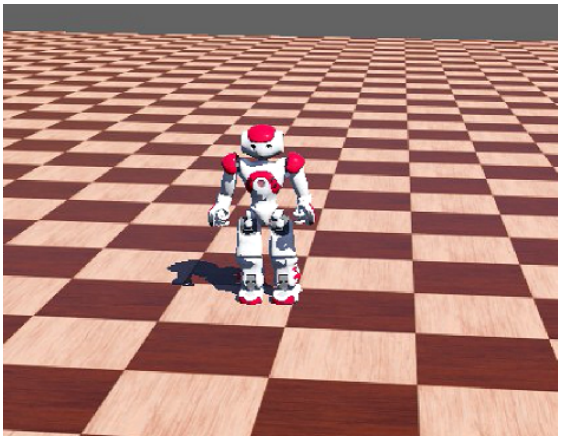


Fig. 2. Robot NAO en simulador Webots.

radianes y configurar la posición objetivo de las articulaciones mencionadas en la tabla 1. Cada articulación se linearizó de acuerdo a su posición angular mínima y máxima para que tanto la lectura como la escritura quedara en el intervalo $[-1.0, 1.0]$. Para obtener la posición del motor en dicho intervalo se utiliza la siguiente ecuación:

$$pos_linearizada = \frac{pos_{actual} - pos_{min}}{pos_{max} - pos_{min}} * 2 - 1. \tag{6}$$

También se hizo un ajuste en el controlador para que las articulaciones "RHipYawPitch" y "LHipYawPitch" se comporten de manera simétrica, como sucede en el robot real.

Una vez que se logró manejar el robot con código de Python en el simulador, se integró el framework de Gym y se creó el entorno de entrenamieto para poder correr los algorítmos de RL. Luego de lograr la integración exitosamente, es posible utilzar la librería de Stable Baselines [4] para implementar y programar en pocas líneas de código los agentes que aprenderán a controlar el robot. Esta librería ya tiene las arquitecturas y redes neuronales de varios algoritmos implementadas y utiliza TensorFlow como backend, lo que nos permitió utilizar TensorBoard como herramienta de evaluación y depuración. [1]

3.2. Entrenamiento para mantener el balance

Como una primera tarea a lograr establecimos el lograr mantenerse en pie sin información previa de como hacerlo. El algoritmo recibe como estado las posiciones de los motores y todos los sensores mencionados en la tabla 2 y puede tomar acciones con una frecuencia de 10Hz.

Logramos que se entrenaran 1M de timesteps del modelo en cuestión en poco menos de 10 horas, lo que significa que, aproximadamente, el entrenamiento se

Tabla 1. Articulaciones del robot NAO.

HeadYaw	HeadPitch	RShoulderPitch
RShoulderRoll	RElbowYaw	RElbowRoll
RWristYaw	RPhalanx1	RPhalanx2
RPhalanx3	RPhalanx4	RPhalanx5
RPhalanx6	RPhalanx7	RPhalanx8
LShoulderPitch	LShoulderRoll	LElbowYaw
LElbowRoll	LWristYaw	LPhalanx1
LPhalanx2	LPhalanx3	LPhalanx4
LPhalanx5	LPhalanx6	LPhalanx7
LPhalanx8	RHipYawPitch	RHipRoll
RHipPitch	RKneePitch	RAnklePitch
RAnkleRoll	LHipYawPitch	LHipRoll
LHipPitch	LKneePitch	LAnklePitch
LAnkleRoll		

Tabla 2. Sensores del robot NAO.

Acelerómetro	Giroscopio
Unidad Inercial	Sensores de presión
Bumpers	Ultrasónicos

hace con un factor de velocidad de 2.8x. El episodio inicia con una posición de las articulaciones que varía uniformemente en un rango de $\pm 0.25\,\%$ y el episodio termina si el robot cae. Para las pruebas, se congeló todo el tren superior (Cabeza, cuello, brazos) para reducir la complejidad del problema a solo 11 grados de libertad y lograr converger a un resultado en una cantidad menor de iteraciones.

Se utilizó el algorítmo PPO2 de Stable Baselines[3] con los parámetros por defecto para hacer pruebas preliminares y demostrar que el robot efectivamente está aprendiendo.

Primero definimos una función de recompensa muy general, donde cada 100 ms se otorgan 5 puntos simplemente por el hecho de que no se haya caído, sumado a la altura a la que el robot se encuentra en ese momento. Con esta función de recompensa, después de 40k pasos se llegó a un comportamiento poco estable en donde el robot brincaba para tratar de obtener la mayor altura, pero en promedio solo duraba 1.2 segundos de pie.

Posteriormente, se ajustó la función para descontar puntos por tener demasiada aceleración, demasiada velocidad, por utilizar torque (para reducir el consumo de energía y obtener movimientos más suaves) y por tener valores de gravedad en otro eje que no sea $-z$.

Específicamente se utilizó la siguiente función de recompensa r:

[3] Algoritmo descrito anteriormente, optimizado para usar GPU en un ambiente vectorizado, contra PPO1 de la misma librería que usa MPI, ya deprecado.

$$r = 2 - d_{torque} - d_{height} - d_{accel} - d_{vel} - d_{distance} - d_{pose}, \tag{7}$$

donde d denota el descuento calculado a partir de las siguientes ecuaciones (8-13):

$$d_{torque} = \frac{\sum_j \text{abs}(torque_j)}{60}, \tag{8}$$

$$d_{height} = (\text{abs}(z - 0.33) * 10)^2, \tag{9}$$

$$d_{accel} = \text{abs}(a_x) + \text{abs}(a_y) + \text{abs}(a_z + 9.81), \tag{10}$$

$$d_{vel} = \text{abs}(v_x) + \text{abs}(v_y) + \text{abs}(v_z), \tag{11}$$

$$d_{distance} = \sqrt{(x^2 + z^2)}, \tag{12}$$

$$d_{pose} = \frac{\sum_j \text{abs}(position_j - position'_j)}{12}. \tag{13}$$

Después de aproximadamente 100k pasos de simulación el robot aprendió a tirarse al piso lo más rápido posible para recibir el puntaje menos negativo posible. Ya que los descuentos resultaban siempre más grandes que la recompensa de mantenerse de pie. Para tratar de compensarlo, se modificó la función de la siguiente manera:

$$r = 5 - w_1 d_{torque} - w_2 d_{height} - w_3 d_{accel} - w_4 d_{vel} - w_5 d_{distance} - w_6 d_{pose}, \tag{14}$$

donde $[w_1, w_2, w_3, w_4, w_5, w_6]$ tomaron los valores $[0.5, 2, 1, 1, 0.5, 1]$.

Logramos que el robot se estabilizara por un par de segundos y después de 4M de iteraciones obtuvo una recompensa promedio por episodio de 42.6 y con una duración de pie promedio de 2.1 segundos. Bajo estas mismas recompensas, antes de entrenar la red, el modelo, que era aleatorio, tenía un promedio de recompensa por episodio de -8.35 puntos y una duración de 290 ms.

Además, para probar la robustez del algoritmo, se probaron diferentes hiper-parámetros. Un ejemplo de dos tamaños de red neuronal sobre la misma función de recompensa entrenada por un millón de episodios se puede ver en la figura 3. Se observa que, aunque aprenden con diferentes oscilaciones y ruido, al final la recompensa promedio es muy similar. De igual manera, todos los hiper-parámetros que se probaron no resultaron en diferencias en las recompensas estadísticamente significativas.

Por último, cambiamos la función de recompensa a que fuera más restrictiva, específica y que obligara al robot a encontrar una pose estable. Para esto definimos la pose de un paso izquierdo, ajustando ligeramente las posiciones de los motores hasta obtener un estado estable donde el centro de masa está sobre el

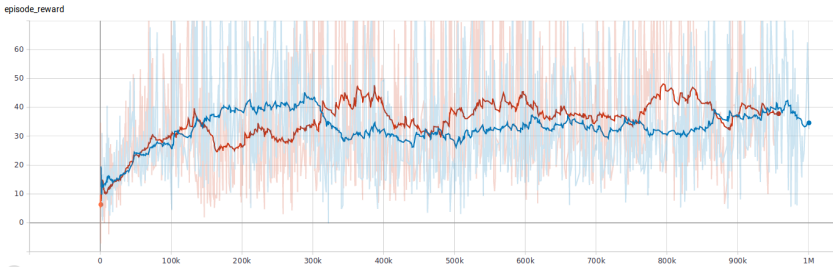


Fig. 3. Comparación de actuación de dos redes neuronales de diferentes tamaños sobre la misma función de recompensa.

eje z. Basándonos en el LLC implementado en DeepLoco[7], definimos la función de recompensa como:

$$r = -w_1 d_{torque} - w_2 d_{height} - w_3 d_{distance} - w_4 d_{pose}, \tag{15}$$

donde los pesos se ajustaron a $[w_1, w_2, w_3, w_4] = [0.1, 0.3, 0.1, 0.5]$ y el descuento por la posición esperada se modificó a:

$$d_{pose} = e^{-(\sum_j \text{abs}(position_j - position'_j))^2}. \tag{16}$$

Después de 400k episodios, el NAO logró una recompensa promedio por episodio de 7.42, y logró mantenerse de pie hasta 5 segundos. En la figura 4 se puede ver la diferencia entre las recompensas promedio del algoritmo sin entrenar y el agente entrenado. Así mismo, en la figura 5 se puede ver la posición semi-estable que alcanzó el robot después de esta etapa de entrenamiento.

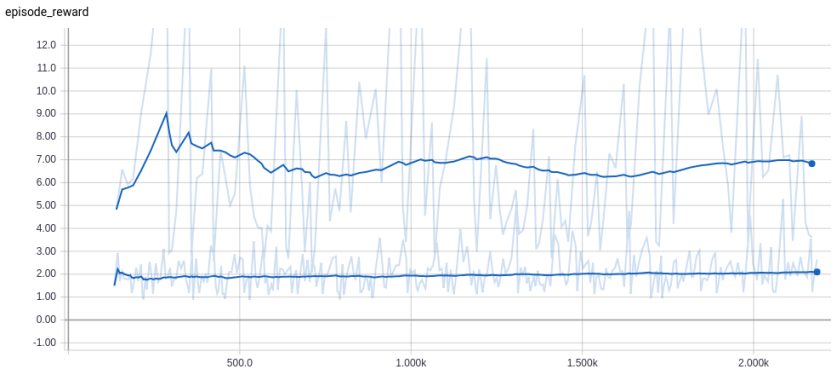


Fig. 4. Comparación de recompensas promedio por episodio entre agente sin entrenamiento y agente entrenado.

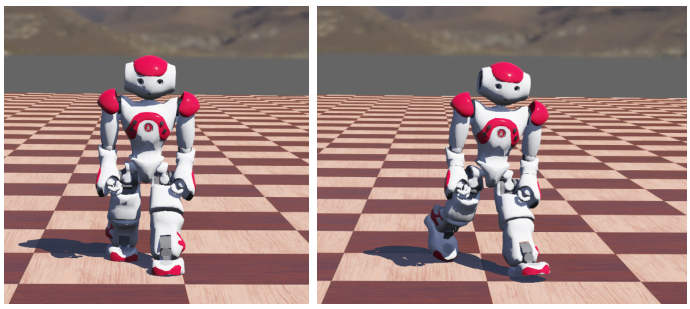


Fig. 5. Del lado izquierdo, la pose objetivo con la que se planteó el descuento. A la derecha el robot entrenado después de 400k episodios. En este episodio se mantuvo de pie y sin moverse por más de 4 segundos.

4. Conclusiones

4.1. Conclusiones del entrenamiento de prueba

En un análisis cualitativo de los resultados, se puede observar que el robot no cumple con la tarea de manera existosa aún. Sin embargo, sí se puede apreciar que el robot aprende a moverse de manera suave gracias al castigo por usar torque, que logra mantener el balance por mayor tiempo y se observa que está aumentando las recompensas promedio de cada episodio. Esto muestra que de alguna manera ha aprendido a mantenerse de pie por un breve intervalo de tiempo a pesar de las diferentes funciones de recompensa. Sin embargo, no hemos encontrado todos los factores más importantes a incluir para que el robot llegue a un estado estable, por lo que se cae después de unos segundos.

Al evaluar las pruebas realizadas, se puede concluir que:

- Para lograr una tarea continua (como mantenerse de pie), el entorno no debe dar una recompensa o castigo escaso, es decir, en el estado terminal del episodio, ya que se comporta como ruido.
- Vivir un timestep más debe ser estocásticamente mejor en recompensa que aventarse al suelo. Dicho en otras palabras, los castigos deben de ser, en promedio, no mayores a la recompensa de seguir de pie. Ya que de lo contrario, el algorítmo converge a terminar el episodio tan pronto como sea posible para evitar perder más puntos.
- Mantener el centro de masa y la aceleración en el eje z es indispensable para que el robot se mantenga de pie, por lo que la función de recompensa debe dar más peso a esto.

4.2. Trabajo a futuro

De acuerdo a los objetivos específicos del proyecto, solo se hemos cumplido por completo con el primero, aunque se ha progresado con el segundo

y tercero. El siguiente paso es seguir explorando recompensas y entrenando el robot para posteriormente establecer los estándares de integración de RL con el NAO y poder evaluar distintos algoritmos. Posteriormente, para poder entrenar las siguientes tareas de un carácter más complejo y comparar resultados contra la programación tradicional, debemos implementar la red neuronal, las recompensas y el algoritmo del controlador de alto nivel.

La velocidad de entrenamiento también es un reto a mejorar, puesto que se requieren tiempos de entrenamiento de al menos 1M de timesteps para lograr resultados similares a los de los humanoides de Roboschool (entorno ejemplo de Gym), que tienen 16 grados de libertad. Para esto, proponemos hacer uso de procesamiento en paralelo en una o varias computadoras.

Referencias

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, último acceso: 2019/05/11
2. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym (2016)
3. Gil, C., Calvo, H., Sossa, H.: Learning an efficient gait cycle of a biped robot based on reinforcement learning and artificial neural networks. Applied Sciences 9(3), 502 (Feb 2019), https://doi.org/10.3390/app9030502, último acceso: 2019/05/11
4. Hill, A., Raffin, A., Ernestus, M., Gleave, A., Traore, R., Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y.: Stable baselines. https://github.com/hill-a/stable-baselines (2018), último acceso: 2019/05/11
5. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning (2015)
6. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning (2013)
7. Peng, X.B., Berseth, G., Yin, K., van de Panne, M.: Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. ACM Transactions on Graphics (Proc. SIGGRAPH 2017) 36(4) (2017)
8. Schulman, J., Levine, S., Moritz, P., Jordan, M.I., Abbeel, P.: Trust region policy optimization (2015)
9. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation (2015)
10. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
11. SoftbankRobotics: NAO the humanoid robot (2018), https://www.softbankrobotics.com/emea/en/nao, último acceso: 2019/05/11
12. Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning 8(3-4), 279–292 (May 1992), https://doi.org/10.1007/bf00992698

13. Webots: http://www.cyberbotics.com, http://www.cyberbotics.com, commercial Mobile Robot Simulation Software

ISSN 1870-4069

Formulaciones para el cálculo eficiente del número de hoyos de un objeto binario 2-D

Juan H. Sossa[1,2], Elsa Rubio[1], Víctor H. Ponce[1], Luis E. Falcón[2]

[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Ciudad de México, México

[2] Tecnológico de Monterrey, Campus Guadalajara, Zapopan, Jalisco, México

humbertososa@gmail.com, erubio@cic.ipn.mx, vponce@cic.ipn.mx, luis.eduardo.falcon@tec.mx

Resumen. El determinar el número de hoyos o de huecos de un objeto es de particular interés en aplicaciones diversas en el campo del análisis de imágenes, por ejemplo, en el control de calidad de piezas forjadas industrialmente. En la literatura aparecen reportados pocos métodos. La mayoría utiliza al menos cuatro o cinco operaciones aritméticas. En este artículo se presentan dos formulaciones para el cálculo exacto del número de hoyos de un objeto binario 2-D. La primera formulación es útil para el caso de objetos 4-conectados, la segunda puede ser utilizada en el caso de objetos con conectividad tipo 8. Ambas formulaciones emplean un número mínimo de comparaciones; son demostradas teóricamente y validadas numéricamente en su operación a través de con un conjunto de objetos o formas de diferente complejidad y cantidad de hoyos.

Palabras clave: objeto conectado, forma conectada, Imagen binaria, bit-quad, número de hoyos.

Formulations for the Efficient Computation of the Number of Holes of 2-D Binary Object

Abstract. Determining the number of wholes of an object is of particular interest in several applications in the area of image analysis, for example, quality control of industrial machined parts. In literature we do not find too many methods. Most of them utilize at least four or five arithmetic operations. In this paper two formulations to compute the number of holes of a 2-D binary objet are provided. The first formulation is valid for the case of 4-connected objects, the second one can be used in the case of 8-connected shapes. Both formulations employ a reduced number of comparisons, and are theoretically demonstrated and numerically validated by using a set of objects of different complexity and number of holes.

Keywords: connected object, connected shape, binary image, bit-quad, number of holes.

1. Introducción

La determinación del número de hoyos (huecos) de un objeto puede ser de particular interés en aplicaciones diversas en el análisis de imágenes, por ejemplo, en el control de calidad de piezas forjadas industrialmente. Otras aplicaciones relacionadas con el cálculo del número de hoyos de una forma son, por ejemplo, en el reconocimiento de objetos [1], el umbralado de imágenes [2], el conteo de formas irregulares [3], el reconocimiento de placas de automóvil [4], y el filtrado de imágenes en niveles de gris [5].

El encontrar el número de hoyos de un objeto permite calcular el número de Euler de dicho objeto. En el caso dos-dimensional, por ejemplo, el número de Euler e de un objeto binario (un objeto, forma o región compuesta por pixeles k-conectados, con $k \in \{4,8\}$) viene dada, de acuerdo a lo dicho [1] y [3] como sigue:

$$e = 1 - h, \tag{1}$$

donde h es el número de hoyos del objeto.

La función de **MATLAB bweuler** (`syntax eul=bweuler(BW,n)`) permite calcular el número de Euler de una imagen binaria 2-D. Se basa en la formulación introducida a principios de los setentas en [6]. El argumento `n` puede tomar un valor de 4 o 8. Cuando `n=4`, la función **bweuler** permite calcular el número de Euler de una forma 4-conectada, mientras que cuando `n=8`, la misma función es útil para el caso de objetos 8-conectados.

Si ahora se quisiera calcular el número de huecos de una forma binaria 4(8)-conectada bastaría con calcular:

$$h = 1 - e. \tag{2}$$

Recientemente, en [7], los autores introdujeron la siguiente formulación para encontrar el número de huecos de un objeto 2-D:

$$h = 1 - (np - nc + nt - nd). \tag{3}$$

En este caso:

- np el número de pixeles del objeto.
- nc el número de lados que conectan a los np pixeles del objeto.
- nt el número de los llamados *tetra pixeles* encontrados en el objeto, y
- nd el número de los llamados pixeles diagonalmente conectados que se pueden encontrar en un objeto 8-conectado o estrictamente 8-connectado.

Como contribución al estado del arte en el cálculo del número de hoyos de un objeto 2-D, en este trabajo se describen dos formulaciones; ambas emplean un número muy limitado de operaciones. La primera formulación es una útil para el caso de objetos 4-conectados, mientras que la segunda es útil en el caso de regiones de píxeles 8-conectados. En ambos casos se demuestra formalmente su operación, además, las dos formulaciones se validan numéricamente a través de ejemplos muy sencillos. Finalmente, se demuestra la aplicabilidad de las propuestas con un conjunto limitado de imágenes de objetos de diferente complejidad y deferente número de huecos.

El resto de este artículo se organiza como sigue. En la sección 2 se presenta un conjunto de definiciones útiles para el seguimiento de la lectura del documento. La

sección 3 se enfoca a describir las formulaciones propuestas junto con su demostración formal, así como una validación con ejemplos sencillos. La sección 4 se orienta a presentar los resultados experimentales junto con una discusión de estos resultados. Finalmente, en la sección 5 se ofrecen las conclusiones y los trabajos a futuro a desarrollar relacionados con la presente investigación.

2. Fundamentos

En esta sección se presenta un conjunto de definiciones que permitirán seguir la lectura del presente trabajo. Cuando sea necesario se darán ejemplos para dar completitud a la explicación.

Definición 1. Sean p_1 y p_2 dos pixeles de una imagen binaria $I(x, y)$ compuesta de celdas cuadradas. Si p_1 y p_2 comparten un lado, se dice que ambos pixeles están conectados por un lado, ver la Figura 1(a)); de otra forma, si p_1 y p_2 están conectados por una de sus esquinas, entonces se dice estar conectados por una esquina, ver la Figura 1(b)); de otra forma, p_1 y p_2 no están conectados (Figura 1(c)).

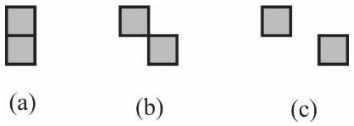


Fig. 1. (a) Dos pixeles conectados por un lado. (b) Dos pixeles conectados por una esquina. (c) Dos pixeles no conectados.

Definición 2. Un objeto o forma binaria S_n compuesta de n pixeles en una imagen binaria $I(x, y)$ es cualquier región cuyos elementos están conectados por sus lados, esquinas o una combinación de ambos. Si todos los pixeles de la forma S_n están conectados por sus lados, entonces S_n se dice ser conectada por sus lados o 4-*connectada*, por otro lado, si todos los pixeles S_n están conectados por sus lados o por sus esquinas entonces S_n se dice estar 8-*connectada*. En resumen, si todos los pixeles de S_n están conectados solo por sus esquinas, entonces S_n se dice estar estrictamente 8-*connectada*.

Ejemplo 1. La Figura 2 muestra tres formas con 9, 9, y 8 pixeles, respectivamente. La primera forma está 4-conectada, la segunda está 8-conectada, mientras que la tercera está estrictamente 8-conectada.

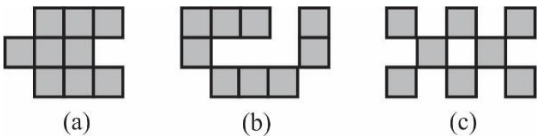


Fig. 2. (a) Una forma 4-conectada. (b) Una forma 8-conectada. (c) Una forma estrictamente 8-conectada.

Definición 3. Un arreglo de pixeles binarios de tamaño 2×2 con valores 0 o 1 se llama *bit-quad*. De acuerdo a [5], los 16 bit-quads mostrados en la Figura 3 se pueden definir.

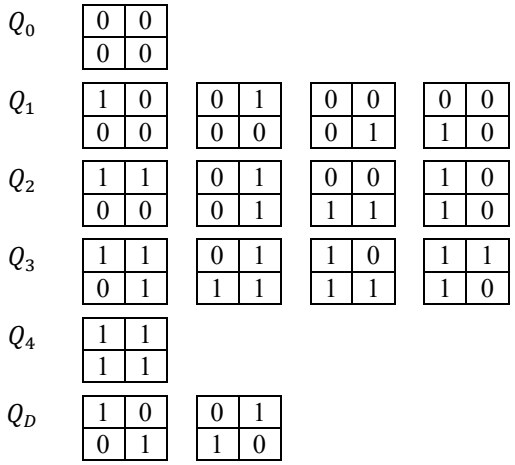


Fig. 3. 16 Bit-quads según [6].

De ahora en adelante:

$$Q_{00} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

$$Q_{10} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, Q_{11} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, Q_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, Q_{13} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{20} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}, Q_{21} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, Q_{22} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, Q_{23} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{30} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, Q_{31} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, Q_{32} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, Q_{33} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}.$$

$$Q_{40} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

$$Q_{\mathrm{D0}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q_{\mathrm{D1}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

La notación $\#Q_{xy}$, denota el número de veces que el bit-quad Q_{xy} es encontrado en una imagen $I(x, y)$ cuando dicho Q_{xy} es desplazado posición a posición a lo largo de la imagen, de arriba abajo y de izquierda a derecha.

Ejemplo 2. La Fig. 4 muestra un imagen de 6×6 con un objeto 4-conectado. Suponga el caso del bit-quad $Q_{10} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Conforme este bit-quad es movido a lo largo de la imagen, según se describió anteriormente, se puede verificar que Q_{10} es encontrado tres veces, luego $\#Q_{xy} = 3$. Las posiciones, en la Figura 4, donde Q_{10} fue encontrado aparecen en gris.

Antes de introducir nuestras formulaciones para calcular el número de hoyos de un objeto 4(8)-conectado, consideremos el ejemplo mostrado en la Figura 5 para entender la operación de la Ecuación (3). Para la forma mostrada en esta figura, el lector puede fácilmente verificar que $np = 25, nc = 29, nt = 2,$ y $nd = 0,$ luego $h = 1 - (25 - 29 + 2 - 0) = 3$ huecos según los esperado.

0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	**1**	0
0	1	**1**	0	0	0
0	**1**	0	0	0	0
0	0	0	0	0	0

Fig. 4. Una imagen binaria con un objeto 4-conectado.

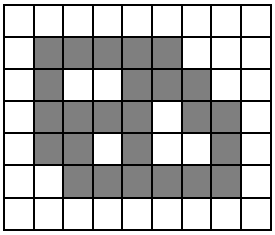


Fig. 5. Una imagen binaria con un objeto 4-conectado compuesto de 25 pixeles y 3 huecos.

3. Las nuevas formulaciones

En esta sección se introducen las formulaciones que permiten el cálculo del número de hoyos de un objeto binario mediante una combinación mínima de bit-quads. En la sección 3.1, se da una formulación para el caso de formas 4-conectadas, mientras que en la sección 3.2 se procede igual, pero para el caso 8-conectado.

3.1. Formulación para calcular el número de huecos para formas 4-conectadas

La siguiente proposición permite determinar el número de hoyos de un objeto 4-conectado:

Proposición 1. Sea S_n cualquier forma 2-D 4-conectada según la Definición 2. El número de hoyos h de S_n según la Ecuación (1) puede ser calculada como:

$$h = 1 - (\#Q_{10} - \#Q_{33}). \tag{4}$$

Prueba. Por inducción matemática en términos del número de pixeles n de S_n, para el caso base de un objeto S_1 consistente de un solo pixel, $h = 1 - \#Q_{11} = 0$, valor que satisface la Ecuación (4).

Paso de inducción: Asúmase que la Ecuación (4) se satisfice para una forma S_k consistente de $n = k$ pixeles 4-conectados, con $k \geq 1$, entero. Luego, por la prueba de inducción, el número de hoyos de S_k viene dado por h_k, donde:

$$h_k = 1 - \#Q_{10} + \#Q_{33}. \tag{5}$$

Ahora, necesitamos demostrar que la Ecuación (4) es aún válida para S_{k+1}, donde S_{k+1} se obtiene al adicionar un pixel 4-conectado a S_k. Denótese como $\#Q'_{10}$ y $\#Q'_{33}$ el número total de bit-quads de S_{k+1} aumentados o disminuidos con respecto a S_k, entonces, se va a demostrar que:

$$h_{k+1} = 1 - \left(\#Q_{10} + \#Q'_{10}\right) + \left(\#Q_{33} + \#Q'_{33}\right) = h_k - \#Q'_{10} + \#Q'_{33} \tag{6}$$

es el número total de hoyos de S_{k+1}.

Se tienen tres casos diferentes, dependiendo de la forma en que un pixel adicional 4-conectado afecta el número de hoyos de S_k.

Caso 1. Supóngase que el pixel adicional 4-conectado, no cambia el número de hoyos h_k de S_k. Sin embargo, podría o no podría afectar el número de bit-quads: Si S_k y S_{k+1} tienen el mismo número de Q_{10} y Q_{33}, entonces $\#Q'_{10} = \#Q'_{33} = 0$ y

$$h_{k+1} = h_k = h_k - \#Q_{10} + \#Q_{33}. \tag{7}$$

Referirse a la Figura 6(a)); si S_k y S_{k+1} tienen diferentes conjuntos de bit-quads, entonces, de la Figura 6(b), se puede observar que $\#Q'_{10}$ y $\#Q'_{33}$ podrían aún ser iguales a cero, y (6) sería de nuevo verdadera; o $\#Q'_{10}$ y $\#Q'_{33}$ pueden aumentar de manera simultánea sus valores por 1, dejando (6) como verdadera, Figura 6(c).

Caso 2. El número de hoyos decrece en S_{k+1}. Para una forma 4-conectada, este puede ser solo el cado de un hoyo de tamaño de un pixel de S_k, así $\#Q'_{10} = 0$ y $\#Q'_{33} = -1$. Entonces

$$h_{k+1} = h_k - \#Q'_{10} + \#Q'_{33} = h_k - 1. \tag{8}$$

Luego, (6) es verdadera. Ver la Figura 6(d).

Caso 3. El número de hoyos aumenta en S_{k+1}. Para el caso de una forma 4-conectada, esto implica que $\#Q'_{33} = 1$ y $\#Q'_{10} = 0$, implicando que (6) es verdadera, Figura 6(e); o $\#Q'_{33} = 0$ y $\#Q'_{10} = -1$, y de (6):

$$h_{k+1} = h_k - \#Q'_{10} + \#Q'_{33} = h_k - (-1) + 0 = h_k + 1. \tag{9}$$

Indicando que la expresión (6) de nuevo es verdadera. Ver la Figura 6(f).

En resumen, si se asume que la ecuación (5) es válida para la forma S_k 4-conectada, entonces la hipótesis inductiva implicará que es válida también para S_{k+1}. Esto es, (4) es válida para toda forma 4-conectada, S_n, $n \in \{1,2,\dots\}$. ■

Para validar numéricamente la Ecuación (4), úsese la imagen mostrada en la Figura 7(a). Como se puede apreciar el objeto se compone de 25 pixeles 4-conectados y 3 huecos. Para este objeto, según la Proposición 1, se deberían obtener 3 hoyos. Procediendo: $h = 1 - (\#Q_{10} - \#Q_{33}) = 1 - (1 - 3) = 3$ según lo deseado.

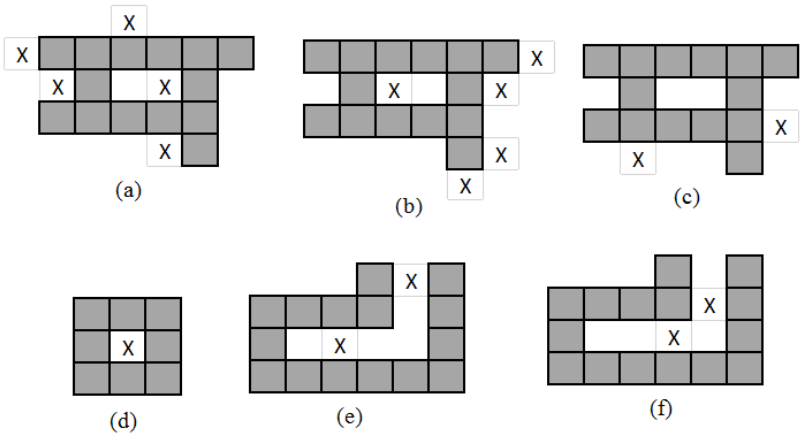


Fig. 6. Formas 4-conectadas S_k en gris. Imágenes mostrando diferentes opciones para generar S_{k+1}, al adicionar un pixel X. (a), (b) y (c) al no modificar el número de huecos. (d), (e) and (f) al modificar el número de huecos.

Las posiciones donde los bit-quads Q_{10} y Q_{33} fueron encontrados en la Figura 7(a) se muestran, respectivamente, con una "×" y un círculo "○".

Comentario 1. Solo dos comparaciones, sobre cada pixel de la imagen, se requieren para determinar de manera exacta el número de hoyos de una forma binaria 4-conectada.

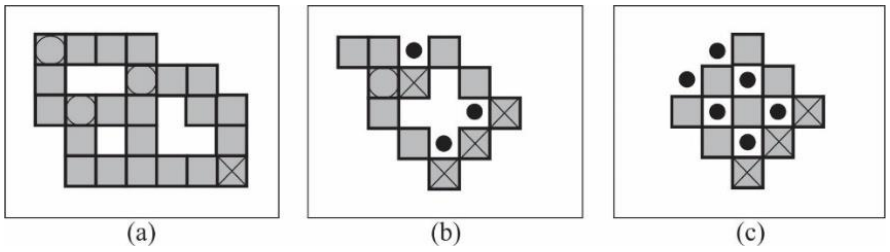


Fig. 7. (a) Forma 4-conectada para probar la ecuación (4). (b), y (c) formas para probar la ecuación (11).

3.2. Formulación para calcular el número de hoyos de una forma 8-conectado

Los objetos 2-D 8-conectados se caracterizan por el hecho de que sus pixeles se encuentran conectados ya sea por sus lados y/o por sus esquinas como se muestra en las Figuras 2(b) y 2(c). La siguiente proposición permite calcular, de manera exacta, el número de hoyos de cualquier objeto 2-D 8-conectado.

Proposición 2. Sea S_n cualquier objeto binario 2-D. El número de hoyos h de una forma S_n según la Ecuación (1) puede ser calculado como:

$$h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) \quad (10)$$

Prueba. Igual que para el caso de la Proposición 1 por inducción matemática. ■

Comentario 2. Para el cálculo del número de hoyos de un objeto 8-conectado, se utilizan los mismos bit-quads que para el caso 4-conecado junto con el bit-quad: Q_{D1}. Luego, entonces, tres comparaciones sobre cada pixel de la imagen se requieren para determinar de manera exacta el número de hoyos de un objeto binario 8-conectado.

Para validar numéricamente validar la Ecuación (10), considérense los objetos mostrados en las Figuras 7(b) y 7(c), respectivamente. Como se puede ver, la primera forma tiene un hoyo en forma de cruz, mientras que le segundo objeto contiene cuatro huecos del tamaño de un pixel.

De acuerdo a la Proposición 2, para el primer objeto $h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) = 1 - (4 - 1 - 3) = 1$, según lo deseado. De la misma manera, para el segundo objeto $h = 1 - (\#Q_{10} - \#Q_{33} - \#Q_{D1}) = 1 - (3 - 0 - 6) = 4$, según lo esperado.

Las posiciones donde los bit-quads: Q_{10}, Q_{33} y Q_{D1} son encontrados en las Figuraas 7(b) and 7(c) aparecen marcados, respectivamente, con una "×", un círculo blanco "○", y un círculo negro"●".

El lector puede fácilmente verificar que el mismo número de hoyos pueden ser encontrados mediante la Ecuación (3), la diferencia reside en las operaciones locales utilizadas.

Comentario 3. Si se omite el uso del bit-quad Q_{D1} en el cálculo del número de hoyos del objeto, se obtiene un cálculo incorrecto. Este hecho fue encontrado después de un análisis con varios objetos con un número de píxeles pequeño.

4. Resultados y discusión

En esa sección se presentan varios experimentos para demostrar la funcionalidad de las formulaciones propuestas. Durante el primer experimento se demuestra la validez de la Ecuación (4) al ser aplicada a objetos 4-conectados. Durante el segundo experimento se muestra la validez de la misma ecuación cuando las formas son sujetas a transformaciones imagen como traslaciones, rotaciones y cambios de escala.

4.1. Experimento 1

Es este experimento se demuestra la validez de la Ecuación (4). Para esto, se usa el conjunto de objetos 4-conectados mostrado en la Figura 8. Se pudiera usar cualquier objeto con cualquier número de hoyos, pero mientras dicho objeto aparezca 4-conectado, la Proposición 1 garantiza el cálculo exacto del número de huecos. Por supuesto, se pudiera hacer lo mismo con la Ecuación (10) para el caso de objetos 8-conectados.

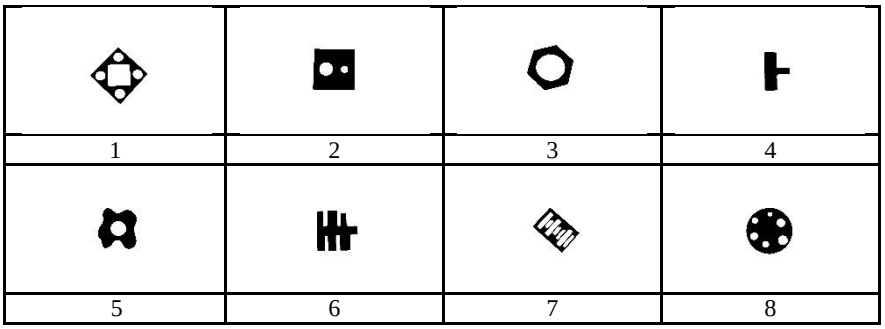


Fig. 8. Las ocho formas 4-conectadas usadas para probar experimentalmente la validez de la Ecuación (4).

Tabla 1. Valización numérica de las Ecuaciones (4) y (3).

	Forma número							
	1	2	3	4	5	6	7	8
Valores verdaderos de h para las ocho imágenes de prueba (caso 4-conectado)	5	2	1	0	1	0	7	6
Valores calculados de h a través de (4)	5	2	1	0	1	0	7	6
Valores calculados de h a través de (3)	5	2	1	0	1	0	7	6

La Tabla 1 resume los resultados. La segunda fila de esta tabla muestra los valores verdaderos de h para las ocho 8 imágenes. La tercera fila muestra los valores calculados mediante la Ecuación (4). Según lo estipulado por la Proposición 1, en todos los ocho casos, los valores calculados coinciden con los verdaderos. Por comparación, en la cuarta fila se muestran los valores obtenidos mediante la Ecuación (3).

Todos los experimentos fueron corridos en una computadora con un Intel Xeon (R) CPU 2.00 GHZ × 12 núcleos y 16 GB de RAM. Como sistema operativo se usó Linux Ubuntu 14.04.3 LTS (x86_64) con un núcleo genérico Linux 3.19.0-43.

4.2. Experimento 2

En este experimento se demuestra de nuevo la validez de la Ecuación (4) pero cuando el objeto bajo consideración aparece en la imagen bajo el efecto de una combinación de transformaciones imagen (traslaciones, rotaciones y cambios de escala, etcétera). Para esto, se usa el séptimo objeto de la Figura 8. La Figura 9 muestra ocho posibles versiones transformadas de este objeto.

Debido al umbralado imperfecto, algunas veces hoyos parásitos muy pequeños pueden aparecer en la imagen resultante. Estos hoyos parásitos pueden afectar el cálculo deseado del número de huecos de la forma. Pueden ser removidos de manera eficiente por la aplicación de un filtro morfológico (una cerradura) según se explica, por ejemplo, en [8]. En el caso que nos atañe, un elemento de estructura como el mostrado en la Figura 10 funcionó en muchos de los casos.

La Tabla 2 ilustra los resultados. Como se puede apreciar, debido a que las ocho formas aparecen 4-conectados, según lo planteado en la Proposición 1, el número deseado hoyos h fue correctamente obtenido.

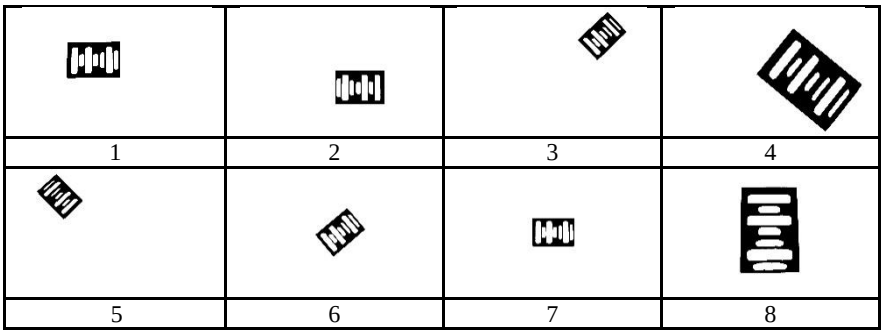


Fig. 9. Ocho transformaciones imagen del objeto 7 de la Figura 8 para probar la validez de la Ecuación (4).

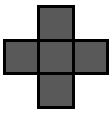


Fig. 10. Elemento de estructura en forma de cruz utilizado para eliminar los huecos parásitos debidos a un umbralado defectuoso.

Tabla 2. Validación numérica de la Ecuación (4) sobre las ocho versiones transformadas del objeto 7.

	Transformación número							
	1	2	3	4	5	6	7	8
Valores verdaderos de h para las ocho imágenes de prueba (caso 4-conectado)	7	7	7	7	7	7	7	7
Valores calculados de h mediante la Ecuación (4)	7	7	7	7	7	7	7	7

Es claro que si el hoyo ruido no es removido por la aplicación del filtro morfológico, se obtendrá una valor no deseado de h. Para mejor explicar esto, supóngase la imagen umbralada de tamaño 8×8 mostrada en Figura 11. En esta imagen, el hoyo parásito aparece marcado con una "**h**". Al aplicar la Ecuación (4) a este objeto ruidoso, se puede verificar que $h = 1 - (\#Q_{10} - \#Q_{33}) = 1 - (5 - 6) = 2$, lo cual es correctamente calculado pero no el valor deseado de 1.

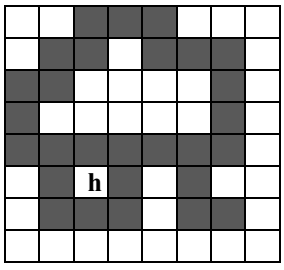


Fig. 11. Imagen binaria para mostrar la no deseada operación de la Ecuación (4) en la presencia de ruido.

Comentario 4. En cuanto al número de operaciones, se puede ver que las dos formulaciones propuestas utilizan un número mejor de operaciones. Quitando el número de operaciones aritméticas usadas por las tres ecuaciones, se observa que la Ecuación (4) usa dos comparaciones, la Ecuación 10 usa tres, mientras que la Ecuación (3) utiliza cuatro. Las dos formulaciones presentadas en este artículo son ligeramente más económicas que la propuesta en [7].

5. Conclusiones e investigación actual

Se han presentado dos formulaciones para calcular de manera eficiente el número de hoyos de una forma binaria 2-D, una útil para el caso 4-conectado y la otra útil para caso de objetos 8-conectados.

Ambas formulaciones emplean un número reducido de comparaciones a diferencia de otras propuestas que se valen de más comparaciones, por ejemplo, la utilizada por el entorno MATLAB que usa 10 bit-quads. En particular, la formulación para obtener el número de hoyos de un objeto 4-conectado requiere de dos bit-quads. La formulación para el caso 8-conectado requiere un bit-quad más.

Se valida numéricamente la formulación para el caso 4-conectado con ejemplos de objetos realistas de diferente complejidad y número de huecos.

Actualmente, se está trabajando en la extensión de la formulación propuesta para el caso de objetos tridimensionales compuestos por voxeles.

Agradecimientos. Los autores agradecen al Centro de Investigación en Computación del Instituto Politécnico Nacional y al CONACYT (proyectos SIP 20190007, y CONACYT Proyecto 65 en el marco de la convocatoria: Fronteras de la Ciencia 2015 por el apoyo económico brindado para llevar a cabo la presente investigación.

Referencias

1. Yang, H. S., Sengupta, S.: Intelligent shape recognition for complex industrial tasks. IEEE Control Syst. Mag., pp. 23–29 (1988)
2. Snidaro, L., Foresti, G. L.: Real-time thresholding with Euler numbers. Pattern Recognition Letters, vol. 24, pp. 1533–1544 (2003)
3. Lin, X., Ji, J., Gu, G.: The Euler number study of image and its application. In: Proc. 2nd IEEE Conference on Industrial Electronics and Applications, pp. 910–912 (2007)
4. Al Faqheri, W., Mashohor, S.: A real-time Malaysian automatic license plate recognition (M-ALPR) using hybrid fuzzy. International Journal of Computer Science and Network Security 9(2), 333–340 (2009)
5. Climent, J., Oliveira, L.S.: A new algorithm for number of holes attribute filtering of grey-level images. Pattern Recognition Letters 53, pp. 24–30 (2015)
6. Gray, S.B.: Local properties of binary images in two dimensions. IEEE Trans. Comput. 20(5), 551–561 (1971)
7. Sossa, H.: On the Number of Holes of a 2-D Binary Object. In: IAPR International Conference on Machine Vision Applications (MVA 2015), May 18-22, 2015, Tokyo, JAPAN, pp. 299–302 (2015)
8. Gonzalez, R., Woods, R.E.: Digital Image Processing. Fourth Edition. Pearson (2017)

ISSN 1870-4069

Un algoritmo genético con fertilización in-vitro aplicado a la sintonización de un regulador difuso

J. Fernando García-Mejía[1,2], Luis Antonio Gonzalez[2], Carlos E. Torres-Reyes[1], J. Arturo Pérez Martínez[1], J. Antonio Garcia-Mejía[1]

[1] Universidad Autónoma del Estado de México, Centro Universitario UAEM Atlacomulco, Atlacomulco, Estado de México, México
[2] Universidad Centro Panamericano de Estudios Superiores, Zitácuaro, Michoacán, México
{fgarciam, cetorresr, japerezm, jagarciame}@uaemex.mx, lage49@yahoo.es

Resumen. La teoría de la evolución de las especies formulada por Charles Darwin y las leyes de la herencia descritas por Gregory Mendel sirvieron de andamiaje teórico para que John Holland desarrollara uno de los primeros algoritmos evolutivos para solución de problemas de optimización: el Algoritmo Genético, el cual ha servido como base para la creación de nuevas estrategias evolutivas que tienen la finalidad de reducir tiempos de convergencia y estabilidad, es decir la repetitividad de los resultados con respecto al número de ejecuciones. En este trabajo se propone un algoritmo denominado Algoritmo Genético con Fertilización In Vitro, que tiene como base el Algoritmo Genético simple y los principios de reproducción asistida descritos por Robert Edwards, como medio de optimización de un problema mono objetivo de 19 variables correspondiente a la sintonización de funciones de membresía de un regulador difuso del tipo Mamdani aplicado a una celda termoeléctrica. Los resultados obtenidos son analizados por medio de tests estadísticos, con lo cual se confirma que el algoritmo propuesto tiene mejores resultados y es diferente a un Algoritmo Genético.

Palabras clave: algoritmo genético, sintonización de funciones de membresía, algoritmo genético con fertilización in vitro.

In Vitro Fertilization Genetic Algorithm Applied to the Tuning of a Fuzzy Regulator

Abstract. The theory of species evolution formulated by Charles Darwin and the laws of inheritance described by Gregory Mendel served as theoretical scaffolding for John Holland to develop one of the first evolutionary algorithms for solving problems of optimization: the genetic algorithm, which has served as a basis for the creation of new evolutionary strategies that have the purpose of reducing times of convergence and stability, namely the repeatability of the results with respect to the number of executions. In this paper an algorithm called In Vitro Fertilization Genetic Algorithm is proposed, which is based on the simple Genetic Algorithm and the principles of assisted reproduction described by Robert Edwards, as a means of optimization of a mono objective problem of 19 variables corresponding to the tuning of membership functions of a Mamdani

type fuzzy regulator applied to a thermoelectric cell. The results obtained are analyzed by means of statistical tests, which confirms that the proposed algorithm has better results and is different from a Genetic Algorithm.

Keywords: genetic algorithm, tuning of membership functions, in vitro fertilization genetic algorithm.

1. Introducción

Los algoritmos evolutivos, (Evolutionary Algorithms, o EA, por sus siglas en inglés) se pueden definir como una colección de métodos heurísticos, es decir reglas obtenidas de un conocimiento previo de cierto dominio de un problema [1], que tienen pasos definidos inspirados en la mayoría de los casos por procesos biológicos como la reproducción y mutación de los seres vivos, la recombinación y la selección biológica [2] los cuales fueron propuestos por Charles Darwin en el libro "El origen de las especies" [3].

En síntesis, un EA son pasos definidos con la capacidad de evolucionar, los cuales se aplican en la optimización. Estos tienen tres características principales

- Tienen una población. Un EA mantiene un conjunto de soluciones, llamado población, representada por números binarios, en trabajos recientes suele usarse la codificación por números reales, para optimizar o aprender el problema en forma paralela. Este es un principio básico del proceso evolutivo.
- Tienen una función objetivo. Esta es una expresión matemática que suele modelar el problema de optimización. El número de sus variables determina el cómo se genera la población, cuyos elementos son evaluados en esta.
- Operadores de recombinación. Es un conjunto de operaciones aplicadas a los elementos de la población, esto permite producir variaciones en su código, lo cual repercute en la exploración del espacio de soluciones.

Los EA funcionan a partir de una colección de posibles individuos que pueden ser representados por medio del vector $P(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ que se denominan población, cada elemento $P(t)$ representa una posible solución factible a un problema de optimización numérica o combinatoria.

Esta población se somete a operaciones de recombinación y trasformación, posteriormente a un proceso de selección, que se realiza de manera iterativa [4], después de un cierto número de iteraciones se espera que el mejor individuo (aquel que genere el valor más alto de la función de aptitud) converja a un determinado punto que será considerado la solución a un problema de optimización [5].

Los EA se caracterizan por partir de un conjunto de soluciones iniciales que se van transformando por la acción de un conjunto de operadores, los cuales se encargan de "refinar" a las soluciones hasta que estas converjan a un determinado punto denominado solución. La estructura de un algoritmo evolutivo se muestra en la Fig 1, la cual permite observar el carácter iterativo del algoritmo [6].

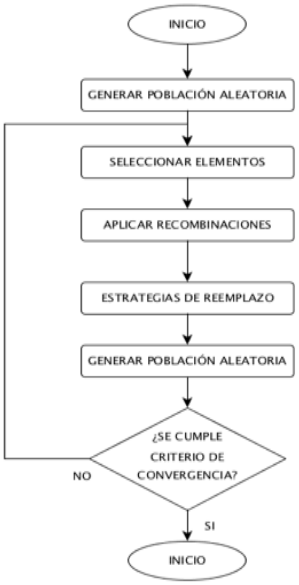


Fig. 1. Esquema genérico de un algoritmo evolutivo basado en los principios biológicos descritos por Darwin y Mendel.

Los algoritmos evolutivos son un área de las tecnologías de la información que continúa en desarrollo, esto es justificable por el teorema "No Free Lunch" desarrollado por David Wolpert y William Macready [7], el cual determina en términos generales que no existe un algoritmo universal de solución de problemas de optimización. Es decir, una técnica que fue exitosa en un entorno no necesariamente funcionará de igual forma en otra situación [8].

1.1. Algoritmo genético

John Holland de la Universidad de Michigan en la década de los 60's desarrollo el algoritmo evolutivo por excelencia, el Algoritmo Genético [9] con inspiración en los trabajos sobre la evolución de las especies desarrollados por Charles Darwin [10] (Wang, 2009) donde se estipulan las condiciones de supervivencia del individuo más apto en un entorno. Otro referente teórico en el desarrollo de los GA son las leyes de la herencia desarrolladas por Gregory Mendel [11].

Las ventajas del uso de Algoritmos Genéticos sobre otras técnicas evolutivas son las siguientes [12, 13, 14].

- Encuentran aproximaciones a problemas sin soluciones prácticas.
- Ideales a problemas complejos de índole adaptativo.
- Habilidad de manipulación para muchos parámetros simultáneamente, dado que los problemas reales no son definibles en términos de una sola variable.

Para su implementación no es necesario poseer un conocimiento a fondo de los problemas que se intentan resolver, solo se realizan cambios aleatorios en las posibles soluciones, evaluándose con la función objetivo para verificar si estos llegan a una mejora.

Algunas limitaciones y desventajas de los algoritmos genéticos con respecto a otras técnicas evolutivas son [12, 13, 14].

- Una incorrecta selección de la función objetivo repercute en la capacidad del algoritmo para encontrar la solución correcta del problema.
- Se debe delimitar el tamaño de la población, ritmo de la mutación y cruzamiento ya que, si la población es pequeña el algoritmo puede no explorar todo el espacio de soluciones.
- Que el algoritmo puede converger de forma prematura, si la solución óptima aparece demasiado pronto, mermando las soluciones y llegando a un óptimo local en lugar de explorar todo el espacio y llegar al óptimo global.
- No se deben de utilizar en problemas de solución analítica, ya que los métodos analíticos tradicionales consumen mucho menos tiempo y potencia de la computadora que los algoritmos genéticos.

La Fig. 2 muestra el diagrama de flujo de un Algoritmo Genético simple o Canónico, para la solución de un problema de optimización mono objetivo:

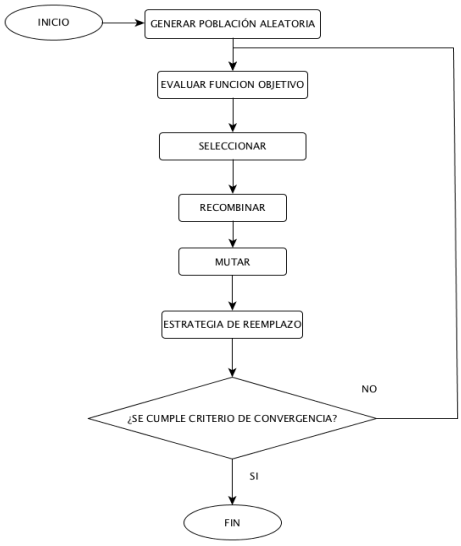


Fig. 2. Diagrama de flujo de un Algoritmo Genético Canónico propuesto por John Holland.

1.2. Algoritmo genético con fertilización in vitro

Como se describió en las secciones anteriores un Algoritmo Genético simple es una técnica útil en la solución de problemas de optimización, tanto numéricos como combinatorios, es el algoritmo evolutivo por antonomasia y por lo tanto objeto de estudio continuo en las ciencias computacionales. Un ejemplo de esto es el artículo *In Vitro Fertilization Genetic Algorithm Applied To Multidimensional 0-1 Knapsack Problem*, desarrollado por Camilo Jr., profesor de la Universidad Federal de Gois [23].

Camilo Jr. propone una variante del GA denominándolo Algoritmo Genético con Fertilización In Vitro IVF/GA (In Vitro Fertilization/ Genetic Algorithm, IVF/GA, por

sus siglas en inglés) que soluciona los problemas de convergencia lenta de los GA a través de la emulación computacional de un proceso biológico denominado Fertilización In Vitro, una técnica que empezó a ser desarrollada en la década de los 60's del siglo XX por Robert Edwards (Premio Nobel de Medicina en 2010) a partir de la fecundación en un laboratorio con condiciones controladas de óvulos de ratón para posteriormente ser implantados en una hembra de la misma especie.

Las investigaciones realizadas por Edwards, fundamentaron el diseño de un nuevo método heurístico denominado IVF/GA realizado por Camilo Jr. En esta nueva técnica se explora la posibilidad de usar los cromosomas que no fueron seleccionados para recombinación por medio de analogías con el proceso de reproducción asistida de Fertilización in Vitro. De manera original la técnica se basó en la codificación binaria de los cromosomas que forman la población del algoritmo.

La propuesta que se presenta en este trabajo explora un área de oportunidad no explorada de manera original en el trabajo de Camilo, la cual consiste en cambiar las cadenas binarias por vectores $\vec{x} \in \mathbb{R}$. Esto permite solucionar problemas de optimización numérica de manera más eficiente con respecto a codificaciones binarias donde los genes $g \in \mathbb{Z}$. La Fig. 3 muestra el diagrama de flujo de un Algoritmo Genético con Fertilización In Vitro y resalta el módulo In Vitro.

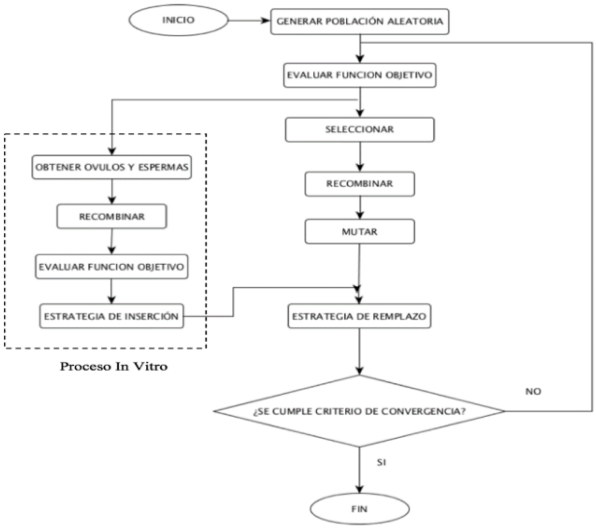


Fig. 3. Algoritmo Genético con Fertilización In Vitro, este módulo se observa resaltado por medio de línea punteada.

2. Caso de prueba

Las Celdas Termoeléctricas (Thermo Electric Cooler, TEC, por sus siglas en inglés), son una alternativa usada en la refrigeración móvil, dado que su funcionamiento no depende de gases Hidrofluorocarbonatados, los cuales han sido prohibidos en algunos países por el daño ecológico que causan. Una TEC (cuyo esquema se muestra en la

Figura 1están constituidas por materiales semiconductores tipo P y N, encapsulados en una superficie cerámica elaborada a partir de óxido de berilio (BeO) mezclado con Alúmina (Al_2O_3), una alternativa a este es el nitrato de aluminio ($Al(NO_3)_3$) [15].

El fenómeno de transferencia de energía en forma de calor (enfriamiento) puede modelarse por medio de una función de transferencia en términos de la variable compleja s tal como lo muestra la expresión 1 [16]:

$$\frac{\tilde{T}_L(s)}{\tilde{I}(s)} = G_I(s) = -6.4061\left(\frac{0.064s + 0.00854}{s^2 + 0.5964s + 0.00855}\right). \quad (1)$$

El comportamiento descrito en la expresión 1 puede modificarse por medio del uso de técnicas de control inteligente como lo muestra [17], el enfriamiento de un diodo de avalancha mediante una TEC, controlada para alcanzar una referencia dada mediante un controlador proporcional difuso de algoritmo de Takagi-Sugeno del tipo Una Entrada, Una Salida (Single Input, Single Output, SISO, por sus siglas en inglés), el cual es simulado en simulink a partir de un modelo matemático linealizado en función de la temperatura y corriente de consumo, consiguiendo la estabilización del mismo a -19.44ºC en 35.91 segundos.

El modelo matemático empleado en [18]y [19] solo funciona para una referencia predeterminada y ciertas condiciones iniciales. Con esta finalidad se continúa la línea de trabajo de los controladores PID, combinándolos con redes neuronales artificiales [18] y se propone el uso de un esquema basado en una red neuronal de 3 capas con funciones de activación de base radial. En este desarrollo se utiliza un cuantificador difuso, el cual tiene como función normalizar las salidas del observador mediante el uso de funciones sigmoidales. Teniendo un tiempo de establecimiento para una referencia de -20 grados en 60 segundos. En [19] se controla una TEC mediante un modelo matemático que hace uso de la potencia de disipación de la misma, utilizando un controlador difuso de algoritmo de Mamdani del tipo SISO, con funciones de pertenencia del tipo gaussiana, con lo cual se busca reducir la velocidad de respuesta en los cambios de temperatura.

En [20] se simula un radiador para un nano satélite en el cual uno de sus componentes es una celda TEC la cual es controlada por medio de un algoritmo de Mamdani del tipo MIMO donde las entradas son las temperaturas tanto ambiente como la de la cara fría y las salidas son salidas de corriente, esto tiene como finalidad controlar el calor generado.

En este trabajo se propone el diseño de un regulador difuso aplicado a una TEC a partir de un Algoritmo Genético con Fertilización In Vitro, cuya población es codificada por medio de números reales. Con lo cual se espera mejoras en los tiempos de establecimiento, sobre impulso y valor final del sistema de control, así como mejoría en la estabilidad y convergencia con respecto a un Algoritmo Genético simple.

3. Metodología

El esquema metodológico empleado para mostrar la utilidad de un Algoritmo Genético con Fertilización In Vitro con respecto a la versión canónica de este como medio de sintonización de funciones de membresía de un controlador difuso cuyo esquema se muestra en la Fig. 4, en la cual como se puede observar se plantea un

sistema de Múltiples Entradas- Una Salida (Multiple Inputs Single Output,MISO por sus siglas en inglés) cuyas funciones de membresía y conjuntos difusos para las variables de entrada son definidos por lo mostrado de acuerdo en la tabla 1, para las variables de salida estos parámetros son definidos en la tabla 2.

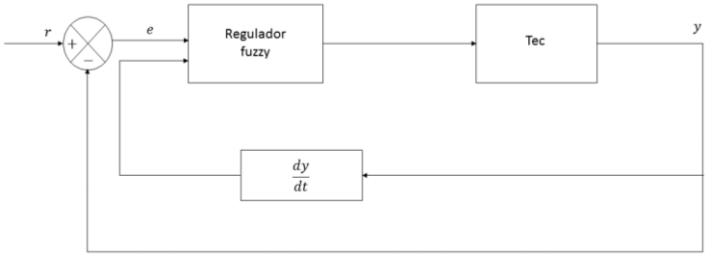


Fig. 4. Diagrama esquemático del regulador difuso propuesto.

Tabla 1. Variable de entrada y sus conjuntos difusos.

Variable	Conjuntos difusos		Función de membresía
er (error)	ne	Error negativo	$\mu_{sigmoide}(er;\ e_1, e_4)$
	c	Error Cero	$\mu_{gaussbellf}(er;\ e_3, e_4, 0)$
	pe	Error Pequeño	$\mu_{sigmoide}(er;\ e_5, e_6)$
ds (incremento de salida)	ns	Incremento negativo	$\mu_{sigmoide}(er;\ e_7, e_8)$
	nc	Sin incremento	$\mu_{gaussbellf}(er;\ e_9, e_{10}, e_{11})$
	np	Incremento Pequeño	$\mu_{sigmoide}(er;\ e_{12}, e_{13})$

Tabla 2. Variable de salida y sus conjuntos difusos (Elaboración propia).

Variable	Conjuntos difusos		Función de membrecía
pe (pendiente de control	pn	Pendiente negativa	$\mu_{sigmoide}(pe;\ e_{14}, e_{15})$
	pc	Pendiente cero	$\mu_{gaussbellf}(er;\ e_{15}, e_{16}, e_{17})$
	pp	Pendiente positiva	$\mu_{sigmoide}(er;\ e_{18}, e_{19})$

De tal forma que la Fig. 5 muestra la FAM (Memoria Asociativa Difusa), la cual relaciona las variables de entrada con la pendiente de control que se constituye como señal de control para la regulación de la TEC.

	ns	nc	np
ne	pn	pn	pn
c	pc	pc	pc
pe	pc	pc	pp

Fig. 5. Reglas de inferencia representadas en una Memoria Asociativa Difusa.

Con base en lo anterior se propone un cromosoma que tiene la forma:

$$\text{cromosoma} = [e_1, e_2, \ldots e_{19}]^T.$$

Con una función objetivo que puede describirse como se muestra en la ecuación 2:

$$f_{obj} = \max\left(\frac{1}{\int \frac{1}{T}(\text{referencia} - \text{salida})^2}\right). \quad (2)$$

Después de proponer el esquema del cromosoma y una función objetivo susceptible a ser maximizada, es necesaria la creación de cromosomas, cuyos genes son creados por medio de una distribución de probabilidad uniforme de acuerdo a la tabla 3:

Tabla 3. Condiciones de generación de cromosomas**.**

Cromosomas de er y ds	Cromosomas de pe
$e_1 = -(0.05 + (\text{random}[0,1]) * 1.9)$	$e_{14} = (-1 + (\text{rando}m[0,1]) * -9)$
$e_2 = ((\text{random}[0,1]) * -20)$	$e_{15} = e_{16}$
$e_3 = -(1 + (\text{random}[0,1]) * (20))$	$e_{16} = -((\text{random}[0,1]) * (1))$
$e_4 = (2 + (\text{random}[0,1]) * (20))$	$e_{17} = (1 + (\text{random}[0,1]) * (9))$
$e_5 = (0.05 + (\text{random}[0,1]) * (1))$	$e_{18} = (1 + (\text{random}[0,1]) * (10))$
$e_6 = (10 + (\text{random}[0,1]) * (60))$	$e_{19} = e_{15}$
$e_7 = -(1 + (\text{random}[0,1]) * (29))$	
$e_8 = -((\text{random}[0,1]) * -20)$	
$e_9 = (-0.1 + (\text{random}[0,1]) * (60))$	
$e_{10} = (1 + (\text{random}[0,1]) * (10))$	
$e_{11} = (-1 + (\text{random}[0,1]) * (2))$	
$e_{12} = (1 + (\text{random}[0,1]) * (9))$	
$e_{13} = (-1 + (\text{random}[0,1]) * (2))$	

Para probar el efecto del algoritmo IVF/GA se propone un conjunto de experimentos los cuales permiten discernir si existen diferencias estadísticamente significativas de la técnica propuesta con respecto a un GA convencional. La descripción de los experimentos se muestra en la tabla 4.

El operador propuesto para este conjunto de experimentos es el operador de cruzamiento por mezcla (Blend Crossover, BLX- α, por sus siglas en inglés) un operador de recombinación que a partir de dos cromosomas progenitores C^{H1}, C^{H2} genera un descendiente de forma aleatoria a la vez de acuerdo a la expresión 3 [21]:

$$C^H = \text{rand}[(h_{\min} - I * \alpha), (h_{\max} + I * \alpha)], \quad (3)$$

donde C^H es el cromosoma descendiente $h_{min} = min(C_i^1, C_i^2)$, $h_{max} = max(C_i^1, C_i^2)$, $I = h_{max} - h_{min}$, $\alpha = rand[0,1]$ con distribución uniforme. Cabe destacar que se pueden producir los descendientes que sean necesarios.

Tabla 4. Experimentos propuestos para el regulador difuso.

No	Técnica	Porcentaje de cruza	Porcentaje de mutación	Número de Cromosomas iniciales	Número de óvulos	Número de espermas	Número de gametos insertados
1	GA	80%	1%	60	-		-
2	GA	60%	1%	60			
3	IVF/GA	80%	1%	60	1	24	15
4	IVF/GA	60%	1%	60	1	24	15

El operador de mutación empleado es la mutación Gaussiana es encargado de modificar a un cromosoma específico C elegido al azar por medio de una distribución de probabilidad gaussina de media 0 y varianza definida como lo muestra la expresión 4 para cada gen g [22]:

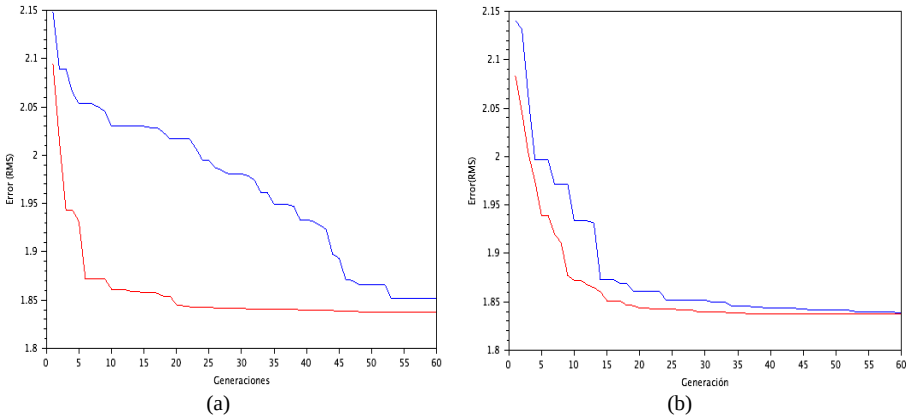
$$\sigma_k = \frac{T - t}{T}\frac{\left(g_k^{max} - g_k^{min}\right)}{3}, \tag{4}$$

donde t es la generación actual, T es el máximo número de generaciones contempladas en el algoritmo de tal forma que el cromosoma mutado puede definirse como lo muestra 5.

$$C' = C + N(0, \sigma_k). \tag{5}$$

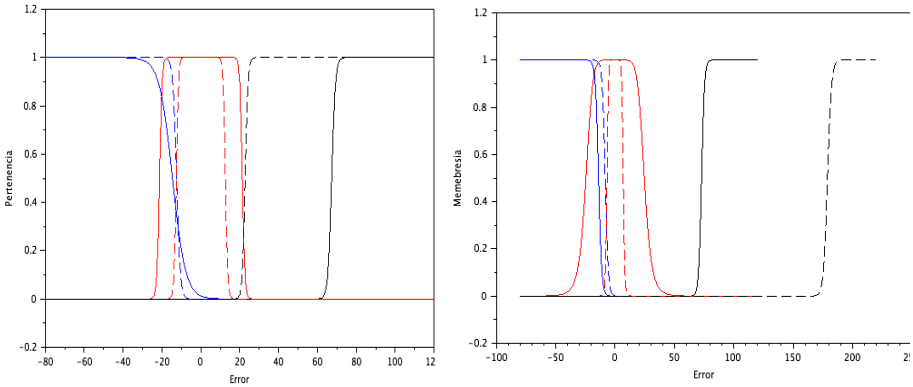
4. Resultados

En esta sección se muestran los resultados del diseño de funciones de membresía por medio de los algoritmos GA y el IVF/GA. El primer aspecto a verificar es la convergencia, la figura 6a muestra este criterio para los experimentos 2 y 4, mientras que la figura 6b los muestra para los experimentos 1 y 3.



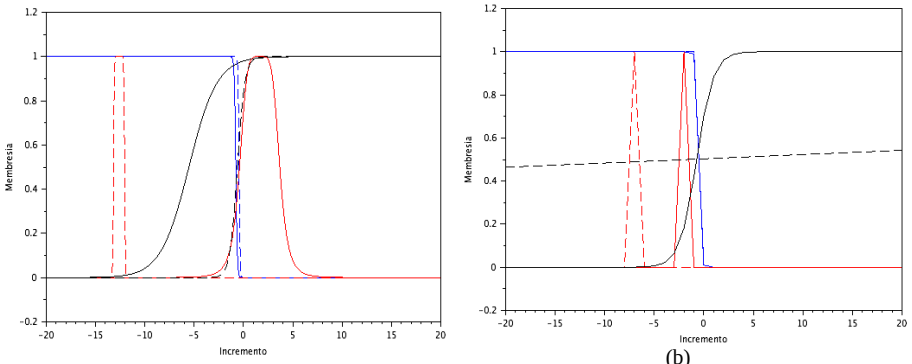
(a) (b)

Fig. 6. Algoritmo Genético con Fertilización In Vitro, este módulo se observa resaltado por medio de línea punteada.



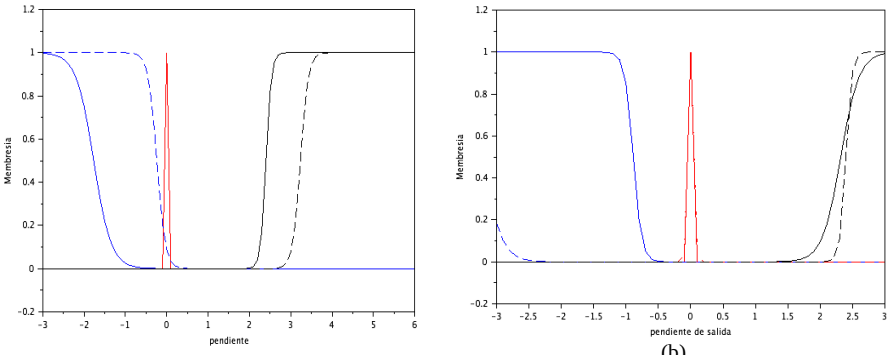
(a) (b)

Fig. 7. Conjuntos difusos variable error GA (azul) e IVF/GA (rojo).



(a) (b)

Fig. 8. Conjuntos difusos variable incremento de salida GA (azul) e IVF/GA (rojo).



(a) (b)

Fig. 9. Conjuntos difusos variable pendiente de salida GA (azul) e IVF/GA (rojo).

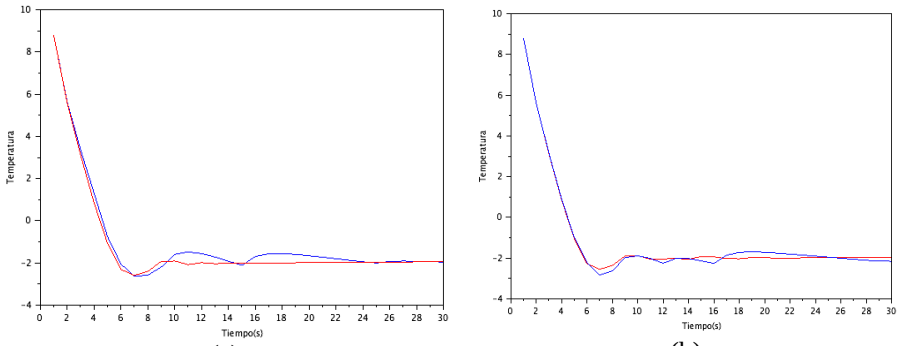


Fig. 10. Desempeño de los controladores propuestos en este trabajo.

Los conjuntos difusos para la variable error son mostrados en la figura 7a. En color azul, rojo y negro se representan las etiquetas lingüísticas error negativo, error cero y error positivo respectivamente. Cabe destacar que estos fueron ajustados por un GA con un porcentaje de cruza del 60% que se muestra en línea continua (experimento 2) y en línea punteada el IVF/GA con el mismo porcentaje (experimento 4). Los resultados de los experimentos 1 y 3 se muestran en la figura 7b.

Los conjuntos difusos para la variable incremento de la salida son mostrados en la Figura 8a. En color azul, rojo y negro se representan las etiquetas lingüísticas incremento negativo, incremento cero e incremento positivo respectivamente. Al igual que en la Figura 7 los conjuntos son ajustados por un GA con un porcentaje de cruza del 60% que se muestra en línea continua y en línea punteada el IVF/GA con el mismo porcentaje. La figura 8b muestra los resultados para los experimentos 1 y 3.

Los conjuntos difusos para la variable incremento de la salida son mostrados en la Figura 9a. En color azul, rojo y negro se representan las etiquetas lingüísticas incremento negativo, incremento cero e incremento positivo respectivamente. Al igual que en la Figura 7 los conjuntos son ajustados por un GA con un porcentaje de cruza del 60% que se muestra en línea continua y en línea punteada el IVF/GA con el mismo porcentaje. La figura 9b muestra los resultados para los experimentos 1 y 3.

La figura 10a muestra la gráfica de salida de temperatura, en el caso de los experimentos 2,4 la 10b indica la salida de los experimentos 2 y 4. Así mismo la tabla 5 muestra los criterios de desempeño que permite valorar el controlador propuesto.

La tabla 5 muestra los índices de desempeño que permite valorar la acción de los controladores diseñados en esta propuesta. En color rojo se indica cual s el controlador que tiene el mejor desempeño.

Tabla 5. Experimentos para el regulador difuso (Elaboración propia).

%Cruza	Técnica empleada	Error RMS	Porcentaje de sobreimpulso	Tiempo de establecimiento	Error estado estable
60%	GA	1.8519856	31.391	25	-0.0431
	IVF/GA	1.8374284	29.386	19	-0.0589
80%	GA	1.8390043	41.3206	26	0.1550
	IVF	1.837053	27.389	22	-0.0323

Al igual que los experimentos anteriores se procedió a realizar un estudio estadístico con la finalidad de determinar la estabilidad y si existen diferencias estadísticamente significativas entre la aplicación de un GA y el IVF/GA propuesto en la presente disertación doctoral.

En la tabla 6 se muestran los resultados estadísticos realizados sobre 30 ejecuciones (por razones de costo computacional), lo cual permite determinar cuál algoritmo cumple con los mejores criterios de estabilidad a través del cálculo de la repetitividad (desviación estándar entre media aritmética).

Además, se aplicó un test de Kolmogorov-Smirnov (con la finalidad de comprobar el comportamiento estocástico de los algoritmos propuestos).

Tabla 6. Estadísticos de los experimentos sobre el regulador difuso

Técnica	Valor Medio	Desviación Estándar	Repetitividad	Significancia asintótica (K-S)
GA 60%	1.868189	0.028763	1.53961	0.360
GA 80%	1.860960	0.034463	1.851932	0.067
IVF/GA 60%	1.853108	0.023630	1.275155	0.041
IVF/GA 80%	1.848904	0.0236300	1.132434	0.047

5. Conclusiones

Como se muestra en la tabla 5, la mejor alternativa se constituye por medio del ajuste de funciones objetivos realizados con el IVF/GA con un porcentaje de cruza de 80%. Los estudios estadísticos descritos en la tabla 6 muestra que el Algoritmo Genético tiene un comportamiento que corresponde a una distribución normal, mientras que la propuesta realizada por medio de un IVF/GA no tiene este comportamiento, dado que los valores de significancia de Kolmogorov-Smirnov es menor a un valor de 0.05. Al aplicar el test de Anova sobre los datos obtenidos con el GA se obtiene el siguiente resultado: No existen diferencias estadísticamente significativas al variar los operadores descritos. Por lo tanto, no es necesario aplicar un test post-hoc.

Cuando no existe el comportamiento con distribución normal el test estadístico a usar es Wilcoxon, el resultado de esta prueba arroja una significancia de 0.645 por lo tanto no existen diferencias estadísticas significativas al variar los porcentajes de cruza de un IVF/GA. Esto hace necesario el uso de una prueba post hoc.

Por lo tanto, se asume que se trata de dos algoritmos distintos dado que sus efectos son externamente diferentes. Pero de manera interna no existen diferencias con respecto a cambios en operadores. Esto, en términos de estadística, expresa que si existen diferencias estadísticamente significativas entre sintonizar un controlador Fuzzy con un GA vs IVF/GA. El Algoritmo Genético con Fertilización In Vitro puede sintonizar las funciones de pertenencia de un controlador difuso tal como lo muestran los resultados de las secciones que forman este documento.

Referencias

1. Ergun, H., Van Hertem, D.: HVDC Grid Planning. En: Van Hertem, D., Gomis-Bellmunt, O., Liang, J.: HVDC Grids: For offshore and Supergrid of the Future, pp. 143–172. New Jersey, Estados Unidos: IEEE Press (2016)
2. Chacón Montes, P.: Algoritmos Evolutivos: algoritmos genéticos y de cuasiespecie. En I. Ramos Salavert, Vida Artificial, pp. 87–111. Castilla, España: Universidad de Castilla-La Mancha (1995)
3. Vasiljevic, D.: Classical and Evolutionary Algorithms in the Optimization of Optical Systems. New York: Springer Science+Business Media (2002)
4. Robusté Antón, F.: Logística del transporte. Barcelona, España: Edicions de la Universitat Politècnica de Catalunya (2005)
5. Chi, Z., Yan , H., Pham, T.: Fuzzy Algorithms: with Applications to Image Processing and Pattern Recognition. New Jersey, Estados Unidos: World Scientific Publishing (1996)
6. Sumathi, S., Surekha, P.: Computational Intelligence Paradigms: Theory and Applications using Matlab. Boca Raton, Florida, Estados Unidos: CRC Press Taylor & Francis
7. Group.Ergun, H., Van Hertem, D.: HVDC Grid Planning. En: D. Van Hertem, O. Gomis-Bellmunt, & J. Liang, HVDC Grids: For offshore and Supergrid of the Future, pp. 143-172. New Jersey, Estados Unidos: IEEE Press (2010)
8. Lloyd, S.: How Smart Is the Universe? En: J. Brockman, Intelligent Thought: Science Versus the Intelligent Design Movement, pp. 179–191. New York, Estados Unidos: Vintage Books (2006)
9. Du, K.-L., Swamy, M.N.: Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature. Suiza: Springer (2016)
10. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithm. Frome, Reino Unido: Luniver Press (2010)
11. Wang, X., Ruan, D., Kerre, E.E.: Mathematics of Fuzziness - Basic Issues. Berlín, Alemania: Springer (2009)
12. Flake, G.W.: The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation. Cambridge, Massachusetts, Estados Unidos: MIT Press (1998)
13. Munakata, T.: Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More. Londres, Reino Unido: Springer (2008)
14. Rajagopal, K.: Operations Research. Nueva Delhi, India: PHI Learning Pvt (2012)
15. Carter, J.N.: Introduction to Using Genetic Algorithms. In: Nikravesh, M., Aminzadeh, F., Zadeh, L. A.: Soft Computing and Intelligent Data Analysis in Oil Exploration, pp. 51–76. Amsterdam, Holanda: Elsevier (2003)
16. Garimella, S.V., Harirchian, T.: Encyclopedia of Thermal Packaging: Thermal Packaging Techniques, vol. 1. (A. Bar-Cohen, Ed.) Singapur, Singapur: World Scientific (2013)
17. Shaojing, S., Qin, Q.: Temperature Control of Thermoelectric Cooler Based on Adaptive NN-PID. In: 2010 International Conference on Electrical and Control Engineering pp. 2245–2248. Wuhan: IEEE. doi:10.11019/iCECE.2010.553 (2010)
18. Sumathi, S., Hamsapriya, T., Surekha, P.: Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab. Berlín, Alemania: Springer (2008)
19. Zang, H.-Q., Li, Q.: The automatic temperature system with fuzzy self-adaptive PID control in semiconductor laser. In: 2009 IEEE International Conference on Automation and Logistics, pp. 1691–1694. Shenyang: IEEE. doi:10.1109/ICAL.2009.5262694 (2009)
20. Aly, A.A., Abo El-Lail, A.S.: Fuzzy Temperature Control of a Thermoelectric Cooler. In: 2006 IEEE International Conference on Industrial Technology, pp. 1580–1585. Mumbai: IEEE. doi:10.1109/ICIT.2006.372492 (2006)

21. Li, Y.-Z., Lee, K.-M., Wang, J.: Analysis and Control of Equivalent Physical Simulator for Nanosatellite Space Radiator. IEEE/ASME Transactions on Mechatronics, 15, pp. 79–87. doi:10.1109/TMECH.2009.2016957 (2010)
22. Bakshi, U.A., Bakshi, V.U.: Automatic Control System. Pune, India: Technical Publications Pune (2008)
23. Bansal, K.: Manual of Intrauterine Insemination (IUI), In Vitro Fertilization (IVF) and Intracytoplasmic Sperm Injection (ICSI). Nueva Delhi, India: Jaypee Brothers Medical Publishers (2011)
24. Camilo, C., Yamanaka, K.: In Vitro Fertilization Genetic Algorithm Applied to Multidimensional 0-1 Knapsack Problem. In: The 2011 World Congress in Computer Science, Computer Engineering and Applied Computing. Las Vegas: World Academy of Sciences (2011)

ISSN 1870-4069

Modelado de un sistema multi-agente para el monitoreo de residuos peligrosos en la industria manufacturera

Carlos A. Soto[1], Juan P. Soto[2], Adrián Vázquez[2], Elvira Rolón[3], Julio C. Rolón[3]

[1] Instituto Tecnológico de Sonora Unidad Guaymas, Guaymas, Sonora, México

[2] Universidad de Sonora, Hermosillo, Sonora, México

[3] Universidad Autónoma de Tamaulipas Centro Universitario Tampico-Madero, Facultad de Ingeniería "Arturo Narro Siller", Tampico, Tamaulipas, México

csoto@iswug.net, {juanpablo.soto,adrian.vazquez}@unison.mx, {erolon,jrolon}@docentes.uat.edu.mx

Resumen. En la actualidad, las tecnologías de la información y la comunicación se han convertido en una parte inevitable de la planificación, diseño y la gestión de procesos en el sector industrial. El uso de sistemas multi-agente para gestionar el manejo de residuos peligrosos facilita a las compañías de manufactura una mejor planificación, monitoreo y recolección de dichos residuos. Los agentes software han demostrado ser una solución eficiente, ya que pueden desarrollar tareas en nombre de los usuarios. Además, los agentes pueden comunicarse entre ellos y utilizar técnicas de razonamiento para la solución de conflictos. Por esta razón, en este trabajo se propone el uso de agentes software para el monitoreo residuos peligrosos en la industria manufacturera.

Palabras clave: sistema multi-agente, residuos peligrosos, control de residuos.

Modeling a Hazardous Waste Monitoring System in the Manufacturing Industry

Abstract. Nowadays, information and communication technology has become a necessary component in the planning, design and management of the different processes in the industry sector. To manufacturing companies, the use of multiagent systems aiming to develop hazardous waste monitoring systems facilitates the planning, monitoring, collection, and management of hazardous waste. Intelligent agents have proven to be an efficient solution, since they can do tasks on behalf of the users. Moreover, these agents can use different intelligent techniques and communicate among themselves. For this reason, this work proposes the use of software agents for hazardous waste monitoring in manufacturing companies. This article will describe the analysis and design of our proposal using the INGENIAS methodology.

Keywords: multi-agent system, hazardous waste, waste control.

1. Introducción

Los residuos son materiales o productos cuyo propietario o poseedor desecha y que se encuentra en estado sólido o semisólido, o es un líquido o gas contenido en recipientes o depósitos, y que puede ser susceptible de ser valorizado o requiere sujetarse a un tratamiento o disposición final de acuerdo a lo dispuesto en la Ley y demás ordenamientos que de ella se deriven [1].

La generación y manejo de residuos no es exclusiva del sector industrial, sin embargo, derivado de la naturaleza de las mismas, son las que generan mayor cantidad de residuos comparadas con otros sectores. Es por esto, que requieren un mayor control con el fin de paliar su impacto en el medio ambiente.

Hoy en día las tecnologías de información son herramientas de gran ayuda para la solución de situaciones que conlleva un impacto ambiental, ya que permiten a los usuarios facilitar la toma de decisiones relacionadas con la logística y el manejo de residuos [2] [3]. Al utilizar las tecnologías de información se simplifica el manejo de los residuos y con ello el cumplimiento de las normatividades aplicables para el sector industrial.

Además, el paradigma de agentes constituye una metáfora de los sistemas con agentes de interacción con propósito, y esta abstracción es similar a la forma en que los humanos pensamos y actuamos respecto a nuestras propias actividades [4]. Además, los agentes pueden mejorar el rendimiento de los individuos, así como el del sistema general en el que se encuentran [5,6].

Los agentes tienen entre otras, las siguientes propiedades útiles [7]:

- Autonomía: los agentes operan sin la intervención directa de humanos u otros y tienen algún tipo de control sobre sus acciones y estados internos.
- Habilidad social: los agentes interactúan con otros agentes (posiblemente humanos) a través de algún tipo de lenguaje de comunicación.
- Proactividad: los agentes toman la iniciativa para lograr sus propios objetivos. Es decir, pueden exhibir comportamientos flexibles, proporcionando conocimiento tanto "reactivo" a solicitud del usuario, o "proactivamente", anticipándose a las necesidades de conocimiento del usuario.

Un sistema basado en agentes o sistema multi-agente es aquel en el que el nivel de abstracción utilizado es el agente. En principio, un sistema basado en agentes podría especificarse en términos de agentes, pero no implementarse con un entorno específico de desarrollo software orientado a agentes. Sin embargo, lo ideal es realizar tanto el diseño como la implementación en términos de agentes.

Un sistema basado en agentes puede estar constituido por un único agente (single-agent system) o por múltiples agentes (SMA) [8]. La mayor diferencia entre estos sistemas se basa en los patrones de comunicación. Un SMA se comunica con la aplicación y el usuario, así como con otros agentes del sistema. Sin embargo, en los sistemas basados en un único agente los canales de comunicación están abiertos solamente entre el agente y el usuario.

Las principales características de un SMA son [9]:

- Proveen la infraestructura adecuada para la comunicación entre agentes.

- Son normalmente diseñados para ser sistemas abiertos sin ningún diseño centralizado.
- Los agentes que componen un SMA son autónomos y pueden ser cooperativos o de naturaleza competitiva.

Es por esto, que en el presente trabajo se propone un modelo multi-agente que permita el monitoreo constante de los residuos generados hasta su disposición final. Además, dicho modelo permitirá alertar y tener al tanto a las personas indicadas para la toma de decisiones en caso de presentarse alguna situación atípica que ponga en riesgo al personal o a la población.

El resto de este documento está organizado de la siguiente manera: la sección dos describe la metodología INGENIAS, la cual es la utilizada para modelar nuestra propuesta; la sección tres describe la arquitectura y el modelo utilizando la metodología descrita en la sección anterior; finalmente, la conclusión y el trabajo futuro son presentados.

2. Metodología INGENIAS

Saber diseñar agentes y encajarlos en un sistema no es suficiente ya que [10]:

- Un sistema debe satisfacer las necesidades del cliente que lo solicitó.
- Hay que tomar decisiones fundamentales como elegir las cualidades que se quieren presentes en los agentes (utilizando las diferentes arquitecturas), decidir qué entidades del sistema van a ser agentes o no y organizarlo todo según la plataforma de desarrollo.

Debido a que este proceso no es trivial, se han estudiado las metodologías más importantes para decidir cuál se adapta mejor a las necesidades del proyecto. Como conclusión consideramos a INGENIAS la más adecuada para este proyecto debido a que es una de las metodologías más actualizadas y completas que existen, propone un lenguaje visual para generar los diferentes modelos o vistas [11] y además estos modelos están soportados por sus correspondientes meta-modelos y esto facilita la comprobación automática de inconsistencias en el diseño.

A continuación, se describe brevemente dicha metodología. INGENIAS [10] es una metodología de ingeniería del software orientada a agentes (ISOA) para el desarrollo de sistemas multi-agente que parte de ideas propuestas en MESSAGE y UML [10].

Por otro lado INGENIAS concibe el desarrollo de un SMA como la representación computacional de un conjunto de modelos y, a su vez, cada modelo trata de mostrar una visión parcial del conjunto del SMA [12].

Así, esta metodología dispone de cinco modelos especificados a través de cinco meta-modelos que giran en torno a dos entidades: el agente y la organización. Estos meta-modelos son:

- Meta-Modelo de Agente: Describe a los propios agentes representando las responsabilidades de cada uno y el comportamiento de los mismos.
- Meta-Modelo de Interacción: Trata la coordinación y comunicación entre agentes.

- Meta-Modelo de Tareas y Objetivos: Relaciona el estado mental del agente con las tareas que debe ejecutar.
- Meta-Modelo de Organización: Define cómo se agrupan los agentes, la funcionalidad del sistema, así como las restricciones que hay que imponer sobre el comportamiento de los agentes.
- Meta-Modelo de Entorno: Define los elementos que se encuentran alrededor del SMA.

En la figura 1 se puede ver la relación existente entre cada meta-modelo y cómo las entidades agente y organización son la parte central de la metodología.

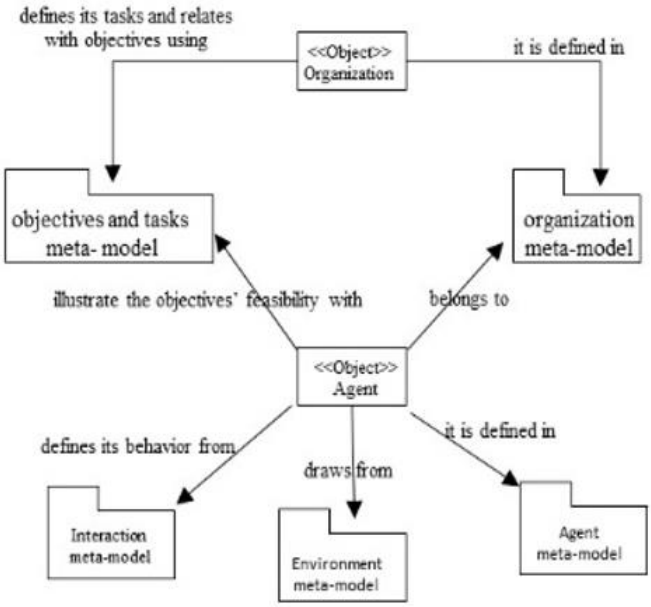


Fig. 1. Relación entre meta-modelos [10].

3. Arquitectura multi-agente

Las principales funcionalidades de un sistema de control y supervisión son: monitorear las variables de operación, procesar dichas variables para generar el control, transmitir las instrucciones de control, permitir las instrucciones de reconfiguración, detectar y diagnosticar condiciones anormales de operación. Estas funcionalidades y tareas pueden ser distribuidas y expresadas a través de una estructura de interacción concurrente y colaborativa.

Con la implementación de agentes inteligentes se puede definir una arquitectura de funcionamiento como se muestra en la figura 2, en la que se define la descripción particular de los elementos que conforman a un agente en concreto y cómo estos elementos interactúan entre sí para lograr su correcto funcionamiento. La arquitectura

determina los mecanismos que utiliza un agente para reaccionar a los estímulos, actuar, comunicarse, etc.

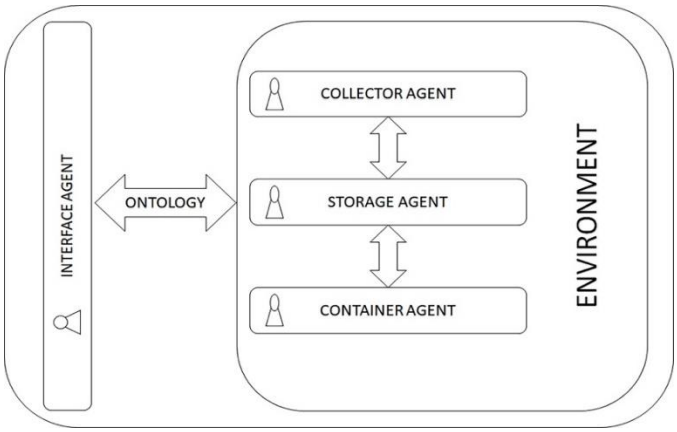


Fig. 2. Arquitectura Multi-agent.

La interacción de agentes (Fig. 2) para el manejo de residuos está definida desde la generación del residuo por el Agente Generador (collector agent), la cual inicia con el depósito en contenedores de los residuos. Una vez que están los residuos en contenedor se dispone al Agente Almacén, el cual gestionará el tiempo para el manejo del recurso definido legalmente para que posterior a esto el Agente Recolector pueda cumplir su actividad hasta la disposición final del residuo.

Para el lograr una comunicación efectiva entre los agentes se utiliza una ontología compartida definida para la comunicación entre ellos.

3.1. Modelo de la arquitectura del sistema

En esta sección se ilustran los meta-modelos de los agentes que describen los roles y las tareas de cada uno de ellos con INGENIAS.

Agente Contenedor. Es el agente a cargo de la gestión de residuos dentro de la industria. Dicha gestión estará definida por las políticas o regulaciones del Agente de Contenedor (Fig. 3). La función principal de este agente es supervisar el nivel de capacidad y tener la información disponible cuando se solicite.

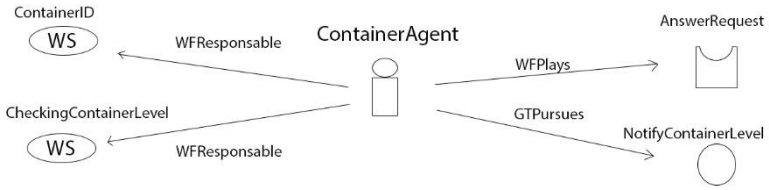


Fig. 3. Agente Contenedor.

La información de monitoreo se utiliza para responder a las solicitudes recibidas (función AnswerRequests). Para responder satisfactoriamente a estas solicitudes, debe asignar un identificador al contenedor (Tarea ContainerID) y monitorearlo (tarea CheckingContainerLevel) con el objetivo de notificar el nivel cuando sea necesario (objetivo NotifyContainerLevel).

Agente Almacén. La responsabilidad de este agente (Fig. 4) es el mayor soporte del sistema, ya que se coordinará con el agente contenedor para la recepción de residuos. El agente recibe el producto para poder realizar sus operaciones internas y cumplir sus objetivos en tiempo de almacenamiento, capacidad de almacenamiento y disposición, la que se realizará al coordinarse con el agente recolector en las actividades necesarias para la entrega y recolección de los contenedores.

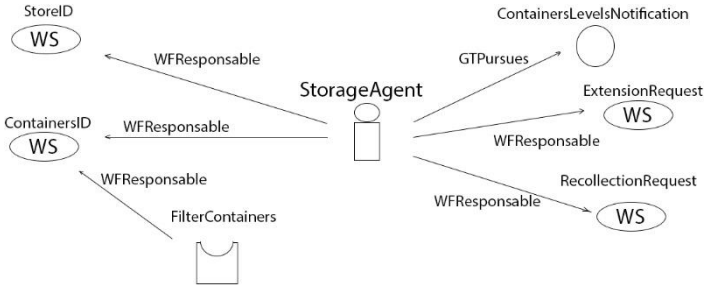


Fig. 4. Agente Almacén.

El objetivo principal del Agente de Almacenamiento es la sincronización con el agente contenedor para llevar a cabo las actividades solicitadas y lograr la recolección de residuos generados (NotifyContainerLevels). Para lograr este objetivo, debe identificar la ubicación de los contenedores en su área (FilteringContainers) a fin de obtener la administración de los diferentes contenedores que se manejan en el almacén (StoreID and ContainerID).

Existen diferentes situaciones en las que se puede solicitar la recolección de cualquiera de los residuos (CollectionRequest) Las razones principales se deben a que el tiempo de gestión de los residuos en el almacenamiento expira, o porque el contenedor está en su capacidad máxima permitida.

Inicialmente, el tiempo que se define para el almacenamiento es de 180 días calendario desde que se coloca el primer residuo dentro del contenedor. Sin embargo, las regulaciones permiten una extensión de tiempo de almacenamiento de residuos.

Las tareas del sector industrial son diversas, y el flujo de residuos puede organizarse dependiendo de la naturaleza de cada uno de ellos. En las empresas de residuos de bajo flujo, se considera una actividad para la recolección de residuos; si hay poca producción de residuos, el agente podrá solicitar una extensión (ExtensionRequest) no mayor de 150 días al comienzo de la recolección.

Agente Recolector. El Agente Recolector se encarga de sincronizar al agente almacén con la logística de la recolección de residuos de la industria, para el manejo correspondiente de los mismos.

El objetivo primario de este agente (Fig. 5) es la recolección de residuos (PickitupResidue). Para esto, el agente debe sincronizarse con el agente de almacenamiento (AcceptRequests). Las tareas que finalizarán la recopilación se definirán mediante la identificación del recolector y la supervisión de almacenamiento (CollectorID and CheckingStore), ya que el Agente recolector debe sincronizar la recopilación cuando sea necesario.

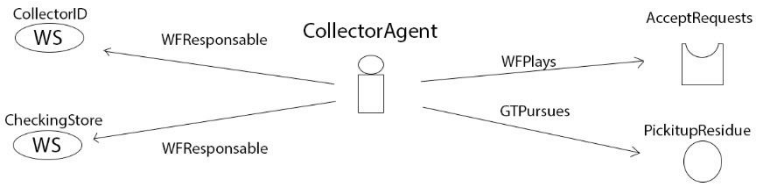


Fig. 5. Agente Recolector.

Agente Interfaz. El agente de interfaz es responsable de organizar la información de los diferentes agentes y presentarla a los usuarios. En otras palabras, permite a los usuarios ver el cumplimiento de los objetivos y metas de los agentes en tiempo real.

La figura 6 muestra el plan para el agente interfaz, cuyo objetivo es recopilar la información (ObtainInfo) de los usuarios del sistema para permitir la interacción, para conseguir este objetivo lleva a cabo las tareas de identificar a los usuarios que interactúan con el sistema (UserID), Gestionar la información capturada (CaptureInfo) y consultar y mostrar las actividades que desea el usuario (RequestInfo). El rol de este agente es filtrar la información (FilterInfo) para tener consistencia entre los resultados obtenidos del agente y las solicitudes de los diferentes usuarios. Para finalizar las solicitudes de los usuarios, el agente debe considerar las tareas de identificación de los usuarios y solicitar la información recopilada por el usuario.

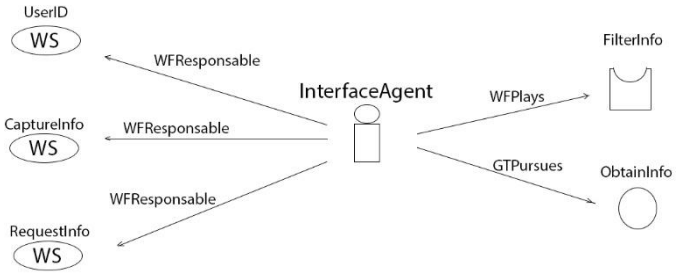


Fig. 6. Agente Interfaz.

Modelo de Organización. La organización que existe en el sistema multi-agente (SMA) está determinada por el objetivo principal, que es la eliminación adecuada de residuos. Para lograr este objetivo, el SMA utiliza la interacción entre los agentes ppertenecientes a la misma organización, comenzando con la reunión de los recursos ambientales en el grupo de agentes de supervisión (MonitoringGroup), que se encuentra en la aplicación del Agente de interfaz, ya que se muestra en la figura 7.

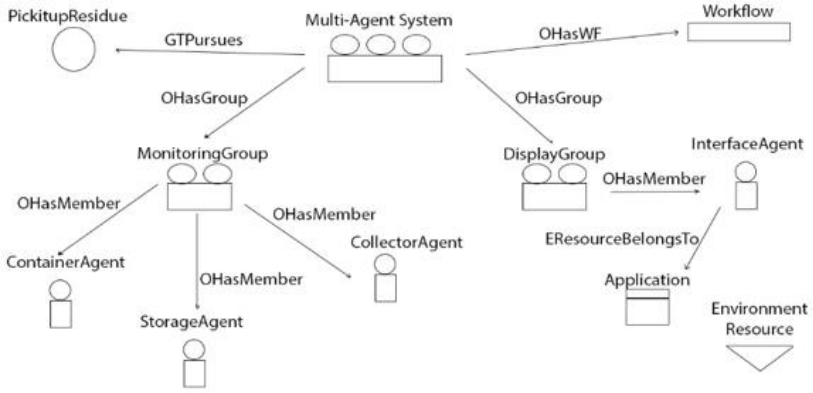


Fig. 7. Modelo de Organización.

La interfaz con los diferentes usuarios del sistema se realiza en la Organización del grupo de visualización (DisplayGroup), que se ha creado para satisfacer las necesidades de información de los usuarios.

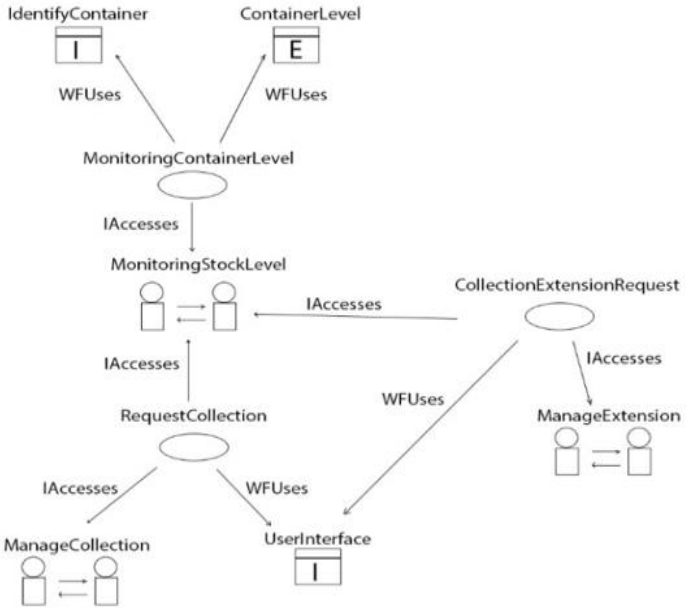


Fig. 8. Modelo de Metas y Tareas.

Modelo de Metas y Tareas. El objetivo de este modelo es mostrar el cumplimiento de los objetivos y tareas en el sistema multi-agente (SMA). En este modelo (Fig. 8) desde el momento en que comienza la recolección de cualquier residuo, comienza el proceso de monitoreo de los contenedores dentro del almacenamiento de la industria (MonitoringContainerLevel); en consecuencia, el almacenamiento se sincroniza con los

otros agentes para finalizar el período de almacenamiento de residuos (CollectionExtensionRequest) o la recolección de residuos (CollectionRequest). Todos los procesos, desde la creación hasta la eliminación final de residuos, estarán disponibles para ser monitoreados y consultados por el usuario.

Modelo de Entorno. El entorno del sistema multi-agentes propuesto (Fig. 9) está definido por los componentes internos del sistema sugerido. Los usuarios interactúan con el sistema y las aplicaciones indican los niveles de los diferentes contenedores que integran el sistema.

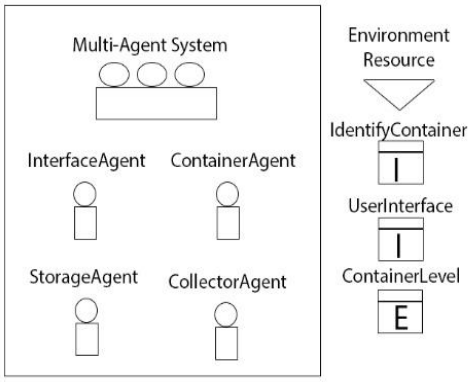


Fig. 9. Modelo de Entorno.

4. Conclusión y trabajos futuros

Este artículo propone una arquitectura multi-agente para el control de residuos peligrosos en empresas de manufactura con el objetivo de aumentar la eficiencia del uso de residuos y evitar al máximo los errores humanos en la gestión de los mismos.

La principal contribución de este trabajo es describir cómo se utilizó la metodología de INGENIAS para desarrollar nuestra arquitectura. Este modelo se encuentra actualmente en el proceso de validación y prueba piloto, y hay propuestas para desarrollarlo como parte del trabajo futuro. Una de esas propuestas es adaptar el modelo al proceso de fabricación en Maquilas Tetakawi's, una compañía de manufactura ubicada en Guaymas, México.

Por último, se contempla el desarrollo de un prototipo. Para esto, se tiene contemplado el uso de la plataforma JADE [13].

Referencias

1. Ley General para la Prevención y Gestión Integral de los Residuos (2003)
2. Anagnostopoulus, T., Kolomvatsos, K., Anagnostopoulos, C., Zaslavsky, A., Hadjiefthymiades, S.: Assessing dynamic models for high priority waste collection in smart cities. Journal of Systems and Software, pp. 178–192 (2015)

3. Popa, C., Carutasu, G., Cotet, C., Carutasu, N., Dobrescu, T.: Smart City Platform Development for an Automated Waste Collection System, In Sustainability, vol. 9, p. 11 (2017)
4. Greenberg, S., Mercer, S.: A multi-agent architecture for knowledge sharing. In: Sixteenth European Meeting on Cybernetic and Systems Research, Viena (2001)
5. Sen, S.: Reciprocity: A foundational principle for promoting cooperative behavior among self-interested agents. In: Second Int. Conf. Multi-Agent Systems (1996)
6. Kanj, H., Flaus, J.: Agent-based framework for mitigating hazardous materials transport risk. In: IEEE International Conference on Envolving and Adaptative Intelligent Systems, pp. 1–8 (2015)
7. Wooldridge, M., Jennings, N.: Intelligent Agents: Theory and Practice (1995)
8. Wooldridge, M., Ciancarini, P.: Agent-Oriented Software Engineering: The State of the Art., vol. 1957, Berlin: Springer (2001)
9. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence (1999)
10. Pavón, J., Gómez-Sanz, J., Fuentes, R.: The INGENIAS Methodology and Tools. Henderson-Sellers, pp. 236–276 (2005)
11. Gómez-Sanz, J., Fuentes, R., Pavón, J., García-Magariño., I.: INGENIAS development kit: a visual multi-agent system development environment. In: Proceedings of the 7th Conference on Autonomous Agents and Multi-agent Systems, Berlin (2008)
12. Gascuena, J., Navarro, E., Fernandez-Sotos, P., Fernandez-Caballero, A., Pavon., J.: IDK and ICARO to develop multi-agent systems in support of Ambient Intelligence. Journal of Intelligent & Fuzzy Systems, vol. 28, p. 1 (2015)
13. Java Site: JAVA Agent DEvelopment Framework, https://jade.tilab.com/. Último acceso: 2019/04/20

ISSN 1870-4069

Propuesta de red de distribución para reciclaje de desechos eléctricos y electrónicos

José-Raúl Morales-Contreras, Santiago-Omar Caballero-Morales, Patricia Cano-Olivos, Diana Sánchez-Partida

Universidad Popular Autónoma del Estado de Puebla A.C.,
Departamento de Posgrado en Logística y Dirección de la Cadena de Suministro,
Puebla, México
joseraul.morales@upaep.edu.mx, santiagoomar.caballero@upaep.mx,
patricia.cano@upaep.mx, diana.sanchez@upaep.mx

Resumen. La eliminación de desechos eléctricos y electrónicos, conocidos como tecno-basura, es uno de los retos más importantes para el medio ambiente. Esto dados los diferentes metales pesados que contienen estos desechos. Si bien existen centros especializados de reciclaje con la infraestructura para separar metales que pueden ser re-utilizados para reducir su impacto ambiental, su transporte hacia dichos centros es un problema que requiere planeación. El presente trabajo define una ruta logística para distribución de tecno-basura a 24 centros de reciclado en la Ciudad de Puebla. Esto con la finalidad de contribuir a la disminución de los niveles de tecno-basura en la ciudad. Para ello se emplearon herramientas heurísticas y programación en Octave. Los resultados respaldan la factibilidad de esta metodología en el contexto real para transporte de estos desechos.

Palabras clave: tecno-basura, red de distribución, heurísticas, optimización.

Distribution Network Approach for Recycling of Electrical and Electronic Waste

Abstract. The disposal of electrical and electronic waste, known as e-waste, is one of the most important challenges for the environment. This is due to the different heavy metals within this waste. Although there are specialized recycling centers with the infrastructure to separate metals that can be reused to reduce their environmental impact, the transportation of this waste to these centers is a problem that requires planning. This work defines a logistic route for the distribution of e-waste to 24 recycling centers in the City of Puebla. This is proposed to contribute to decrease the levels of e-waste in the city. To accomplish this, heuristic tools and programming with Octave were considered. The results support the feasibility of this methodology in the real context of transportation of e-waste.

Keywords: techno-trash, distribution network, heuristics, optimization.

1. Introducción

Hoy en día el término tecno-basura (e-waste en inglés) es ampliamente utilizado y hace referencia a un conjunto de residuos que son considerados de alto riesgo para la salud por los compuestos tóxicas que despiden [17]. Entre estos se encuentran el mercurio, plomo, cadmio, níquel, selenio, arsénico, cromo y bromo, que si no son tratados adecuadamente se convierten en un riesgo para la salud además de contribuir a la emisión de gases de efecto invernadero [5,24].

Principalmente estos compuestos están presentes en computadoras, teléfonos celulares, televisores y electrodomésticos que han llegado al final de su vida útil [3]. La tecno-basura resultante ha tenido un crecimiento constante. Tan solo en México, una persona genera 2.5 kg por año y cerca del 50% se deposita en rellenos sanitarios o en tiraderos que no tienen un control de lo que se vierte en ellos [17].

Bajo este contexto se han realizado propuestas para facilitar la recolección y tratamiento de este tipo de residuos. En [2] se propuso un modelo de programación estocástica para el diseño de una red logística inversa bajo la cantidad de retorno, la relación de clasificación (calidad) y las incertidumbres de los costos de transporte asociada a posibles localizaciones de centros de recolección y centros de clasificado.

Por su parte, en [23] se analizó el impacto de la planeación de la capacidad sobre el ciclo de vida del producto para el funcionamiento de las dimensiones de sustentabilidad en la responsabilidad social de la logística inversa utilizando sistemas dinámicos.

En [16] se presentó el diseño de una red logística inversa para la recolección de residuos de equipo eléctrico y electrónico en la región española de Galicia. Para ello, se propuso un enfoque jerárquico en donde la primera fase consistió de la formulación de un problema de ubicación de instalación mediante programación lineal entera mixta. En la segunda fase se formuló el problema de enrutamiento de vehículos de recolección para dicha instalación con flota heterogénea y se resolvió mediante un algoritmo heurístico basado en ahorros. En la tercera fase se realizó un estudio de simulación de las rutas de recolección para evaluar el rendimiento general del sistema de recuperación.

En [15] se desarrolló un modelo lineal entero mixto para un diseño de red logística inversa basada en la huella de carbono. La función objetivo de este modelo resuelto mediante el software Lingo minimizó costos considerando la recolección, eliminación, transporte, apertura fija y costos de emisiones de CO_2.

En [11] se abordó la problemática nacional de Brasil sobre los residuos sólidos debido a la falta de métodos y regulaciones acerca de los desechos electrónicos. Para ello se utilizaron metodologías como SODA (Strategic Options Development Analysis), PSM (Problem Structuring Methods), VFT (Value Focused Thinking Approach), SSM (Soft System Methodology) y SCA (Strategic Choice Approach) para análisis estratégicos y mejorar la toma de decisiones conjuntas en la cadena de suministro para los fines de procesamiento de desechos electrónicos.

En [22] se buscó minimizar el costo total asociado al transporte, el costo operativo, costo fijo para nuevas instalaciones, costo de disposición final y costo del relleno sanitario, así como optimizar los ingresos de la venta de materiales recuperados. Mediante su propuesta se determinó el número y la ubicación de los sitios de almacenamiento y desmontaje necesarios con un incremento en la tasa de recuperación del 20%. Por otra parte, en [1] se determinaron los centros de recolección necesarios

para recolectar productos usados cuyo valor puede ser recuperado mediante procesos especiales. Esto puede motivar a las empresas a participar en las operaciones de recolección [14].

Otro trabajo relevante es el presentado en [18] en donde se identificaron y analizaron criterios para seleccionar socios en cadenas de suministro inversas para los fines de recolección y tratamiento de desechos. Para ellos se utilizó una metodología integrada por herramientas como AHP (Analytic Hierarchy Process) y VIKOR (del serbio VIseKriterijumska Optimizacija I Kompromisno Resenje o Multicriteria Optimization and Compromise Solution).

En [21] se propuso un sistema de apoyo al proceso de decisión que permitiera identificar el tipo de sistema o red más adecuada para la recuperación de productos fuera de uso. De esta manera se determinó que productos cuyo valor es más susceptible al paso del tiempo suelen requerir una configuración de red que permita recuperar el producto en un plazo de tiempo corto (responsiva). En tanto, si el valor del producto no es tan sensible al tiempo se puede tener un plazo de tiempo mayor para su procesamiento. También se consideraron los procesos de recuperación económica más apropiados a cada producto como reutilización, re-fabricación, y reciclaje.

En cuanto a la aplicación de modelos de ruteo son escasos los trabajos reportados que implementen, por ejemplo, el modelo del Problema de Ruteo Vehicular Capacitado (CVRP) específicamente para el diseño de redes de logística inversa. Existen recopilaciones como en [4] que presentan diferentes métodos matemáticos para la solución óptima de redes de distribución haciendo énfasis en las emisiones en las rutas óptimas.

En una dirección similar, existen trabajos que consideran el CVRP considerando flotas de vehículos híbridos (eléctricos) como soporte a una red logística verde [25]. El CVRP, al ser considerado como un problema de complejidad computacional NP-duro, usualmente requiere de métodos heurísticos para su resolución con instancias de mediana y alta escala. En el caso de [25] se utilizó un algoritmo de recocido simulado.

Por último, en [13] se integró el costo económico y la reducción del costo de emisión para un problema de enrutamiento de vehículos verdes con múltiples depósitos capacitados. Para esto se ocupó un modelo de programación lineal entera para un conjunto de instancias a pequeña escala utilizando la herramienta Lingo. Cabe mencionar que también se desarrolló una meta-heurística basada en la optimización de colonias de hormigas (ACO, por sus siglas en inglés) para resolver instancias de problemas tanto a pequeña como a gran escala en un tiempo razonable.

El presente trabajo aporta en el contexto de diseño de una red de transporte mediante el CVRP para la recolección de desechos electrónicos y su distribución a centros de procesamiento para la recuperación de su valor y correcta eliminación. Esto dado que, al igual que en lugares como España, Brasil, Turquía, en México hay mucha generación de este tipo de desecho y una tasa muy baja de eliminación responsable mediante el reciclado (5.1% al año 2013). Actualmente, México se considera el segundo país de América Latina con más desechos eléctricos y electrónicos [19,20].

En el caso particular del Estado de Puebla, se recolectan en promedio 120 unidades de residuos electrónicos y eléctricos diario, siendo el cuarto estado con mayor recolección de este tipo de residuos a nivel nacional [12]. Sin embargo, estudios han comprobado que este tipo de desechos denominados tecno-basura ha superado la capacidad de reciclaje de los servicios sanitarios de la entidad [10].

Por lo tanto, se plantea que el presente trabajo brinde soporte al cuidado del medio ambiente en la entidad al proveer de una metodología para el diseño de red de recolección y distribución para el procesamiento correcto de tecno-basura. Esto con el fin de evitar que los desechos terminen en barrancas, rellenos sanitarios o tiraderos clandestinos, y que los compuestos dañinos que emiten no tengan un contacto directo con los recursos naturales.

2. Contexto del problema

En la zona metropolitana de Puebla existen 3 empresas autorizadas para el reciclado de residuos mismas que no son suficientes para el tratamiento de hasta 187 mil 677 toneladas que se generan de forma mensual [6]. Por lo que, la Secretaría de Desarrollo Rural, Sustentabilidad y Ordenamiento Territorial (SDRSOT) ha tenido que aprobar las autorizaciones de otros centros de recolección de empresas o centros de reciclaje privado para el manejo especial de residuos, es decir, para la separación y tratamiento de basura reciclable [6].

Sin embargo, a pesar de haber aumentado los centros de reciclado, estos se enfocan en el tratamiento de los residuos que les lleva la población, por lo que no existe un sistema de distribución que envíe los desechos de los Servicios Sanitarios de Puebla a los centros de reciclado para disminuir la acumulación de residuos eléctricos y electrónicos del establecimiento. Esto, a pesar de existir legislaciones que obligan a las entidades federativas a hacerse cargo de la recolección, acopio, almacenamiento, transporte, tratamiento y disposición final de la basura en general [8].

Bajo este contexto, se sugiere el diseño de una red de logística inversa que considere el transporte de residuos electrónicos y electrodomésticos desde la ubicación de los Servicios Sanitarios de Puebla que es en donde se concentra la mayor cantidad de residuos de la ciudad hasta los principales centros de reciclado de tecno-basura por medio de vehículos sujetos a cierta capacidad. Esto para el uso más eficiente de los recursos estatales o de empresas privadas que deseen participar en este proceso.

3. Metodología

3.1 Obtención de datos

Datos concernientes a las ubicaciones de los centros de reciclaje disponibles se basó en la información principalmente de la zona metropolitana de Puebla y zonas cercanas. Para el caso de estudio de la zona metropolitana del Estado de Puebla, la recolección de datos fue llevada a cabo mediante una minuciosa investigación en diferentes bancos de datos disponibles para el acceso público, a través de dos fuentes oficiales que son el INEGI y la Secretaría de Desarrollo Rural, Sustentabilidad y Ordenamiento Territorial (SDRSOT), de donde se obtuvo la información sobre las empresas que cuentan con permiso para el procesamiento de los residuos de aparatos eléctricos y electrónicos (RAEE). De esta manera la figura 1 presenta los centros identificados.

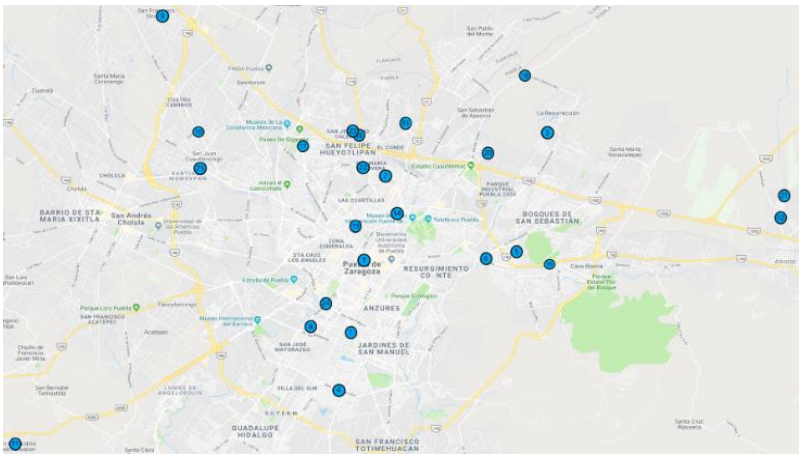


Fig. 1. Localización de 24 centros de reciclado y un centro de distribución en la zona metropolitana de Puebla.

A partir de esta información se estimó la llegada de desechos máxima a recibir por cada centro. Dado que el presente trabajo está enfocado en los desechos eléctricos y electrónicos, se consideraron las estadísticas más recientes con respecto a este rubro.

Al año 2011 se reportó que en México se generaban aproximadamente 300,000 toneladas de tecno-basura. Siendo el valle de México aquél con la mayor generación anual con 94,000 toneladas en tanto que las demás entidades generaban entre 36,000 y 10,000 toneladas al año [17]. Con una tasa estimada de crecimiento de 6% anual esto daría al año 2019 un aproximado de 451,090 toneladas. En el caso de Puebla se considera una cantidad aproximada de 12,000 toneladas anuales.

En cuanto a la capacidad de los vehículos para transporte se ha reportado un aproximado de entre 8 y 10 toneladas [7].

3.2 Procesamiento de datos

Para poder resolver el problema de ruteo es necesario hacer un procesamiento inicial de los datos. En primer lugar, se determinó la distancia relativa entre cada centro de reciclado y el centro de distribución. Para ello, se hizo uso de la distancia Euclidiana y de programación en Octave. La figura 2 presenta el código realizado para la obtención de este recurso en tanto la figura 3 presenta la matriz simétrica de distancias obtenida.

En segundo lugar, se estimó la llegada promedio diaria de toneladas a reciclar en cada centro de reciclado. Esto para hacer una eficiente la distribución de los desechos a lo largo del mes.

Previamente se estimó que la entidad de Puebla genera un aproximado de 12,000 toneladas anuales de tecno-basura. Esto es equivalente a 1000 toneladas al mes lo cual daría un estimado de 1000/30 días = 33.3 toneladas diarias, o 33.3 toneladas / 24 centros = 1.38 toneladas diarias a recibir por cada uno de los centros de reciclado. Para la capacidad de vehículos se estiman 8 toneladas de carga [7]

```
coordenadas; % Arreglo nx2 en donde se tienen las coordenadas x y y de cada
             % una de las n instalaciones

numero_nodos=length(coordenadas(:,1));

matriz_distancia=[];

for i=1:numero_nodos
  nodoA=coordenadas(i,:);
  for j=1:numero_nodos
    nodoB=coordenadas(j,:);
    distEuc=sqrt( (nodoB(1)-nodoA(1))^2 + (nodoB(2)-nodoA(2))^2 );
    matriz_distancia(i,j)=distEuc;
  end
end
```

Fig. 2. Código Octave para estimación de matriz de distancias Euclidianas entre cada una de las 25 localizaciones de la red.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	0.000	0.061	0.107	0.023	0.071	0.059	0.028	0.016	0.141	0.096	0.135	0.171	0.083	0.049	0.081	0.075	0.074	0.119	0.041	0.087	0.174	0.063	0.077	0.015	0.085
2	0.061	0.000	0.064	0.084	0.058	0.050	0.033	0.064	0.105	0.073	0.175	0.152	0.021	0.015	0.071	0.018	0.034	0.066	0.022	0.040	0.153	0.009	0.021	0.054	0.071
3	0.107	0.064	0.000	0.127	0.047	0.053	0.086	0.117	0.154	0.134	0.236	0.095	0.055	0.065	0.050	0.072	0.094	0.023	0.082	0.024	0.094	0.072	0.075	0.107	0.134
4	0.023	0.084	0.127	0.000	0.086	0.076	0.051	0.027	0.158	0.112	0.126	0.182	0.105	0.071	0.094	0.098	0.094	0.140	0.063	0.108	0.186	0.086	0.099	0.034	0.100
5	0.071	0.058	0.047	0.086	0.000	0.012	0.059	0.084	0.163	0.130	0.206	0.102	0.065	0.048	0.013	0.075	0.091	0.067	0.063	0.040	0.105	0.067	0.078	0.076	0.125
6	0.059	0.050	0.053	0.076	0.012	0.000	0.047	0.072	0.155	0.120	0.194	0.114	0.060	0.038	0.024	0.068	0.082	0.071	0.052	0.041	0.117	0.059	0.071	0.064	0.115
7	0.028	0.033	0.086	0.051	0.059	0.047	0.000	0.033	0.121	0.080	0.151	0.161	0.055	0.022	0.071	0.048	0.049	0.094	0.013	0.063	0.163	0.035	0.049	0.022	0.072
8	0.016	0.064	0.117	0.027	0.084	0.072	0.033	0.000	0.131	0.086	0.122	0.185	0.086	0.055	0.095	0.075	0.069	0.126	0.042	0.095	0.188	0.064	0.076	0.011	0.074
9	0.141	0.105	0.154	0.158	0.163	0.155	0.121	0.131	0.000	0.047	0.173	0.249	0.102	0.117	0.176	0.088	0.073	0.141	0.109	0.135	0.248	0.096	0.085	0.126	0.060
10	0.096	0.073	0.134	0.112	0.130	0.120	0.080	0.086	0.047	0.000	0.138	0.226	0.079	0.082	0.144	0.061	0.040	0.127	0.070	0.111	0.226	0.064	0.059	0.082	0.014
11	0.135	0.175	0.236	0.126	0.206	0.194	0.151	0.122	0.173	0.138	0.000	0.306	0.193	0.171	0.216	0.177	0.158	0.241	0.155	0.213	0.310	0.170	0.176	0.131	0.127
12	0.171	0.152	0.095	0.182	0.102	0.114	0.161	0.185	0.249	0.226	0.306	0.000	0.148	0.147	0.090	0.165	0.185	0.112	0.163	0.115	0.009	0.161	0.168	0.177	0.224
13	0.083	0.021	0.055	0.105	0.065	0.060	0.055	0.086	0.102	0.079	0.193	0.148	0.000	0.034	0.077	0.018	0.040	0.049	0.044	0.033	0.148	0.023	0.020	0.075	0.081
14	0.049	0.015	0.065	0.071	0.048	0.038	0.022	0.055	0.117	0.082	0.171	0.147	0.034	0.000	0.062	0.033	0.044	0.072	0.017	0.042	0.149	0.022	0.036	0.044	0.078
15	0.081	0.071	0.050	0.094	0.013	0.024	0.071	0.095	0.176	0.144	0.216	0.090	0.077	0.062	0.000	0.088	0.105	0.072	0.076	0.049	0.094	0.080	0.091	0.087	0.139
16	0.075	0.018	0.072	0.098	0.075	0.068	0.048	0.075	0.088	0.061	0.177	0.165	0.018	0.033	0.088	0.000	0.022	0.068	0.035	0.050	0.165	0.012	0.003	0.065	0.062
17	0.074	0.034	0.094	0.094	0.091	0.082	0.049	0.069	0.073	0.040	0.158	0.185	0.040	0.044	0.105	0.022	0.000	0.089	0.036	0.071	0.186	0.024	0.020	0.061	0.040
18	0.119	0.066	0.023	0.140	0.067	0.071	0.094	0.126	0.141	0.127	0.241	0.112	0.049	0.072	0.072	0.068	0.089	0.000	0.087	0.032	0.109	0.071	0.069	0.116	0.130
19	0.041	0.022	0.082	0.063	0.063	0.052	0.013	0.042	0.109	0.070	0.155	0.163	0.044	0.017	0.076	0.035	0.036	0.087	0.000	0.058	0.165	0.022	0.036	0.032	0.064
20	0.087	0.040	0.024	0.108	0.040	0.041	0.063	0.095	0.135	0.111	0.213	0.115	0.033	0.042	0.049	0.050	0.071	0.032	0.058	0.000	0.115	0.048	0.052	0.085	0.111
21	0.174	0.153	0.094	0.186	0.105	0.117	0.163	0.188	0.248	0.226	0.310	0.009	0.148	0.149	0.094	0.165	0.186	0.109	0.165	0.115	0.000	0.162	0.167	0.180	0.224
22	0.063	0.009	0.072	0.086	0.067	0.059	0.035	0.064	0.096	0.064	0.170	0.161	0.023	0.022	0.080	0.012	0.024	0.071	0.022	0.048	0.162	0.000	0.014	0.054	0.063
23	0.077	0.021	0.075	0.099	0.078	0.071	0.049	0.076	0.085	0.059	0.176	0.168	0.020	0.036	0.091	0.003	0.020	0.069	0.036	0.052	0.167	0.014	0.000	0.067	0.060
24	0.015	0.054	0.107	0.034	0.076	0.064	0.022	0.011	0.126	0.082	0.131	0.177	0.075	0.044	0.087	0.065	0.061	0.116	0.032	0.085	0.180	0.054	0.067	0.000	0.071
25	0.085	0.071	0.134	0.100	0.125	0.115	0.072	0.074	0.060	0.014	0.127	0.224	0.081	0.078	0.139	0.062	0.040	0.130	0.064	0.111	0.224	0.063	0.060	0.071	0.000

Fig. 3. Matriz de distancias entre cada instalación. En la localización 1 se identifica el centro de distribución en tanto que en las localizaciones 2-25 se identifican los 24 centros de reciclado.

3.3 Método de solución

Como se revisó en la Sección 1, muchos de los trabajos en el aspecto de redes para reciclado y procesamiento de desechos consideran el problema de ruteo vehicular o CVRP. Este problema se formula matemáticamente de la siguiente manera [9]:

$$min \sum_{i \in V} \sum_{j \in V} C_{ij} X_{ij} \,. \tag{1}$$

Sujeto a:

$$\sum_{i \in V} X_{ij} = 1 \ , \forall \, j \ \in V \backslash \{0\}, \tag{2}$$

$$\sum_{J\in V} X_{ij} = 1 \ , \forall \ i \ \in V\backslash\{0\}, \tag{3}$$

$$\sum_{i\in V} X_{i0} = K, \tag{4}$$

$$\sum_{j\in V} X_{0j} = K, \tag{5}$$

$$\sum_{i\notin S}\sum_{j\in S} X_{ij} \geq r(S) \ \forall \ S \subseteq V\backslash\{0\}, S \neq \emptyset, \tag{6}$$

$$X_{ij} \in \{0,1\}, \forall \ i,j \ \in V. \tag{7}$$

En donde C_{ij} es la distancia entre cada localización *i* y *j*, X_{ij} es la variable de decisión binaria que es igual a "1" si el camino entre *i* y *j* es parte de la solución y es igual a "0" en caso contrario, *K* es el número de rutas / vehículos a considerar para dar servicio a todas las instalaciones, *V* es el conjunto de nodos en la red en donde "0" es el centro de distribución, y *S* es el subconjunto de nodos de *V* que pertenece a las ubicaciones a visitar (centros de reciclado). De esta manera, (1) establece la función objetivo que minimiza los tiempos de ruteo entre cada una de las instalaciones en la red, (2) y (3) son restricciones que establecen que cada centro o nodo cliente (p.e., centros de reciclado) es visitado sólo una vez, las restricciones (4) y (5) establecen que los vehículos que salen del centro de distribución son los mismos que regresan al final de sus rutas de distribución. Finalmente, (6) es la restricción asegura que haya conectividad entre cada ubicación y (7) la establece la naturaleza de la variable de decisión.

En base a esta formulación, el resultado consistirá de rutas que en su conjunto darán servicio a cada nodo en la red, cubriendo sus requerimientos sin exceder la capacidad de los vehículos al mismo tiempo que se minimizara el tiempo o distancia de traslado entre los mismos. La figura 4 ilustra esta solución.

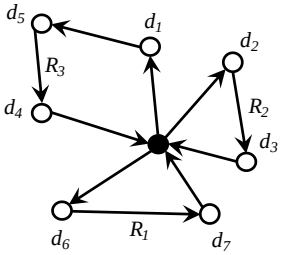


Fig. 4. Modelo de ruteo CVRP.

Para obtener la solución al CVRP se realizó la programación de una adaptación del meta-heurístico presentado en [9] mediante el software Octave. Nuestra adaptación se presenta en la figura 5.

Este meta-heurístico fue considerado dado que puede implementarse de una manera rápida mediante software libre como lo es Octave. Esto en contraste con el uso de software que requiere algún tipo de licencia.

A diferencia del meta-heurístico presentado en [9], nuestra adaptación incluye mejora en las dos etapas del mismo, esto es, en la creación de una solución inicial y en la obtención de la solución final.

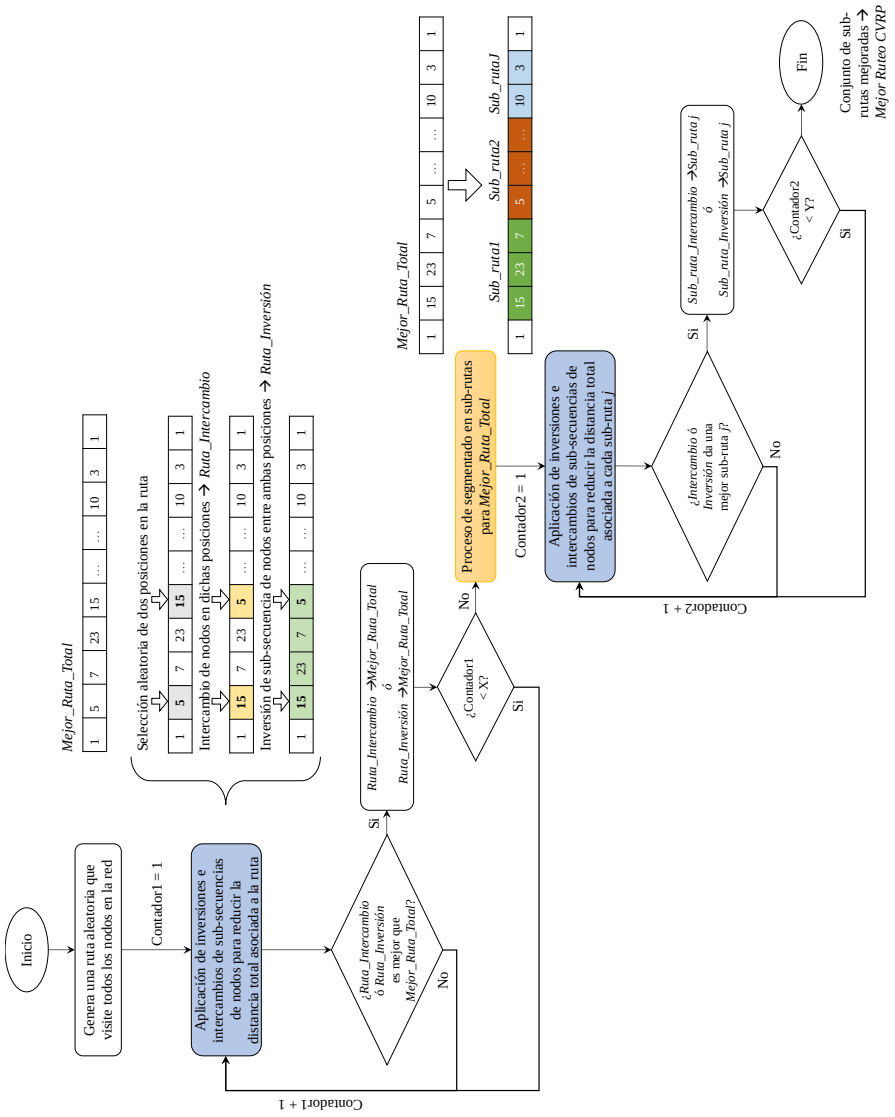


Fig. 5. Meta-heurístico implementado para dar solución al CVRP.

De manera general este meta-heurístico inicialmente crea una sola ruta para visitar a todos los centros de reciclado. Esto se hace sin considerar los requerimientos de los centros y las capacidades de los vehículos. Esta ruta se mejora mediante operaciones heurísticas de inversión e intercambio de nodos/subsecuencias de nodos, que son

algoritmos heurísticos para el mejoramiento de una solución. En medida que se aplican estos procesos se obtiene una ruta cuya distancia total sea mínima.

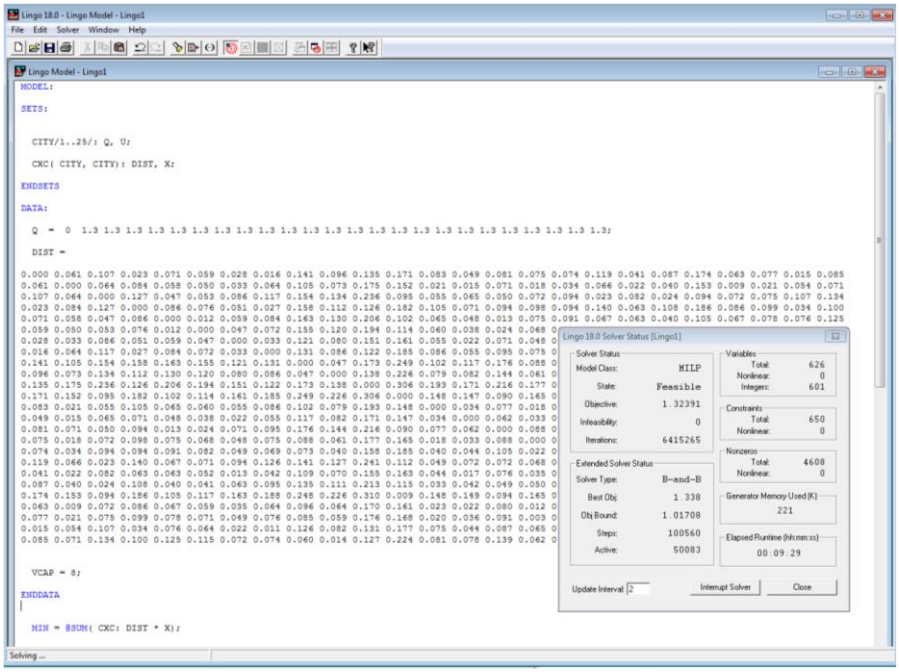


Fig. 6. Implementación en Lingo para evaluar el desempeño del meta-heurístico para el CVRP.

La mejor ruta obtenida después de X=100,000 iteraciones de los procesos de inversión/intercambio se utilizó para la siguiente etapa que consiste en el segmentado en sub-rutas. En esta etapa se consideran las restricciones de demanda y capacidad de los centros y vehículos respectivamente. Al igual que en la etapa inicial de creación, las sub-rutas segmentadas se mejoran mediante intercambio e inversión de nodos. Este proceso se ejecutó durante Y = 10,000 iteraciones.

La implementación se realizó en un equipo con sistema operativo Microsoft Windows 10 Pro con procesador AMD A8-7410 APU y 6 GB RAM. Para evaluar el desempeño del algoritmo se hizo la implementación del modelo matemático del CVRP en el software de optimización Lingo. La Figura 6 presenta de manera general esta implementación, la cual utiliza el algoritmo de Branch & Bound (B-and-B) para obtener una solución óptima.

4. Resultados

La Figura 7 presenta el esquema de ruteo obtenido para hacer la distribución a los 24 centros de reciclado diariamente. Este resultado se obtuvo en un tiempo de 30 segundos. Como se observa, se determinaron 4 rutas (4 vehículos) de distribución con

el esquema de cobertura presentado en la Tabla 1. Estas rutas recorren una distancia Euclidiana de 1.3651.

Tabla 1. Resultado de ruteo diario para distribuir a los 24 centros de reciclado.

Ruta 1	1	6	5	15	12	21	18	1
Ruta 2	1	25	10	9	17	23	2	1
Ruta 3	1	3	20	13	16	22	19	1
Ruta 4	1	14	7	24	8	11	4	1

En comparación, Lingo proporcionó una solución con 5 rutas y una distancia Euclidiana de 1.3347 en un tiempo aproximado de 30 minutos. El error de la solución obtenida mediante el meta-heurístico fue de [(1.3652 - 1.3347) /1.3347] × 100% = 2.28% lo cual es mínimo y representa una ventaja en cuanto a tiempo de obtención, principalmente para problemas de ruteo con mayor número de nodos. Finalmente, el meta-heurístico, al proporcionar una solución con menos rutas, puede favorecer la reducción en costos operativos.

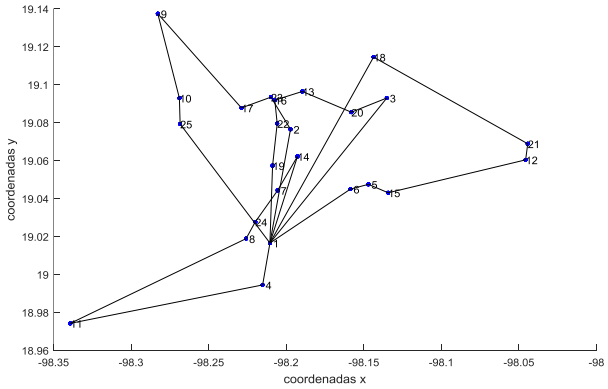


Fig. 7. Rutas determinadas para distribuir a los 24 centros de reciclado en Puebla.

5. Conclusiones

En el desarrollo del presente proyecto se conoció acerca del impacto de los residuos eléctricos y electrónicos en el medio ambiente y para el ser humano. De igual manera se conocieron las limitadas acciones e infraestructura para llevar a cabo el reciclado de estos desechos de manera efectiva.

Por lo anterior, se propuso una cadena de logística inversa para procurar que todos los desechos eléctricos y electrónicos que llegan a los servicios sanitarios de Puebla sean reciclados y de esta forma evitar lo más posible el contacto de las sustancias tóxicas que segregan con el medio ambiente.

Se determinó una solución para abarcar la zona metropolitana de Puebla, así como algunos puntos aleñados de manera efectiva. Con estas rutas se espera que se pueda transportar la tecno-basura hacia los centros de reciclado disponibles.

Aunque este proyecto se enfoca exclusivamente a los desechos eléctricos y electrónicos, es recomendable desarrollar un proyecto similar enfocado hacia cada uno de los desechos de la basura.

Para ello, sería necesario que la población se organizara para separar todos y cada uno de los desechos que se generan, con la finalidad de que la separación y reciclado de la basura se garantice y sobre todo sea efectivo.

Como trabajo futuro se pretende estimar de manera más específica la cantidad de tecno-basura generada en la entidad y la disponibilidad de infraestructura para su transporte. De igual manera, incluir los diferentes desechos y calendarios de recolección y entrega para una mejor planeación en base a la densidad de transporte en la entidad.

Referencias

1. Aras, N., Aksen, D., Gönül Tanuğur, A.: Locating collection centers for incentive-dependent returns under a pick-up policy with capacitated vehicles. European Journal of Operational Research 191(3), 1223–1240 (2008)
2. Ayvaz, B., Bolat, B., Aydın, N.: Stochastic reverse logistics network design for waste of electrical and electronic equipment. Resources, Conservation and Recycling 104, pp. 391–404 (2015)
3. Bazurto, J.A., Osorio, D., Usme, E., Satiba, S.: La tecno-basura un problema contemporáneo (16/11/2010), http://latecnobasura.blogspot.com/. Último acceso: 2018/12/19
4. Carroll, P., Keenan, P.: Chapter 10 - Decision Making Using Exact Optimization Methods in Sustainable Transportation. In: Faulin, J., Grasman, S.E., Juan, A.A., Hirsch, P. (eds.): Sustainable Transportation and Smart Logistics, pp. 263–283. Elsevier (2019)
5. Colprensa: Así afecta la basura electrónica a la salud y al medioambiente (25/12/2017). El Heraldo, https://www.elheraldo.co/salud/asi-afecta-la-basura-electronica-la-salud-y-al-medioambiente-440278. Último acceso: 2018/12/18
6. David, S.: Sólo tres empresas reciclan electrónicos en Puebla (14/11/2017). Tierra Baldía: Comunicación y Periodismo, http://tierrabaldia.com.mx/noticia/1083/solo-tres-empresas-reciclan-electronicos-en-puebla/. Último acceso: 2019/01/20
7. Dorantes, R.: Entregan 15 nuevos camiones recolectores de basura en V Carranza (15/05/2017). Excelsior, https://www.excelsior.com.mx/comunidad/2017/05/15/1163663. Último acceso: 2019/03/30
8. Secretaría General del Gobierno de los Estados Unidos Mexicanos: Ley General para la Prevención y Gestión Integral de los Residuos. Diario Oficial de la Federación (2018)
9. Duran, M., Caballero-Morales, S.O., Martínez-Flores, J.L., Cano-Olivos, P.: Improvement of Pick-Up Routes for an International Shipping Enterprise by Using a Heuristic Method. Revista Latino-americana de Inovação e Engenharia de Produção 4(6) 58–68 (2016)
10. González-Soto, E.: Tecno-basura agobia a Puebla y buscan mitigar sus efectos (1/04/2018) E-consulta.com, http://www.e-consulta.com/nota/2018-04-01/medio-ambiente/tecnobasura-agobia-puebla-y-buscan-mitigar-sus-efectos. Último acceso: 2019/03/30
11. Guarnieri, P., Camara-e-Silva, L., Levino, N.: Analysis of electronic waste reverse logistics decisions using Strategic Options Development Analysis methodology: A Brazilian case. Journal of Cleaner Production 133, pp. 1105–1117 (2016)
12. INEGI: México en Cifras (1/01/2018), http://www.beta.inegi.org.mx/app/areasgeograficas/?ag=21#tabMCcollapse-Indicadores. Último acceso: 2018/04/30
13. Jabir, E., Panicker, V.V., Sridharan, R: Design and development of a hybrid ant colony-variable neighborhood search algorithm for a multi-depot green vehicle routing problem. Transportation Research Part D: Transport and Environment 57, pp. 422–457 (2017)

14. Juan-Martín, M.: Ahora, Sony te da hasta $10,000 por tu vieja televisión (1/04/2018). Paréntesis.com, https://www.parentesis.com/noticias/Informacion/Ahora_Sony_te_da_hasta_10000_por_tu_vieja_television#. Último acceso: 2019/03/30
15. Kannan, D., Diabat, A., Alrefaei, M., Govindan, K., Yong, G.: A carbon footprint based reverse logistics network design model. Resources, Conservation and Recycling 67, pp. 75–79 (2012)
16. Mar-Ortiz, J., Adenso-Diaz, B., González-Velarde, J.L.: Design of a recovery network for WEEE collection: the case of Galicia, Spain. Journal of the Operational Research Society 62(8), 1471–1484 (2011)
17. Méndez, E.: México se inunda de tecno-basura; produce 300 mil toneladas al año (27/02/2011). Excélsior, https://www.excelsior.com.mx/node/717925. Último acceso: 2018/12/18
18. Prakash, C., Barua, M.K.: A combined MCDM approach for evaluation and selection of third-party reverse logistics partner for Indian electronics industry. Sustainable Production and Consumption 7, pp. 66–78 (2016)
19. Porter, N.: En México, sólo 10% de electrónicos se recicla de forma responsable (8/05/2018) La Capital, http://www.lacapital.com.mx/noticia/63156-En_Mexico_solo_10_de_electronicos_se_recicla_de_forma_responsable. Último acceso: 2018/06/17
20. Redacción EC: ¿Qué países generan más basura electrónica? (20/04/2015). El Comercio. https://elcomercio.pe/tecnologia/actualidad/paises-generan-basura-electronica-374949. Último acceso: 2018/06/18
21. Ribas-Vila, I., Rubio, S.: Decisiones en el diseño de redes de logística inversa: propuesta de un modelo de decisión. En: Proceedings of the 7th International Conference on Industrial Engineering and Industrial Management – XVII Congreso de Ingeniería de Organización, pp. 1025–1033 (2013)
22. Shih, L.-H.: Reverse logistics system planning for recycling electrical appliances and computers in Taiwan. Resources, Conservation and Recycling 32(1), 55–72 (2001)
23. Sudarto, S., Takahashi, K., Morikawa, K., Nagasawa, K.: The impact of capacity planning on product lifecycle for performance on sustainability dimensions in Reverse Logistics Social Responsibility. Journal of Cleaner Production 133, pp. 28–42 (2016)
24. Valencia, M.: ¿Sabe qué pasa con los dispositivos electrónicos que dejó de usar? (21/04/2018). Dinero, http://www.dinero.com/internacional/articulo/las-consecuencias-de-la-basura-electronica/257532. Último acceso: 2019/03/30
25. Yu, V.F., Redi, A.A.N.P., Hidayat, Y.A., Wibowo, O.J.: A simulated annealing heuristic for the hybrid vehicle routing problem. Applied Soft Computing 53, pp. 119–132 (2017)

ISSN 1870-4069

On Generating Random Graphs Based on Fuzzy Proximity Relationships between Vertices

Mohamed Amine Omrani, Wady Naanaa

Faculty of Sciences of Tunis,
Computer Science Department,
LIMTIC Laboratory-ISI, Ariana, Tunisia
mohamedamine.omrani@fst.utm.tn, wady.naanaa@fsm.rnu.tn

Abstract. Uncertain graphs are getting more and more important. They allow to tackle fuzzy situations in numerous frameworks. This paper investigates the issue of generating random graphs based on uncertain proximity relationships between vertices and the goal is to construct the most likely graph. The Constraint Programming paradigm was used to provide a systematic way to release uncertain graphs while maximizing and minimizing the paths that separate certain vertex pairs. The proposed approach allowed to generate uncertain graphs at a reasonable time. This is confirmed by experimental results obtained from a series of tests on several instances. Our solution lays the basis for further real world applications in different fields, such as analytical chemistry, telecommunication networks, and civil engineering.

Keywords: random graphs, fuzzy proximity, constrain programming.

1 Introduction

In mathematical context, a graph is a non-linear data structure consisting of a set of vertices and edges that link vertices together. One of the most attractive features of graphs is their adequacy to model pairwise relationship between objects. Indeed, such structural information would be useful to represent and solve relevant computational problems by providing a fairly accurate description of the problem at hand. Thereby, a random graph is a graph that is generated by a random process. Historically, the first model of random graphs was defined by *Paul Erdõs* [3] to give a probabilistic construction of a graph with large girth and large chromatic number. Later, there has been a shift of emphasis toward generating random graphs with prescribed feature constraints such as vertex degree [16, 15], imposed or forbidden induced sub-graphs [18], proximity relationships between vertices [17]. Graph theory is a vast subject in which the goals are relying on various graph properties.

The issue of random graph generation is not recent, but is still important because its applications are renewable in real-life situations. The growing interest in this fascinating issue can be attributed to several factors. One factor is the realization that networks are everywhere. From social networks such as

Facebook, Twitter, Instagram or Linkedin, the World Wide Web and the Internet to the complex interactions between isomers in the molecular structures, we face the challenge of elucidating their structure. By and large natural networks grow in an unpredictable manner and this is often modeled by a random construction. Another factor is the realization by Computer Scientists that NP-hard problems are often easier to solve than their worst-case suggests and that an analysis of running times on random instances can be informative.

However, generating graphs is commonly a complicated task since it amounts to the assembly of a set of vertices in all possible ways to ensure exhaustivity while being consistent with a set of structural constraints such as the number of vertices, edges, and connections between them. Figure 1 illustrates with an example case the task of generating constrained random graphs under constraints. Let G be a random simple graph to generate defined by a given degree sequence $\Delta = \langle 2, 2, 2, 3, 3, 4, 4 \rangle$, a set of 7 vertices, a set of 10 edges and other hard constraint such as imposing a distance of length two separating vertices 1 and 4. The goal is to generate all graphs that satisfy the given input data set. This configuration gives 249 simple undirected graphs of which 19 graphs are connected graphs with no symmetry. Of these, only two graphs respect the proximity relationship constraint between vertices 1 and 4.

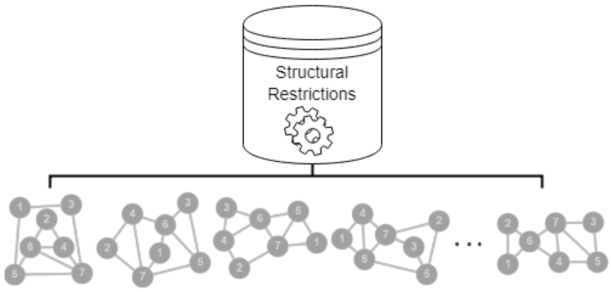


Fig. 1. Generating constrained graphs based on degree sequence $\Delta = \langle 2, 2, 2, 3, 3, 4, 4 \rangle$.

Often, the conditions under which the data are obtained are less favorable in reality. We have, therefore, to deal with data tainted by uncertainties due to a misinterpretation of real situation. Generally, a large set of graphs are consistent with a same degree sequence. In such a situation, determining the graph that models the true real-world situation is not always obvious because of the huge set of candidate graphs. Therefore, it appears that an intelligent assembly process is necessary to deal with the combinatorial nature of the problem. The second challenge comes from the use of uncertain data in order to narrow the set of candidate graphs. It is possible that after having listed the desired structural constraints, there is no way to satisfy them all simultaneously. In this case, the

problem is said to be over-constrained. A way to get around such a situation is to choose which constraints to ignore. On the other hand, when we get a huge set of graphs that simultaneously satisfy all the constraints, such graphs appear equally good, and another challenge consists in benefiting from the available uncertain information to select the more likely graph among the numerous candidate graphs.

Although several studies have been dedicated to random graphs in recent years, some focusing on graph properties [10], others on generation methods [18, 7]. They are often highly dedicated to specific situations and encounter difficulties in taking into account the uncertainty nature of some information. In addition, the assumed constraints are unlikely to occur in real-world situations. On the other hand, to our knowledge almost half of the structural problems contain uncertain information as input data. Therefore, an approach for dealing with uncertain graphs will be of great interest.

The current research was carried out to shed light on the problem of generating uncertain graphs, where the uncertainty nature comes from Uncertain Proximity Relationships (UPRs) between vertices. A UPR is a proximity relationship between two vertices for which the topological distance is not known but bounded (above or below) by a given constant. What is meant by topological distance is the number of edges that composes the shortest path which connects the concerned vertex pair. UPRs are of two types: UPRs to maximize (max-UPRs) and UPRs to minimize (min-UPRs). The min-UPRs type includes uncertain proximity favoring short topological distances separating vertices. Structural preferences here state that the shorter the overall min-UPRs distances is, the more likely the graph is to correspond to the true graph. The second type of uncertain proximity, that is max-UPR, is basically the first type reversed, favoring a long topological distance between each vertex pair involved in a max-UPR. As mentioned above, an UPR is specified by an upper or a lower bound on the topological distance separating a given vertex pair.

This paper proposes a weighted constraint-based approach providing a systematic way to generate uncertain graphs, while enforcing uncertain proximity relationship preferences of certain prescribed vertex pairs. More concretely, we explore here the use of the Constraint Programming (CP) paradigm to generate graphs on the basis of a given degree sequence, while maximizing the sum of all topological distances of max-UPRs preferences and minimizing those of min-UPRs.

Our interest in UPRs is motivated by their importance as basic data structures for many real-life applications. For example, in analytical chemistry, the molecular structure can be considered as a graph and the problem of generating molecular structure consists in finding all molecular graphs that are consistent with a dataset derived from different kinds of spectra. Often, this dataset imposes a set of uncertain inter-atoms proximity relationships for which the topological distance between interacting atoms cannot be accurately known but rather bounded by given constants [2]. This issue, besides being interesting, is useful in a variety of situations and is of great interest not only for chemists, but also to

biologists, computer scientists, economists, electrical engineers, mathematicians, neuroscientists and sociologist [23, 2]. From the theoretical point of view, this problem is NP-hard on general graphs. And to the best of our knowledge, research on this topic are scarce. A key part of this contribution is to rely on uncertain proximity relationships to guide the construction process to the most likely graphs.

The contribution of the paper is three folds. First, we propose a new formulation, precisely the one resorting to Constraint Programming (CP), a paradigm that has proved to be an adequate framework for dealing with various combinatorial optimization problems. So, generating uncertain graphs is encoded as a Weighted Constraint Satisfaction Problem (WCSP) in an easy and declarative way that enables an efficient integration of various pieces of structural restrictions. Second, we show how this encoding can be employed to get the best we can from min-UPR restrictions, as well as, from max-UPR restrictions during the graph generation process. Third, we have succeeded in defining a single objective function, developed as a soft constraint of the weighted constraint-based formulation, that combines the two contradictory preferences, max-UPR and min-UPR to finally get the most likely uncertain graphs.

The rest of the paper is organized as follows: Section 2 and 3, are respectively devoted to giving preliminary notions of graph theory and constraint programming. Section 4 illustrates our weighted constraint-based approach for generating uncertain graphs which optimizes the inter-vertices distances revealed by UPRs. In Section 5, we report a series of experimental results that show the validity and the efficiency of our approach on several uncertain graph generation instances.

2 Basic Graph Notions

To begin with, we shall expose the theoretical definitions and terminologies needed to cope with the graph theory framework. In what follows, $G{=}(V, E)$ will designate the graph comprising a non-empty finite set V of vertices related together with a collection, $E{\subseteq}\binom{V}{2}$ of edges, each of which is an unordered pair of vertices (*undirected*). If $\{i, j\} \in E$, then i and j are *adjacent*, and the edge $\{i, j\}$ is *incident* to both i and j. For conciseness, $\{i, j\}$ will be denoted by i-j. The *neighborhood* of vertex i is $N(i) = \{j | i\text{-}j \in E\}$. We set $n = |V|$ and $m = |E|$. The degree of a vertex $i \in V$, denoted by $deg(i)$, is the number of edges incident to i.

The integer sequence $\Delta = \langle\ deg(1), \ldots, deg(n)\ \rangle$ is known as the *degree sequence* of G. Giving a real-world example, in a computer network, a pair of computers is in E, if and only if, they are connected by some medium. If E contains multiple copies of an edge, that is, E is a multi-set, then G is a multi-graph.

A *chain* of G is a sequence of vertices $(1, \ldots, \ell)$ such that, i-$i{+}1 \in E, 1 \leqslant i < \ell$, where ℓ is the length of the chain. A *path* or in G is a chain $(1, \ldots, \ell)$ in which $i \neq j$, for all $1 \leqslant i, j < \ell$. A path in G of length ℓ is called an ℓ-path of G. Note that the number of vertices in a ℓ-path is equal to the number of edges.

G is said to be *connected* if, for every $i, j \in V$, with $i \neq j$, there is an path in G from i to j. The *topological distance* between vertices i and j, denoted $d(i, j)$, is the length of the shortest path separating i and j in G.

An *uncertain proximity relationship* of length ℓ between i, j of type $s \in \{0, 1\}$ will be denoted by $\text{upr}^s_{i,j,\ell}$. If $(s = 1)$ then the UPR is of type max-UPR, otherwise $(s = 0)$ the UPR is of type min-UPR. Therefore, $\text{upr}^s_{i,j,\ell}$ suggests that the distance separating i and j in G verifies $d(i, j) \leqslant \ell$, if $s = 0$ and $d(i, j) \geqslant \ell$, otherwise.

We can now pose the graph generation problem as follows: Given a set V of n vertices, a corresponding degree sequence Δ of n integers and a set of uncertain proximity relationships (UPRs), is there a graph G comprised of the vertices in V that realizes Δ and satisfies the UPRs? In other words, can we construct an edge set E such that Δ is the degree sequence of $G = (V, E)$ and the set of UPRs is satisfied?

3 Constraint Programming Background

Constraint Programming (CP) is a convenient paradigm to model and solve highly combinatorial problems, that draws on a wide range of techniques from Artificial Intelligence, Operations Research, Algorithms, and elsewhere. A constraint is a restriction, so, the basic idea in Constraint Programming (CP) is that the user states the constraints and the decision variables, and a constraint solver is used to solve them [5]. The best CP definition that can be given here is the one described by *Eugene Freuder*: *Constraint Programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem; the computer solves it.* The following equation describes the principle of the CP paradigm, consisting of two mainly interconnected components, *Modelling* and *Solving*:

$$CP = Modelling + Solving. \tag{1}$$

Modelling is the mental process of mapping an informal description of a problem into a formal description in a particular formal system [9]. After this mapping, the role of constraint solvers begins. They search the solution space systematically, as with backtracking or branch and bound algorithms, or use forms of local search which may be incomplete.

There are many real-life problems that require decision-making in the presence of constraints. Such problems can be modeled as Constraint Satisfaction Problems (CSPs). More formally, as described by Tsang [22], a CSP is a problem that can be defined by a triple (X, D, C), where X is a finite set of variables, each of which is associated with a finite domain D, and a set of constraints C that restricts the values that variables can simultaneously take.

A general-purpose constraint solver is used to solve the CSP by finding an assignment of values to all the decision variables which satisfy the constraints, while reducing the CSP problem to another one that is equivalent but more simpler by means of inference. There is a wide choice of solvers available in the

literature, which include Choco [14], Gecode [6], ECLiPSe [1], ILOG CP [11], Minion [8]. The role of constraint solvers is to assign a value to each variable satisfying all the constraints. Constraint solvers explore the solution space either systematically, as with backtracking or branch and bound algorithms, or use forms of local search [12]. In systematic search, the constraints are not merely treated as tests, but play an active role by helping to discover the inconsistencies early on, via the so-called constraint propagation. This latter process allows constraint solver to eliminate parts of the search space and then makes the search shorter.

The so-called ILOG CP solver, developed by IBM, is the one we use in this work. It represents a main component of the integrated development environment (IDE) CPLEX Optimization Studio. CPLEX Optimization Studio consists of the OPL modelling and scripting language for developing optimization models, and the IDE for running and testing optimization models. It also includes the optimization engines CPLEX Optimizer and CP Optimizer for solving Mathematical Programming (MP) and Constraint Programming (CP) models. In particular, the modelling language OPL allows to state Constraint Programming (CP), Logic Programming (LP) and Mixed Integer Programming (MIP), as well as combinations of all of them. A strong side of OPL is that it allows a mathematical encoding of the problem that is separate from the data, while offers several basic branching strategies such as depth-first search which is the default search strategy, and a number of other strategies including a best-first strategy.

3.1 Valued CSP formalism (VCSP)

Many real-life combinatorial problems can be naturally modeled and often efficiently solved using constraint programming. However, in some real situations, the classical CSP framework does not help. Indeed, it may occur that after having listed the desired constraints against the decision variables, there is no way to satisfy them all simultaneously. In this case, the instance is said to be over-constrained. In contrast, in other situations, all the constraints can be easily satisfied, which results in several solutions. Such solutions appear equally good, and there is no way to select among them. These scenarios often occur when constraints are used to formalize desired properties rather than requirements that cannot be violated. Such desired properties are not faithfully represented by classical constraints but should rather be considered as preferences whose violation should be avoided as far as possible. To cope with similar situations, classical constraints have been generalized to soft constraints, providing one way to allow constraint relaxation. Therefore, a general formalism, called Valued Constraint Satisfaction Problem (VCSP), has extended the CSPs to handle soft constraints.

The Valued Constraint Satisfaction Problem (VCSP) formalism can be used to solve optimization problems as well as over-constrained problems as they allow constraint relaxation. A VCSP is simply obtained by annotating the constraint of classical CSP with valuations denoting the costs of violations. More precisely, VCSP extends the CSP framework (X, D, C) by associating weights (or simply

costs) to tuples (combinations of values that can simultaneously be taken by variables). Generally, costs are specified by means of valuation structure defined as a triple $S = (E, \oplus, \preceq)$, where E is the set of cost values totally ordered by $\preceq$. The maximum and minimum costs are denoted by $\top$ and $\bot$, respectively. $\oplus$ is a commutative, associative and monotonic operation on E used to aggregate costs. The commutativity and associativity of $\oplus$ ensure that the evaluation of an instantiation does not depend of the order in which the evaluations are done. Monotony ensures that valuations cannot decrease when constraint violations become more important.

More formally, a VCSP is defined by a quadruplet (X, D, C, S) as follows [13]:

- $X = \{x_1, \ldots, x_n\}$ is a finite set of variables;
- $D = \{D_1, \ldots, D_n\}$ is a finite set of value domains so that D_k is the domain of x_k;
- $C = \{C_1, \ldots, C_m\}$ is a set of valued constraints. A constraint is a pair (X_k, θ_k), where $X_k = \langle x_{k_1}, \ldots, x_{k_r} \rangle \subseteq X$ is the constraint scope and $\theta_k : \prod_{x \in X_k} D_x \rightarrow E$ is a cost function;
- $S = (E, \oplus, \preceq)$ is a valuation structure.

A variable $x_i \in X$ must be assigned a value from its domain D_i. If a valued constraint is defined by a cost function whose domain is limited to $\{\top, \bot\}$ then it is a hard constraint, otherwise it is a soft constraint. The arity of a valued constraint is the size of its scope. The arity of a problem is the maximum arity over all its constraints. The valuation of an assignment t to a subset of variables $V \subseteq X$ is given by:

$$\Theta_P(t) = \bigoplus_{(X_t, \theta) \in C, X_k \subseteq V} \theta(t{\downarrow}X_k), \tag{2}$$

where $\downarrow$ denotes the projection of t on the variables of X_k. Therefore, an optimal global solution for a VCSP on n variables is an n-tuplet t, such that $\Theta_P(t)$ is minimal over all possible tuples. A VCSP has many variations which mainly differ mainly by the valuation structure. The choice of the most appropriate valuation structure depends on the characteristics of the problem to be formulated in terms of VCSPs. Accordingly, a classical CSP can be seen as a VCSP variation with a valuation structure $S = (E, \oplus, \preceq)$, where $E = \{0, +\infty\}$, with standard integer ordering $\preceq$ and $\oplus$ being the classical sum. The maximum and minimum costs are respectively 0 and $+\infty$. In addition to classical CSPs, other VCSPs variations have been proposed in the literature. Among them, Fuzzy CSPs (FCSP) [20], Probabilistic CSPs (Prob-CSP) [4], Possibilist CSPs (Pos-CSP) [21] and Weighted CSPs (WCSP) which is the VCSP variant chosen here to solve the problem of uncertain graph generation.

3.2 The Weighted CSP variant (WCSP)

In Weighted Constraint Satisfaction Problems (WCSPs), tuples come with an associated cost. This allows modeling optimization problems where the goal is

to minimize the total cost of the proposed solution. More formally, a Weighted CSP, is a Valued CSP where the valuation structure $S = (E, \oplus, \preceq)$ such that E contains a set of integers $\{0, \ldots, k\}$ with standard integer ordering $\leqslant$. The aggregation operation $\oplus$ is defined by $a \oplus b = min(k, a+b)$. Moreover, for any integer pair a and b such that $b \leqslant a$, substraction $\ominus$ is defined as follows:

$$a \ominus b \quad \begin{cases} a \text{ - } b, & \text{if } (a \neq k), \\ k, & \text{otherwise.} \end{cases} \tag{3}$$

Without loss of generality, we assume that there exits a zero-ary valued constraint $C_\varnothing = (\varnothing, \theta_\varnothing)$ dedicated to maintaining a lower bound for the minimum cost of the problem.

4 Uncertain Graph Generation as WCSP

As stated in the introduction, the goal of this work is to generate the most likely graph by assembling a set of vertices in all possible ways to ensure exhaustivity. The assembled graphs must satisfy hard structural constraints coming from the degree sequence, Δ, while optimizing a set of prescribed soft restrictions derived from uncertain proximity relationships data (UPRs). In what follows, we encode the likely graph to be generated as a simple adjacency matrix, and then we attempt to post hard and soft constraints ensuring Δ and UPRs.

Representing a graph with a given degree sequence is straightforward. We opted for the adjacency matrix and we tried to post constraints in order to enforce the available uncertain proximity relationships between certain vertex pairs. To reach this goal based on uncertain connectivity information, we used the Weighted CSP framework. A judicious choice that will allow us to express soft restrictions. Before we proceed, we notice that the n vertices of the degree sequence Δ will be indexed by the integers from 1 to n.

4.1 The Basic Model

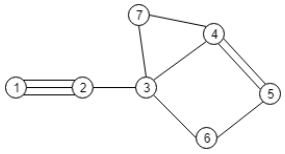
As stated above, a Weighted CSP is defined by a quadruplet (X, D, C, S). For the purpose of encoding constrained uncertain graphs, the variable set, X, will be composed of two subsets, which are the *Adjacency variables* and the *proximity variables*, let us refer to them as A and Λ; we therefore, have $X = A \cup \Lambda$.

The first variable subset, that is A, contains the variables that handle the adjacency of vertices in the graph. Hence, as shown in Figure 2, for a undirected multi-graph G with n vertices, we create $n \times n$ variables denoted by $a_{i,j}$, a variable for each ordered pair of vertices (i, j), which will be referred to as adjacency variables. Setting $a_{i,j}$ to a non-zero value means that there is one or more edges between vertex pair i and j; whereas, setting $a_{i,j}$ to 0 means that there is no edge connecting i and j. The value domain of each variable $a_{i,j}$ depends of the degrees, $deg(i)$ and $deg(j)$, relative to i and j.

The second subset of X, that is Λ, contains the decision variables that will encode UPRs. Recall that a UPR is a set of uncertain proximity relationship

$$\begin{pmatrix} 0&3&0&0&0&0&0\\ 3&0&1&0&0&0&0\\ 0&1&0&1&0&1&1\\ 0&0&1&0&2&0&1\\ 0&0&0&2&0&1&0\\ 0&0&1&0&1&0&0\\ 0&0&1&1&0&0&0 \end{pmatrix}$$

A



G

Fig. 2. An adjacency matrix A (left) giving the values of the adjacency variables corresponding to the degree sequence $\Delta = \langle 3, 4, 4, 4, 3, 2, 2\rangle$. The resulting target graph is G (right).

on vertex pairs for which the topological distance is not known but bounded by a given constant. More precisely, an element of UPR can be a max-UPR or a min-UPR. For any element of UPR (a min-UPR or a max-UPR), involving vertices i and j with bounding constant ℓ, we introduce a set $\Lambda_{i,j}$ of ℓ variables. The domain of each of these variables is set to the list of vertices that may correspond to the intermediate vertices on the shortest chain connecting vertex i to vertex j. Therefore, we have $\Lambda_{i,j} = \{\lambda_{i,j,1}, \ldots, \lambda_{i,j,\ell}\}$. The domain of each variable $\lambda_{i,j,k}$ could contain any vertex except i because the target graphs are loopless.

In what follows, we detail the constraint set C, of the Weighted CSP quadruplet, which defines the constraints related to the simultaneous processing of all structural restrictions, (*hard*) and (*soft*), in the problem.

4.2 Hard Constraints

Since the target graph comprises n vertices, is undirected and loopless, then we post the following hard constraints:

1. Since our uncertain graphs are undirected, we post $n(n\text{-}1)/2$ binary constraint. For each pair $i < j$, stating that $a_{i,j}{=}a_{j,i}$.
2. To forbid self-loops, we add the constraint $a_{i,i}{=}\ 0$, for each $1{\leqslant}i{\leqslant}n$.
3. Given the degree $deg(i)$ of vertex i, we can post a constraint $\sum_{j=1}^{n} a_{i,j}{=}deg(i)$, for each vertex i to ensure that the vertices have the correct degree.
4. If a pair of vertices can be related by more than one edge, this will result in a multigraph and then we may have $a_{\mathrm{i,j}} \geqslant 2$, for some vertex pairs (i, j).
5. For each UPR of length ℓ and type s, denoted $\mathrm{upr}^{s}_{i,j,\ell}$ involving the variable set $\Lambda_{i,j} = \{\lambda_{i,j,1}, \ldots, \lambda_{i,j,\ell}\}$, there is a hard restriction portion on the distance separating i and j in the target graph, depending on parameter s. Hence, $d(i, j) \leqslant \ell$ if UPR is a min-UPR ($s = 0$) and $d(i, j) \geqslant \ell$ otherwise ($s = 1$). This constraint can be expressed by the following two logical formulas:

$$(s = 0) \implies \mathrm{card}(\lambda_{i,j,k} \neq j) \leqslant \ell, \qquad k : 1, \ldots, \ell, \tag{4}$$

$$(s = 1) \implies \text{card}(\lambda_{i,j,k} \neq j) \geqslant \ell, \qquad k : 1, \ldots, \ell. \tag{5}$$

4.3 Soft Constraints: UPRs

Recall that a Uncertain Proximity Relationship (UPR) can be min-UPR or max-UPR. Expressing a UPR is a delicate task on account of the desired soft preferences (maximizing and/or minimizing topological distances). In addition to the two hard constraints expressed by equations (4) and (5), the encoding of this task requires adding many variables and constraints according to the type of prescribed proximity relationship $s \in \{0, 1\}$. Each UPR, either be min-UPR or a max-UPR, that constraint a pair of vertices (i, j), favors respectively short or long topological distance that separates i from j in the target graph. The goal here is to minimize or maximize that distance as much as possible with respect to parameters ℓ and s.

In the Weighted CSP formulation proposed below, we provide a solution to handle the uncertain nature of these preferences.

According to the type of UPR, min-UPR or a max-UPR, the idea consists in placing j, respectively, as close, or as far, as possible to vertex i through an objective based on cost functions. The objective value will vary depending on the number of intermediate vertices between i and j. This can be achieved as follows: Every $\text{upr}^s_{i,j,\ell}$ is defined by a subset of ℓ variables $\Lambda_{i,j}$. The domain of each $\lambda_{i,j,k}$ is set to $[1..n]\backslash i$. In addition, we post, for each k: 1, ...,ℓ-1, a ℓ-unary constraint $(\lambda_{i,j,k}, \theta^s_{i,j})$ for which the cost functions that define the weighted constraint are set as follows:

$$\theta^s_{i,j}(v) \begin{cases} s, & \text{if } \ v = j, \\ \\ 1\text{-}s, & \text{otherwise.} \end{cases} \tag{6}$$

To connect the set of variables $\Lambda_{i,j}$ to the adjacency variables, we post for each variable $\lambda_{i,j,k}$, $1 \leqslant k \leqslant \ell$-1, the following ternary constraint:

$$(\ \lambda_{i,j,k}{=}u \quad \wedge \quad \lambda_{i,j,\text{k}+1}{=}v) \implies a_{u,v}{\neq}0. \tag{7}$$

Also, the beginning of the separating chain entails the following constraint:

$$\lambda_{i,j,1}{=}u \quad \Rightarrow \quad a_{i,u}{\neq}0. \tag{8}$$

We can notice that the sum of costs of every variables subset $\Lambda_{i,j} \subset \Lambda$ depends on the topological distance separating i and j at hand. In fact, the sum of the costs over all the $\lambda_{i,j,k}$ is equal to the distance between i and j minus one. This coding is extended to obtain the most probably uncertain graph, that is, the one that minimizes the overall costs Φ, imposed by UPR set. This global objective can be expressed as follows:

$$\Phi(I) = \sum_{(\lambda_{i,j},\theta^s_{i,j})} \sum_{k=1}^{\ell} \theta^s_{i,j}(I(\lambda_{i,j,k})). \tag{9}$$

Note that, if the pair of vertices i and j are directly related in the uncertain graph, the global objective value is equal to zero.

Example 1. Let's take the problem of constructing G, a uncertain graph composed of eight vertices $\{1, \ldots, 8\}$ and defined by $\Delta = \langle 2, 2, 2, 2, 2, 2, 2, 2 \rangle$ as degree sequence. Let's focus on the following four min-UPR restrictions which state the following uncertain proximity relationship to minimize: $\{\text{upr}^0_{8,3,3}, \text{upr}^0_{7,4,4}, \text{upr}^0_{1,2,3}, \text{upr}^0_{1,6,4}\}$.

This problem can be outlined by the basic model as described in Subsection 4.1 with $\Lambda = \{\ \Lambda_{8,3},\ \Lambda_{7,4},\ \Lambda_{1,2},\ \Lambda_{1,6}\}$ as proximity relationship variables to minimize. The applicability of the proposed WCSP encoding on this graph instance, using the CP Optimizer library developed by IBM (https://www.ibm.com) which provides a constraint programming engine, performs well, giving good results. Figure 3 shows the obtained uncertain graph with its corresponding statistics. Three seconds as CPU time and 0.79 MB of ROM memory are required to generate G having a global objective $\Phi = 4$, knowing that 1025 graph candidates are obtained before the integration of min-UPRs restrictions.

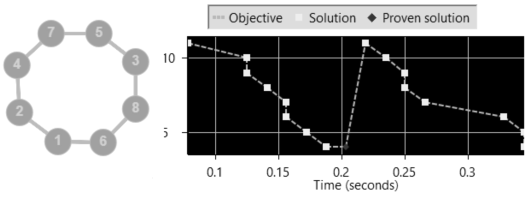


Fig. 3. min-UPR results (first Attempt).

Example 2. Let's again take the problem of constructing the most likely graph, with the same input data given in Example 1, which are the set of 8 vertices $\{1, \ldots, 8\}$ and the degree sequence $\Delta = \langle 2, 2, 2, 2, 2, 2, 2, 2 \rangle$. This time, our soft restrictions concern max-UPR and composed by four prescribed max-UPR, which are $\text{upr}^1_{8,4,5},\ \text{upr}^1_{7,3,5}$, $\text{upr}^1_{6,2,5},\ \text{upr}^1_{6,4,5}$.

As seen in Figure 4, by encoding the max-UPRs constraints described below based on set $\Lambda = \{\Lambda_{8,4},\ \Lambda_{7,3},\ \Lambda_{6,2},\ \Lambda_{6,4}\}$, we have gotten the most likely graph G among 1025 candidates. The experiment statistical results indicate that it takes 0.28 seconds of CPU running and 1.05 MB of Memory to generate G. The global objective in output,Φ, is equal to 12.

Although widely accepted, one limitation of our previous WCSP encoding to maximize UPRs is found. For every max-$\text{upr}^s_{i,j,\ell}$ with $s = 0$, it must be imposed that the path, taken into account by each subset of variables $\Lambda_{i,j}$, represents the shortest path connecting i to j in the target uncetain graph G. Therefore, for

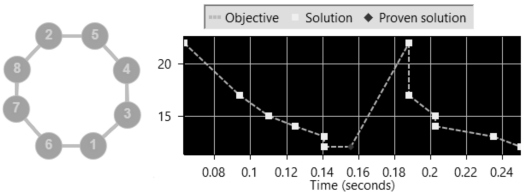


Fig. 4. max-UPR results (second Attempt).

each $\lambda_{i,j,k}$ the following hard constraint must be added to the set of constraints:

$$(\lambda_{i,j,k} = u \wedge u \neq j) \Rightarrow (\bigvee_{k'=1}^{k\text{-}1} a_{\Lambda_{i,j,k'},\Lambda_{i,j,k'+1}} = 0) \vee (a_{i,\lambda_{i,j,1}} = 0) \vee (a_{u,j} = 0). \tag{10}$$

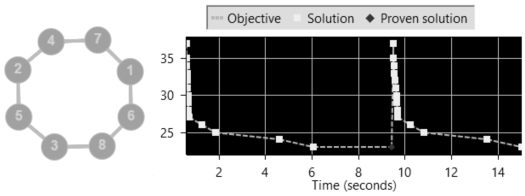


Fig. 5. max-UPR results (third attempt).

The results illustrated in Figure 5 confirm that it is necessary to add equation 10 below. Experiment statistics indicate that our solution for max-UPR takes 18.3 seconds of CPU time and 2.76 MB of ROM to effectively deal with the problem cited in the example 2 resulting graph G with a global objective $\Phi = 17$. To this end, we give a last example which takes into account uncertain proximity relationships information in a general way to ensure the generation of the most likely uncertain graph. However, valid results are achieved when applying our constraint-based encoding.

Example 3. Let's one more time take the same problem proposed in Example 1 and refined in Example 2. We focus here on both min-UPR restrictions of Example 1 and simultaneously on max-UPR restrictions defined as input data in Example 2. Hence, we have $\{\text{upr}^0_{8,3,3}, \text{upr}^0_{7,4,4}, \text{upr}^0_{1,2,3}, \text{upr}^0_{1,6,4},\}$ to minimize and $\{\text{upr}^1_{8,4,5}, \text{upr}^1_{7,3,5}, \text{upr}^1_{6,2,5}, \text{upr}^1_{6,4,5},\}$ to maximize.

This problem can be outlined by the basic model as described in Subsection 4.1 that is made of two subsets which are the *Adjacency variables* (A) and the

proximity variables (Λ). Thus, $\Lambda = \{\Lambda_{8,3},\ \Lambda_{7,4},\ \Lambda_{1,2},\ \Lambda_{1,6},\ \Lambda_{8,4},\ \Lambda_{7,3},\ \Lambda_{6,2},\ \Lambda_{6,4}\}$.

The applicability of the proposed WCSP encoding on this graph generation instance, using the CP Optimizer library developed by IBM (https://www.ibm.com) which provides a constraint programming engine, performs well, giving good results.

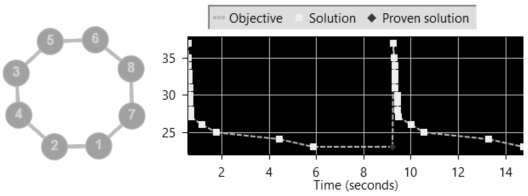


Fig. 6. UPR final results (fourth Attempt).

Figure 6 shows the obtained uncertain graph with its corresponding statistics. Three seconds as CPU time and 2.8 MB of ROM memory are required to generate G with a global objective $\Phi = 36$. Know that 1025 graph candidates are obtained before the integration of UPRs restrictions. The results of the experiment found clear support for the soft constraint-based uncertain graph generation.

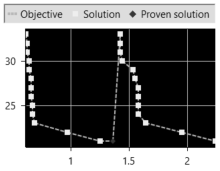
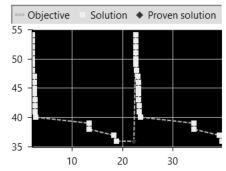
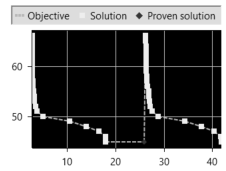
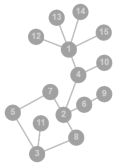
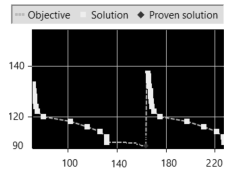
5 Experimental Results

To evaluate the effectiveness of our approach, we have tested it on several simple graph instances. All tests have been done on a computer (Intel Core i5-8250U CPU 1.60GHz 1.80GHz, 8 GB of RAM) running windows 10 (64 bit). During graph generation, we used the ILOG CP Optimizer solver developed by IBM (https://www.ibm.com), which provides a constraint programming engine.

Table 1 summarizes the experimental results obtained. For each instance of degree sequence Δ, presented in a simplified version such as $\langle 4_2, 3_3, 2_2, 1_1\rangle$ is equivalent to $\langle 4, 4, 3, 3, 3, 2, 2, 1\rangle$, we first reported the corresponding number of solutions ($\sharp$). We also included the uncertain proximity input data ($\text{upr}^s_{i,j,\ell}$) with their results in terms of memory usage (ROM) in MB and CPU time in seconds (CPU).

In addition, the column entitled (Cost Φ), is related to the cost function (global objective) of the Weighted-CSP formulation proposed in this paper. The last two remaining columns show the results, obtained by the program after considering UPRs, in terms of the most likely graph solution and respectively its experimental statistic. What is worth noting in our program is that, when we consider the uncertain proximity relationships in graph instances, we got the most likely graph admitting in its structure the most plausible topological distances over all pairs of vertices in interaction.

Table 1. Experimental results obtained from our Weighted CSP formulation.

Degree Sequence	$\mathfrak{u}$	$\mathbf{upr}^{s}_{i,j,\ell}$	**ROM**	**Cost** Φ	**CPU**	**Graph Solution**	**Statistics**
$\langle 4_2, 3_2, 2_2, 1_2\rangle$	2480	$\Lambda^0_{8,1,3}$, $\Lambda^0_{6,1,3}$ $\Lambda^0_{6,5,4}$, $\Lambda^0_{2,7,4}$ $\Lambda^1_{7,8,5}$, $\Lambda^1_{6,8,5}$ $\Lambda^1_{6,4,5}$, $\Lambda^1_{3,5,5}$	27.68	21	2.30		
$\langle 5_1, 4_1, 3_3, 2_2, 1_2\rangle$	11960	$\Lambda^0_{1,9,3}$, $\Lambda^0_{2,3,3}$ $\Lambda^0_{8,1,3}$, $\Lambda^0_{6,1,3}$ $\Lambda^0_{6,5,4}$, $\Lambda^0_{2,7,4}$ $\Lambda^1_{1,8,5}$, $\Lambda^1_{2,4,5}$ $\Lambda^1_{7,8,5}$, $\Lambda^1_{6,8,5}$ $\Lambda^1_{6,4,5}$, $\Lambda^1_{3,5,5}$	71.37	36	42.81		
$\langle 5_1, 4_1, 3_2, 2_3, 1_3\rangle$	21080	$\Lambda^0_{1,10,3}$, $\Lambda^0_{2,4,3}$ $\Lambda^0_{8,2,3}$, $\Lambda^0_{6,2,3}$ $\Lambda^0_{6,5,4}$, $\Lambda^0_{2,7,4}$ $\Lambda^1_{1,7,5}$, $\Lambda^1_{2,3,5}$ $\Lambda^1_{7,8,5}$, $\Lambda^1_{6,4,5}$ $\Lambda^1_{6,3,5}$, $\Lambda^1_{3,4,5}$	102.12	45	50.09		
$\langle 5_1, 4_1, 3_2, 2_4, 1_7\rangle$	62917	$\Lambda^0_{1,10,3}$, $\Lambda^0_{2,4,3}$ $\Lambda^0_{8,2,3}$, $\Lambda^0_{6,2,3}$ $\Lambda^0_{6,5,4}$, $\Lambda^0_{2,7,4}$ $\Lambda^0_{10,12,3}$, $\Lambda^0_{10,13,3}$ $\Lambda^0_{10,15,4}$, $\Lambda^0_{11,2,4}$ $\Lambda^1_{1,7,5}$, $\Lambda^1_{2,3,5}$ $\Lambda^1_{7,8,5}$, $\Lambda^1_{6,4,5}$ $\Lambda^1_{6,3,5}$, $\Lambda^1_{3,4,5}$	252.12	90	229.12		

The results obtained from a series of tests on several instances can only validate the proposed approach, allowing to generate uncertain graphs at a reasonable time. Recall that our purpose is not to generate all graphs as quickly as possible, but to encode a soft constraint that can be used with an external search procedure on problems with side constraints, to consider proximity relationships preferences.

In particular, we have not considered connectivity or symmetry, and there are many symmetries in these problems. We view symmetry as a separate feature that can be maintained independently. Paper [19] describes some explorations of constraint programming in graph generation, concentrating on forcing the graphs to be connected.

6 Conclusions

In this paper, we used a variant of the Valued CSP approach to address the graph generation problem under uncertainty. In particular, we have presented a Weighted CSP formulation that allowed the integration of hard and soft numerous structural constraints in a single framework. This study has shown the advantage of adding fuzzy information or preferences, found in uncertain proximity relationships between pairs of vertices, to select the most probably graphs. From a practical point of view, we used the programming library named ILOG CP. The experimental study showed that our constraint-based solution is successful in coping with the hard problem of uncertain graph generation. As a future work, we plan to take account of more complex uncertain data like fuzzy degrees or uncertain subgraphs. Such data are frequently encountered in real-world situations and may be very helpful in further limiting the set of candidate solutions. To this end, once enhanced with the capability of handling more uncertain data, our solution could be more useful in several real-world applications.

References

1. Apt, K.: Constraint logic programming using ECLiPSe. Cambridge University Press, Cambridge (2007)
2. Elyashberg, M.: Computer-based structure elucidation from spectral data : the art of solving problems. Springer, Heidelberg (2015)
3. Erdös, P.: Graph theory and probability. Canadian Journal of Mathematics 11, 34–38 (1959)
4. Fargier, H., Lang, J.: Uncertainty in constraint satisfaction problems: A probabilistic approach. pp. 97–104 (2005)
5. Freuder, E.C., O'Sullivan, B.: Grand challenges for constraint programming. Constraints 19(2), 150–162 (Apr 2014)
6. G Tack, M.L., Schulte, C.: Gecode: generic constraint development environment. http://www.gecode.org/ (2009)
7. Gent, I.P., Macintyre, E., Prosser, P., Smith, B.M., Walsh, T.: Random constraint satisfaction: Flaws and structure. Constraints 6(4), 345–372 (Oct 2001)
8. Gent, I., Jefferson, C., Miguel, I.: Minion: A fast, scalable, constraint solver. In: ECAI 2006: 17th European Conference on Artificial Intelligence. pp. 98–102. Frontiers in Artificial Intelligence and Applications, IOS Press (2006), eCAI 2006: 17th European Conference on Artificial Intelligence, Riva del Garda
9. Hnich, B., Hnich, B.: Function variables for constraint programming. Tech. rep. (2003)
10. van der Hofstad, R.: Random Graphs and Complex Networks. Cambridge University Press (2016)
11. IBM: Ibm ilog solver v6.7. https://www.ibm.com/analytics/cplex-cp-optimizer, accessed: 2019-05-02
12. Jussien, N., Debruyne, R., Boizumault, P.: Maintaining arc-consistency within dynamic backtracking. In: Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming. pp. 249–261. CP '02, Springer-Verlag (2000)

13. Kolmogorov, V., Krokhin, A.A., Rolinek, M.: The complexity of general-valued csps. CoRR abs/1502.07327 (2015)
14. Laburthe, F.: Choco: Implementing a cp kernel. CP00 Post Conference Workshop on Techniques for Implementing Constraint Programming Systems (TRICS) (01 2000)
15. Mihail, M., Vishnoi, N.K.: On generating graphs with prescribed vertex degrees for complex network modeling. In: 3rd Workshop on Approximation and Randomization Algorithms in Communication NEtworks (2003)
16. Milo, R., Kashtan, N., Itzkovitz, S., Newman, M.E., Alon, U.: On the uniform generation of random graphs with prescribed degree sequences. arXiv preprint cond-mat/0312028 (2003)
17. Omrani, M.A., Naanaa, W.: A constrained molecular graph generation with imposed and forbidden fragments. In: Proceedings of the 9th Hellenic Conference on Artificial Intelligence - SETN16. ACM Press (2016)
18. Omrani, M.A., Naanaa, W.: Graph generation with imposed and forbidden patterns. In: IADIS International Conference Applied Computing - AC'17. pp. 179–186. IADIS Digital library (2017)
19. Prosser, P., Unsworth, C.: A connectivity constraint using bridges. In: Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy. pp. 707–708. IOS Press, Amsterdam, The Netherlands, The Netherlands (2006), http://dl.acm.org/citation.cfm?id=1567016.1567174
20. Ruttkay, Z.: Fuzzy constraint satisfaction. In: Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference. IEEE
21. Schiex, T.: Possibilistic constraint satisfaction problems or "how to handle soft constraints?". CoRR abs/1303.5427 (2013)
22. Tsang, E.: Foundations of Constraint Satisfaction. Books On Demand (2014)
23. Yu, K., Wang, X., Li, Q., Zhang, X., Li, X., Li, S.: Individual morphological brain network construction based on multivariate euclidean distances between brain regions. Frontiers in Human Neuroscience 12 (may 2018), https://doi.org/10.3389/fnhum.2018.00204

Electronic edition
Available online: http://www.rcs.cic.ipn.mx