Advances in Applications of Artificial Intelligence and Optimization

Research in Computing Science

Series Editorial Board

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 104,** octubre 2015. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. : 04-2005-121611550100-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor responsable: *Grigori Sidorov, RFC SIGR651028L69*

Research in Computing Science is published by the Center for Computing Research of IPN. **Volume 104**, October 2015. Printing 500. The authors are responsible for the contents of their articles. Printed in Mexico City, in the IPN Graphic Workshop – Publication Office.

Volume 104

Advances in Applications of Artificial Intelligence and Optimization

Obdulia Pichardo Lagunas (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2015

ISSN: 1870-4069

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

Indexed in LATINDEX, DBLP and Periodica

Printing: 500

Printed in Mexico

Editorial

This volume of the journal "Research in Computing Science" contains selected papers on the research areas related to artificial intelligence and its applications to a variety of tasks.

The papers were carefully chosen by the editorial board on the basis of the at least two reviews by the members of the reviewing committee or additional reviewers. The reviewers took into account the originality, scientific contribution to the field, soundness and technical quality of the papers. It is worth noting that various papers for this special issue were rejected.

The papers of this volume are related to several areas of application of artificial intelligence. First of all, there are tasks related to optimization: scheduling of the detection-and-rescue operations, task planning for a multicomputer system, bi-objective purchasing, and pickup and delivery problem. Other papers are related to application of neural networks (to broiler's growth and particle tracking velocimetry analysis), fuzzy systems (to grade assignment and digital signal distribution) and cellular automata (to compression problem). Finally, electronic component of an intelligent hospital bed is described.

I would like to thank Mexican Society for Artificial Intelligence (Sociedad Mexicana de Inteligencia Artificial) and MICAI 2015. Also, I am grateful to Polytechnic University of Morelos (Upemor), Tecnologico de Monterrey Campus Cuernavaca, Electrical Research Institute (IIE) and the National Center for Research and Technology Development (CENIDET) for their support during preparation of this volume.

The entire submission, reviewing, and selection process, as well as preparation of the proceedings, were supported for free by the EasyChair system (www.easychair.org).

Obdulia Pichardo Lagunas
October 2015

Table of Contents

Page

A Fast Algorithm for Scheduling Detection-and-Rescue Operations Based on Data from Wireless Sensor Networks 9
Boris Kriheli, Eugene Levner, Michael Bendersky, and Eduard Yakubov

Objective Analysis in Task Planning and Allocation of Multicomputer Systems 23
A. Velarde M.

Solution of a Bi-Objective Purchasing Scheduling Problem with Constrained Funds using Pareto Optimization 41
José Francisco Delgado-Orta, Laura Cruz-Reyes, Alejandro Palacios-Espinosa, and Christian Ayala-Esquivel

The Pickup and Delivery Problem: a Many-objective Analysis 51
Abel García-Nájera and Antonio López-Jaimes

Fuzzy System for Grade Assignment in Competence Assessment Based Educative Models 61
Fabio Tomás Moreno Ortiz, Antonio Hernández Zavala, and Omar Rodríguez Zalapa

Improving Performance of Particle Tracking Velocimetry Analysis with Artificial Neural Networks and Graphics Processing Units 71
Rubén Hernández Pérez, Ruslan Gabbasov, and Joel Suárez Cansino

A Simulation of the Broiler Growth Rate Using Artificial Neural Networks 81
Raquel Salazar, Fernando Rojano, and Abraham Rojano

Methodology for the Model for Failure Prediction in a Digital Signal Distribution 91
José Cruz Ramos-Báez, María de Lourdes Martínez-Villaseñor, and Dafne Rosso-Pelayo

A Secure Compression Scheme for Real-time Applications Using 2D-WT and Cellular Automata 103
M. T. Ramírez-Torres, J. S. Murguía, M. Mejía Carlos, and J. A. Aboytes-González

Electronic System of an Intelligent Machine: the Case of an Assistive Bed Device 115
E. Vázquez-Santacruz, C. Morales-Cruz, and M. Gamboa-Zúñiga

A Fast Algorithm for Scheduling Detection-and-Rescue Operations Based on Data from Wireless Sensor Networks

Boris Kriheli[1,2], Eugene Levner[1,2], Michael Bendersky[1*], and Eduard Yakubov[1]

[1] Holon Institute of Technology, Holon, Israel

[2] Ashkelon Academic College, Ashkelon, Israel

[*]Corresponding author: michaelb@hit.ac.il

Abstract. The need for the search, detection and rescue of disaster survivors arises in many emergency situations in military and civil applications. Suppose a number of people are trapped in ruins after an earthquake or tsunami. Their medical condition depends on their location, detection time and the time of the rescue operation. In order to efficiently detect and perform the needed rescue operations, a network of wireless sensors is used which provide acoustic, seismic, electromagnetic, gravimetric and other information. The information is processed automatically to yield prior probabilities of location and expected rescue times for each potential target. The acquired information from the sensors is imperfect because under extraordinary and severe circumstances, two types of errors may occur: (i) a "false-negative detection test" – it is a case when a target is overlooked during the test; and (ii) a "false-positive detection", or "false alarm" – when a not-a-target location is wrongly classified as a sought target. Therefore, non-zero probabilities of overlooking a hidden target and a "false alarm" exist. We suggest a two-phase solution to the problem of scheduling detection and rescue operations. First, the disaster area is divided into sub-areas and available rescue teams and sensors are assigned. Second, a schedule is found for the rescue teams to perform the rescue operations (in parallel). We seek to find the best coverage of the disaster sub-areas served by rescue teams and to schedule the search-and-rescue operations in each sub-area while minimizing the search-and-rescue time and maximizing the number of saved lives within a given search time limit. The problem is formulated as a non-standard two-stage assignment / scheduling problem and a fast combinatorial real-time algorithm is suggested.

Keywords: disaster management, detection-and-rescue problem, wireless sensor network, imperfect inspections, best coverage, scheduling, fast on-line algorithm

1 Introduction

The need for search, detection, and rescue (DAR) of disaster survivors arises in many emergency situations in military and civil applications. Suppose that a large number of people are trapped in ruins after an earthquake, a tsunami wave, or a terrorist attack. Their medical condition and survival probabilities depend on their location, the time

needed to locate them and the evacuation (rescue) time. For DAR operations to be efficient, a computer-aided network of wireless sensors of different types is used which provide acoustic, seismic, electromagnetic, gravimetric and other information about the targets (see [6, 11, 17]).

Real-time monitoring and quick response are the most essential requirements in the design of an emergency response system. Different types of sensors are used together and the collected information is incorporated into a wireless sensor networks (WSNs) thus allowing for the communication between both sensors and human rescue teams. For example, temperature and movement-detection sensors are used to monitor the location of people, satellite cameras can track the spread of the disaster and depict the disaster area map while ultrasonic sensors measure the range to targets in the environment and report dynamic changes of maps due to the changes of built structures through destruction of debris. The use of such heterogeneous tools must be supported by innovative planning or scheduling tools in order to exploit and integrate the capabilities of each sensor and provide an optimal use of all available resources.

In this work, we consider scenarios that require locating and identifying multiple stationary and dynamic targets. We assume the presence of a relevant communication infrastructure enabling the command center and the rescue teams to continuously exchange information. In order to plan an effective team deployment over the search area, it is necessary to rapidly gather as much information as possible about the targets and the area, and use this information to define joint search-and rescue mission plans. A mission plan consists of a sequence of actions to be performed by an agent for a certain time duration as defined by environmental factors and geographical locations.

Mission planning is modeled as a mixed integer linear programming problem (MILP) in which the model simultaneously allocates predefined sub-areas of a disaster area to be explored and specifies the schedule of the actions that each agent should follow. The resulting plans guarantee optimal results for the search activities. A number of constraints are included to model cooperation and connectivity relationships among agents (sensors and human rescue teams). For example, at the beginning of the search process, the agents are uniformly spread over the area, while in later stages they are focused on specific subareas according to importance.

Initially, the data from the sensors is collected by the network and integrated to define prior probabilities of location, the damage scale and expected rescue times for each potential target. The problem presented in this paper can be partitioned into two stages. First, the disaster area is divided into sub-areas and available rescue teams are assigned to each sub-areas in which they will perform in parallel their DAR missions. At the second stage, a detailed schedule of operations is planned ahead for each rescue team. Notice that the detection-and-rescue operations at the second stage are implemented simultaneously by several rescue teams. The goal is to find the best coverage of the disaster area by mobile rescue teams and to schedule the search-and-rescue operations in each sub-area in order to minimize the search-and-rescue time and maximize the number of saved lives within the given limits of the search-and-rescue time. This problem is a natural extension of similar search-and-rescue problems studied in [7-9, 11, 13, 16].

The automatic information-gathering system gathers information from sensors scattered over a geographical region to help the rescue teams to find the targets in minimum time. The inspections are imperfect because under uncertain environmental

circumstances, two types of errors may occur: (i) a "false-negative" detection test – a target object is overlooked during the test; and (ii) a "false-positive" detection or a "false alarm", which wrongly classifies a clean location as a sought target. Hence, non-zero probabilities of overlooking the hidden target as well as that of a "false alarm" exist. We propose to model the DAR problem as a scheduling problem involving several search teams working in parallel, and subject to time/budget and probabilistic constraints. The general problem of selecting the best schedule is NP-hard thus, the proposed solution is an approximation or an "almost-optimal" solution.

The remainder of the paper is organized as follows. In Section 2, we provide a review of related works and approaches for using smart sensor networks to detect/rescue hidden objects while focusing on detecting and rescuing of human survivors. In Section 3, we provide a formal formulation of the problem and propose a mathematical model. In Section 4 we propose a solution using a fast algorithm (without significant computational load). A numerical example is given in Section 5 and Section 6 contains a summary along with future research directions.

2 Related Work

Planning of search-and-detection operations has been researched thoroughly in the area of operational research and artificial intelligence. The pioneer work done by Bernard Koopman done during World War II aimed to provide efficient methods for detecting hidden submarines. See [2], [15] and [19] for a detailed survey and the bibliography of the discrete search literature. In recent years, the problem of planning and scheduling of detection operations has become critical in light of increasing growth of natural and human-made disasters and the usage of a WSN has become popular. A WSN is an advanced technology for collecting diverse data from multiple sensors. A typical WSN system is distributed within the sensor field and consists of a number of sensor nodes, such as seismic, acoustic and magnetic anomalies. See [1] for a comprehensive survey regarding the main factors influencing the WSN design. The WSN collects thousands of raw data and works as a centralized or decentralized fusion system (see [18]). In the centralized case, the data is collected by individual sensors and sent through the sink node to a central dedicated fusion node, task manager node for processing while in the decentralized case the information is collected and analyzed by a set of autonomous devices.

We consider a situation where the basic functions of the WSNs are to monitor and control environmental parameters related to the detection-rescue and collectively transfer the data obtained through the network to a central location. In WSNs, the mobile agents are added into the system to improve its performance and act as automatic carriers of data. [4] provides more examples and details of modern applications of WSNs including battlefield surveillance, detection of enemy intrusion and detection and rescuing of hidden targets. Many search-planning algorithms are based on a cellular partitioning of the disaster area (see [7] and the references within). In [3], a multi-scale grid is used for representing the environment. [10] studied the usage of UAVs (unmanned aircraft vehicles) for DAR missions. Other research has studied the use of autonomous teams of robots for DAR (see [14]). MILP models has been successfully

used in search planning problems and mission assignment ([5]). An advantage of a MILP formulation is that, given exact input data, an optimal solution can be provided. Compared to latter works, we put an emphasis on the parallel work of several search-and-rescue teams and solve both task allocation and scheduling problems.

To conclude, we consider a different objective function and corresponding mathematical formulations of the problem. This problem is a natural extension of similar search-and-rescue problems studied in [7-9, 11, 13, 16]. Our contribution is threefold: (i) a new two-stage decomposition methodology partitioning the initial mission planning problem into an assignment and scheduling components aimed to enhance the efficiency of DAR missions performed by several teams of networked agents (sensors and human teams); (ii) a novel generalized assignment problem (used at the first stage) including disjunctive and resource constraints in the context of DAR missions; (iii) a novel scheduling problem (of the second stage) and the design of a new fast scheduling algorithm.

3 Problem Description and Mathematical Formulation

As said above, the goal of the present study is two-fold. First, we find the best coverage of the disaster area by a mobile rescue teams and, second, we optimally schedule the search-and-rescue operations in each sub-area in order to minimize the search-and-rescue time and maximize the number of saved lives within the given limits of the search-and-rescue time. At the first stage, the disaster area is divided into sub-areas and available rescue teams are assigned to the disaster sub-areas in which they will perform in parallel their DAR missions. At the second stage, a detailed schedule of operations is planned ahead for each rescue team. The DAR operations at the second stage are implemented simultaneously by several rescue teams.

3.1 The coverage of the Disaster Area

The planning process starts with discretizing the known disaster area into a set of squared environmental cells representing the spatial elements that should be served by the available WSN and the rescue teams. Without loss of generality, the disaster area is decomposed into a uniform cell grid, the cells' set being denoted by A, $|A| = n$. In this simple, but effective scenario, the disaster area is uniformly partitioned in as much equal sub-areas as possible within the available time reserve and personnel resources. In real-world scenarios, the disaster area is usually irregular and cluttered; we represent the non-uniform effect on both mobility/effectiveness of the rescue teams, on the one hand, and sensing of the WSN throughout the field, on the other. For this purpose, we assume that the total number of available human teams is known and equals M while the total number of available sensors is denoted by S.

We are now ready to formulate the area coverage-planning problem as a generalized assignment problem with resource and precedence constraints. As will be seen next, the problem is a special case of the MILP class.

Define, f_{hj} - performance effectiveness function, corresponding to a human rescue team h, $h=1,2..,m$, assigned to perform the detection-and-rescue missions in cell j, $j=1,2..,n$. f_{hj} is characterized by the expected number of detected/saved human lives during performing the DAR mission (in cell j) which, in turn, depends on the local sub-area characteristics, the agent skills and the search time. Therefore, the entire performance of the mission planning for the effective coverage by the agents strictly depends on the allocation of the agents to the sub-areas. These characteristics are estimated by the rescue/evacuation manager based on the data provided by the WSNs. This issue is particularly relevant in the case of the heterogeneous sensors and teams working simultaneously ("in parallel"). We take into account disjunctive conditions stating that each cell can be served by a human team and/or by an automated device, like a mobile robot or an unmanned aerial vehicle UAV. Precedence relations are imposed according to which, in any cell, first the sensors measurements are to be performed, after which human teams are able to start their rescue mission.

In addition, define B and T the total budget at hand and the total time for the DAR operation respectively and by c_{hj}, t_{hj} and d_{hj} the cost, the required time, and sensor cost required to perform a DAR in cell j by team h. Also, let k_j be the number of rescue teams in sub area j (can be larger than 1). Finally, let X_{ij} and Y_{sj} be binary variable defined as follows:

$$X_{ij} = \begin{cases} 1 & \text{rescue team } i \text{ assgined to cell } j \\ 0 & \textit{else} \end{cases}$$

and

$$Y_{sj} = \begin{cases} 1 & \text{sensor } s \text{ is assgined to cell } j \\ 0 & \textit{else} \end{cases}$$

Then the constrained multi-agent coverage problem (CMACP) can be formulated as presented in (1)-(7).

$$\max \sum_{h=1}^{m} \sum_{j=1}^{n} f_{hj} \cdot x_{hj}$$

subject to

$$\sum_{h=1}^{m} \sum_{j=1}^{n} x_{hj} \leq M \qquad (1)$$

$$\sum_{h=1}^{m} \sum_{j=1}^{n} c_{hj} \cdot x_{hj} \leq B \qquad (2)$$

$$\sum_{h=1}^{m} \sum_{j=1}^{n} d_{hj} \cdot y_{hj} \leq C \qquad (3)$$

$$\sum_{h=1}^{m}\sum_{j=1}^{n} t_{hj} \cdot x_{hj} \le T \quad (4)$$

$$\sum_{h=1}^{m} x_{hj} \le k_j \quad (5)$$

$$\sum_{h=1}^{m} x_{hj} = 1 \quad (6)$$

$$x_{hj} \le y_{kj}, \forall k, h, j \quad (7)$$

$$x_{ij}, y_{kj} \in \{0,1\}, \forall k, h, j, i$$

The first constrain is immediate since there is a total of M teams while (2)–(4) represent the total budget of the human teams, the total budget of the WSN system and the total time given to perform the rescue operation. (5) follows from the definition of k_j and (6) corresponds to the fact that every rescue team should be assigned to a sub area. Since sensor measurement must precede the rescue mission in all sub-areas, we have (7).

The presented generalized assignment problem with precedence and resource constraints is a special class of the MILP problem. We have used the MILP solver (a commercial optimization package called GAMS) and obtained an optimal solution in under 5 minutes for small and medium size instance $(m \le 20, n \le 100)$.

3.2 The Scheduling of Detection-and-Rescue Operations in Each Sub-Area

After completing phase 1, i.e., assigning the agents to the different sub area (cells) we can continue to phase 2 and define the sequence of detection-and-rescue operations. When defining the sequence of operations, the most important goal is to maximize the number of saved human survivors (targets) and then protection of property.

We consider the following scenario. The targets are clustered, that is, located in groups of linked sites (cells) where the targets in each cluster are processed simultaneously while each group is inspected and rescued non-stop from one cluster to the other. Since the coverage of the area into the cells is sufficiently fine-grained, we may assume that each cell contains one target (at the most). If the number of rescue teams is K (known in advance since it is defined by the resource constrain), a cluster of K targets can be processed simultaneously. At the first step we determine the cluster of size K that contains the maximum of expected number of potential survivors in its cells (and will be processed by K rescue teams). After the first cluster is processed, the K teams are assigned to the next cluster (again, containing K targets). The targets are detected and rescued until the given time reserve T_0 is exhausted, or all targets are discovered and saved. The problem is to efficiently detect and rescue the targets so as to maximize the possible performance (the number of saved lives) of the detection-and rescue mission.

For simplicity, we consider the following special case of scheduling the human rescue teams, the scheduling of automated search teams and heterogeneous smart sensors being handled along the same line. Any inspection of any cell (either containing the target or not) is imperfect. This means that a prior probability α_i of a false alarm and a prior

probability β_i of overlooking the target are given. This implies that each cell may be examined more than once. It follows that a detection sequence will be finite but repetitions of the same cells are possible. Each rescue team performs a set of sequential operations in order to identify and rescue the target. The times and expected efficiency of lifesaving during the operations being given, the goal of the detection and rescue process is to determine a search strategy which the rescue team employs to locate and rescue the maximum number of targets within the given reserve of detection-and-rescue time.

A disaster area contains m squared sub-areas. Each are contains mi potential target locations, $m_i<m$, $i=1,2,...,N$ and is characterized by the following known parameters:

- p_i - prior probability that location i contains the target;
- α_i - prior probability of a "false alarm" , or a false-positive outcome, the conditional probability that an inspection declares that a target is found in cell i whereas, in fact, this location does not contain a target;
- β_i - prior probability of overlooking, or a false-negative outcome, the conditional probability that an inspection declares that location i has no target but in fact it has;
- t_i - expected time to inspect cell i by one of the teams
- c_i - expected number of potential survivors in cell i.

Each sequential inspection strategy specifies a finite sequence

$$s = \langle S[0], s[1], s[2], \ldots, s[n], \ldots \rangle$$

where s[n] denotes the cluster's index which is inspected by K parallel teams at the n^{th} step of sequence s, $s[n] \in \{1,2,...\}$ and s[0] is an initializing sub-sequence of locations which will be defined below.

Given the above input data, the optimal search scenario is specified by the following conditions:

i. the clusters are inspected sequentially;
ii. for any search strategy and any cluster, the outcomes of inspections are independent;
iii. the stopping rule is defined as follows:

For any integer h, define - a_{ih} - the conditional probability that cluster i contains the target given that it contains the target in h inspections. a_{ih} depends on the given p_i, β_i. In addition, let H_i ("height") be the minimal positive integer such that $a_{ih} \geq CL$ where CL is a priori given confidence level. It should be noted that all of the H_i's can be computed by the rescue manager before the search process starts.

Given a sequence *s* of inspections, the search ends when either the search-rescue time reserve expires, or, at some step, all clusters return the outcome of "*the target is claimed to be in location* i *for the* H_i^{th} time in s".

For a given sequence s, we shall use the following notation:

- $T_{s[n]} = T\left(s[n], s\right)$ – time (accumulated) spent to detect the target in $s[n]$ on the n^{th} step of strategy s;
- $T_{s[n]} = \max_m t_{s[m]}$ where maximum is taken over all the teams working in cell;
- $P_{\underline{S}[n]}$-the probability that targets, located in cell $s[n]$, are detected $H_{s[n]}$ times before the n^{th} step of strategy s. $H_{s[n]}$ and $P_{\underline{S}[n]}$ depend on α_i and β_i, and guarantee required confidence level; in practice, $H_{s[n]}$ is equal 1 or 2. This concept and its relationship with the confidence level CL is described below.
- $c_{s[n]}$ – lifesaving efficiency in location (cluster) $s[n]$.

The expected (linear) total lifesaving efficiency, $F(s)$, is defined as follows:

$$\max \qquad F(s) = \sum_{n=1}^{\infty} P_{s[n]} c_{s[n]} T_{s[n]}$$

$$\text{subject to} \qquad T_{s[n]} \le T_0 \quad .$$

In the above notation, the stochastic scheduling problem is to find a sequence s^* that maximizes the expected efficiency $F(s)$ subject to the search time reserve.

One should note that the above formulation gives rise to three special cases: when $\alpha_i = \beta_i = 0$ for every i, the problem is known as the perfect inspections problem researched in the finite-horizon scheduling literature. If all α_i's are zero but $\beta_i \neq 0$ for all i's we have the false-negative inspections and if $\alpha_i \neq 0, \beta_i = 0$ for every i we have the false-positive inspections. In addition, when the problem is minimization, and the time reserve constrain is relaxed, the model is much simpler and can be solved using a proposed method in [12].

4 Problem Analysis and Algorithm

We begin by defining

$B_i = \{\text{Inspection declares that cluster } i \text{ has a target}\}$,
$C_i = \{\text{Cluster } i \text{ really contains the target}\}$

and using the notations in Section 3 we have $p_i = P(C_i)$, $\alpha_i = P\left(B_i \middle| \overline{C_i}\right)$ and $\beta_i = P\left(\overline{B_i} \middle| C_i\right)$.

Now, the probability that the target is discovered in cell i, defined f_i, is equal to

$$f_i = P(B_i) = P(C_i)P(B_i / C_i) + P(\overline{C}_i)P(B_i / \overline{C}_i) = p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i,$$

while the probability to correctly detect the target in cell i within a single inspection is equal to

$$P(C_i / B_i) = \frac{P(C_i)P(B_i / C_i)}{P(C_i)P(B_i / C_i) + P(\overline{C}_i)P(B_i / \overline{C}_i)} = \frac{p_i \cdot (1-\beta_i)}{p_i \cdot (1-\beta_i) + (1-p_i)\alpha_i}$$

Theorem 1. Given a sequence s, the conditional probability a_{ih} - the probability that location i contains the target given the probability it contains the target in h inspections is given by:

$$a_{ih} = P\left(C_i / B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{(h)}\right) = \frac{P(C_i) \cdot P\left(B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{(h)} / C_i\right)}{P\left(B_i^{(1)} \cap B_i^{(2)} \cap \ldots \cap B_i^{h}\right)}$$

$$= \frac{p_i \cdot (1-\beta_i)^h}{p_i \cdot (1-\beta_i)^h + (1-p_i)\alpha_i^h}$$

Corollary. Given a predetermined confidence level CL for the probability a_{ih} defined above, H_i is the minimal integer satisfying

$$a_{ih} = \frac{p_i \cdot (1-\beta_i)^{H_i}}{p_i \cdot (1-\beta_i)^{H_i} + (1-p_i)\alpha_i^{H_i}} \geq CL \text{ for any } i.$$

Inspections in each cluster M_i are done in parallel by different rescue teams in pre-specified times. The search strategy is a finite sequence of clusters (more exactly, their index), where, at step n, the cluster $s[n]$ is inspected and rescued:

$$s = \langle S[0], s[1], \ldots, s[n], \ldots \rangle.$$

Denote by $s[k,n]$ the number of a cell in cluster $s[n]$ inspected at the nth step of strategy s. Denote by $s^*[k,n]$ the total number of inspections of cell $s[k,n]$ counting from the first inspection up to its inspection in cluster $s[n]$ inspected at the n^{th} step of strategy s. Notice that $s^*[k,n]$ can be easily computed for all k as soon as the

sequence s is known up to its n^{th} step. Denote by $c_{s[k,n]}$ the rescue effectiveness assigned to cell $s[k,n]$. Let $T_{s[k,n]}$ be the time spent for inspection of all cells of all the clusters in strategy s up to location $s[k,n]$,

$$T_{s[k,n]} = \sum_{i=1}^{N} t_i (H_i - 1) + \max T_{[m,s]} + \sum_{i=1}^{k} t_{s[k,n]},\ n \geq 1$$

The search effectiveness attributed to strategy s is

$$F(s) = Exp\left(R(s)\right) = \sum_{n=1}^{\infty} \sum_{k=1}^{j_k} R_{s[k,n]}(s) \cdot P\left(R(s) = R_{s[k,n]}(s)\right) =$$

$$= \sum_{n=1}^{\infty} \sum_{k=1}^{j_n} c_{s[k,n]} T_{s[k,n]} \begin{pmatrix} s^*[k,n]-1 \\ H_{s[k,n]}-1 \end{pmatrix} \left(1 - f_{s[k,n]}\right)^{s^*[k,n]-H_{s[k,n]}} \cdot f_{s[k,n]}^{H_{s[n]}}$$

Theorem 2. The strategy s^* is an optimal strategy for the max-efficiency search problem iff the ratios

$$Q_{s[n]} = \frac{\sum_{k=1}^{j_n} c_{s[k,n]} \cdot P_{s[k,n]}}{T_{s[n]}} = \frac{\sum_{k=1}^{j_n} c_{s[k,n]} \cdot P_{s[k,n]}\left(s^*[k,n]\right)}{T_{s[n]}}$$

$$= \frac{\sum_{k=1}^{j_n} c_{s[k,n]} \cdot \begin{pmatrix} s^*[k,n]-1 \\ h_{s[k,n]}-1 \end{pmatrix} \left(1 - f_{s[k,n]}\right)^{s^*[k,n]-h_{s[k,n]}} \cdot f_{s[k,n]}^{h_{s[k,n]}}}{T_{s[n]}}$$

are arranged in non-decreasing order of the magnitude.

The proof is by the interchange argument and skipped here.

5 Example

Consider the problem of searching a target in a stochastic setting described in [3]. The rescue team has limited time (to perform the search and rescue operation) and limited memory (the only saved information is the information on how many times a target has been detected in each visited cell up to a current step in the search sequence). The search stops as soon as the limit of the search time is exhausted. The area of interest is divided into N possible locations containing the hidden targets. In our example, we consider an area divided into four sub-areas with one cell in each ($M = \{c_1, c_2, c_3, c_4\}$) two teams ($K = 2$) and $T_0 = 24$ (hours). In addition, there are three clusters,

$C_1 = \{c_1, c_2\}$, $C_2 = \{c_3, c_3\}$ and $C_4 = \{c_3, c_4\}$. The input data is given in Table 1 below and the confidence level is 95%.

Table 1. Input Data

Cells	c_1	c_2	c_3	c_4
$p_i = P(C_i)$	0.5	0.6	0.75	0.01
$\alpha_i = P(B_i \mid C_i)$	0.07	0.10	0.12	0.10
$\beta_i = P(\overline{B}_i / C_i)$	0.03	0.07	0.05	0.04
t_i	5	8	10	10
c_i	20	10	12	2

Using Table 1 and (1)-(2) we can compute f_i and H_i for $i = 1, 2, 3, 4$. For example, for the first cell ($i = 1$), we have $f_1 = 0.52$, $a_{11} = 0.932692$ and a_{12} is equal to 0.994819. Following, H_1 equals 2 for a CL of 95%.

Table 2 below presents the values of f_i and H_i for all four cells.

Table 2. Computation of f_i and H_i

Cells	c_1	c_2	c_3	c_4
f_i	0.52	0.618	0.8025	0.0106
H_i	2	2	1	4

The optimal strategy is as follows:

$$\langle S[0], C_1, C_1, C_2, C_1, \ldots \rangle = \langle 1, 2; 1, 2; 1, 2; 3, 3, 1, 2, \ldots \rangle$$

where $S[0] = \langle 1, 2 \rangle$. The search process rapidly converges and stops after three steps demanding 23 hours: probability that the process does not stops at step 1 is 1; that it does not stop at step 2 is 0.4687, at step 3 is 0.1668, at step 4 is 0.0111, and at step 5 is zero.

6 Conclusion

In this work, we present a fast algorithm to solve the two-stage detection-and-rescue planning problem. In order to optimize the scheduling process, we use a greedy strategy, an index-based strategy, which is proven to be optimal when the objective is to maximize the lifesaving efficiency. The "best cluster" is selected at each stage, and

the process is proved to be rapidly converging. Our solution is both simple and computationally efficient. When the confidence level is pre-defined, such local search strategies guarantees an optimal (max-efficiency) search sequences. In addition, using the suggested greedy methods can be applied to other search scenarios (e.g., with moving targets, agents-with-memory, etc.) and combining it with dynamic programming and biology-motivated heuristics can be a perspective direction for solving more complicated detection-and-rescue planning problems.

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Computer networks 38(4):393–422 (2002)
2. Benkoski, S.J., Michael G. Monticino, James R. Weisinger: A survey of the search theory literature. Naval Research Logistics 38(4):469–494 (1991)
3. Chung T.H., Burdick, J.W.: Analysis of search decision making using probabilistic search strategies. IEEE Transactions on Robotics 28 (1) (2012)
4. Dargie, W., C. Poellabauer: Fundamentals of Wireless Sensor Networks: Theory and Practice. John Wiley and Sons, 330 pp. (2010)
5. Feo Flushing, E., L. Gambardella, G. A. Di Caro: GIS-based Mission Support System for Wilderness Search and Rescue with Heterogeneous Agents. (2013)
6. Gelenbe, E., F.J.Wu: Large-scale simulation for human evacuation and rescue. Computers and Mathematics with Applications 64:3869–3880 (2012)
7. Kress, M., Lin, K.Y., Szechtman, R.: Optimal discrete search with imperfect specificity. Mathematical Methods of Operations Research 68(3):539–549 (2008)
8. Kress, M., Royset, J.O., Rozen, N.: The eye and the fist: Optimizing search and interdiction. European Journal of Operational Research 220(2):550–558 (2012)
9. Kriheli, B., Levner, E. Search and detection of failed components in repairable complex systems under imperfect inspections. Lecture Notes in Computer Science, v.7630, pp. 401–412 (2013)
10. Kriheli, B., E. Levner, A. Spivak: Optimal search for hidden targets by unmanned aerial vehicles under imperfect inspections. In: Proceedings of the 13 IFAC Symposium on Manufacturing, Snt Petersburg, Russia (2013)
11. Lambrou, T. P., Panayiotou, C. G.: Area coverage vs event detection in monitoring applications using mixed sensor networks. In: 8th World Congress of the InternationalFederation of Automatic Control, IFAC WC 2011, Milano, Italy (2011)
12. Levner, E., B. Kriheli, A. Elalouf, D. Tsadikovich: A fast scheduling algorithm for detection and localization of hidden objects based on data gathering in wireless sensor networks. In: Proc. the 13th Mexican International Conference on Artificial Intelligence, MICAI-2014, Lecture Notes in Computer Science (2014)
13. Sato, H., Royset, J.O.: Path optimization for the resource-constrained searcher. Naval Research Logistics 57:422–440 (2010)
14. Shima, T., C. Schumacher: Assignment of Cooperating UAVs to Simultaneous Tasks using Genetic Algorithms. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, pp. 1–14 (2005)
15. Stone, L.D.: Theory of Optimal Search. New York: Academic Press (1969)
16. Song, N.O., D. Teneketzis: Discrete search with multiple sensors. Math. Methods Oper. Res. 60(1):1–13 (2004)

17. Tang, H., Elalouf, A., Levner, E., Cheng, E.: Efficient computation of evacuation routes in a three-dimensional geometric network. Computers and Industrial Engineering 76:231–242 (2014)
18. Warston, H., Petersson, H.: Ground surveillance and fusion of ground target sensor data in a network based defense. In: Proceedings of the 7th International Conference on Sensors (2004)
19. Washburn, A.R.: Search and Detection. INFORMS, New York (2002)
20. Wu, Q., Rao, N.S.V., Barhen, J., Iyengar, S.S., Veishavi, V.K., Qi, H., Chakrabarty, K.: On computing mobile agent routes for data fusion in distributed sensor networks. IEEE Transactions on Knowledge and Data Engineering 16(6):1–14 (2004)

Objective Analysis in Task Planning and Allocation of Multicomputer Systems

A. Velarde M.

Instituto Tecnológico El Llano, Aguascalientes,
Mexico

Abstract. Parallel computing systems with multiple processing elements have the ability to run a set of different tasks at the same time. To achieve such a parallelism, these systems use typically an algorithm for task planning and another algorithm for task assignment. The planner algorithm must solve the problem of how many and which tasks remaining in a queue must be executed, how many processors should be used to execute the selected tasks, and which tasks must be executed first; meanwhile, the assignment algorithm must determine what free processors in a mesh will be used to execute the selected tasks. The objectives of both algorithms are maximizing the use of all processors and the adjacency of the processors assigned to a same task, as well as minimizing the waiting times of the tasks in a queue. Nonetheless, in the purpose of maximizing and minimizing all the objectives at the same time, several conflicts may occur among them, causing degradation in the performance of a parallel computing system. In this paper, it is presented an analysis of how the objectives in the task planning and assignment may conflict, specifically in multicomputer systems. The analysis is carried out by using a multi-objective optimization algorithm, through which each objective is evaluated to determine its effect in the performance of a parallel computing system. With the results of the evaluated objectives, a scale of priorities is proposed.

Keywords: Task planning, parallel computing

1 Introduction

Multicomputer systems with architectural meshes, interconnection topologies in 2D and 3D, denominated multicomputers in 2D or 3D mesh for commercial purposes and researching, have been the most common parallel systems due to their simplicity, scalability, structural regularity and simple implementation [1,17,3] in research and industry environments.

Various parallel computers commercial and experimental, such as the IBM BlueGene/L [4] and the Intel Paragon [5] have been built based on these two architectures. Some of the commercial multicomputer systems are Multiple Instruction Multiple Data (MIMD), with architectures that permit processor sub mesh partitions, and have the advantage of supporting multiple processes. Each of which can be assigned to an independent processor sub mesh for execution.

In an MIMD mesh, that supports multiple users, a task must be assigned to a free processor submesh, that corresponds to the size required of the operating system. The tasks solicit different computing requirements, and processor submeshes with different sizes within the mesh. When a task is finished executing, the submesh that it occupied is freed up for the next assignment process, this is known as consecutive assignment. The task assignment problem in multicomputer systems can be approached on two levels: on a task level and on a programming level [1,8]. For this research the task level assignment is used.

The main problem with efficient utilization of the processors in dynamic mesh multiuser systems, is the planning of computing resources [6,7,8]. Mesh resource planning, through hardware partitioning involves two components: task planning and a task assignment to the mesh. The function of task assignment, is to choose the next task or tasks for the queue that will be assigned to a free sub mesh to be executed. The function of submesh assignment, is to localize free submeshes that are to be assigned to the selected tasks for planning [6]. In figure 1, where the busy processors are shown with dark circles and the free processors in white, we can see a joining of 6 tasks in the queue to be ingresses into a 2D mesh with a processor size 8×8.

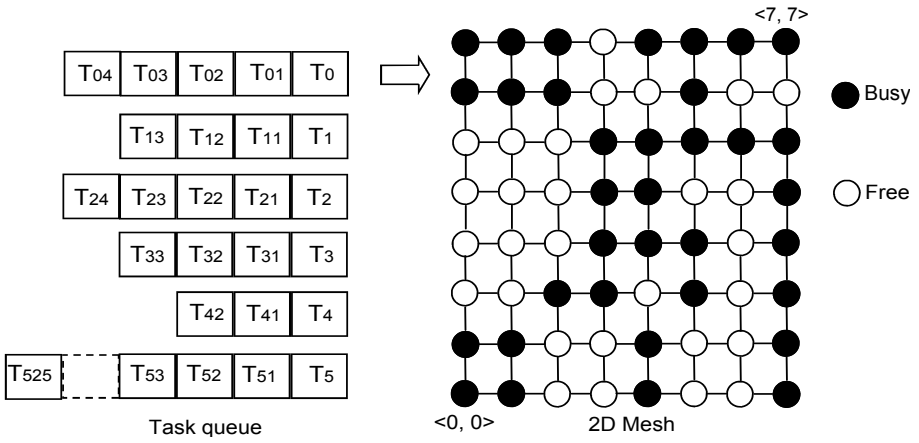


Fig. 1. System structure for task execution in a multi-computer 2D mesh system.

In task assignment for mesh processors there are two different methods: the continuous assignment method, in which the assignments are carried out only in adjacent processors within the mesh, and the non continuous assignment method which allows tasks to be assigned to processors, that are not found to be adjacent to the mesh. In the carrying out of planning and assignment functions,

independent from the type of assignment that is to be used, the following 6 criteria are aimed to be minimized or maximized [9]: system utilization, processor performance, average stationary time, wait time, remain time, result coefficient and the performance coefficient. These objectives, upon being evaluated with task loads in the system, generally result in counter-position, due to the fact that bettering one result in turn worsens another. Thus provoking the parallel system into being highly efficient under one criteria, and in others having efficient results far under the established norms [11,12].

In [13], five of these six objectives are evaluated through an Evolutionary Distribution Algorithm (EDA), especially being the case with the Unified Marginal Distribution Algorithm (UMDA). The object evaluation in [13], is meant to obtain the assignment which is best adjusted, to certain threshold values that are established as optimum values in the planning and assignment tasks. However, with the individualized evaluation process, the results are contrasted producing a multi-objective problem.

An example of the aforementioned is presented when the adjacency between processors assigned to tasks is maximized, thus making that the time in which the tasks remain, and wait in the queue becomes maximized, encouraging degradation in response times that the parallel system authorizes to the users.

In [10], a multi-objective problem is defined as, that in which involves the optimizing of a number of objectives simultaneously. With these types of problems, the objectives are in conflict with each other, the optimal solution of each function that corresponds with each objective (function objective) is different from the rest. In solving these problems, with or without the presence of constraints, it results in a set of interchangeable optimal solutions, popularly known as Pareto optimal solutions [10]. Due to the multiplicity in solutions, these problems were proposed to be solved appropriately using Evolutionary Optimization Algorithms (EOA), those in which use a population focus in the search engine procedure. Evolutionary Optimization Algorithms, use a population based approach, in which more than one solution is involved in an iteration, and evolve a new population of solutions at each iteration [10]. Multi-objective optimization problems give rise to a set of solutions, which require further processing to achieve a single main solution. To perform the first task, a natural proposition is to use an EOA, because the use of a population in one iteration helps an EOA to simultaneously find multiple non-dominated solutions, representing an exchange between objectives in a single simulation run. So considering that planning and allocating tasks in a parallel system, is a multi-objective problem, in this paper, five of the conflicting objectives in the planning and allocation of tasks on a system multicomputers are raised (defined below).

The objectives that the algorithm for scheduling must cover during implementation and proposed in [13] are:

1. Reducing the remain time of the tasks in the queue.
2. Decrease task starvation ,which would avoid discrimination in the allocation of tasks that require a lot of processors (great tasks). This is caused because

the tasks that require a small amount of processors (small jobs), are being continuously assigned.

The task allocation algorithm during its execution, is responsible for covering the following objectives proposed in [13]:

1. Reduce the number of assignments to the mesh of processors performing the tasks allocation algorithm.
2. Maximizing the use of the mesh processors, i.e., decrease the percentage of processors that remain free after the allocation algorithm places, one or more tasks in the mesh of processors (external fragmentation) [11].
3. Maximize contiguity between processors (assign the set of free allocate processors as close together as possible), to minimize the distance in the communication path, and avoid interference between them [12]; this is done in order to get a good parallel algorithm to decrease communication time, and maximize processing time [12].

The selection of the five objectives presented above obeys the completion of a state of the art review, of the research work related to different techniques or proposed planning and task allocation methods, seeking, isolated, to optimize one or more of the six criteria or objectives for planning and tasking.

In this paper, we address the problem of planning and allocation of tasks to a mesh of processors as a multiobjective problem, using for this the evolutionary algorithm outlined in [13] and explained in this section: UMDA algorithms (from this research), evaluating each objective function for analysis of the results, and determine which are the determinants or is the better decision to assign, and schedule them in a system of multicomputers objectives.

For the completion of the objective contrast analysis, a multicomputer Liebres InTELigentes system for teaching programming, systems of high performance computing in higher education institutions, [14] was used to perform the analysis of the contrast between the objectives. The results of each of the objective functions evaluated, obtained with different workloads in the queue of the target system, and with processors mesh sizes 4×4, 8×8 and 16×16. When the values of the objective functions are similar, a priority scale is established for the proposed set of objectives, in this way, a successive application of this scale takes places until the best allocation is found, as set forth in [13].

This research is divided as follows: a section about previous works, where the research conducted in the area of the allocation of processors is presented, a section of basic concepts where the terms related to a 2D architecture mesh are defined; a section titled UMDA algorithm where performance and features of this algorithm are proposed, as well as it is presented in [13] and how it adapts to this research this algorithm. Processes for the analysis of the contrast between the objectives, the planning and allocation of tasks in a 2D mesh and contrast of the objectives during the planning and allocation of tasks, is explained by a set of examples and how the objectives are opposed, during the planning and allocation of tasks on a system multicomputers. In the section of experiments,

tests for the analysis of conflicting goals and resolve the opposition that occurs in the goals set during the planning, and allocation of tasks in the screen are explained. In the last section, the conclusions are drawn from the analysis and experiments conducted are presented.

2 Previous work

Several investigations have been conducted to develop strategies for assignments, in both contiguous and non-contiguous parallel computing systems. This section describes, for reasons of space, only the most significant non-contiguous allocation techniques, that have been developed within different researches. It has been possible to extract the characteristics of the methods, and define the scale priorities of the different objectives proposed in this paper, which seek to become maximized or minimized.

The first adjustment technique FF (First Fit) [15], through the pursuit of free submesh, and determining the sub-mesh that best fits the application, seeks to find the maximum adjacency between processors to lower latency communication between tasks. In the paging technique [15], the iterative process of division of the sub-mesh in partitions of equal size $2i$, where i is a positive integer representing the index of the page parameter, seeks to assign a task to a selected page, allowing the job to run with a total of processors avoiding interference adjacency messages by disjoint processors. MBS [15], a process of division of the mesh to obtain overlapped square submeshes with potency lengths of 2, recursively will decrease to be suited to a request, this causes the work to be embedded on a set of 100% adjacent processors. In ANCA [16], it first tries to assign the task to a sub-mesh of adjoining processor, if it fails, the application is partitioned into sub-partitions of equal size recursively, until it is able to assign subpartitions in locations available to the mesh. In the Random strategy [15], tasks are assigned to the mesh depending on a random number and all free processors are considered in the allocation, with this type of arbitrary allocation use of all available processors and the elimination of any kind of fragmentation that can occur are sought, however, a high communication interference occurs between tasks.

Newer techniques have similar connotations to the original proposals, through the use of an initial strategy to assign the tasks, but when the allocation is not able to be done, a second strategy that replaces the first to achieve the objective of allocation is activated. Examples of such techniques, include search strategy and friendly Multiple Adaptive (Adaptive Scan and Multiple Buddy AS & MB) [15], Allocation Contiguous No Quick (QNA for its acronym in English Quick Non-Contiguous Allocation) [18], and strategy allocation proposed in [17]. In AS & MBS, it seeks to assign the task to a sub-grid of equal size as the one requested, if it does not exist, the MBS strategy is activated to perform the division process of requirements [19].

The strategy proposed in citeBani, the FF method is used in conjunction with the BF method, as follows: if a task requests a sub-mesh sized 4×4 and

the application can not be granted, the request size is reduced to a multiple of 2, for this case 2×2 mesh size requested, and so on until the request has the minimum number of processors, in this case 1×1. When the first technical fault occurs, a second technique is the BF is activated through this technique, a search is performed in free submeshes that best fit it, i.e., with the exact number of processors that the task requires [20]. In fact, 2 alternative techniques are applied within the method of allocation to improve the condition of contiguity, by maintaining a good level of closeness between processors, to run the same task and reduce communication latency that is caused by no contiguity between the processors.

3 Basic Concepts

The proposed system is of multicomputers connected in a 2D mesh with a job queue waiting for admission to the mesh, and allowances are established as a dynamic quadratic assignment. The following definitions formally describe a system of this type.

Definition 1. An n-dimensional mesh has $k_0 \times k_1 \times ... \times k_{n-2} \times k_{n-1}$ nodes, where k_i is the number of nodes along the $i-th$ dimension and $ki \geq 2$. Each node is identified by n coordinates: $0(a), 1(a), ..., n-2(a), n-1(a)$, where

$$0 \leq i(a) < k_i for 0 \leq i < n.$$

Two nodes a and b are neighbors only if $i(a) = i(b)$ for all dimensions except for a dimension j, where $j(b) = j(a) \pm 1$. Each node in a mesh refers to a processor and two neighbors that are connected by a direct communication link.

Definition 2. A 2D mesh definition, which is referenced as $M(W, L)$ consists of $W \times L$ processors, where W is the width of the mesh and L is the height of the mesh. Each processor is denoted by a pair of coordinates (x, y), where

$$0 \leq x < W and 0 \leq y < L.$$

A processor is connected by a bidirectional communication link to each of its neighbors. For each 2D mesh $= P_{ij}$.

Definition 3. In a 2D mesh, $M(W, L)$, a sub-mesh: $S(w, l)$ is a two-dimensional mesh belonging to $M(W, L)$ with a width w and a height l, where

$$0 < w \leq w and 0 < l \leq L,$$

and $S(w, l)$ are represented by coordinates (x, y, x', y'), where (x, y) is the lower left corner of the sub- mesh, and (x', y') is the upper right corner. The node in the lower left corner is called the base node of the sub-mesh and the upper right corner is the end node. In this case $w = x' - x + 1$ and $l = y' - y + 1$. The size of $S(w, l)$ is: wxl processors.

Definition 4. In a 2D mesh $M(W, L)$, a sub-mesh available $S(w, l)$ is a sub-mesh that meets the conditions: $w \geq \beta \geq \alpha$ assuming that the allocation of

$S(\alpha, \beta)$ required where the allocation refers to select a set of processors to an arriving task.

Definition 5. Let ϑ be a set of system tasks, such that $\vartheta = J_1, J_2, \ldots, J_n$, where n is the number of tasks at time t y ϑ_k a set of sub-tasks of the task k where: $\vartheta_k = j_{k1}, j_{k2}, \ldots, j_{kf(k)}$ y $f(k)$ is the total number of sub-tasks of the task j. For each task each task j and each sub-task $f(k) \in j$ a processor $m_i \in P$ is had that should run j and sub-task $j_{kf}(k)$, consuming an uninterrupted time of $t \in \mathbb{N}$.

Definition 6. Given two matrices of size $n \times n$: a flow matrix F whose (i, j) elements represent flows between i and j tasks and an array of distances D whose (i, j) represent the distance between sites i and j. An assignment by the vector p, which is a permutation of the numbers $1, 2, ..., n$. $p(j)$ is where the task j is assigned. Thus, the quadratic assignments can be written as:

$$min_p \in \sum_{i=1}^{n} \sum_{i=1}^{n} f_{ij} dp(i)p(j).$$

Definition 7. An optimization problem, is one whose solution involves finding a set of candidate alternative solutions that best meet objectives. Formally, the problem consists of the solution space S and objective function f. Solving the optimization problem *(S, f)* it is to determine an optimal solution, namely, a feasible solution $x^* \in S$ such that $f(x^*) \leq f(x)$ for any $x \in S$. Alternative solutions can be expressed by assigning values to some finite set of variables $X = Xi : i = 1, 2, ..., n$. If U_i is denoted the domain or universe (set of possible values) of each of these n variables, the problem is to select each variable X_i domain U_i value x_i assigned that, subject to certain restrictions, optimizes an objective function f. The universe of solutions is identified with the set:

$$U = x = (x_i : i = 1, 2, \ldots, n) : x_i \in U_i.$$

The problem constraints reduce the universe of solutions to a subset of $S \subseteq U$ solutions, called feasible space.

Definition 8. Utilization. It is defined as the fraction of time in which the system was used. And it is given by:

$$U_G = W_G/(C_G * m_G),$$

where W_G is the amount of work the system, C_G is the end time of execution of all tasks in the system, m_G is the total number of processors in the system.

Definition 9. Processing Performance (throughput). The number of tasks completed per unit of time in the system, and it is given by:

$$n/C_G,$$

where n is the total number of tasks in the system.

Definition 10. Mean turnaround time. The average time it takes for all tasks upon entering the queue until their execution is ended. It is calculated as:

$$\frac{1}{n} \sum_{j=1}^{n} t_{\mathrm{t}}^{\mathrm{j}},$$

where $t_{\mathrm{t}}^{\mathrm{j}} = c^j - r^j$, c^j is the time of completion of the task and r^j is the delivery time of the task j.

Definition 11. Waiting time. It is defined as the average waiting time before starting the task execution. It is calculated as:

$$\frac{1}{n}\sum_{j=1}^{n} t_{\mathrm{w}}^{\mathrm{j}},$$

where

$$t_{\mathrm{w}}^{\mathrm{j}} = t_{\mathrm{s}}^{\mathrm{j}} - r^j,$$

where $t_{\mathrm{s}}^{\mathrm{j}}$ is the start time of execution of the task j.

Definition 12. Coefficient response (response rate). It is defined as the average of the response factors of all tasks. It is defined as:

$$\frac{1}{n}\sum_{j=1}^{n}(t_{\mathrm{w}}^{\mathrm{j}} + P^j)/P^j,$$

where P^j is the runtime and $t_{\mathrm{w}}^{\mathrm{j}}$ is the waiting time of the task j.

Definition 13. Competitive ratio. A measure of system performance defined as:

$$p = c_g/c_{LB},$$

where c_g is the time of completion and c_{LB} is the minimum time to complete tasks, calculated as: $maxw_G/m_g, t_{\mathrm{g}}^{\max}$, where $t_{\mathrm{g}}^{\max}$ the maximum runtime of the n tasks.

4 The UMDA

The EDA (Estimation of Distribution Algorithms), are evolutionary algorithms that use a collection of candidate solutions for accomplishing search paths avoiding local minimums [21,22]. These algorithms use the estimation and simulation of the joint probability distribution, as a mechanism of evolution, instead of directly manipulating the individuals that represent solutions to the problem. EDA algorithm starts randomly generating a population of individuals that represent solutions to the problem. Three types of operations are performed iteratively on the population [21,22]. The first type of operation is the generation of a subset of the best individuals in the population. Secondly, a learning process from a probability distribution model is made from selected individuals. Third, new individuals are generated by simulating model the distribution obtained. The algorithm stops when a certain number of generations are reached or when performance fails to improve significantly.

To estimate in each generation the distribution of joint probability, from selected individuals, we use the algorithm of the univariate marginal distribution (UMDA by its acronym, Univariate Marginal Distribution Algorithm). Thus, the joint probability distribution is factored as the product of independent univariate distributions [21,22], that is:

$$p_l(x) = p(x|\, D_{\text{l-1}}^{\text{Se}}) = \prod_{\text{i=1}}^{\text{n}} p_l(x_i).$$

Each univariate probability distribution is estimated from marginal frequencies:

$$p_l(x_i) = \frac{\sum_{j=1}^{n} \delta_j(X_i = x_i|\, D_{\text{l-1}}^{\text{Se}})}{N},$$

where

$$\delta(X_i = x_i|\, D_{\text{l-1}}^{\text{Se}}) = \left\{ \begin{array}{l} 0\, if\, is\, the\, i-th\ D_{\text{l-1}}^{\text{Se}},\ X_i = x_i \\ 1\, in\, other\, case \end{array} \right\}.$$

The pseudo code for UMDA proposed in [21,22] is shown in table 1.

Table 1. The pseudo code for UMDA [21,22]

$D_0 \leftarrow$ Generate M individuals (the initial population) random
Repeat for $l = 1, 2, ..$ until stop criteria
$D_{\text{l-1}}^{\text{Se}} \leftarrow$ select $N \leq M$ Individuals of D_{l-1} according to the selection method:
$p_l(x) = p(x|\, D_{\text{l-1}}^{\text{Se}}) = \prod_{\text{i=1}}^{\text{n}} p_l(x_i) \prod_{\text{i=1}}^{\text{n}} \frac{\sum_{j=1}^{n} \delta_j(X_i = x_i|\, D_{\text{l-1}}^{\text{Se}})}{N} \leftarrow$
Estimate the joint probability distribution D_l
Sample M individuals (the new population) of $P_l(x)$

In this paper, the application of UMDA evolutionary algorithm is carried out as follows:

1. A set of tasks is dynamically extracted from the queue that fit in the free submeshes, this set of tasks represents a possible assignment (individual); this process is repeated until n number of individuals (user-defined), that constitute a population.
2. For each individual in the population, the five objectives are evaluated to determine the subset of assignments (subset of the best individuals) that show results closest to maximization or minimization established for each objective function.
3. The probability distribution model learning process, is produced from selected individuals representing the best assignments to the mesh of processors.
4. A new generation of individuals occurs by simulating the distribution model obtained in the previous step.
5. An algorithm stop mechanism is activated when minimizing or maximizing of the objective functions.

During the process of evaluating each target for each of the individuals, the contrasts are shown in the results, due to the improved results from a function other objective result worsen. The following section explains through examples how the objectives are opposed.

5 Process for the Analysis of the Contrast between the Objectives of the Planning and Allocation of Tasks in a 2D mesh

To make the contrast analysis of the 5 goals, extracted from research and previously discussed in previous sections, first, the results were considered [12,13] of the trials of the UMDA that evaluates each target separately. Second, additional trials were conducted of the same algorithm to determine the contrast of each of the targets within the same group. The formal approach of the found contrasts are detailed in the following section. Upon completion of the experiments, and based on the results a scale of priorities is formed, that in which is used in this research as a determiner to find the best assignment of tasks to the processor mesh.

5.1 Opposition of the Objectives during Planning and Allocation of Tasks

In this section, the found contrasts are explained through a set of examples between the objectives that are pursued to meet the optimal utilization of the processors of the mesh. A formal approach to them is also performed.

Objectives 1 and 2, which seek to minimize the number of assignments to the mesh of processors, to minimize the time that jobs remain in the queue, is at odds with the objectives of minimizing the use of processors in the mesh and minimizing starvation of tasks. To illustrate how these four objectives are opposed, consider that at time t, the allocator 29 reports free processors (as shown in figure 1), with this information the scheduler determines that the set of the 5 tasks T_0, T_1, T_2, T_3 and T_4 are candidates to fill 21 processors in the mesh, or assign the task requiring 26 T_5 and T_4 processors requesting task 3. Assign the set of 5 tasks releases the same number of positions in the queue for the entry of new tasks, thus reducing the number of accesses to the queue to find more tasks, this allows a greater number of users and tasks to be served and the waiting time of tasks in the head of the queue is decreased; but in opposition to this, an external fragmentation of 8 processors is generated and starvation in a cycle of tasks increases, upon not being served a task that requires a large number of processors.

Now, if the T_4 and T_5 tasks assigned do not produce starvation nor external fragmentation, a smaller number of tasks can be accepted in the queue, so the number of assignments to the screen increases and therefore so do the time tasks must wait to enter the mesh of processors.

Objective 3, which seeks to maximize the use of the processors in the mesh, contrasts with objective 5, which maximizes the adjacency of the occupied processors in the 2D mesh. To illustrate the contrast between these objectives, consider the example above. By searching for the lowest communication cost, all of the 5 selected tasks: T_0, T_1, T_2, T_3 and T_4 are assigned in contiguous processors, as follows: T_0 task is assigned to the sub-mesh $< 4, 0 >< 5, 2 >$ regardless of the

processor to $< 4, 2 >$, the T_1 task is assigned to the sub-mesh $< 2, 0 >< 3, 1 >$, the task T_2 is assigned to the sub-mesh $< 0, 5 >< 2, 6 >$ regardless of the processor to $< 2, 5 >$, the task T_3 is assigned in submesh $< 0.2 >< 1, 3 >$, and T_4 task is assigned to the sub-mesh $< 6, 3 >< 7, 4 >$ regardless of the processor to $< 7, 4 >$. If the proposed method detects the increase of starvation in the system, the T_5 task will be assigned to mesh with the task T_1 or to task $T3$ that is selected to occupy all of the processors, after a search in the queue and to avoid external fragmentation; in this way the 29 free processors in the mesh will remain busy. If targets 3 and 5 are opposed one can deduce that to assign the task T_1 or T_3 and T_5, the use of processors in the mesh is maximized, but the adjacency between processors is minimized, and in contrast, if the set of 5 tasks is assigned, the adjacency between processors is maximized, but an 8 processor external fragmentation occurs.

Objective 1 minimizes allocations to the mesh of processors, runs counter to *objective 5*, which maximizes the adjacency between processors. The contrast between these two objectives appears when you intend to assign a large number of tasks in the mesh of processors, and processors to which tasks are assigned are not close enough together or contiguous, to avoid producing very high communication costs. If we consider assigning the set of 5 tasks, T_0, T_1, T_2, T_3 and T_4, the number of assignments made to the mesh is minimized, but if the positions of the free processors in the mesh are adjacent, occur tasks will be assigned to the the mesh in a very disjointed way, causing the adjacency of processors to be minimal and communication costs between tasks to be very high.

Objective 2 which seeks to minimize the waiting time of tasks in the queue, runs counter to the objective of maximizing the adjacency between processors. Using the same example in the previous section and considering that the *objective 2*, sets to minimize the waiting time of tasks in the queue, upon assigning the largest number of tasks in the mesh the wait time fora set of tasks is minimized. In the allocation that occurs at time *t*, fewer jobs wait in a queue. For such cases if two objects are affected simultaneously by a third objective, and we manage to improve them, the algorithm will decide, based on this option when planning and allocating, considering the generated cost involved in external fragmentation and increased task starvation.

Objective 4, which aims to minimize the starvation of tasks, runs counter to the objective 5, which maximizes the adjacency between processors. The decision to allocate a greater number of tasks to minimize waiting times and maximize the use of processors, seems like a viable option, but if we consider a third goal in conflict that tries to reduce task starvation in the system, we find then that we are in favor of two objectives (2 and 3), and sacrifice also 2 (objectives 4 and 5).

Based on the explained examples, we have found that maintaining a strict control of tasks that fit in the mesh, to meet the proposed objectives, produces exhausting searches in the queue, and calculations to locate tasks in the best position in the mesh [12]. Rather than seeking the best positions of tasks in the mesh, you should perform an analysis of the objectives, which seek to optimize,

because when trying to locate submeshes of sizes that the tasks required for the sole purpose of being contiguous, without considering a evaluation of other objectives, can lead to poor results in response times and system performance.

Based on the above explanations, in this paper a scale of priorities for the objectives is made, to be considered during the planning and allocation of tasks in multicomputers systems. It is noted that this scale of priorities, is considered in the algorithm proposed in this paper, and with this the results explained in the section of results were obtained.

6 Scale of Priorities of the Objectives

This paper presents a stratification of the proposed objectives, based on the results obtained with the algorithm UMDA and observations made in the above experimentation have been performed in order to determine the best allocations, should similar or identical l values be found when compared to those of objective's rankings.

Stratification proposal is as follows:

1. Objective 5, which sets to maximize adjacency between processors, is considered the major goal in the allocation for five situations that arise during task assignments to the mesh of processors, in experiments: the first factor to consider is the communication time tasks consume during execution, because the non-adjacent processors generate very high communication costs and even more when to perform tasks that require large quantities of processors within the mesh. Although the search for free submeshes is a tedious process and consumes time from the processor, it is a task that should be extensive at any time.
2. Objective 3, maximize the use of processors to reduce external fragmentation, is considered secondly because of its importance in the allocation of the processors in the mesh. It is importance, is that it serves as a support for Objective 1, the experiments conducted allow us to observe that in order to maximize the use of processors, allocations should be made in the greater number of processors that are adjacent within the mesh of processors. To meet this objective, the algorithm that makes finding free submeshes must be big enough.
3. Objective 4, minimize task starvation, it is located on the third level of importance, because being able to meet the two prior objectives this allows a safe handling of tasks, that require large numbers of processors in the mesh, thus avoiding task starvation. If the free search algorithm submeshes provides sets of adjacent free processors, it is possible to avoid task starvation upon placing them within the mesh.
4. Objective 1, minimize the number of assignments to the mesh of processors, i.e. minimize the number of planifications, that the algorithm must perform with the tasks that remain in the mesh, it is considered as the fourth objective of importance to the evaluation and is considered to include Objective 2,

because achieving being able to minimize the number of assignments, to the mesh of processors reduces the time that tasks waiting in the queue

5. Objective 2 is considered a level 5 of importance, which sets to decrease the waiting time of tasks in the queue.

7 Results

The experiments were performed with different workloads in the Liebres Inteligentes [14] system and with different sizes in the queue. Size loads of 256, 512, 1024 and 2048 tasks are considered in the system. The lengths of the queue are carried out in 10, 20, 30 and 50 tasks with their respective subtasks. The number of subtasks for each task is 255 at most, considering that the size of the mesh of processors is $< 16 \times 16 >$. In figure 2, the results obtained for each objective function are shown. For reasons of space, only the loads on the system and the values obtained for the objective function are shown, up to 800 tasks.

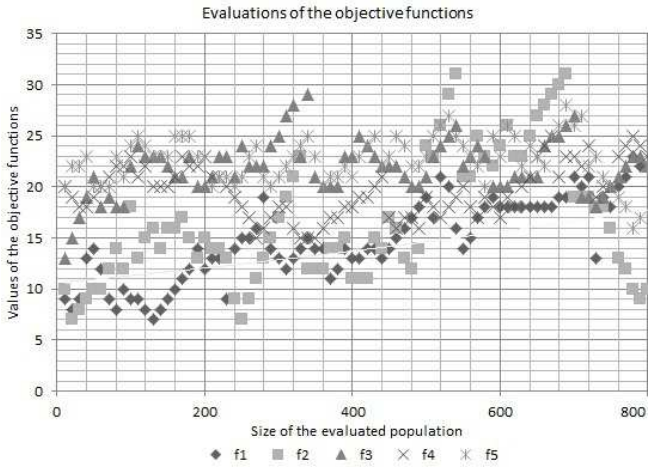


Fig. 2. Experiment 1, the chart shows the values obtained for the objective functions

In the X coordinate, different system loads are shown and in the Y coordinate obtained values of the objective functions are given. The performance of the functions, shown at the bottom of the graph, identify the symbol used for each function, the order of the functions is ascending, not by degree of importance.

In the following paragraphs the results for each objective, and the impact they have over the functions of planning and allocation of tasks are explained.

The values for function 5, maximize adjacency between processors that seek to assign tasks with the highest degree of contiguity, show that with values of less than 100 task workloads, no acceptable values are had, but as the number of task execution increases, values significantly improve. The function is observed in this part, the number of subtasks processed exceeds the average, ie exceeds 128 subtasks task.

Function 4, minimizing starvation task has a high tendency, because the tasks that are to be processed contain a large number of subtasks, causing fewer tasks with subtasks to be addressed quick through the system. As large tasks are evicted, starvation tends to stabilize at acceptable levels causing a greater fluency in job processing.

The 3rd function, maximizes the use of processors, considered one of the most important functions for the system, it has a tendency to group values, with the results of the function 5, but as the number of tasks to be processed increases, there is a dispersion in a middle point whose tendencies show that upon obtaining higher adjacency the use of processors decreases, especially when the system starts to process jobs with a large number of subtasks.

Function 2, decreases the time that a task expected to be attended in line, shows a very clear trend, when tasks are processed with fewer resource requirements, waiting times are very short, due to the planning carried out by the algorithm . Otherwise, when the tasks are processed containing a large number of requirements, waiting times are higher, which undoubtedly, upon increasing the number of resources, this trend is easily improved. This is shown in the graph when loads between 200 and 400 jobs are processed; the values of the functions are fired very easily.

Finally, function 1, which seeks to minimize the number of assignments that the algorithm performs, shows very poor booting trends, but as the implementation progresses, their values are significantly improved. It´s tendency with the values of the function 2, makes it a dependent function that takes a curve to the values acquired in function 2. As function 2 has better values that represent a decrease of time a task has to wait to be served in the queue, the number of assignments that the allocator algorithm performs is substantially reduced. Function 2 in conjunction with function 1, has a high degree of importance on the results that the system puts out, so it is important to consider them as priorities.

Figure 3 shows another exemplifying embodiment of the system. With a greater number of tasks in the system, value trends are similar to the previous graph. With this example, it is intended that the functions obtain values that correspond to a greater load in the system. The analysis that is done in this experiment, allows us to observe that the functionality of the planning algorithm is feasible when used with heavy system loads.

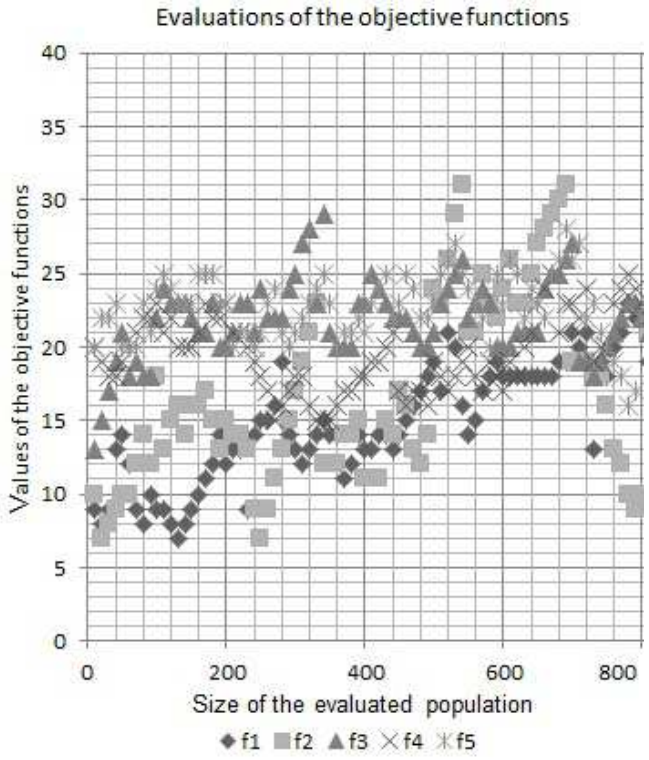


Fig. 3. Experiment 2, the chart shows the values obtained for the objective functions

8 Conclusions

Multicomputers systems are a viable option for parallel processing, because of their growth in terms of computing power and distributed storage. The inherent problems associated with their architecture are the planning and allocation of tasks. For allocating tasks to the mesh of processors many techniques based on different strategies have been proposed through: geometric figures that move across the screen to locate the free submeshes, adjustments to free submesh application sizes and techniques based on random assignments. Most of these techniques make use of planning policies, based on the first to arrive is the first to be served (FIFO First Input, First Output), that is, pre-planning is not used in the queue, furthermore it only seeks to solve one problem: the one that deals with adjacent or contiguous allocation to be able to reduce message passing between tasks and subtasks.

The method presented in this document deals with the problem of planning

and allocation of tasks on a multicomputer system as a multiobjective problem, conducting an analysis of how each goal impacts system performance, by means of using an evolutionary algorithm. The objective of this analysis is to show the values that functions present when goals are opposed, both in the planning and allocation of the mesh processors.

With the obtained results using the Liebres Inteligentes multicomputer system it is possible to deduce that in order to evaluate an allocation technique of processors in a mesh, it is necessary that this technique evaluates at least 5 different objectives, because in this way, you can determine that not only one problem will be solved, but a set of values that balance a solution will be had. An approach that seeks to solve only one objective, is not feasible, for example, one that seeks to solve task adjacency and allows system response time is not considered in the formulation of the solution.

Finally, it is very important to mention the work that is to be performed by the free submeshes search algorithm, within the mesh of processors, because it is what supports, monitors and fulfills the most important objectives within the proposed stratification in this paper research.

References

[1] Grama A., Gupta A., Karypis G., Kumar V.: Introduction to Parallel Computing. Second Edition. Addison Wesley (2003)

[2] Bani S., Ababneh I., Ould M.: A Performance Comparison of the Non-Contiguous Allocation Strategies in 2D Mesh Connected Multicomputers. International Conference On Communication, Computer And Power (ICCCP'09) MUSCAT (2009)

[3] Ababneh I., Bani-Mohammad S.: A new window-based job scheduling scheme for 2D mesh Multicomputers. Simulation Modeling Practice and Theory 19, pp. 482–493 (2011)

[4] Adiga N.R., Almasi G., Almasi G.S., Aridor Y., et al.: An Overview of the BlueGene/L Supercomputer. Team IBM and Lawrence Livermore National Laboratory.

[5] Bokhari. S. H.: Communication Overhead on the Intel PARAGON, IBM SP2 & MEIKO CS-2. [Online] http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19960004071.pdf

[6] Ahmad S. E.: Processor Allocation with Reduced Internal and External Fragmentation in 2D Mesh-based Multicomputers. Journal of Applied Science 11(6), pp. 943–952 (2011)

[7] Das D., Pradhan D. K.: Job Scheduling in Mesh Multicomputers. IEEE Transactions On Parallel And Distributed Systems, 9(1) (1998)

[8] Foster I.: Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering. Addison Wesley (1995)

[9] Heiss H. U.: Dynamic Partitioning of Large Multicomputer Systems. Proc. Int. Conf. on Massively Parallel Computing Systems (IEEE MPCS94), Ischia (1994)

[10] Deb K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley (2001)

[11] Velarde A., Ponce de Leon E., Díaz E., Padilla A.: Planning and Allocation of processors in 2D meshes. Doctoral Consortium. Mexican Internacional Conference on Artificial Intelligence MICAI 2010 Pachuca Hidalgo, México (2010)

[12] Velarde A., Ponce de Leon E., Díaz E., Padilla A.: Dynamic quadratic Assignment to Model Task Assignment Problem to Processors in 2D Mesh. Advances in Soft Computing Algorithms. Research in Computing Science, Vol. 54, pp. 199–218 (2011)

[13] Velarde A., Ponce de Leon E., Diaz E.: Planning and Allocation Tasks in a Multicomputer System as a Multi-objective Problem. Advances in Intelligent Systems and Computing 227. EVOLVE 2013, International Conference, Leiden, The Netherlands, Springer (2013)

[14] Velarde A.: Liebres InTELigentes: Sistema de Multicomputadoras para la enseñanza de la programación de los sistemas de cómputo de alto rendimiento en Instituciones de Educación Superior. Articulo aceptado en: Congreso Internacional de Investigacion en Ciencias y Sustentabilidad de AcademiaJournals, Tuxpan, Ver., Mexico (2015)

[15] Lo V., Windisch K., Liu W., Nitzberg B.: Non-contiguous processor allocation algorithms for mesh-connected multicomputers. IEEE Transactions on Parallel and Distributed Systems, vol. 8, no. 7, pp. 712–726 (1997)

[16] Chang C.Y., Mohapatra P.: Performance improvement of allocation schemes for mesh-connected computers, Journal of Parallel and Distributed Computing, vol. 52, no. 1, pp. 40–68 (1998)

[17] Bani A. S.: Submesh Allocation in 2D Mesh multicomputers: Partitioning at the Longest Dimension of Request. The International Arab Journal of Information Technology, Vol. 10, No.3, pp. 245–252 (2013)

[18] Procsimity V4.3 User's Manual. University Oregon (1997)

[19] Zolfaghari R.: Efficient Algorithm for Processor Allocation in Mesh Multicomputers Network with Limitations and Assumptions. IJCEM International Journal Of Computational Engineering & Management, Vol. 16, no. 4 (2013)

[20] Suzaki K., Tanuma H., Hirano S., Ichisugi Y., Connelly C., Tsukamoto M.: Multi-tasking Method on Parallel Computers which Combines a Contiguous and Non-contiguous Processor Partitioning Algorithm. Proceedings of the 3rd InternationalWorkshop on Applied Parallel Computing, Industrial Computation and Optimization, Lecture Notes in Computer Science, Springer, London, pp. 641–650 (1996)

[21] Larrañaga P., Lozano J. A., Mühlenbein H.: Algoritmos de estimación de distribuciones en problemas de optimización combinatoria. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial (2003)

[22] Lozano J. A., Larrañaga P.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic.

Solution of a Bi-Objective Purchasing Scheduling Problem with Constrained Funds using Pareto Optimization

José Francisco Delgado-Orta[1], Laura Cruz-Reyes[2], Alejandro Palacios-Espinosa[3], and Christian Ayala-Esquivel[1]

[1] Universidad del Mar, Oaxaca, Mexico

[2] Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Tamaulipas, Mexico

[3] Universidad Autónoma de Baja California Sur, La Paz, Mexico

fdelgado@zicatela.umar.mx, cruzreyeslaura@gmail.com, palacios@uabcs.edu.mx, christiancae@zicatela.umar.mx

Abstract. In this paper the Purchasing Scheduling Problem (PSP) with limited funds is presented. PSP is formulated through the optimization of two objectives based on the inventory-supply process: maximization of satisfied demands and minimization of purchasing costs. The problem is solved using two variants of the Ant Colony System algorithm (ACS), designed under Pareto's optimization principle in which elements of multi-objective representation for computing a feasible solution are incorporated to the basic design of ACS. Experimental results reveal that the Pareto approach improves solutions over the ACS in 8%, obtaining an efficiency of 80% solving the set of PSP instances as purchasing plans. This reveals the advantages of developing evolutionary algorithms based on multi-objective approaches, which can be exploited in planning and scheduling systems.

Keywords: Purchasing scheduling problem, multi objective optimization, ant colony system algorithm

1 Introduction

The purchase of goods is an essential activity for companies and business. It is the process that involves supply based on searches of items in physical facilities, information of products to check inventory stocks, objects or items in big catalogs and supply of goods on supplier locations. All these activities are periodically executed based on customer demands and the inventory control, associated with the availability

of economic resources and the storage space in warehouses. In this manner, the Purchasing Scheduling Problem (PSP) (introduced in [1]), establishes a mathematical approach to compute purchasing schedules when demands are variable. Industrial application of PSP is defined as a graph-based problem with several objectives, for example maximization of demand satisfaction, minimization of purchasing costs, maximization of inventory supplies and minimization of supply times.

In addition, multi-objective formulation of PSP faces additional constrains such as penalties to influence a schedule with a subset of desired elements, which implies a quality factors in purchasing related with customer preferences [2, 3], critical supply times [4], negotiations in economical lots of orders [5], categorization of products to be purchased [6], and availability of physical space at warehouse facilities [7] when stock must be supplied. For this reason, selection of appropriated goods to be supplied for inventory has become a complex and multi-objective task, whose approach determines the efficiency of a purchasing plan. It is desirable to optimize economical resources in the companies able to produce, distribute and sell their products according to the supply chain.

2 The Purchasing Scheduling Problem

The Purchasing Scheduling Problem (PSP) is defined through a catalog of products like a weighted graph $G = (V, E)$, where $V = \{P \cup S\}$ consists of a set of n products (P) per m suppliers (S). The set E is formed by pairs (p,s), $p \in P$ and $s \in S$. Each pair has a cost c_{ps} to purchase a product p from any s supplier. Purchasing process is organized through orders $P_k \in P$ (or demands), where k represents a decision maker (a purchaser) with a number n_k of products to be satisfied with an available fund a_k. In these concepts, PSP optimizes two objectives: maximization the amount of satisfied products (for each order P_k) and minimization of purchasing costs (c_{ps}) in an inventory cycle. These objectives introduce the field of multi-objective computation.

3 Multiobjective Optimization

Muitivariant and multiobjective nature of real problems present a challenge to development of efficient algorithms. As a consequence, computation of optimal solutions in a multi-objective problem (MOP) is computationally intractable [8] when large-scale instances are solved. As a consequence, optimal solution of MOP is not possible to compute because MOP is represented by a set of objectives in conflict. This is why computation of solutions in a MOP consists of establish the set Pareto front $PS = \{s_1 , s_2 ,.., s_m\}$ with s_m solution vectors of the problem, where feasibility of solutions is given in terms of dominance and efficiency of Pareto.

Dominance is defined according to the analysis of objectives in pairs. It establishes that objective $s_j \in PS$ dominates a vector $s_j' \in PS$ if and only if $s_j \leq s_j', \forall j \in \{1,..., p\}$, with at least one index j for which the inequality is strict

(denoted by $s_j \prec s_j'$). Efficiency of Pareto defines a feasible solution s_j, for which there does is no other solution s_j' such as $z(s_j) \prec z(s_j')$. It implies that s_j is a non-dominated solution (or Pareto optimal). PSP implies the solution of two objectives based on warehouse operations, in which these represent opposite decisions. It defines a multi-objective scene of PSP in terms of a graph-based problem, needed to compute efficient solutions for the related MOP in PSP.

4 PSP Formulation

PSP is formulated through the next data sets:

The general inventory catalogue sets:

P: is the set of products in an inventory catalog with *n* products.

P_k: is the set of products to be purchased with n_k products, $P_k \in P$, k=1,2,…,s

S: is the set of suppliers in the product catalog with *m* suppliers.

The model uses the next variables:

k is the number of orders in each inventory cycle. k = 1,2,…, s

c_{ij} is the cost to purchase a product *i* from a supplier *j*.

a_k represents the available funds for each order *k*.

x_{ijk} is an integer variable {0,1}. It has a one value if a product *i* is assigned to the supplier *j* in the order *k*, zero in otherwise.

Objectives of PSP are defined with the *f* and *g* coefficients, a normalized objective values in the domain [0,1], where *f* represents a profit in terms of satisfied demands and *g* indicates a uniform reference with regard to the assigned c_{ij} values for each assigned product. These values are based on the utility principle proposed in [9;10;11], defined through expressions (1) and (2).

$$\max f = \frac{1}{\sum_{k=1}^{s} n_k} \sum_{k=1}^{s} \sum_{i=1}^{n_k} \sum_{j=1}^{m} x_{ijk} \ , \tag{1}$$

$$\min g = 1 - \left[\frac{1}{\sum_{i=1}^{n_k} \sum_{j=1}^{m} c_{ij}} \sum_{k=1}^{s} \sum_{i=1}^{n_k} \sum_{j=1}^{m} c_{ij} x_{ijk} \right]. \tag{2}$$

Solution of a multiobjective problem is defined in [12] as a single-objective based on a utility value, following a decomposition strategy. For this reason, objective *g* is inverted and solved as a maximization objective. As a result, PSP is defined in the general model of expressions (3)-(5):

$$\text{maximize } z = f + g \,, \tag{3}$$

Subject to:

$$\sum_{j=1}^{m}\sum_{i=1}^{n_k} c_{ij}x_{ijk} \leq a_k \qquad k = 1,2,...,s \tag{4}$$

$$x_{ijk} \in \{0,1\} \qquad i = 1,2,...,n,\ j = 1,2,...,m;\ k = 1,2,...,s \tag{5}$$

The *z* value of equation (3) has a one-value when all products have been assigned (*f* is optimal and the dominant objective); in the other hand, a zero-value indicates that *g* is the dominant objective. Expressions (4)–(5) establishes constrains of available funds in the integer model.

5 The Ant Colony System Algorithm

The Ant Colony System (ACS) algorithm [13] is a well-known method to solve graph-based problems. Construction procedures of solutions in ACS are based on selection of arcs (*i*,*j*) of a graph. Ants travel around the roads, leaving an amount of pheromone τ_{ij}, used to determine the desirability of the roads η_{ij}. These parameters are used by artificial ants to generate desirable routes, such as the feedback process of natural ants that looks for the shortest paths between the anthill and the food sources. Evolutive process (iterative) of ACS permits evaporation of pheromone trails to converge towards the most feasible routes, which optimize objectives of the problem. General ACS procedure is presented in Fig. 1.

```
1  Procedure ACS_Algorithm ()
2      Initialize_parameters (τ_ij , η_ij )
3      While(isReached(stopCriteria)) do
4          constructionProcedure( τ_ij , η_ij )
5          updateOfPheromoneTrailsProceduure( τ_ij )
6      End_of_while
7  End Procedure
```

Fig. 1. The AntColonySystem Procedure

The *constructionProcedure* in *ACS_Algorithm* builds routes with the desirable nodes in the problem using a transition rule. It defines a basic multi-objective ant colony system algorithm, defined in [1], and based on the multi-objective formulation of PSP. This algorithm creates solutions through of selection of arcs of *i* products that are purchased to the *j* suppliers, where selection of the next *i*-th product is randomly performed in each order P_k. When a product has been selected, the supplier *j* is chosen using the parameter q_0 of equation (6).

$$j = \begin{cases} \arg\max_{u \in N_i^k} \left\{ [\tau_{ij}]^{\alpha} * [\eta_{ij}]^{\beta} \right\} & q < q_0 \\ f(p_{ij}^k) & \text{otherwise} \end{cases} \quad (6)$$

When $q<q_0$, a deterministic selection is performed using the τ_{ij} and η_{ij} , and the constructive parameters α and β of ant algorithms; otherwise a roulette is executed through the computation of the function *f* of expression (7). This function is used to explore the neighborhood $N_k(i)$ of suppliers for the *i*-th product to be selected. This exploration is performed through the p_{ij} values.

$$p_{ij}^k = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha} [\eta_{il}]^{\beta}} & j \in N_i^k \\ 0 & \text{Otherwise} \end{cases} \quad (7)$$

When an ant chooses a feasible arc (*i*,*j*), local evaporation of pheromone is performed using the τ_0 values and the $\rho \in [0,1]$ parameter of expression (8). This process is executed while ants have feasible arcs to select in *constructionProcedure*.

$$\tau_0 \leftarrow (1-\rho)\tau_{ij} + \rho\tau_0 \quad (8)$$

Once those ants have completed their solutions in *constructionProcedure*, global updating of pheromone (the *updateOfPheromoneTrailsProceduure*) is performed according to equation (9).

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}\ ,\ \forall (i,j) \in T_k, T_k \subseteq P_k \quad (9)$$

Where $\Delta\tau_{ij}$ represents the amount of deposited pheromone, which is computed like a measure cumulative uniform of the selected products by an ant in an order T_k. Expressions (6)-(9) define the heuristics of the basic design for the ACS algorithm, commonly used in single-objective problems (such as the aggregation described in the PSP formulation). However, solution of PSP requires a diversification of the search in the solution space, needed to reach the best solutions according to the Pareto's efficiency principle. Reason why, the knowledge based on Pareto's approach is incorporated to the ACS design to solve the related multi-objective problem.

6 The Pareto Optimization approach

Pareto Optimization has been used in optimization to obtain a Pareto Optimal Set to solve multiobjective problems [14]. All solutions in the Pareto's set are non-dominated solutions. In this way, Pareto Ant Optimization establishes the set of non-dominated solutions with a number of ant colonies that have the same number of ants. In its evolutive process, solutions of ants are compared and the pheromone updating process

is applied to the non-dominated solutions. Pareto Ant Optimization defines two variants over the basic design of ACS to solve multi-objective problems. It is based on multi-objective heuristic rules that permit to guide the ants in the *constructionProcedure* over different regions of the solution space. The first variant (P-ACO[1]) defines a modified Δ rule used in global updating of pheromone. It is computed through equation (10).

$$\Delta\tau_{ij} = \frac{1}{n_k}\left(\frac{f+g}{\theta}\right)^2 \tag{10}$$

The Δ values of expression (10) introduce a profit/cost relationship between objectives using a θ value, used according to [15] as a balancing parameter in selection of arcs. Where n_k is the size of the problem. The $\Delta\tau_{ij}$ values in global updating process ensure a faster convergence for ant algorithms in based-graph problems. However, an appropiated θ value can determine a better efficiency in solutions of a MOP.

The second variant (P-ACO[2]) consists of introducing pheromone values per each objective τ_{ij}^k. In addition of pheromone values, also heuristic values η_{ij}^k are added according to the *k*–th objective. This strategy permits a further exploration for each single-objective in cases where arcs are selected in non-deterministic way. It is used to determine solutions in the Pareto's front. Consequently, P-ACO[2] algorithm defines the τ_{ij}^f, τ_{ij}^g, η_{ij}^f and η_{ij}^g values, which represents the uniform profit/cost values for objectives in PSP. These parameters define the multi-objective selection rule of expression (11), which defines the third form to compute solutions in the *constructionProcedure* of ACS.

$$p_{ij}^k = \frac{\left[\left(\tau_{ij}^f\right)^{1-\lambda}\left(\tau_{ij}^g\right)^{\lambda}\right]^{\alpha}\left[\left(\eta_{ij}^f\right)^{1-\lambda}\left(\eta_{ij}^g\right)^{\lambda}\right]^{\beta}}{\sum_{l\in N_i^k}\left[\left(\tau_{ij}^f\right)^{1-\lambda}\left(\tau_{ij}^g\right)^{\lambda}\right]\left[\left(\eta_{ij}^f\right)^{1-\lambda}\left(\eta_{ij}^g\right)^{\lambda}\right]} \tag{11}$$

Where $\lambda \in [0,1]$ represents the relative importance of the different objectives according to [16]. Selection of arcs (*i*,*j*) related to the heuristic rule of equation (11) implies that updating pheromone requires a multi-objective definition in design of the P-ACO[2] variant. Therefore, $\Delta\tau_{ij}^k$ values are introduced as a performance measure of the current solution with regard to the *k* objectives of the problem, used in global upgrading of pheromone. Once that all arcs (*i*,*j*) are selected in *constructionProcedure*, the P-ACO[2] variant performs an upgrading rule (incorporated in *updateOfPheromoneTrailsProceduure*) that uses the Δ values per each *k* objective ($\Delta\tau_{ij}^f$ and $\Delta\tau_{ij}^g$), defining a dual updating process according to expression (9). Where the Δ values are computed like the inverse of the maximum profit $\Delta\tau_{ij}^f$ and minimum

cost $\Delta\tau_{ij}^{g}$ (best ant solutions) respectively. These variants in the multi-objective scene diversify the construction of solutions for PSP, providing to the ACS algorithm different exploration degrees to build feasible solutions for the orders (T_k) according to the formulation of PSP.

7 Architecture of Solution

The proposed approach follows the architecture of Fig. 2. In which the constructive process of purchasing schedules of PSP is described. Architecture consists of two modules: Preprocessing and Optimization. Preprocessing module is used to extract information of PSP sets of a database model (proposed in [17]). This action generates a PSP input instance which consists of a plain-text file, used to establish the solver independent to the database. It permits the use of the architecture in several purchasing scenarios, giving support to the staff of the purchasing personal.

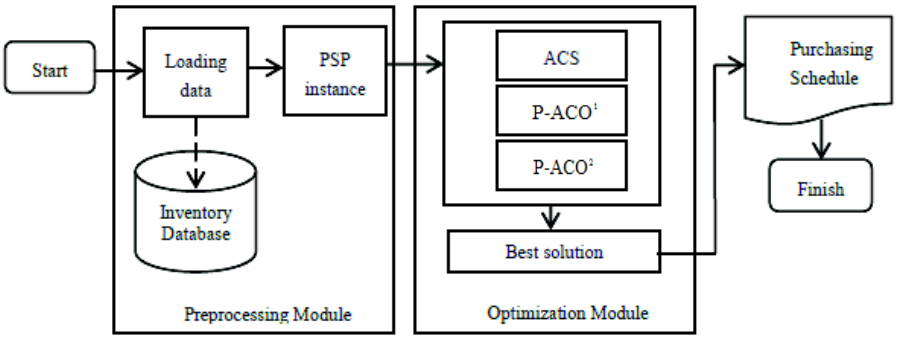


Fig. 2. Proposed solution methology.

Optimization module receives a PSP input instance and executes the basic ACS defined in [1], and the two Pareto-based variants described in this work. In each execution, the best solution of a determined algorithm is presented like a purchasing schedule to decision makers. It represents an optimized solution with regard to the objectives and constrains of PSP, whose efficiency is then analyzed by the purchasers to establish the decision to buy.

8 Experiments and Results

Due to real instances of PSP were unavailable, a dataset of ten orders was built using a pseudo-random number generator. It uses the queries of web catalogs, stored in a model inventory database. The generator creates the orders with different prices and suppliers for products, maximum and minimum prices for products and available funds.

Parameters of generator are shown in Table 1. It supposes an inventory cycle with ten orders (purchasers), where some products have costs more expensive than available funds.

Table 1. Input parameters for instances generated for PSP.

Orders (k)	Number of Products (n_k)	Min price	Max price	Available Funds
1	126	14.00	10999.00	25000.00
2	123	73.00	60000.00	50000.00
3	63	29.00	120000.00	40000.00
4	146	99.90	15980.00	65000.00
5	70	3.00	60000.00	30000.00
6	194	56.90	20000.00	80000.00
7	128	75.00	18799.00	75000.00
8	119	95.00	18000.00	55000.00
9	108	3.00	88996.00	48000.00
10	126	14.00	11499.00	40500.00

Instances were solved in an Apple MacBook Pro device model A1286, four-core processor (2.4 Ghz per core), 8 GB of RAM memory, 750 GB hard disk under Mac OS X 10.9 Mavericks. ACS and its multiobjective variants (P-ACO[1] and P-ACO[2]) were developed in Java Standard Edition 8 with Eclipse Luna. At each execution, an accumulated sum ($\sum$) is stored with the number of times in which the values of best solutions are reached. It indicates the exploration degree of each algorithm.

Table 2. Solutions of PSP with ACS and ACO variants based on Pareto approach.

Algorithm	Iterations	1000	5000	10000	15000	20000	30000
ACS	f	0.8301	0.8513	0.8431	0.8212	0.8205	0.8452
	g	0.1150	0.1005	0.1030	0.1278	0.1288	0.1026
	$\sum$	22.31	24.17	23.32	26.47	25.22	25.53
	t	895.75	8753.96	15830.87	9840.30	18963.43	16732.78
P-ACO[1]	f	0.8643	0.8513	0.8216	0.8807	0.8895	0.8772
	g	0.0818	0.0963	0.1230	0.0795	0.0751	0.0894
	$\sum$	16.45	12.29	18.73	17.55	17.34	22.73
	t	667.71	5542.52	8754.93	5503.87	13927.54	29453.01
P-ACO[2]	f	0.8801	0.8582	0.8870	0.8925	0.8858	0.8993
	g	0.0849	0.0991	0.0802	0.0755	0.0763	0.0722
	$\sum$	28.72	23.55	30.05	32.92	28.64	30.63
	t	1350.64	6922.03	22532.98	25073.84	21569.23	27954.91

Additionally, average time computation is measured (*t*) in which best solutions of ACS are reached, and the *f* and *g* average values for each instance. Table 2 shows the performance for ACS and Pareto-ACO variants for six tests with 30 executions with: 1000, 5000, 10000, 15000, 20000 and 30000 iterations for each algorithm. Parameter values established to test the algorithms were: $q_0 = 0.5$, ρ=0.1, $\theta = 0.5$, $\lambda = 0.2$,$\alpha = 1$ and $\beta = 2$. In each execution, the best solution is storage to establish the average performance that is presented in Table 2.

Results of Table 2 indicate that ACS reaches a 72% in average of *z* values, in an average time of 12 seconds. It presents a variation coefficient of 0.76 with a Pearson Correlation Coefficient of -0.98. P-ACO[1] algorithm improves in average 5% the results of ACS, but according to the $\sum$ value, the search is more directed towards a faster convergence ($\sum$ average value is less than ACS). It demonstrates that P-ACO[1] variant over the ACS algorithm is able to improve the solutions of PSP. Even though P-ACO[1] reaches a variation coefficient of 0.81, the Pearson correlation is established in -0.94. It reveals a slow deviation P-ACO[1] directing the search in the Pareto front with regard to ACS. However, results demonstrate P-ACO[1] variant diversifies the search and reaches a faster convergence than ACS.

In the other case, results of the P-ACO[2] algorithm demonstrates that introducing pheromone values per each objective give to the ACS enough exploration degree to improve objective results of ACS by 8%, and 3% in average results of the ACS and P-ACO[1] variants, reaching the 80% in average in *z* values. Effects of this are shown in the $\sum$ column. This exploration average is supported by a variation coefficient of 0.83 and a correlation coefficient of -0.98, which establishes a diversified and further search in the Pareto Front. Efficiency of P-ACO[2] can be also observed in the average time of computation for the best solutions (17 seconds in average). Although it presents slower convergence, it is proved that P-ACO[2] variant represents the best alternative when testing configurations for the ant algorithms, described in terms of Pareto efficiency solving PSP.

9 Conclusions and Future Works

In this paper, the Purchasing Scheduling Problem was approached with three variants of the Ant Colony System Algorithm. The first variant (ACS) represents an efficient strategy when the search in the problem looks for a single objective, providing good solutions. In the same manner, it was proved that hybridization of the ACS algorithm using the Pareto principles is helpful in discovering different regions of the solution space, giving better solutions with the test parameters.

Consequently, an alternative to compute the Pareto optimal values can be approached using some neighborhood techniques, such as the classical 3-Opt and Cross Exchange operators over the P-ACO[1] and P-ACO[2] algorithms, well-known operators that usually improve results of the ACS algorithm. Additionally, results of P-ACO[1] and P-ACO[2] algorithms to determine the importance of the constructive parameters (θ and λ) in the proposed Pareto's approach.

Efficiency of PSP solutions and speed of computation show the advantages of developing evolutionary algorithms to integrate them in complex decision-making systems. They can be used as planning tools to develop ERP systems (Enterprise Resource Planning), reliable information technology resources to implement in industrial and organizational environments.

References

1. Delgado, O. et al.: An Ant Colony System Metaheuristic for Solving a Bi-Objective Purchasing Scheduling Problem. Research in Computing Sciences 82, pp. 21–30 (2014)
2. Zahra, L., Shahnorbanun, S., Muriati, M.: A Product Quality-Supply Chain Integration Framework. Journal of Applied Sciences 13(1):36–48 (2013)
3. Wallin, C. et al.: What is the "right" inventory management approach for a purchased item? International Journal of Operations and Production Management 26(1):50–68 (2006)
4. Nagurney, A., Yu, M.: Supply Chain Network Design under Profit Maximization and Oligopolistic Competition. Transportation Research 46:281–294 (2010)
5. Hang, Y., Fong, S, Yan, Z.: CSET automated negotiation model for optimal supply chain formation. World Review of Science, Technology and Sustainable Development 7(1):67–78 (2006)
6. Guo, H. et al.: Product Feature Categorization with Multilevel Latent Semantic Association. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1087–1096 (2009)
7. Hall, N.: The Warehouse-Inventory-Transportation Problem for Supply Chains. Innovative European Journal of Operation Research 237:690–700 (2014)
8. Doerner, K. et al.: Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. Annals of Operation Research 131(4) (2004)
9. Kumar, R. et al.: Evolution of Hyperheuristics for the Biobjective 0/1 Knapsack Problem by Multiobjective Genetic Programming. In: Conor Ryan & Maarten Keijzer, Ed., GECCO, ACM, pp. 1227–1234 (2006)
10. Falkenauer, E.: A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics, 2:5–30 (1996)
11. Martello, S., Toth, P.: Lower bounds and reduction procedures for the bin packing problem. Discrete Applied Mathematics 28:59–70 (1990)
12. Hamdi, A., Mukheimer, A.: Modified Lagrangian Methods for Separable Optimization Problems. Abstract and Applied Analysis, Vol. 2012, pp. 1–20 (2012)
13. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, pp. 53–66 (1997)
14. Doerner, K. et al.: Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. Annals of Operations Research (2004)
15. Chaharsooghi, S., Meimand Kermani, A.H.: An effective ant colony optimization algorithm (ACO) for multi-objective resource allocation problem (MORAP). Applied Mathematics and Computation, vol. 200, pp. 167–177 (2008)
16. Zhao, F., Tang, J., Tang, Y.: A New Approach based on Ant Colony Optimization to determine the Supply Chain Design for a Product Mix. Journal of Computers 7(3):736–743 (2012)
17. Coronel, J.A.: Solution of Assignation problem in purchasing of goods using metaheuristic algorithms. Bachelor's Thesis, Universidad del Mar campus Puerto Escondido (2014)

The Pickup and Delivery Problem: a Many-objective Analysis

Abel García-Nájera and Antonio López-Jaimes

Universidad Autónoma Metropolitana, Unidad Cuajimalpa,
Departamento de Matemáticas Aplicadas y Sistemas, México, D.F.,
Mexico

{agarcian,alopez}@correo.cua.uam.mx

Abstract. The pickup and delivery problem (PDP) considers a set of transportation requests, which specify the quantity of product that has to be picked up from an origin and delivered to a destination. There exist a number of vehicles available to be used for completing these tasks. PDP consists of finding a collection of routes with minimum cost, such that all transportation request are serviced. Traditionally, cost has been associated with the number of routes and the total travel distance. However, in many applications, some other objectives emerge, for example, the minimization of travel time and the maximization of the collected profit. If we consider all these four objectives equally important, PDP can be tackled as a many-objective problem. In this paper we are interested in analyzing this many-objective problem in order to study some of its properties, specifically, (*i*) the change of difficulty when the number of objectives is increased, and (*ii*) the conflict degree between each pair of objectives. In order to analyze these topics, we compare the performance of a recently proposed multi-objective evolutionary algorithm against that of the well-known ϵ-MOEA, which has shown good results in many-objective problems.

Keywords: Many-objective optimization, pickup and delivery problem, evolutionary algorithm

1 Introduction

The pickup and delivery problem (PDP) is part of the class of problems known as vehicle routing problem (VRP) [7], which is well-known to be NP-hard [9]. The PDP considers transportation requests, which are defined between pairs of customers (origin and destination). The problem consists in designing a set of routes with minimum cost to service all transportation requests.

Cost is regularly associated with the number of routes and the travel distance, however, there are several other sources of cost [6]. One generalization of the PDP is the PDP with time windows and selective requests (PDPTWSR), which considers two additional sources of cost, namely the travel time and the

uncollected profit. Moreover, if those four objectives are considered to be equally important, PDPTWSR can be tackled as a many-objective problem.

The PDPTWSR is defined as follows. There is a set $\mathcal{V} = \{0, 1, \ldots, N, N + 1, \ldots, 2N\}$ of $2N + 1$ vertices. Customers are represented by the vertices in the subset $\mathcal{V}' = \mathcal{V} \setminus \{0\}$. Each customer $i \in \mathcal{V}'$ is geographically located at coordinates (x_i, y_i), and has an associated time window $[b_i, e_i]$, during which it has to be serviced, and a service time s_i required to load or unload goods. Customer subset $\mathcal{V}_\mathcal{O} = \{1, \ldots, N\}$ corresponds to the pickup locations, while customer subset $\mathcal{V}_\mathcal{D} = \{N + 1, \ldots, 2N\}$ represents the delivery locations.

The set $\mathcal{TR} = \{1, \ldots, N\}$ represents N transportation requests. Each transportation request $i \in \mathcal{TR}$ specifies the size q_i of the load to be transported and the locations $j \in \mathcal{V}_\mathcal{O}$ and $k \in \mathcal{V}_\mathcal{D}$ where the load will be collected (origin) and delivered (destination), respectively. Finally, each transportation request $i \in \mathcal{TR}$ has an associated profit p_i, hence $P = \sum_{i \in \mathcal{TR}} p_i$ is the total possible profit.

The vertex 0 is located at (x_0, y_0) and has a time window $[0, e_0 \geq \max\{e_i : i \in \mathcal{V}'\}]$. This vertex is the base of a homogeneous fleet of vehicles which have capacity Q, greater or equal to the maximum size of the loads to be transported.

The travel between vertices i and j has associated costs, such as the distance d_{ij} (relating to fuel cost) and travel time t_{ij} (relating to driver salary). Transportation requests are optional to be serviced, this means that origin and destination customers associated to a request might not be visited. Therefore, if they are not visited, there is no profit collected from that request. For the benchmark problems to be considered later, unit velocity and direct travel are assumed, so the times and distances are both simply taken to be the Euclidean distances. Moreover, the profit p_i associated to transportation request $i \in \mathcal{TR}$ will be equaled to the size q_i of the transportation load. For real-world problems, however, the distances d_{ij} are unlikely to be Euclidean, the travel times t_{ij} are unlikely to be simply related to the distances, and profit does not necessarily involves only the load to be transported.

The aim of the problem is to find the set of a minimum number routes which minimize the total cost, the travel distance and the uncollected profit, such that: (i) each route starts and ends at the base, (ii) customers related to each transportation request are visited by only one vehicle or none, (iii) the vehicle load must not, at any time, be negative and must not exceed the vehicle capacity Q, (iv) for each request i, its corresponding pickup location p_i must be visited in the same route and before its corresponding delivery location d_i, and (v) for each request i, its corresponding delivery location d_i must be visited in the same route and after its corresponding pickup location p_i.

Having defined the PDPTWSR, we can define four key objectives this paper will concentrate on minimizing, namely the number of routes or vehicles (f_R), the total travel distance (f_D), the total travel time (f_T), and the uncollected profit (f_P), subject to the constraints explained above.

To the best of our knowledge, the problem under study has not been tackled before. However, the pickup and delivery problem with time windows (PDPTW) has been subject of plenty of investigation. Many approaches for solving the

PDPTW can be found in the literature and Parragh et al. [11] make an excellent survey of many of them. We will review some previous studies which have tested their approaches on the PDPTW benchmark sets of Li and Lim [10].

Li and Lim [10] introduced a metaheuristic based on a tabu-embedded simulated annealing algorithm, which restarts a search procedure from the current best solution after several non-improving search iterations. This restart strategy guides the local search in three neighborhoods defined to solve the general multiple-vehicle PDPTW. This is combined with a metaheuristic based on a K restarts annealing procedure with tabu-list to avoid cycling in the search process. In addition, authors generated several benchmark instances which are used in this study. Bent and Van Hentenryck [1] proposed a two-stage hybrid algorithm. The first stage uses a simple simulated annealing algorithm to decrease the number of routes, while the second stage uses a large neighborhood search to decrease the total travel distance. Ropke and Pisinger [12] presented an adaptive large neighborhood search, which is and extension to the large neighborhood search and the ruin-and-recreate heuristic. The proposed method is composed of a number of competing subheuristics that are used with a frequency corresponding to their historic performance. Hasle and Kloster [5] introduced SPIDER, which is a heuristic approach based on local search. This approach has three phases: construction of initial solutions, tour depletion, and iterative improvement. The construction phase considers extensions of classical constructions heuristics as well as other methods proposed by the authors. In the tour depletion phase, a greedy tour removal heuristic is invoked. A single tour is depleted, and insertion of the unassigned orders in the remaining tours is attempted. The new solution is accepted if all unassigned orders are successfully inserted in the remaining tours. Finally, the iterative improvement phase is based on variable neighborhood descent, using a selection of several intra-tour, inter-tour, and special operators.

In the above reviewed studies, the PDP has been solved considering the minimization of the number of routes first, and then, the total travel distance. More recently, García-Nájera and Gutiérrez-Andrade [3] proposed a multi-objective evolutionary approach for solving the PDPTW, minimizing the number of routes, the travel distance, and the travel time simultaneously. Their approach was able to find many best-known solutions to benchmark instances and outperformed a well-known multi-objective optimizer.

As far a we are concerned, the study of García-Nájera and Gutiérrez-Andrade [3] is the only regarding the solution of the PDPTW considering multiple objectives. The present study aims at remedy this situation, that is, the aim is to analyze the many-objective problem in the sense of how difficult the problem is when more objectives are considered.

The remainder of this paper is organized as follows. Section 2 introduces the main concepts of multi-objective optimization and explains the performance metrics that are used here to compare algorithms. The multi-objective optimizers used in this study are briefly described in Section 3. Then, Section 4 presents the analysis of the results from both algorithms. Finally, we present our conclusions in Section 5.

2 Multi-objective Optimization Problems

Any multi-objective optimization problem can be defined, without loss of generality, as the problem of minimizing $\mathbf{f}(\mathbf{x})$, subject to $g_i(\mathbf{x}) \leq 0, \forall\ i \in \{1,\ldots,p\}$, and $h_j(\mathbf{x}) = 0, \forall\ j \in \{1,\ldots,q\}$, where $\mathbf{x} \in \mathcal{X}$ is a potential solution to the problem, $\mathcal{X}$ is the domain of solutions, $\mathbf{f} : \mathcal{X} \rightarrow \mathbb{R}^m$ are the m objective functions, and the constraint functions $g_i, h_j : \mathcal{X} \rightarrow \mathbb{R}$ delimit the feasible search space.

We say that solution $\mathbf{x}$ *dominates* solution $\mathbf{y}$, written as $\mathbf{x} \prec \mathbf{y}$, if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall\ i \in \{1,\ldots,m\}$, and $\exists\ j : f_j(\mathbf{x}) < f_j(\mathbf{y})$. Consequently, solution $\mathbf{x} \in \mathcal{S} \subseteq \mathcal{X}$ is *non-dominated* with respect to $\mathcal{S}$ if there is no solution $\mathbf{y} \in \mathcal{S}$ such that $\mathbf{y} \prec \mathbf{x}$. Solution $\mathbf{x} \in \mathcal{X}$ is said to be *Pareto optimal* if it is non-dominated with respect to $\mathcal{X}$, and the *Pareto optimal set* is defined as $\mathcal{P}_s = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{x} \text{ is Pareto optimal}\}$. Finally, the *Pareto front* is defined as $\mathcal{P}_f = \{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m \mid \mathbf{x} \in \mathcal{P}_s\}$.

In contrast, with single-objective problems, where one can straightforwardly compare the best solutions from the various approaches studied, multi-objective problems have to compare whole sets of solutions. Many performance indicators have been proposed in the literature, being two of them the *hypervolume* and the *generational distance*, which are explained next.

2.1 Hypervolume Indicator

The hypervolume performance metric $\mathsf{H}(\mathcal{A}, \mathbf{z})$ concerns the size of the objective space defined by the approximation set $\mathcal{A}$, which is limited by setting a suitable reference point $\mathbf{z}$. Formally, for a two-dimensional objective space $\mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), f_2(\mathbf{x})\big)$, each solution $\mathbf{x_i} \in \mathcal{A}$ delimits a rectangle defined by $\big(f_1(\mathbf{x_i}), f_2(\mathbf{x_i})\big)$ and the reference point $\mathbf{z} = (z_1, z_2)$, and the size of the union of all such rectangles is used as the measure. This concept can be extended to any number of dimensions m to give the general hypervolume metric [13].

2.2 Generational Distance Indicator

In order to evaluate the convergence of the algorithms to the Pareto front we adopted the generational distance indicator GD, which is defined by $\mathsf{GD}(\mathcal{A}) = 1/|\mathcal{A}|\big(\sum_{i=1}^{|\mathcal{A}|} d_i^2\big)^{1/2}$, where d_i is the Euclidean distance between each solution $\mathbf{x_i} \in \mathcal{A}$ and the nearest member of $\mathcal{P}_f$.

3 Many-objective Optimization of the PDPTWSR

In this section, the two multi-objective optimizers used in this study for solving the many-objective PDPTWSR are briefly described. The first is the well-known ϵ-MOEA [2], which has been proved to be successful in a number of applications. The second is the multi-objective evolutionary algorithm recently proposed by García-Nájera and Gutiérrez-Andrade [3], hereinafter GN-MOEA, which is able to find appropriate Pareto approximations to the related problem PDPTW.

3.1 ϵ-MOEA

Deb et al. [2] proposed ϵ-MOEA, based on the ϵ-dominance concept introduced in [8] that states: Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$. Then, $\mathbf{x}$ is said to ϵ-dominate $\mathbf{y}$ for some $\epsilon > 0$, denoted as $\mathbf{x} \succ_\epsilon \mathbf{y}$, if and only if $(1+\epsilon) \cdot x_i \geq y_i, \forall\, i \in \{1, \ldots, m\}$.

In ϵ-MOEA, the search space is divided into a number of hyper-boxes and diversity is maintained by ensuring that a hyper-box can be occupied by only one solution. ϵ-MOEA randomly initializes a population. The non-dominated solutions of are copied to an archive population. Two solutions are chosen as parents: one is chosen from the population and one is chosen from the archive population. One offspring solution is created using these parents. If the offspring dominates one or more population members, then the offspring replaces one of them (chosen at random). On the other hand, if any population member dominates the offspring, it is not accepted. When both the above tests fail (that is, the offspring is non-dominated to the population members), the offspring replaces a randomly chosen population member, thereby ensuring that the EA population size remains unchanged. For the offspring to be included in the archive population, the offspring is compared with each member of the archive using ϵ-dominance criterion.

3.2 GN-MOEA

García-Nájera and Gutiérrez-Andrade [3] proposed a problem-specific multi-objective evolutionary algorithm (GN-MOEA) for minimizing three objectives of the PDPTW simultaneously, namely the number of routes, the travel distance, and the travel time. GN-MOEA uses an encoding of list of lists: a route is encoded as a list and a solution as a list of routes. They use the non-dominance sorting criterion [4] to assign fitness to individuals. Solution similarity is used as a diversity measure. This is simply computed as the ratio of the number of arcs that are common in two solutions to the total number of arcs traversed in both solutions. This measure is used in the mating selection process, since one parent is selected according to the diversity measure and the other is selected according the fitness. These parents are selected by using the tournament selection method. Crossover aims at combining routes from both parents, while mutation exchanges transportation requests between routes, and removes transportation requests from one route and inserts them into another.

4 Experimental Study

Our study has two main purposes. Firstly, to determine which of the two multi-objective optimizers described in the previous Section has a better performance on the problem at hand, and secondly, to perform a many-objective analysis of the PDPTWSR. To this end, we carried out two sets of experiments. The first set (RDT) considered the objectives number of routes (f_{R}), travel distance (f_{D}), and travel time (f_{T}), and the minimization of four combinations of these

objetives, namely $f_R f_D$, $f_R f_T$, $f_D f_T$, and $f_R f_D f_T$. The second set of experiments (RDTP), took into account the three previous objectives plus the uncollected profit (f_P), and the minimization of five combinations of these four objectives, namely $f_R f_P$, $f_R f_D f_P$, $f_R f_T f_P$, $f_D f_T f_P$, and $f_R f_D f_T f_P$.

In order to have controlled experiments, we used the PDPTW benchmark sets of Li and Lim [10], which include 56 instances of size $N = 100$, and are divided into six categories: lc1, lc2, lr1, lr2, lrc1, and lrc2. Crossover and mutation operators used in GN-MOEA were set also in ϵ-MOEA for fair comparison. We ran both algorithms 30 times for each problem instance. The GN-MOEA parameters were set to the values reported in [3]: population size = 100, number of generations = 500, tournament size = 5, and mutation probability = 0.1. These values were also set in ϵ-MOEA, plus the number of hyper-boxes = 100.

4.1 Analysis of the Hypervolume Indicator

To compute the hypervolume indicator, we require an appropriate reference point. From the 60 Pareto approximations (30 from GN-MOEA and 30 from ϵ-MOEA) for each instance, we took the maximum value for each objective, and the reference point $\mathbf{z}$ was set 10% above each dimension's maximum value.

For each instance and repetition, we took the non-dominated set and computed the hypervolume covered by those solutions. Then, we applied a statistical t-test (two-sample, one-tailed, unequal variance) to the two vectors of 30 hypervolume values, from the GN-MOEA and ϵ-MOEA, respectively, to test the null hypothesis that data in the vectors are independent random samples from normal distributions with equal means. The summary of the t-test results are shown in Tables 1 and 2, for the sets of experiments RDT and RDTP, respectively. The first main column of these Tables show the instance category and the number of instances comprising that category. These Tables have one main column for each combination of objectives, and each main column has two subcolumns, corresponding to the number of instances in each instance category for which non-dominated solutions from GN-MOEA (GN) and from ϵ-MOEA (ϵ), respectively, covered a statistically larger hypervolume.

From Table 1 we can observe that, in the $f_R f_D$ case, both algorithms performed similarly, since for only three out of 56 instances there was a statistical difference in the size of the covered hypervolume. For the remaining three combinations of objectives, it is clear that GN-MOEA found non-dominated solutions that covered a significantly larger hypervolume for more instances than ϵ-MOEA.

On the other hand, from Table 2 we can see that the non-dominated solutions from ϵ-MOEA covered a significantly larger hypervolume for many instances in the cases $f_R f_P$, $f_D f_T f_P$, and $f_R f_D f_T f_P$, while the Pareto approximations from GN-MOEA delimited a significantly larger hypervolume in more instances for the cases $f_R f_D f_P$ and $f_R f_T f_P$.

Overall we can conjecture that GN-MOEA has a superior performance over ϵ-MOEA when objective $f_R f_D f_T$ are considered, however, when objective f_P is included, ϵ-MOEA outperforms GN-MOEA in many cases.

Table 1. Number of instances for which non-dominated solutions from each optimizer cover a significantly larger hypervolume for the set of experiments RDT.

Inst.		$f_R f_D$		$f_R f_T$		$f_D f_T$		$f_R f_D f_T$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	1	0	3	1	1	0	2	0
lc2	8	0	0	3	0	2	0	2	0
lr1	12	0	1	5	0	10	0	5	0
lr2	11	0	0	6	0	10	0	5	1
lrc1	8	0	1	1	0	4	0	3	0
lrc2	8	0	0	6	0	6	0	2	0
Total	56	1	2	24	1	33	0	19	1

Table 2. Number of instances for which non-dominated solutions from each optimizer cover a significantly larger hypervolume for the set of experiments RDTP.

Inst.		$f_R f_P$		$f_R f_D f_P$		$f_R f_T f_P$		$f_D f_T f_P$		$f_R f_D f_T f_P$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	0	3	3	2	9	0	3	3	0	9
lc2	8	0	5	5	0	8	0	1	4	0	5
lr1	12	0	3	0	7	0	9	0	12	0	12
lr2	11	0	6	5	0	10	0	0	11	0	9
lrc1	8	1	5	2	3	0	5	0	7	0	8
lrc2	8	0	5	3	0	6	0	0	8	0	7
Total	56	1	27	18	12	33	14	4	45	0	50

4.2 Analysis of the Generational Distance Indicator

Since the optimal Pareto front is not known for the benchmark instances, for computing GD we used, as a reference set, the non-dominated solutions resulting from the union of all the approximation sets to $\mathcal{P}_f$ obtained by each algorithm at the end of every run. In order to analyze the difficulty of PDPTWSR when objectives are added we used the comparison of both algorithms to get useful insights about its difficulty. More specifically, if the difference between the performance of the optimizers vary with the number of objectives, that would indicate that PDPTWSR's difficulty vary with the number of objectives.

After computing GD for each approximation set obtained by both algorithms we carried out the Wilcoxon rank-sum test (two-sample, one-tailed) to determine which algorithm yielded the smaller generational distance. We tested the alternative hypothesis that the mean of GD generated by optimizer A is less than that of B. In this test we employed a significance level of 5%. In Table 3 we present a summary of the statistical tests showing the number of instances in which each MOEA achieved a significantly better GD value. For the set of experiments RDTP in Table 3 we can observe that GN-MOEA outperformed ϵ-MOEA in most of the instances. Interestingly, the result for $f_R f_P$ does not agree with that of the hypervolume. By analyzing some approximation sets we realized that in some instances, although GN-MOEA generated solutions very

Table 3. Number of instances for which non-dominated solutions from each optimizer has a significantly better GD for the set of experiments RDTP.

Inst.		$f_R f_P$		$f_R f_D f_P$		$f_R f_T f_P$		$f_D f_T f_P$		$f_R f_D f_T f_P$	
Cat.	No.	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ	GN	ϵ
lc1	9	9	0	9	0	9	0	0	9	0	9
lc2	8	7	0	8	0	8	0	0	2	0	5
lr1	12	12	0	3	3	4	0	0	12	0	12
lr2	11	11	0	9	0	11	0	0	9	0	9
lrc1	8	8	0	7	0	6	0	0	7	0	7
lrc2	8	8	0	8	0	8	0	0	7	0	7
Total	56	55	0	44	3	46	0	0	46	0	49

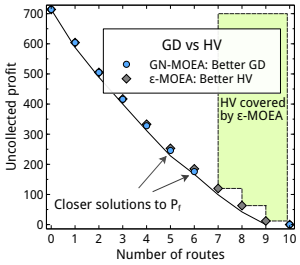


Fig. 1. Approximation sets where GN-MOEA achieves a good GD and ϵ-MOEA a good HV.

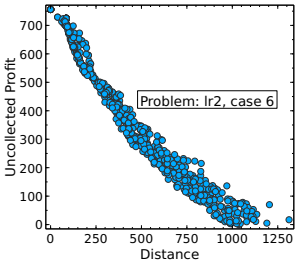


Fig. 2. Conflict between objectives distance and uncollected profit.

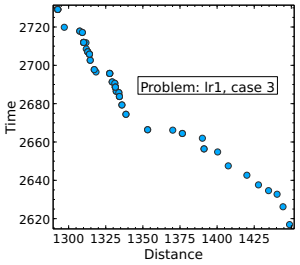


Fig. 3. Conflict between objectives distance and time.

near to the reference set, it did not cover some solutions in the middle of it. In order to illustrate this situation, we present Figure 1, in which solutions with 7, 8 and 9 routes were not found by GN-MOEA. In this approximation sets, GN-MOEA achieved a better GD (4.7 vs 5.15). However, with the same outputs, ϵ-MOEA overcame GN-MOEA in terms of hypervolume. The explanation is that the hypervolume contribution of those middle solutions was more important than the hypervolume advantage produced by solutions close to the reference set. Regarding the difficulty of PDPTWSR as the number of objective increases we have the following observations. In some comparative studies, ϵ-MOEA has shown a good performance in many-objective optimization problems. That is, its convergence ability degrades very slowly when more objectives are added. Therefore, we expect ϵ-MOEA to maintain a similar performance for every number of objectives. The results show that, regarding both hypervolume and generational distance, GN-MOEA outperformed ϵ-MOEA in most of the problem instances with 2 and 3 objectives. In contrast, ϵ-MOEA obtained better indicator values in $f_D f_T f_P$ and $f_R f_D f_T f_P$. We argue that GN-MOEA's convergence ability was affected by the additional objective. Therefore, this could indicate that the difficulty of PDPTWSR increases as more objectives are added.

4.3 Conflict among Objectives of PDPTWSR

One can expect a natural conflict between uncollected profit (f_P) and either time (f_T) or distance (f_D). However, at first sight, there is no conflict between distance and time. In order to estimate the conflict among all pair of objectives we employed the Spearman's rank correlation coefficient (which value is in $[-1, 1]$). That is, a coefficient value close to -1 means that one objective grows while the other decreases. For this purpose, we test the alternative hypothesis (significance value of 0.05) that the Spearman's rank coefficient is negative for a pair of objective values. The results show that, according to correlation, there is conflict between time (f_T) and distance (f_D) for many of the instances: 0, 3, 9, 11, 7, 8 for instance categories lc1, lc2, lr1, lr2, lrc1, lrc2, respectively. In the other hand, there is no conflict at all between f_R and either f_T or f_D, and there is conflict between f_P and f_R, f_D, f_T in all instances. In order to illustrate this situation we present Figures 2 and 3 in which the Pareto front of the best approximation of $\mathcal{P}_f$ is plotted.

5 Conclusions

In this paper we have analyzed some properties of the many-objective PDP, namely the change of difficulty when the number of objectives is increased, the conflict degree between each pair of objectives, and whether the difficulty of particular objectives decreases due to the change of the fitness landscape. To this end, we employed two multi-objective optimizers: the well-known ϵ-MOEA, which has been successful in a number of applications, and the recently proposed GN-MOEA, which showed improved performance over a popular optimizer on some benchmark instances of the PDP with time windows. We ran two sets of experiments: the first to optimize different combinations of the objectives number of routes, travel distance and travel time, and the second to optimize different combinations of the previous objectives plus the uncollected profit.

Our analysis is threefold. First, we computed the hypervolume covered by the non-dominated solutions found by both algorithms. For the first set of experiments, we found that GN-MOEA has a better performance over ϵ-MOEA in three out of four combinations of objectives, and, for the remaining case, there is no difference between both algorithms. For the second set of experiments, ϵ-MOEA outperformed GN-MOEA in three out of five combinations of objectives, and GN-MOEA has a better performance in the remaining two cases.

Secondly, we computed the generational distance the non-dominated solutions found by both algorithms. In this case, GN-MOEA solutions have a shorter generational distance than those from ϵ-MOEA for all four combinations of objectives in the first set of experiments and for three combinations of objectives in the second set, and solutions from ϵ-MOEA have a shorter generational distance for the remaining two combinations of objectives in the second set. These results are consistent with those from the hypervolume, except for the combination $f_{\mathrm{R}} f_{\mathrm{P}}$, for which ϵ-MOEA found solutions with a larger hypervolume and GN-MOEA solutions have a shorter generational distance. After analyzing

these results, we can argue that this situation is due to that, although GN-MOEA generated solutions very near to the reference set, it did not cover some solutions in the middle of it.

Finally, we analyzed the conflict between objectives. We found that there is no conflict at all between f_{R} and either f_{D} or f_{T}, however objectives f_{D} and f_{T} are in conflict in many instances, and objective f_{P} is in conflict with the other three objectives in all the 56 instances.

After these interesting results, we believe that we can continue with our research by investigating why ϵ-MOEA is not able to find better solutions than those from GN-MOEA for the first set of experiments, that is, what are the properties of GN-MOEA that make it a better solver when only objectives f_{R}, f_{D} and f_{T} are considered. To further analyze the many-objective performance of both algorithms, we are planning to include at least two additional objectives.

References

1. Bent, R., Van Hentenryck, P.: A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. In: Principles and Practice of Constraint Programming. pp. 123–137 (2003)
2. Deb, K., Mohan, M., Mishra, S.: Towards a quick computation of well-spread pareto-optimal solutions. In: Proceedings of EMO 2003. pp. 222–236. Springer (2003)
3. Garcia-Najera, A., Gutierrez-Andrade, M.A.: An evolutionary approach to the multi-objective pickup and delivery problem with time windows. In: 2013 IEEE Congress on Evolutionary Computation. pp. 997–1004. IEEE (2013)
4. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley (1989)
5. Hasle, G., Kloster, O.: Industrial vehicle routing. Geometric modelling, numerical simulation, and optimization pp. 397–435 (2007)
6. Jozefowiez, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. Eur. J. Oper. Res. 189(2), 293–309 (2008)
7. Laporte, G.: Fifty years of vehicle routing. Transport. Sci. 43(4), 408–416 (2009)
8. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evolut. Comput. 10(3), 263–282 (2002)
9. Lenstra, J.K., Rinnooy-Kan, A.H.G.: Complexity of vehicle routing and scheduling problems. Networks 11(2), 221–227 (1981)
10. Li, H., Lim, A.: A metaheuristic for the pickup and delivery problem with time windows. In: 13th International Conference on Tools and Artificial Intelligence. vol. 1, pp. 160–167. IEEE Computer Society (2001)
11. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems. J. Betriebswirtschaft 58(2), 81–117 (2008)
12. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transport. Sci. 40(4), 455–472 (2006)
13. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms – A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) PPSN V. LNCS, vol. 1498, pp. 292–304. Springer (1998)

Fuzzy System for Grade Assignment in Competence Assessment Based Educative Models

Fabio Tomás Moreno Ortiz[1,2], Antonio Hernández Zavala[2], and Omar Rodríguez Zalapa[1,2]

[1] Universidad Tecnológica de Querétaro, Mexico

[2] Instituto Politécnico Nacional, Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada Unidad Querétaro, Mexico

{famoreno,orodriguez}@uteq.edu.mx,anhernandezz@ipn.mx

Abstract. Mexico is adopting the competence-based model for education improvement. One of the major problems is to quantify the results of assessment to provide a grade considering subjective data. It commonly results in the assignation of an arbitrary grade which is estimated by the experience of the teacher using the evidence from the students. This paper presents a fuzzy system for calculating the grade assignment at undergraduate courses considering the student competences. The obtained results are compared against the grades that are calculated with the traditional average method and those obtained with the pass-fail checklists method.

Keywords: Fuzzy logic, competence education, assessment, scholar grade

1 Introduction

The twentieth-century educative models were focused more on teaching than learning. Today people need to develop the constant learning ability to be adapted to their life changing circumstances in a globalized environment. An overview of the educational paradigms in Colombia is shown in [1], which are very similar to those of many countries in Latin America.

The Mexican Technological Universities Subsystem (TUS) has adopted the professional competencies (PC) educative system for its educational programs since 2009. The competence-based education (CBE) system, proposes the integration of knowledge, competencies, and attitudes for preparing a student to solve problems throughout his life and when interacting with others. It is expected that a person could be adapted to changing contexts and show evidence of creativity, innovation, motivation, and values [2].

As CBE is a relatively recent model, there is still discussion on the development of curricula and assessment methods, so the need for research to face the challenges and limitations of the model exists [3, 4].

In the CBE, one of the most widespread methods for assessment in Latin America is the evidence portfolio [5, 6]. It is evaluated by observing the results of learning (ROL) and a grade is assigned in function of compliance checklists. For the TUS, the grade is given in alphabetical and numerical scale according to the following levels: SA (Satisfactory = 8), when the student has attained the ROL; DE (Prominent = 9), when the student has achieved the ROL and exceeds the requirements; AU (Autonomous = 10), when the student exceeds the ROL in different contexts [7]. When learning outcomes are not achieved, a grade of NA (Not Proficient) is assigned. The paradigm transition in assessment has been slow due to the habit of evaluating by closed tests which provide numerical results. Traditionally, the test results are averaged to calculate the final grade. In some cases, each periodical assessment is rated in different proportion corresponding to the "weight" it has into the content of the educational program.

Some major problems are quantifying and standardizing the results of the evaluation due to the linguistic nature of ROL, which causes subjectivities in the evaluation [8]. In the TUS the teachers have chosen one of two assessment methods: a) Continue with the traditional method using closed tests, averaging results and rounding the corresponding grade to the alphabetical and numerical scale; or b) assessing the ROL through checklists and grading in a pass-fail scheme.

The average method is not appropriate for the CBE as it only tests the knowledge and neglects the student's performance and attitude. The pass-fail method fits the observation of ROL and the performance criteria, but it commonly causes an injustice feeling in the teacher and frustration in the student. This is because the failure in a single item in the checklists can lead to fail in the subject regardless of the effort to meet the rest of the checklist items.

The fuzzy logic exposed in [9, 10] is a useful tool to assess competencies due to its linguistic nature. In this sense, there are some software tools to quantify those results as in [11, 12]. Fuzzy systems allow to process the measures of assessment instruments through membership functions that fit more to the teacher's linguistic criteria to declare when a level of compliance is acceptable.

In this paper, the use of fuzzy systems, as described in [13], is proposed as an alternative for calculating the grade in competency-based educative models. The results are compared with the traditional methods average and pass-fail.

The paper is organized as follows: Section II describes the current scoring methods and fuzzy method is presented. In Section III, the resulting scores between the three methods are compared and the remarkable changes are highlighted. Section IV discusses which method works best in terms of accreditation rate, and finally section V is related to the conclusions and future work.

2 Methods

Tests were made with two different data sets for the student grades. Due to verification purposes, the first set was constructed with 100 random data values in [0, 100]. The second data set contains the real grades applied to five different university groups. The subject, content, learning outcomes and the teacher were the same for each group and were applied in three different scholar periods. The evaluated parameters were: Attendance (A), assignments (T), practices (P), exams (E) and project (Y).

In order to test the three mentioned methods, their operating conditions were the same for all. Each test used the same weight, thresholds and identical membership functions for each evaluation parameter. The calculation methods were programmed using LabVIEW.

2.1 Average Method

In this method, the average score (c_p) is calculated by the simple average of the five parameter measures:

$$c_p = \frac{A+T+P+E+Y}{5}. \tag{1}$$

The average grade (n_p) is assigned as:

$$n_p = \begin{cases} NA, & c_p \leq 75 \\ SA, & 75 < c_p \leq 85 \\ DE, & 85 < c_p \leq 95 \\ AU, & 95 < c_p \leq 100 \end{cases}. \tag{2}$$

2.2 Pass-fail Method

The threshold for the pass-fail method is set to 80% compliance, corresponding to the minimum mark to pass. Each parameter has assigned a value of 1 (accepted) if its measurement is equal to or greater than the threshold, or 0 (rejected) if it is less than the threshold, as follows:

$$a_t = \begin{cases} 0, & A < 80 \\ 1, & A \geq 80 \end{cases}, \tag{3}$$

$$t_t = \begin{cases} 0, & T < 80 \\ 1, & T \geq 80 \end{cases}, \tag{4}$$

$$p_t = \begin{cases} 0, & P < 80 \\ 1, & P \geq 80 \end{cases}, \tag{5}$$

$$e_t = \begin{cases} 0, & E < 80 \\ 1, & E \geq 80 \end{cases}, \tag{6}$$

$$y_t = \begin{cases} 0, & Y < 80 \\ 1, & Y \geq 80 \end{cases}. \tag{7}$$

The pass-fail score (c_t) is the sum of the parameter values:

$$c_t = a_t + t_t + p_t + e_t + y_t. \tag{8}$$

Finally, the pass-fail grade (n_t) is assigned as:

$$n_t = \begin{cases} NA, & c_t < 3 \\ SA, & c_t = 3 \\ DE, & c_t = 4 \\ AU, & c_t = 5 \end{cases}. \tag{9}$$

2.3 Fuzzy Method

The Mamdani model [14] is used to calculate the fuzzy grade (c_d). The measures from the five input parameters are described by five linguistic variables. Each variable is defined by two membership functions as in figure 1. The membership function for accepted (μ_a) is a triangular function whereas for rejected (μ_r) is a trapezoidal function defined by:

$$\mu_r(x) = \begin{cases} 0, & x < 0 \\ 1, & 0 \leq x \leq 60 \\ \frac{80-x}{20}, & 60 < x \leq 80 \\ 0, & x > 80 \end{cases}, \tag{10}$$

$$\mu_a(x) = \begin{cases} 0, & x < 70 \\ \frac{x-70}{30}, & 70 \leq x \leq 100. \\ 0, & x > 100 \end{cases} \tag{11}$$

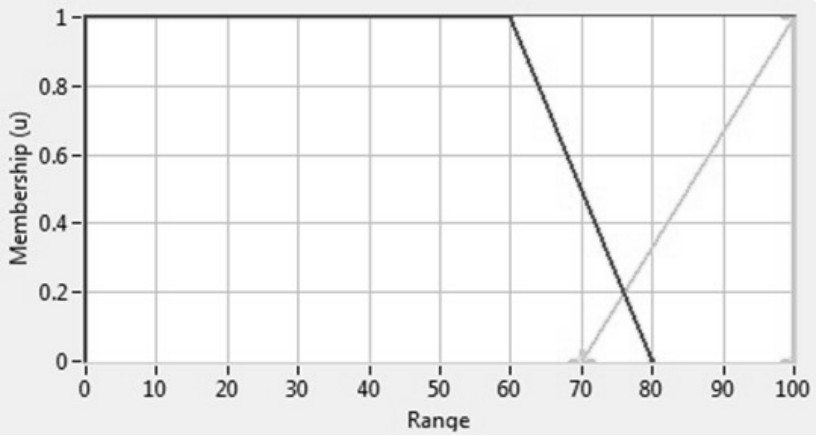


Fig. 1. Input membership functions. μ_r in the left and μ_a in the right.

The output fuzzy set is defined by four membership functions as in figure 2. In this case, a trapezoidal MF is used for no proficient (μ_{NA}), and triangular functions are used for satisfactory (μ_{SA}), prominent (μ_{DE}) and autonomous (μ_{AU}), defined as follows:

$$\mu_{NA}(x) = \begin{cases} 0, & x < 5 \\ 1, & 5 \leq x \leq 7 \\ 8 - x, & 7 < x \leq 8 \\ 0, & x > 8 \end{cases}, \tag{12}$$

$$\mu_{SA}(x) = \begin{cases} 0, & x < 7 \\ x - 7, & 7 \leq x < 8 \\ 9 - x, & 8 \leq x < 9 \\ 0, & x \geq 9 \end{cases}, \tag{13}$$

$$\mu_{DE}(x) = \begin{cases} 0, & x < 8 \\ x - 8, & 8 \leq x < 9 \\ 10 - x, & 9 \leq x < 10 \\ 0, & x \geq 10 \end{cases}, \tag{14}$$

$$\mu_{AU}(x) = \begin{cases} 0, & x < 9 \\ x - 9, & 9 \leq x \leq 10. \\ 0, & x > 10 \end{cases} \tag{15}$$

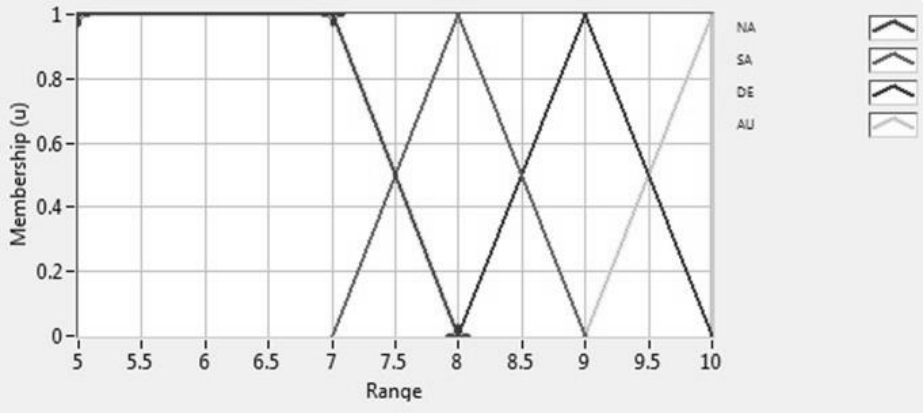


Fig. 2. Output membership functions. From left to the right, μ_{NA}, μ_{SA}, μ_{DE} and μ_{AU}.

The rule set is defined as in the case of the pass-fail criteria, so that, if the student has approved in three items, a fuzzy grade (n_d) of SA is assigned. When the student has four items approved, DE is assigned. When the student has five items approved an AU grade is assigned. In any other case, the assigned grade is NA. Thus, a 32 rule set was obtained according to Table 1. The antecedent for each rule is a compound of the joint of the five items using AND operators as in the form of equation (16). All the rules have the same weight and the consequence implication is the minimum. The defuzzification method is the center of area.

IF (A and T and P and E and Y) THEN (n_d). (16)

Table 1. Rule Set

A	T	P	E	Y	n_d	A	T	P	E	Y	n_d
0	0	0	0	0	NA	1	0	0	0	0	NA
0	0	0	0	1	NA	1	0	0	0	1	NA
0	0	0	1	0	NA	1	0	0	1	0	NA
0	0	0	1	1	NA	1	0	0	1	1	SA
0	0	1	0	0	NA	1	0	1	0	0	NA
0	0	1	0	1	NA	1	0	1	0	1	SA
0	0	1	1	0	NA	1	0	1	1	0	SA
0	0	1	1	1	SA	1	0	1	1	1	DE
0	1	0	0	0	NA	1	1	0	0	0	NA
0	1	0	0	1	NA	1	1	0	0	1	SA
0	1	0	1	0	NA	1	1	0	1	0	SA
0	1	0	1	1	SA	1	1	0	1	1	DE
0	1	1	0	0	NA	1	1	1	0	0	SA
0	1	1	0	1	SA	1	1	1	0	1	DE
0	1	1	1	0	SA	1	1	1	1	0	DE
0	1	1	1	1	DE	1	1	1	1	1	AU

1 = accepted, 0 = rejected.

3 Results

In the first test, the grades were calculated by the three methods using the same random values. The results are shown in figure 3.

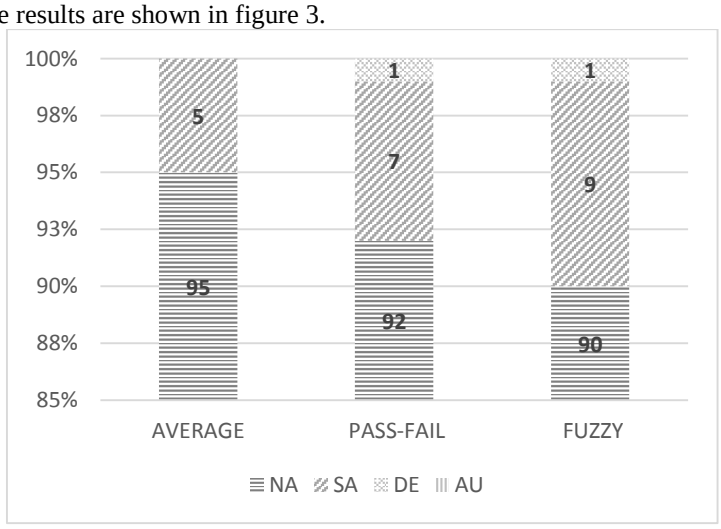


Fig. 3. Grades calculated using random values for the three methods.

Most NA grades were maintained in the three methods. The grades that changed between methods are presented in Table 2. Five students increased one level their grade in relation to the average method, and kept the same in the pass-fail and fuzzy methods (samples from A to E). Two samples were equal in the average and pass-fail methods getting better in the fuzzy method (samples F and G). Only one sample (H) resulted in SA in the average method, which was assigned NA in the pass-fail and the fuzzy methods.

Table 2. Random samples with grade changes between methods

sample	A	T	P	E	Y	n_p	n_t	n_d
A	95	96	100	33	85	SA	DE	DE
B	80	85	76	81	39	NA	SA	SA
C	37	85	30	83	95	NA	SA	SA
D	100	2	90	2	80	NA	SA	SA
E	28	22	88	96	85	NA	SA	SA
F	79	14	98	27	80	NA	NA	SA
G	63	38	97	77	89	NA	NA	SA
H	68	100	65	98	45	SA	NA	NA

Applying the same process and using the real data from 135 students a considerable difference was obtained in the approving index. The fail index is 93% in the average method, 49% in the pass-fail method, and 47% in the fuzzy method. The results are compared in figure 4.

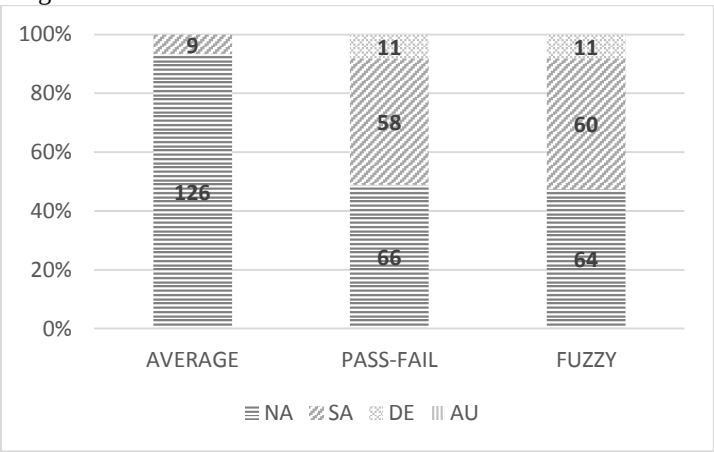


Fig. 4. Grades calculated with real data.

Most of the assigned grades were better by one level in the fuzzy and the pass-fail methods. Eight grades were SA in the average and were DE in the other two methods. Fifty-seven grades changed from NA in the average to SA in the others, reducing the fail index considerably. Table 3 shows the major changes in the grades. Two grades changed from NA to DE (students132 and 135), and two grades were SA only in the

fuzzy method (students 64 and 83). None of the students got in the average method a better grade than in the other methods.

Table 3. Major changes in the grades using real data

student	A	T	P	E	Y	n_p	n_t	n_d
12	96	95	80	83	0	NA	DE	DE
64	78	84	80	50	0	NA	NA	SA
83	93	78	80	25	0	NA	NA	SA
132	86	92	80	100	0	NA	DE	DE
135	100	84	80	100	0	NA	DE	DE

4 Discussion

By comparing the grades that were calculated in the three different methods, it was observed that the fuzzy method has the best approval rate in both cases, with random values as with real data. Increasing the approving rate by a 2%.

Out of the three methods, the average has the highest failing index. Most of the grades were improved in the fuzzy method, except for the H sample in Table 2. It had two high and three low random values, resulting in a better grade in the average than in the other two methods.

The fuzzy method has the lowest NA index, as well as the higher notes compensate those notes that are slight below the threshold limit. As we can see in samples F, G from Table 2 and samples 64 and 83 in Table 3. Many students failed in the average method as well as almost all of them didn't accomplish the project.

5 Conclusions

By observing the measures, it is perceived that the fuzzy method is more balanced and is a good option to calculate the grade. Most of its results are nearer to the pass-fail method, which is applied in the educational programs by using checklists.

The advantage of the fuzzy method is that those students who were very close to the approving limits could do it by compensating the highest notes in other parameters. This method could be able to reduce the injustice feeling in the teacher and the frustration feeling in the student, providing a method that considers the subjectivities.

When rating the measures from checklists through the fuzzy method, the arbitrary assignment of the grade is eliminated reducing the student complaints. The students are certain that all of their evidences in the portfolio were considered for their grade, and the teacher assigns a grade based on student achievement data.

As future work, a comparison between the methods where each parameter has different weight in the grade could be performed. This implies different thresholds in the pass-fail method and distinct membership functions in the fuzzy method.

Acknowledgment. This work was supported in Mexico by the Instituto Politécnico Nacional (IPN) and the Consejo Nacional de Ciencia y Tecnología (CONACyT).

References

1. W. A. Salas Zapata: Formación por comptetencias en educación superior. Una aproximación conceptual a propósito del caso colombiano. Revista Iberoamericana de Educación, vol. 36 (2005)
2. M. Rueda Beltrán: La evaluación del desempeño docente: consideraciones desde el enfoque por competencias. Revista electrónica de investigación educativa, vol. 11, no. 2, pp. 1–16 (2009)
3. M. E. Cano García: La evaluación por competencias en la educación superior. Profesorado, Revista de Currículum y Formación de Profesorado, vol. 12, no. 3, pp. 1–16 (2008)
4. K. Koeppen, J. Hartig, E. Klieme, D. Leutner: Current Issues in Competence Modeling and Assessment. Zeitschrift für Psychologie / Journal of Psychology, vol. 216, no. 2, pp. 61–73 (2008)
5. R. Barragán Sánchez: El Portafolio, metodología de evaluación y aprendizaje de cara al nuevo Espacio Europeo de Educación Superior. Una experiencia práctica en la Universidad de Sevilla. Revista Latinoamericana de Tecnología Educativa, vol. 4, no. 1, pp. 121–140 (2005)
6. E. Barberà: La evaluación de competencias complejas: la práctica del portafolio. EDUCERE Artículos arbitrados, vol. 9, no. 31, pp. 497–503 (2005)
7. Criterios generales para la planeación, el desarrollo y la evaluación, en la implantación de programas educativos por competencias profesionales. Coordinación General de Universidades Tecnológicas, México D.F. (2010)
8. J. M. Jornet Meliá, J. González Such, J. M. Suárez Rodríguez, M. J. Peraless Montolío: Diseño de procesos de evaluación de competencias: consideraciones acerca de los estándares en el dominio de las competencias. Bordón, vol. 63, no. 1, pp. 125–145 (2011)
9. L. A. Zadeh: Fuzzy logic and approximate reasoning. Synthese, vol. 30, no. 3-4, pp. 407–428 (1975)
10. L. A. Zadeh: Fuzzy Logic. Computer, vol. 21, no. 4, pp. 83–93 (1988)
11. C. Fourali. Fuzzy logic and the quality of assessment of portfolios. Fuzzy Sets and Systems, vol. 68, no. 2, pp. 123–139 (1994)
12. M. V. D´Onofrio, M. A. González: Desarrollo de una Herramienta Informática para la Evaluación de las Compentencias Adquiridas en la Gestión de la Mejora de Planes de Estudio. Artículos de las III Jornadas de Enseñanza de la Ingeniería, vol. 1, pp. 66–70 (2013)
13. J. M. Mendel: Fuzzy logic systems for engineering: a tutorial. In: Proceedings of the IEEE, vol. 83, no. 3, pp. 345–377 (1995)
14. E. Mamdani, S. Assilian: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies, vol. 7, no. 1, pp. 1–13 (1975)
15. L. K. Baartmana, T. J. Bastiaens, P. A. Kirschner, C. P. van der Vleuten: Evaluating assessment quality in competence-based education: A qualitative comparison of two frameworks. Educational Research Review, vol. 2, no. 2, pp. 114–129 (2007)
16. J. A. Montero Morales, J. Gómez Urgellès, F. Alías Pujol, C. Garriga Berga, L. Vicent Safont and D. Badía Folguera: Evaluación de competencias subjetivas. Experiencia en la evaluación del rendimiento del trabajo en grupo de los estudiantes. In: VIII Jornada sobre Aprendizaje Cooperativo (2008)

17. C. Makatsoris: An information and Communication Technologies–Based Framework for Enhancing Project Management Education through Competence Assessment and Development. Human Factors and Ergonomics in Manufacturing, vol. 19, no. 6, pp. 544–567 (2009)
18. J. Ma, D. Zhou: Fuzzy set approach to the assessment of student-centered learning. IEEE Transactions on Education, vol. 43, no. 2, pp. 237–241 (2000)
19. I. D. Rudinskiy: Fuzzy Knowledge Evaluation Model as a Methodological Basis for Automation of Pedagogical Testing. IEEE Transactions on Education, vol. 50, no. 1, pp. 68–73 (2007)
20. D. Srinivasan: Efficient Fuzzy Evolutionary Algorithm-Based Approach for Solving the Student Project Allocation Problem. IEEE Transactions on Education, vol. 51, no. 4, pp. 439–447 (2008)

Improving Performance of Particle Tracking Velocimetry Analysis with Artificial Neural Networks and Graphics Processing Units

Rubén Hernández Pérez, Ruslan Gabbasov, and Joel Suárez Cansino

Universidad Autónoma del Estado de Hidalgo,
Centro de Investigación en Tecnologías de Investigación y Sistemas,
Cuerpo Académico de Computación Inteligente,
Mexico

rub3n.hernandez.perez@gmail.com

Abstract Flow analysis has a wide range of applications both in engineering and science, and many techniques have been developed over the years that allow the data extraction and measurement of the flow dynamics. In particular, Particle Tracking Velocimetry (PTV) techniques have shown good results when combined with Artificial Neural Networks (ANN) techniques, especially using Self-Organizing Maps (SOM). In order to improve the performance and reduce the time consumption of proposed SOMs for PTV analysis, the parallel nature of modern architectures such as Graphics Processing Units (GPU) can be used. In this paper we describe how the GPU architecture can be exploited for the implementation of the inherent parallelism of a SOM for PTV analysis, and measure the performance obtained with different optimizations techniques. We show that it is possible to gain a speedup of ~ 5 when running in parallel.

Keywords: Particle tracking velocimetry analysis, artificial neural networks, GPU

1 Introduction

Flow visualization techniques are used to reveal the fluid motions and allow to study the complex phenomena such as turbulence. Techniques such as Particle Tracking Velocimetry (PTV) give us quantitative two-dimensional information about the velocity field by extracting the positions of illuminated particles suspended in the flow and tracing their motion between two or more frames. This information can be used later to compute other flow quantities such as vorticity [9].

The applications of PTV cover a wide range of areas of engineering, science and industry [3]. Some examples are found in medical applications [1], [5], where the PTV analysis provides information for validation and testing of models and procedures that can be used later for surgical treatment of arteries and veins.

This kind of studies demand techniques and tools that enable data extraction from the images of PTV and more important, these tools must have response as fast as possible for PTV analysis, which can be critical, especially in cases when a large number of images is obtained in real time.

The PTV technique can be divided into two steps. First is the image capture and preprocessing where the positions of tracers are determined. The second one is the position match of the tracers between the images (pairing) and measurement of their displacement. Here we study only the second step.

In order to perform an efficient particle tracking when the number of tracers is very large, i.e., the field is crowded, Artificial Neural Networks (ANN) have been proposed as an efficient way to achieve a good level of accuracy [2] and, in particular, Self-Organizing Maps (SOM) have demonstrated to give excellent results [7] for such images. There have been some improvements to the Labonte's algorithm by adding a distance-dependent schema for updating the weights for the neurons [6] and by adding a Delta-Bar-Delta rule to the net to reduce the computation time [4].

Many ANNs, including SOMs, are already implemented in software and there are many available libraries that can be used [8]. The advantage of such implementations is that they can be directly invoked and used as a black-box, nevertheless, they are not necessarily optimized for custom problems and hardware. The problem of performance can be somehow mitigated by using modern microprocessor architectures that include set of instructions for specific tasks and the many-core architecture. Another way to improve the performance is by using special hardware implemented with digital or analogue circuits representing the neurons, but there are many issues yet to be solved like poor flexibility to change the net structure and the way to send and retrieve information from it.

In this work we focus on modern hardware that allows parallel execution, such as Graphics Processing Units or GPUs, the multi-core architecture that offers flexibility and is able to perform arithmetic operations, which represents a cheap alternative to the CPUs. The main characteristic of such architecture is the number of computing cores available which is much larger than any other commercial microprocessor currently available [8]. For programming these devices NVIDIA Corp. provides proprietary NVIDIA Toolkit, which is a set of tools, including it's C/C++ compiler and CUDA libraries. This tool is distributed for free and allows to exploit all supported GPU capabilities.

The multicore architecture allows to take advantage of the parallel nature of the SOM by implementing it on the GPU.

2 A SOM for PTV Analysis

The Labonte's original algorithm [7] starts with the coordinates vectors of particles in two consecutive frames denoted by x_i $(i = 1, ..., N)$ and y_j $(j = 1, ..., M)$, according to this vectors, two sub-nets of N and M neurons are created. Each neuron have two weight vectors denoted by v_i and w_j whose values at the beginning are assigned to x_i and y_j respectively. First the stimulus vector v_i

from the first layer is present to the second layer, then a winner neuron w_c is selected as the one closest to v_i. Having this, the neurons in the second layer are subjected to the next displacement.

$$\Delta W_j(c) = \alpha_j(v_i - w_c), \qquad j = 1, ..., M \tag{1}$$

Where $\alpha \in [0, 1]$ is an scalar value between 0 and 1 given by the condition $\alpha_j = \alpha$ if neuron $j \in S_c(r)$ and $\alpha_j = 0$ if not. In the case of the Ohmi's algorithm [6] the condition changes outside the radius r where the displacement is modified by the distance-dependent Gaussian function:

$$\alpha * \exp\{-(|w_c - w_j| - r)^2/(2r^2)\} \tag{2}$$

For both cases, $S_c(r)$ is the radius of the closed circle centred on the point y_c. Each time the first layer presents the weight vectors as stimuli for the second layer, the second layer is then updated according to the next operation.

$$w_j \leftarrow w_j + \sum_{i=1}^{N} \Delta w_j(c_i), \qquad j = 1, ..., M \tag{3}$$

And in the same way, in the next step, the second layer presents the weight vectors as stimuli for the first layer and update its values with the same formula just in the opposite direction.

$$v_i \leftarrow v_i + \sum_{j=1}^{M} \Delta v_i(c_j), \qquad i = 1, ..., N \tag{4}$$

At each step, the radius r of the circle, within which the neuron weights are changed, is decreased and the amplitude α of the weight translation is increased according to the following equations respectively:

$$r \leftarrow \beta r, \quad 0 < \beta < 1 \tag{5}$$

$$\alpha \leftarrow \alpha/\beta \tag{6}$$

These steps are iterated until r reaches a given value of r_f, which should be small enough to cover only the winner neuron. The value of β is an scalar between 0 and 1. Finally, the matching between particles in the frames is done by a last nearest-neighbour check with a small radius ϵ.

3 Implementation

The need for high performance computing through the use of GPU, is based on the complexity of the algorithm itself, since it involves two subnets interacting with each other every iteration, and each of them has as many neurons as particles in the corresponding frame. The operations performed by all neurons

of v_i against all neurons of w_j are executed in both direction each iteration, implying an exponential behaviour. Besides the number of iterations required by the net depends on the scalar value of the compression factor β and the boundaries delimited by r and r_f as will be shown in section 4.

In order to show the improvement that can be achieved by implementing the proposed SOM in a GPU, we have three approaches. The first one consists in use of the displacement of neurons in equation 3 and simply transforming the loop for processing all neurons (w_j in the first step and v_i in the second step) in parallel for each neuron. This means unrolling the nested loop shown below:

```
for (j=0; j<M; j++) {
   for (i=0; i<N; i++) {
      Calculate displacements...
   }
}
```

Then a parallel form in CUDA, where the code is executed once per neuron will be:

```
j = blockDim.x * blockIdx.x + threadIdx.x;
if (j<M) {
   for (i=0; i<N; i++) {
      Calculate displacementes...
   }
}
```

Here it is necessary to add a condition $i < N$ in order to guarantee that all neurons have one thread assigned.

This quick approach provides substantial speed enhancement in comparison with the traditional serial code and parallel implementations using OpenMP, but it doesn't exploit all the capabilities of the GPU. The architecture of a GPU is composed of blocks of threads that can be referenced by a 9-dimensional index and with these is possible to establish a mapping for all operations between the neurons on v_i and w_j. To illustrate how this can be done, we divide the algorithm in 5 basic steps as shown in Fig. 1.

There are two steps (Calculate Distances and Calculate Displacements) that involve calculation of values between each neuron of v_i and each neuron of w_j. In the first approach, these operations are done by executing in parallel a serial loop for each neuron of v_i. In the new approach the GPU can execute all operations as single threads and is possible for each thread to have two indexes to make reference both to v_i and to w_j and a third index to identify the calculation as unique.

```
idvi = threadIdx.x + blockIdx.x * blockDim.x;
idwj = threadIdx.y + blockIdx.y * blockDim.y;
idviwj = idwj + idvi * N;
```

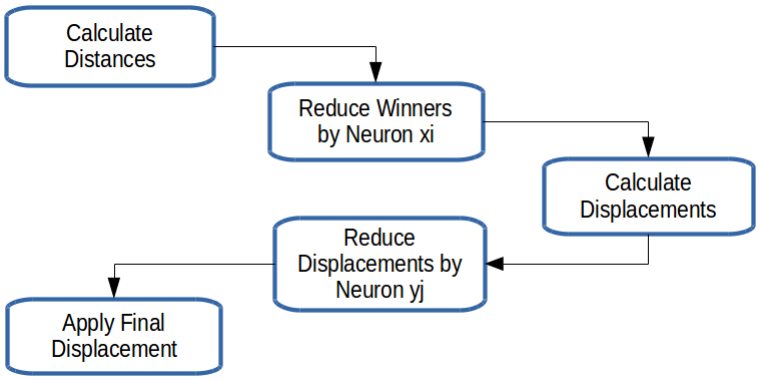


Fig. 1. Steps involved in the iterations over the net. Based on equations 1, 3 and 4, is is possible separate the calculations in well defined segments.

It is necessary to allocate a space of memory of size $N \times M$ to store all calculations between sub-nets and after it is necessary to perform two reduction operations, the first one searching for winners at each neuron v_i, and the second one calculating the cumulative weight displacement for each neuron in w_j. With this approach is possible to obtain a speedup of 3.2X just by calculating all distances and displacements in one step according to GPU capabilities. Note however, that the reduction operations were not parallelized, leaving room for a further improvement.

The parallel reduction is easy to implement but hard to get it right. As an example of optimization, we changed the the serial implementation by a parallel one as a binary three shown in Fig. 2.

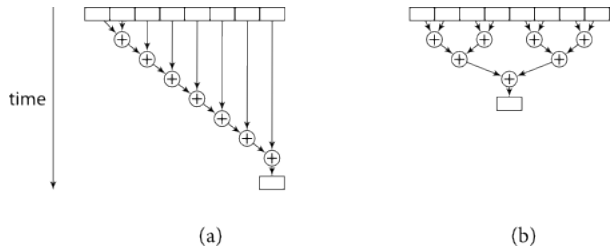


Fig. 2. Difference between simple serial reduction (a) and parallel reduction (b) implementations. It can be easily noted that the parallel reduction requires less execution time.

With this strategy we get a cumulative speedup of 4X with respect to the

original implementation, and as we can observed in Fig. 2, the binary three reduces the number of steps by executing parallel operations at deep nodes of the three. Here is a part of the code implemented in C code that performs this task:

```
idvi = threadIdx.x;
idwj = blockIdx.x;

originalId = idvi + idwj * N;

for(i=512; i>0; i>>=1) {
   if(idwj < i) {
      newId = idwj + (idvi + i) * N;
      Add operations to reduce elements...
   }
}
```

This code reduces blocks of 512 elements at once.

Finally, the last improvement has relation with the calculations of weight displacements. For the Labonte's algorithm it is a simple condition whether to apply or not a displacement based on r, while for Ohmi's algorithm this condition changes and depending on r value, the displacement is applied by the Gaussian function in equation 2. To improve performance of the net, the latter can be replaced by a function that describes the similar behaviour, but avoiding the *if* condition.

In this work we chose a sigmoid function as follows:

$$\alpha * (1 - 1/(1 + \exp\{-(|w_c - w_j| - r) * \lambda\})) \tag{7}$$

In this new equation, the r value is used to displace the function from the center and adds a new parameter λ that can be used to control the smoothness of the curve described by the function. This final approach by replacing conditions in CUDA code, improves performance up to 5.1X with respect to the original implementation.

4 Experiment

All implementations were tested using synthetic images of laminar flow containing 1024 particles in each frame. The performance was measured by changing the value of β, which due to the boundaries defined by r, r_f and equation 5 affects directly the number of performed iterations.

The total number of iterations performed by the net, can be considered as twice the operations performed (see Fig. 1). Each iteration updates weights in both sub-nets vi and wj, so if, for example, the net performed 135 iterations, it actually performed 270 updates over the weight vectors of the 1024 neurons.

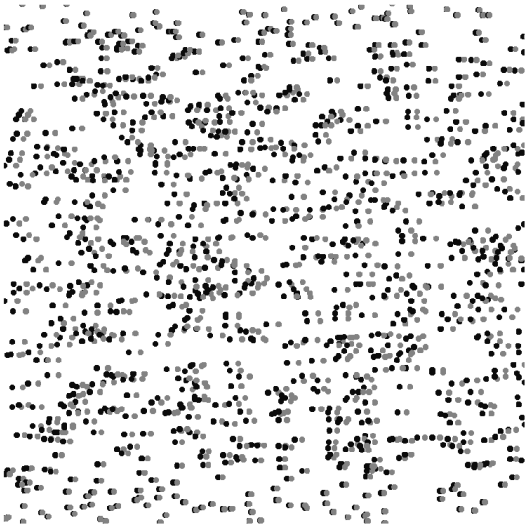


Fig. 3. Example of synthetic image of laminar flow with two frames overlapped. Particles in dark gray are the positions of the first time frame, and light gray dots are the positions of the second frame.

r	rf	β	iterations
100	0.1	0.70	20
100	0.1	0.75	25
100	0.1	0.80	31
100	0.1	0.85	43
100	0.1	0.90	66
100	0.1	0.95	135

Table 1. Values of parameters used to measure the performance of the net. The number of iterations depends on the initial radius r, the final radius r_f and the parameter β. The table shows only the 6 most significant values used for the test. The radius units are in pixels.

The execution time was measured using CUDA events instructions as shown below. It is possible to use CPU or operating system timers, but measurements can be biased by external processes and operating system thread scheduler. Using CUDA instructions eliminate such problems when measuring the GPU execution time. The CUDA event instructions are essentially a GPU time stamp that is recorded at a specified point in execution time:

```
cudaEvent_t begin, end;

cudaEventCreate(&begin);
cudaEventCreate(&end);

cudaEventRecord(begin, 0);

      Iterations over the net...

cudaEventRecord(end, 0);
cudaEventSynchronize(end);

cudaEventElapsedTime(&elapsedTime, begin, end);
```

Finally, the GPU time was measured for each implementation described above and for different β values as a total time used to computation and to copy and retrieve data to/from the memory since we found that for considered size of subnets (1024) the latter does not impact the final performance.

5 Results

The experiment with 4 implementations of the SOM algorithm, shows improvements of 3.2X, 4X and 5.1X in speedup respectively as compared to serial CPU version (see Fig. 4). The first approach shows that high speedup can be obtained by executing operations between frames at the same step, while the second was obtained by parallelizing the inner loop.

Finally, using a Sigmoid function in order to avoid taking decisions as implemented in original Labonte's and Ohmi's algorithms, allows to fully exploit SIMD architecture of the GPU hardware.

6 Conclusions

A GPU can be used to improve performance in a SOM for PTV analysis, however the speedup obtained heavily depends on parallelism and optimisation of the code. The easiest way is to take a serial code and transform it to a parallel one, but full capabilities of a GPU can be exploited by using optimizations of code, using indexes available for identifying threads and avoiding divergence statements. The algorithms from Labonte and Ohmi have proven to be efficient

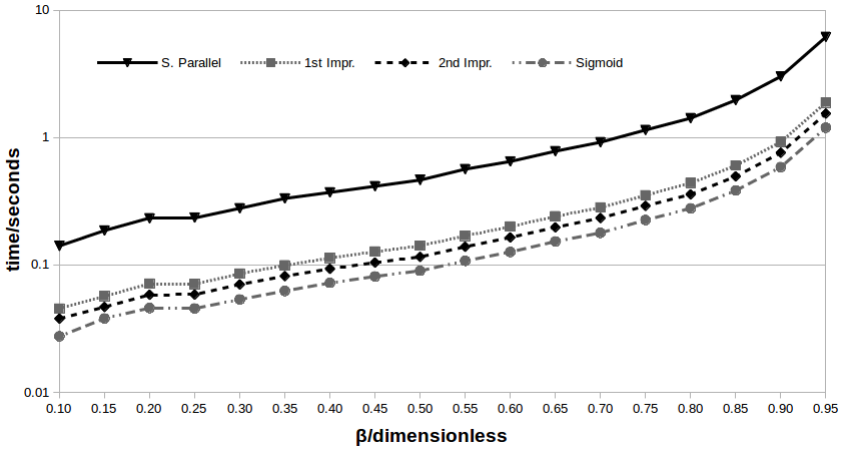


Fig. 4. This chart shows the execution time in seconds for each implementation and each value of β tested.

to deal with the problem of pairing particles, but the functions used to displace weights of neurons can be expressed in terms of a simpler function instead a condition, and it is possible to keep control of the radius and smoothness of the displacements in a more efficient way.

References

1. Grus, T., et al.: Particle image velocimetry measurement in the model of vascular anastomosis. Prague Medical Report 108, 75–86 (2007)
2. Ian Grant, X.P.: An investigation of the performance of multi layer, neural networks applied to the analsis of piv images. Experiments in Fluids 19, 159–166 (1995)
3. J. Hassan, H.Z.: Effects of vortex generator on junction flow. Applied Sciences and Technology (IBCAST) pp. 449–452 (2015)
4. Joshi, S.R.: Improvement of algorithm in the particles tracking velocimetry using self-organizing maps. Journal of the Institute of Engineering 7, 6–23 (2009)
5. Kabinejadian, F., et al.: Particle image velocimetry (piv) flow measurements of carotid artery bifurcation with application to a novel covered carotid stent design. IFMBE Proceedings 39, 1441–1444 (2012)
6. Kazuo Ohmi, A.S.: Cellular neural network based ptv. 13th Int Symp on Applications of Laser Techniques to Fluid Mechanics pp. 26–29 (2006)
7. Labonte, G.: A new neural network for particle-tracking velocimetry. Experiments in Fluids 26, 340–346 (1999)
8. Verber, D.: Implementation of Massive Artificial Neural Networks with CUDA. University of Maribor (2012)
9. Westerweel, J.: Digtal Particle Velocimetry, Theory and Application. Delft University Press (1993)

A Simulation of the Broiler Growth Rate Using Artificial Neural Networks

Raquel Salazar[1], Fernando Rojano[2], and Abraham Rojano[1]

[1] Universidad Autónoma Chapingo, Edo. México, Mexico

[2] Le Magneraud Experimental station, France

raquels60@hotmail.com

Abstract. To simulate the broiler growth the input variables were: day of year, vents opening, wind velocity, external temperature and absolute humidity, the maximum, average and minimum of the internal temperature and absolute humidity. For that purpose, two techniques were applied, a multi-layer perceptron (MLP) static Neural Network (NN) and the Layered Digital Dynamic Network (LDDN) which were applied to a set of experimental data obtained from a broiler cycle of production. The performance for both techniques was compared using: mean squared error (MSE), mean absolute error (MAE) and model efficiency (EF). The model evaluation measurements showed the superiority of the LDDN compared with MLP. The results from the sensitivity analysis found that the variable day of year was the most important variable to predict the broiler growth rate, so using this variable as the only input variable in the model, efficiency of 0.995 was reached for simulation.

Keywords: Day of a year, humidity, temperature, sensitivity analysis

1 Introduction

One of the advantages of modelling with neural networks is the ability to represent nonlinear systems with high complexity. For instance, a complex nonlinear discrete function can be approximated by a multi-layer Perceptron (MLP) model, where using sufficient number of neurons in the hidden layer can reach an acceptable approximation of any nonlinear function difficult to reach by means of using other modelling techniques.

Outnumbered applications of the Artificial Neural Networks (ANN) have been made in order to model agricultural production. One example is an ANN model used [1] to estimate performance in production of hens in a livestock building located in the South of Brazil. The use of the algorithm of back-propagation together with the training process allowed to generate an acceptable ANN model which can be used as a tool for

decision making by the technical staff in different production flocks to be based on scientifically objective criteria. Furthermore, in this work [1] the authors found that the ANN model allowed the simulation of consequences following the contribution percentage from each input to the poultry production.

An ANN model was developed by Galeano and Cerón [2] to estimate the weight of birds based only on the age of the bird, they found a very good performance with a correlation coefficient (R) of 0.99. The authors recommend the ANN as a viable option for modelling the animal production, because ANN has the ability of new variables inclusion and good adjustment between measured and predicted variables.

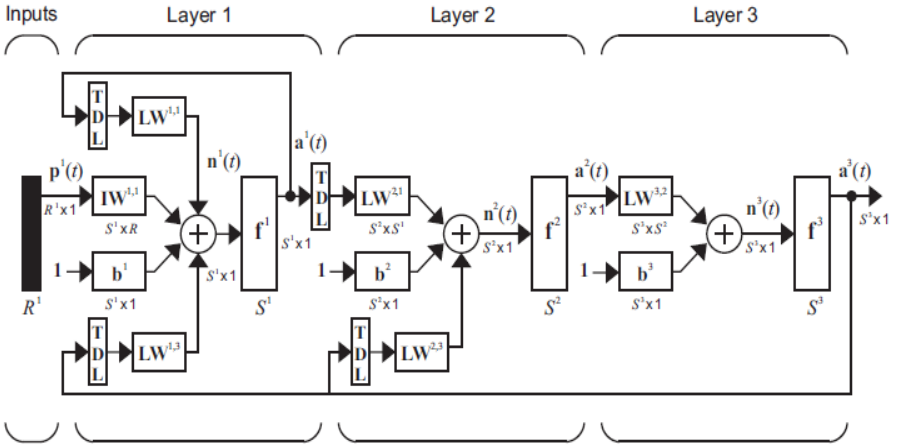
Sefat et al. [3] applied ANN for Modelling the Economic Efficiency of Broiler Production, the independent variables were the amount of consumable inputs (economic value in economic terms) and the dependent variable was the economic performance of production units. They found a NN model for the relationship between costs of different inputs used for production and benefit/cost ratio, they showed that a neural network architecture with 2 hidden layers (4 and 17 neurons in the first and second layers, respectively) provided the best results in estimating the model, with a correlation coefficient R=0.96 and MSE=0.00024. However, according to Masters [4] the additional hidden layers through which errors must be backpropagated makes the gradient more unstable, and the number of false minima increases. The only time that two hidden layers are required is when the network must learn a function having discontinuities.

The static neural networks, described above are not enough for modelling some phenomena, or when the number of patterns is small. Dynamic networks are generally more powerful than static networks (although somewhat more difficult to train). These contain delays and operate on a sequence of inputs. These dynamic networks can have purely feedforward connections, or they can also have some feedback (recurrent) connections. Their response at any given time will depend not only on the current input, but on the history of the input sequence. Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. These have applications in such diverse areas as control of dynamic systems, prediction in financial markets, fault detection, and even the prediction of protein structure in genetics [5].

Dynamic networks can be trained using standard optimization methods. However, the gradients and Jacobian matrix that are required for these methods cannot be computed using the standard backpropagation algorithm. Instead the dynamic backpropagation algorithms that are required for computing the gradients are the backpropagation-through-time (BPTT) and real-time recurrent learning (RTRL). In the BPTT algorithm, the network response is computed for all time points, and then the gradient is computed by starting at the last time point and working backwards in time [6] [7]. This algorithm is computationally efficient for the gradient calculation, but it is difficult to implement on-line. The Layered Digital Dynamic Network (LDDN) is a dynamic neural network that can be arranged in the form displayed in Fig. 1. The LDDN can be trained in the Neural Network Toolbox in Matlab [8].

As with the multilayer network, the fundamental unit of the LDDN is the layer. Each layer in the LDDN is made up of five components: A set of weight matrices that come into that layer (which may connect from other layers or from external inputs), any

tapped delay lines that appear at the input of a set of weight matrices, a bias vector, a summing junction, and a transfer function.



TDL=tapped delay line; LW= weights in the hidden layers; IW= weights in the input layer, b= bias unit; f= transfer functions between layers

Fig. 1. Layered Digital Dynamic Network (LDDN)

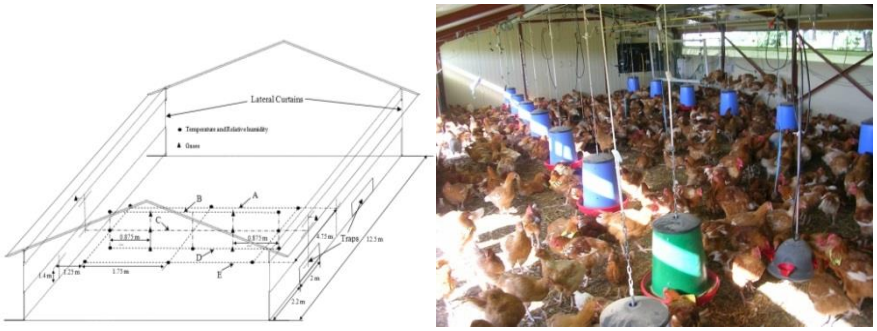


Fig. 2. Broiler geometry and sensor location.

The output of the LDDN is a function not only of the weights, biases, and the current network inputs, but also of outputs of some of the network layers at previous points in time. For this reason, it is not a simple matter to calculate the gradient of the network output with respect to the weights and biases. The weights and biases have two different effects on the network output. The first is the direct effect, which can be calculated using the standard backpropagation algorithm. The second is an indirect effect, since some of the inputs to the network are previous outputs, which are also functions of the weights and biases [9].

In this study we applied the LDNN to model the growth weight rate of the broiler production in an experimental station in France, a comparison between the performance between static and dynamic networks is made for this particular case. Also, a sensitivity

analysis is performed to detect the most important variable affecting the broiler growth rate.

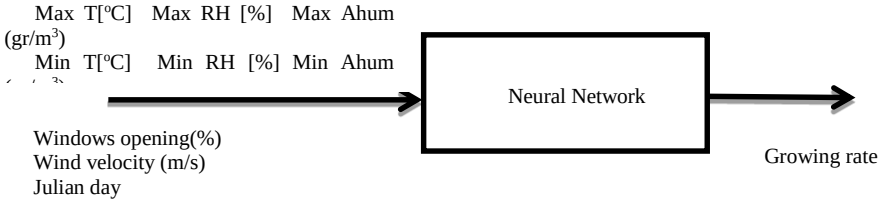


Fig. 3. Inputs and output for the Artificial Neural Network Model

2 Methodology

The data used in this study was collected from a cycle of organic production (84 days) in a broiler house at the experimental station "Le Magneraud" located in western France (46.15 N, -0.69 W). The broiler house had a volume of 158.4 m3 and was naturally ventilated through lateral curtains with a maximum opening of 0.53 m. The house had an eave height of 1.8 m. All the walls and roof were composed of foam, but an additional layer of steel panel was added to the exterior of the roof (see Fig. 2) .

The two heaters located in the house were uniformly distributed along the length, and each had a heating capacity of 4800 W. The experiment started on April 23 2014, the maximum broilers density was used, 750 broilers in an area of 75 m2. The broiler house was occupied with 21,476 animals with a mortality rate during the production cycle less than 3 %. The weight gain found in daily basis is expressed in the following equation (R2 = 0.993).

$$w = 0.0003346 \times d^2 + 0.0236706 \times d - 0.0245480, \quad (1)$$

where d is the n-th day of production cycle and w is the weight gain, in kg [10].

At the beginning of the growing cycle young broilers had a weight of about 0.5 kg, at the end of the growing cycle mature broilers had a weight of about 2.9 kg each. The broilers were kept indoors during the first 35 days; then two trap doors (length and height of 2 m and 0.53 m respectively) were opened to give the chickens access to a backyard during the day (9:00h to 17:00h).

Two types of neural networks were tested, first a static multilayer feedforward perceptron with one hidden layer and backpropagation algorithm The activation function used is expressed in (2):

$$\tan sig(n) = \frac{2}{1+e^{-2n}} - 1, \quad (2)$$

Equation (2) is mathematically equivalent to hyperbolic tangent function (tanh(n)). It differs in that it runs faster than the MATLAB implementation of hyperbolic tangent function (tanh), the results can have very small numerical differences. This function is a good tradeoff for neural networks, where speed is important and the exact shape of

the transfer function is not [11]. The training set size and the hidden layer size are tied together. For the ANN's architecture, only a single hidden layer is used.

The dynamic neural network used in this work was the Layered Digital Neural Network (LDNN). was built with two delays in the input layer and two delays in the output layer, two hidden layers with three and two hidden nodes, the number of iterations was variable to find a good performance.

In both models static and dynamic, the input nodes receive an input vector, this input vector is composed of Julian day, windows opening, maximum, average and minimum temperature, maximum, average and minimum relative humidity, maximum, average and minimum absolute humidity, wind velocity, the output is the weight growth rate of the broilers as displayed in figure 3.

Three methodologies were applied to find out the number of nodes in the hidden layer, the first one was proposed by Hecht-Nielsen ($h = 2n + 1$) cited in Kůrková [11]; the second methodology was developed by Masters [4] ($h = \sqrt{mn}$); finally an additional intermedia criteria was applied ($h = n$); where n is the number of variables in the input layer, m number of variables in the output layer, in our case m = 1.

The artificial neural networks models generated were compared and the best were selected, based on their largest efficiency, lowest mean square error (MSE), as well as on the other statistical performance parameters described below.

3 Evaluation of the Model

The most widely used measure according to Wallach et al. [12] is the mean squared error (MSE), defined as:

$$\mathrm{MSE} = \left(1/n\right) \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

where y_i is the measured value, $\hat{y}_i$ is the corresponding simulated value, and n the number of measurements.

The mean absolute error (MAE) is expressed in (4):

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right| \tag{4}$$

MAE has advantages over MSE if the objective is simply to examine the overall model error.

Model efficiency is defined as:

$$\mathrm{EF} = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} = 1 - \frac{\mathrm{MSE}}{\mathrm{MSE}\bar{y}} \tag{5}$$

If the model is perfect then $y_i = \hat{y}_i$ for each *i* and EF= 1. If one uses the average of observed values as the predictor for every case $\hat{y}_i = \bar{y}$ for all *i* the EF= 0. A model that is a worse predictor than the average of observed values will have EF= 0 [12].

3.1 Sensitivity analysis

A sensitivity analysis was performed in the dynamic model to evaluate the importance of each input variable. One of the most important methods in sensitivity analysis is the backward stepwise method, it consists of step by step adding or rejecting one input variable and examining the effect on the output results. Based on the changes in performance measurements, a largest value in MSE, or a small value for efficiency, due to one input omission shows the most important input variable [13].

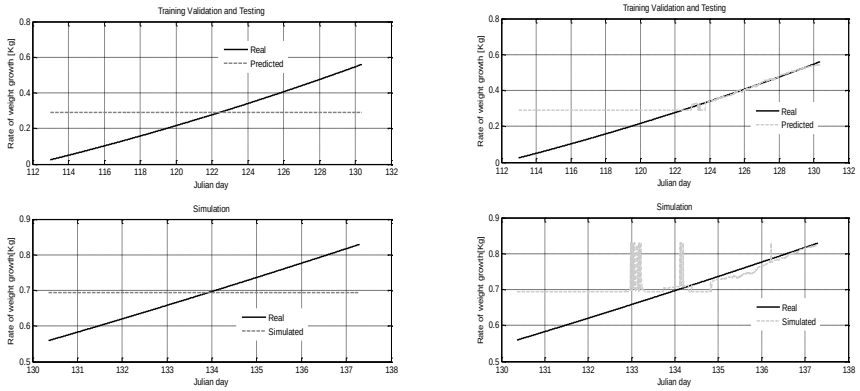


Fig. 4. Training, validation, testing and simulation of the Multilayer perceptron $h = \sqrt{mn} = 3$ (left), $h = \sqrt{mn} = 4$ (right)

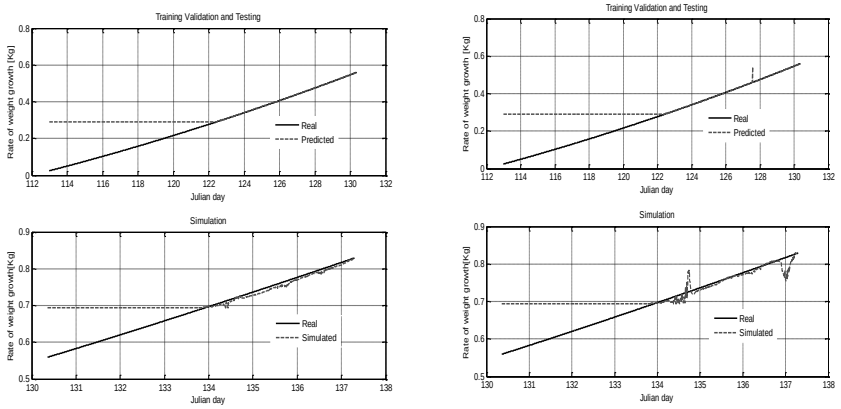


Fig. 5. Training, validation, testing and simulation of the Multilayer perceptron *n=h=12* (left), *h=2n+1=25* (right)

4 Results

The total data patterns available were 3500, the data was divided randomly, 2500 data patterns were used for training validation and testing and 1000 data patterns for simulation which correspond to 137.29 Julian days.

First the feedforward NN was implemented for the three cases of the number of nodes. The results are summarized in Table 1, Fig.4 and Fig.5. The performance of the static neural network is very poor especially when the first criterion is applied. The other two cases have better performance but still their efficiencies are less than 0.5.

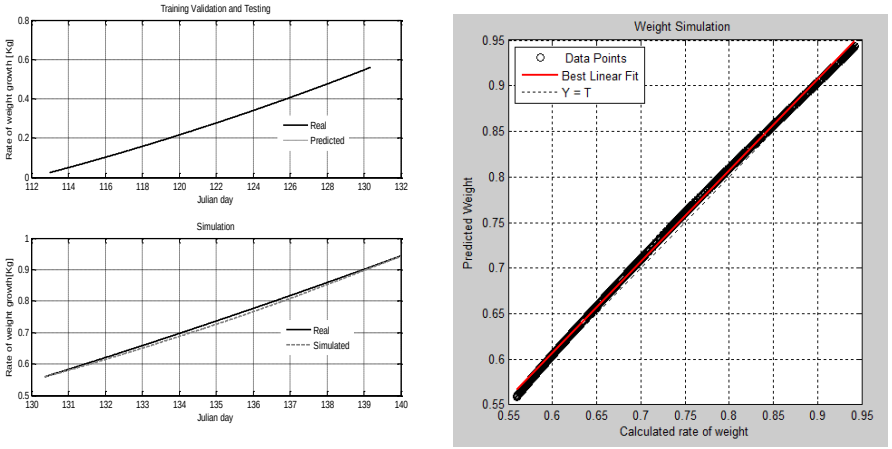


Fig. 6. Training, validation, testing and simulation of the LDNN with 20 iterations

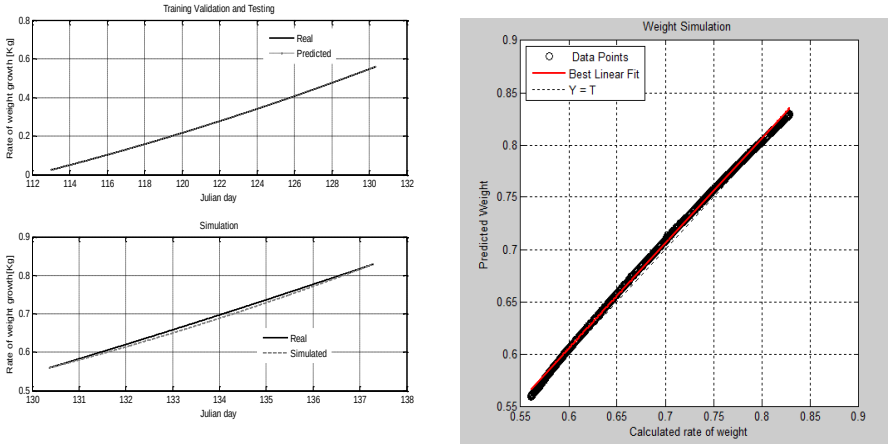


Fig. 7. Training, validation and testing of the LDNN with 50 iterations

A Dynamic Artificial Neural Network (DANN) was chosen because of its memory association and learning capability with sequential and time-varying patterns, which is most likely the biological situation [13]. Table 2, Fig. 6 and Fig. 7 displays a summary of the performance of the LDNN for different iterations. Only with 20 iterations was enough to reach an efficiency equal to one.

It is clear the superiority of the dynamic network in comparison to the static. A sensitivity analysis was performed for each input variable and results are shown in Table 3. The first row, second column of Table 3 display the efficiency of the model when all input variables are taken into account. The last column of Table 3 show a

negative efficiency when Julian day is rejected from the set of input variables, meaning that this is the most important input variable in the model. The efficiency decrease to 0.9875 when relative humidity is not in the model, which makes it the second most important variable. When windows opening or temperature are removed from the variables set the efficiency decrease to 0.9885, meaning that these two variables have the same importance in the model.

Table 1. Multilayer Perceptron Neural Network Model Performance

	Training, validation and testing			Simulation		
Number of neurons in the hidden layer	MSE	MAE	EF	MSE	MAE	EF
h=3 ($h = \sqrt{mn}$)	0.024	0.134	-0.01	0.006	0.067	-0.001
h=4 ($h = \sqrt{mn}$)	0.017	0.108	0.28	0.007	0.075	-0.265
h=12 ($n=h$)	0.013	0.076	0.422	0.003	0.037	0.475
h=25 ($h=2n+1$)	0.013	0.077	0.422	0.003	0.040	0.464

Table 2. Layered Digital Neural Network Model Performance

	Training, validation and training			Simulation		
LDNN Iterations	MSE	MAE	EF	MSE	MAE	EF
20	4.9447×10^{-8}	1.8763×10^{-4}	1	5.6825×10^{-5}	0.0069	0.995
50	7.67×10^{-8}	8.949×10^{-5}	1	3.879×10^{-5}	0.0057	0.993

Table 3. Sensitivity analysis simulation of the LDNN

Efficiency	0.993	0.9885	0.9874	0.9935	0.9885	0.9911	-0.47
Temperature	*		*	*	*	*	*
Relative Humidity	*	*		*	*	*	*
Absolute humidity	*	*	*		*	*	*
Wind opening	*	*	*	*		*	*
Wind Velocity	*	*	*	*	*		*
Julian day	*	*	*	*	*	*	

Table 4. LDNN performance using only Julian day as an input variable

	Training, validation and training			Simulation		
LDNN iterations	MSE	MAE	EF	MSE	MAE	EF
50	3.96×10^{-8}	1.631×10^{-8}	1	5.681×10^{-8}	0.0069	0.995

Given the high importance of the Julian day for the prediction, a run of the LDNN was implemented having only this input variable. Results for 50 iterations are shown in Table 4.

5 Conclusions

In this study two types of Neural Networks are compared for simulation of the growth weight rate for the broiler production in an experimental station in France, a feedforward static neural network, and the Layered Digital NN (LDNN) which is a dynamic NN with recurrent connections and delays in the inputs and output. The input nodes were Julian day, windows opening, maximum, average and minimum temperature, maximum, average and minimum relative humidity, maximum, average and minimum absolute humidity, wind velocity. The best static model reach an efficiency less than 0.5. Some of the arguments of using dynamic neural networks is that each iteration is more complicated because of gradient calculations; however in this particular case only with 20 iterations an efficiency equal to one is reached in training validation and testing and very close to one in the simulation.

Also, the sensitivity analysis shows that the most important input variable for the prediction of broiler growth rate is the Julian day, so the simulation was performed using only this variable and the results showed the superiority of the LDNN compared with static NN. The outcomes out of this model can be applied for prediction of the broiler growth rate using only the Julian day.

References

1. Salle, C.T.P., Guahyba, A.S., Wald, V.B. Silva, A.B., Salle, F.O., Nascimento, V.P.: Use of artificial neural networks to estimate production variables of broilers breeders in the production phase. British Poultry Science 44(2): 211–217 (2003)
2. Galeano, V.L., Cerón, M.M.: Modelación del crecimiento de pollitas Lohmann LSD con redes neuronales y modelos de regresión no lineal. Rev. MVZ Córdova 18(3):3861–3867 (2013)
3. Sefat, M. Y., Borgaee, A. M., Beheshti, B., Bakhoda, H.: Application of Artificial Neural Network (ANN) for Modelling the Economic Efficiency of Broiler Production Units. Indian Journal of Science and Technology, 7(11):1820–1826 (2014)
4. Masters, Timothy: Practical Neural Network Recipes in C++. Academic Press Inc. London. (1993)
5. Laheeb, M. I.: Anomaly Network Intrusion Detection System based on Distributed Time-Delay Neural Network (DTDNN). Journal of Engineering Science and Technology 5(4):457–471 (2010)
6. Abdulla, Z., Kasmuri, N.H.: Comparison of Static and Dynamic Neural Network Models in Predicting Outlet Temperature of Shell and Tube Heat Exchanger Plant. In: IEEE 9th International Colloquium on Signal Processing and its Applications, Kuala Lumpur, Malaysia (2013)
7. Medsker, L.R., Jain, L.C.: Recurrent Neural Networks Design and Applications. The CRC Press International Series on Computational Intelligence (2000)

8. Beale, M.H., Hagan, M.T., Demuth, H.B.: Neural Network Toolbox™ User's Guide. Mathworks Inc. (2015)
9. Vogl, T.P., Mangis, J.K., Rigler, A.K., Zink, W.T., Alkon, D.L.: Accelerating the convergence of the backpropagation method. Biological Cybernetics 59: 257–263 (1988)
10. Rojano, A.F, Bournet, P. E., Robin, P. Hassouna, M.: Test of two different schemes through CFD to include heat and mass transfer induced by animals inside a broiler house. In: Proceedings International Conference of Agricultural Engineering, Zurich (2014)
11. Kůrková, V.: Kolmogorov's theorem and multilayer neural networks. Neural networks 5(3):501–506 (1992)
12. Wallach, D., Makowski, D., Jones, J. W., Brun, F.: Working with Dynamic Crop Models: Methods, Tools and Examples for Agriculture and Environment. Academic Press. 487 p. (2013)
13. Shojaeefard, M.H., Akbari, M., Tahani, M., Farhani, M.: Sensitivity Analysis of the Artificial Neural Network Outputs in Friction Stir Lap Joining of Aluminum to Brass. Advances in Materials, Science and Engineering. Volume 2013, Article ID 574914 (2013)

Methodology for the Model for Failure Prediction in a Digital Signal Distribution

José Cruz Ramos-Báez, María de Lourdes Martínez-Villaseñor, and Dafne Rosso-Pelayo

Universidad Panamericana Campus México, México, D.F., Mexico

{jcramos,lmartine, drosso}@up.edu.mx

Abstract. In the case of Digital Signal Distribution (DSD), machine learning algorithms have contributed to elaborate better ways to enable failure prediction. In this work a nested model for predicting failures in the components involved in DSD failure is presented. The failure can be caused by multiple and different components and also due to correlations between them. We propose a clustering model to isolate component behavior, and subsequently apply predictive models to each cluster. With principal components analysis and cluster analysis we have been able to identify group of failures' causes in this way failures can be segmented and treated properly. We found seven significant features for classification to determine which part is failing. The clustering process generated two groups that allow us to predict if a general failure is going to occur, and the classification process permits us to forecast which component is probably going to present a failure.

Keywords: Digital signal distribution, failure prediction, predictive models

1 Introduction

Failure prediction has interested research communities from different areas for more than three decades. Different offline and online methods have been used to identify risk situations that can prevent the system of deliver the expected service. A survey of online failure prediction methods and propose taxonomy of online prediction methods is presented in [1]. They described four major categories: failure tracking, system monitoring, detected error reporting, and undetected error auditing. In this paper we focus on failure prediction based on Digital Signal Distribution (DSD) systems monitoring. Using undetected error auditing and cluster, determine a fewer of variables for detected failures.

"Online failure prediction is frequently confused with root cause analysis. Having observed some misbehavior in a running system, root cause analysis tries to identify the fault that caused it, while failure prediction tries to assess the risk that the misbehavior will result in future failure" [1].

In computer systems for Digital Sign Processing, there is "a need for real-time performance within the given operational parameters of a target system and, in most cases, a need to adapt to changing data sets and computing conditions"[2]. In complex Digital Signal Distribution (DSD) systems, the need is also to maintain real-time performance and avoid the interruption of system delivery. Maintaining quality service delivery in DSD is vital, given that phone and Internet service is delivered through fiber optic technology to homes and enterprises. When a system failure occurs, it is very important to maintain the system working as customers in residential homes and companies require fiber optics telephone and internet access, which depend on the DSD capacity.

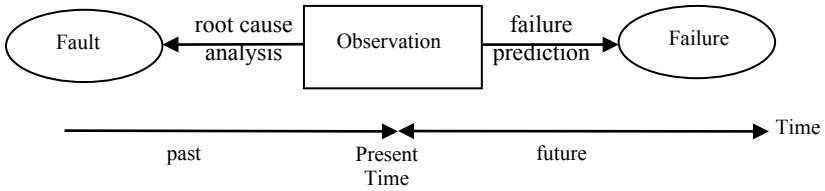


Fig. 1. Distinction between root cause analysis and failure prediction [1]

A methodology to detect and isolate failures in complex Digital Signal Distribution systems is presented, considering faults in: cards; router; VPN; link; FRSW (Finite Range Scattering Wave Function), IC (Integrated Circuit), IT (Information Technology), LANSM (Local Area Network Security Monitor), all of them named as Digital Signal Distribution. The methods included in this work are based in behavior detection and/or fault pattern recognition in big volumes of registers. This work is centered in proactive prediction and management: online failure prediction mainly in order to forecast faults and failures.

Most articles have referred to disk fault detection [3] software [4], using statistical models [5] and [8], and some other model with the results of Machine Learning [6], determine Method prediction [7] and make heuristics models [9].

Our research presents a different approach for DSD system failure detection. We aim to determine with the fewest relevant attributes, if a total failure is going to occur and which component fault is responsible for the failure. We performed principal component analysis to find the most relevant features that enable the failure prediction. Furthermore, we apply clustering processes to make data set segmentation in order to group failures by its behavior. Subsequently we performed a classification first in the whole data set, and afterwards in each of the identified cluster groups in order to find the rules that can describe failure patterns that allow us to detect and predict faults.

The rest of this paper is organized as follows: in section 2, we describe the Digital Signal Distributor considered, and information obtained from system monitoring. We explain our proposed methodology in section 3. We describe our experiments, and discuss the results in section 4. Finally, we conclude and outline our future work in section 5.

2 Digital Signal Distribution

DSD are complex systems composed of multiple parts and elements related with each other. Each component and its relations are probable failure causes. The faults are recollected in a fault log in order to allow failure prediction in the DSD system. In this section we describe DSD system.

2.1 Digital Signal Distribution System

Digital Signal Distribution (DSD) consists in receiving and transmitting digital signals between multiple routers; VPN (Virtual Private Network); links; FRSW (Finite Range Scattering Wave Function); IC (integrated circuit); IT (Information Technology); LANSM (Local Area Network Security Monitor) and cards.

Components of a DSD system are numerous. A DSD system is a receiver and transmitter of digital signals to a certain customers, consisting of wires or fiber optics. Inside of it we can find cards, cables, filters, power supplies, cards memory, ports, etc. Therefor a DSD system is considered a complex system in which a fault in any component may cause that the DSD system suffer a general failure. At present it is not possible to predict what causes a failure to the DSD system, and certainly the one with a failure causes an economic loss by the lack of service.

2.2 Data Description

The data used in this work describe internal components of DSD involved in failures, just some of these failures were resolved, also the data set contains others parts that were involved on DSD system failure. The private data set used contains 11,000 instances detected with errors and has 86 attributes describing components such as card, memory, wires, etc. where the error was present; all data were collected daily during the period 2009 to 2012.

These attributes are numerical and nominal. All of them were taken from various DSD, all DSD consist of the same internal components. The difference is the usage time. We consider that the data collected represent all the possible components and circumstances that can produce a fault. This can be considered as a simplification of the real problem given that unknown causes are possible.

As first criteria in data transformation, from the 86 attributes considered to begin with, we ignored nominal features that are not relevant.

3 Methodology for Failure Prediction in a Digital Signal Distribution

The methods used for failure prediction vary from filter design specific fault, the use of statistical tests and innovations modeling algorithms, and others. We applied a different approach in order to get the best patterns and predict failure in the best way.

We propose a nested clustering and classification model in order to identify faults patterns and behavior to be able to generate failure prediction.

3.1 Data Preparation and Feature Selection

As first step in our methodology we performed a transformation and cleaning process over the data set, ignoring nominal irrelevant features, eliminating not existing values and repeated data. The class feature taken *Closure_code* contains the specific cause of failure, which is determined as the dependent variable *Y*. We use WEKA to find the correlation matrix with these attributes.

Table 1. Correlation matrix, using PCA

1	-0.02	0.03	-0.04	-0.02	0.97	-0.06	0.58	Organization
-0.02	1	-0.05	0.02	0.01	-0.03	-0.15	0.34	Cve_classification
0.03	-0.05	1	-0.05	-0.04	0.03	0.1	-0.53	Closure Code
-0.04	0.02	-0.05	1	0	-0.04	0.01	0.01	Failure time VPN
-0.02	0.01	-0.04	0	1	-0.02	0.01	0.02	Failure time IDE
0.97	-0.03	0.03	-0.04	-0.02	1	-0.05	0.57	Year
-0.06	-0.15	0.1	0.01	0.01	-0.05	1	-0.11	Hour
0.58	0.34	-0.53	0.01	0.02	0.57	-0.11	1	Inc Relation Service Call Id

As second step in our methodology we applied a Principal Component Analysis (PCA). PCA is used in order to emphasize variation and bring out the most significant variables with strong patterns in the dataset. PCA allows better visualization and management of the dataset simplifying the dataset in only 7 attributes with x_i, $Y \in X$ and i=1..7. Attributes' correlations with the dependent variable resulted from PCA, are shown in Table 1. This table reveals relationships between our data set attributes, we obtained a small set of independent principal components from our larger 86 set of related original attributes. In general, higher values are more useful, and we consider excluding low values from the analysis. Table 2 shows the variables.

Table 2. Relevant attributes obtained with PCA

Closure_Code.- the specific component that failed,	Y
Organization- It is used to know the organization that created the incident.	x_1
cve_classification.- The classification of where the incident occurred example: Hardware, software, configuration, etc.	x_2
failure time VPN.- The service downtime.	x_3
failure time IDE.- Downtime in the IDE service, Internet Business Manager.	x_4
Year.- year failure	x_5
Hour.- failure time	x_6
Inc Relation Service Call Id.- It is the relationship you have with another ticket raised with or without affectation.	x_7

3.2 Clustering Process

The third step consists in making data set segmentation in order to group failures by its behavior, the procedure used is a clustering method. We applied K-means with WEKA obtaining two clusters shown in Fig. 2. Clusters C_1 y C_2 represent two separation groups of variable Y which is a nominal feature with $y \in Y$; $x_i, y_i \in C_1$ and xj, yj $\in C_2$ where $x_i \in X$. These clusters help to determine if types of failures can be grouped according to their components and behavior.

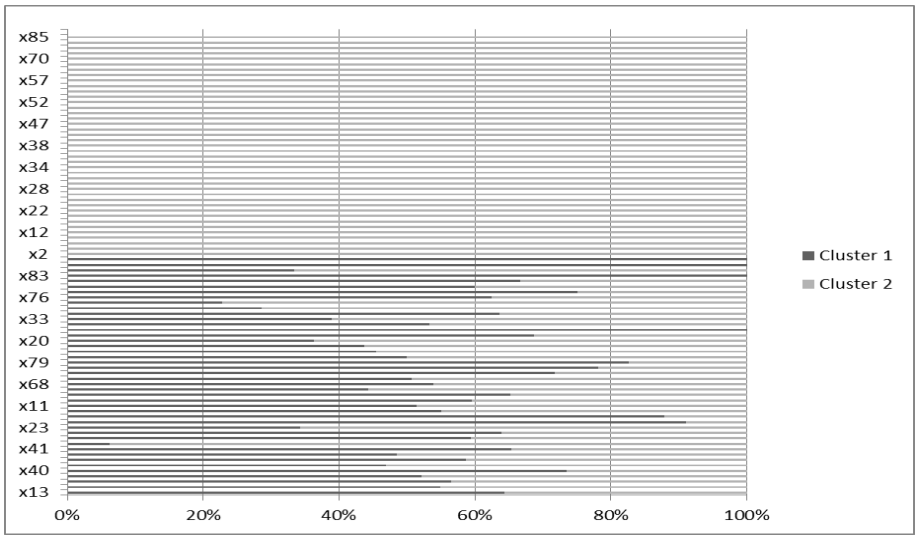


Fig. 2. Cluster using Kmeans

Table 3. Nominal values of *Closure_code*

Variable	Y	Variable	Y
Infrastructure Company	y_{13}	Team strength	y_{65}
Memory card	y_6	Ventilator	y_{60}
Way transmission	y_5	Port	y_{20}
Cabling	y_3	Autoreset	y_{71}
Memory error	y_{16}	Planning hardware	y_{19}
Transmission equipment	y_{41}	Incorrect statement to a third party	y_{33}
support / validation	y_4	Designing	y_{44}
Error in entering commands	y_8	Chassis	y_{76}
Bug	y_{23}	Routing	y_{24}
Air filter	y_{67}	Equipment	y_2
In investigation the root cause	y_{58}	SUCCESSFUL INTERVENTION	y_7

Variable	Y	Variable	Y
Inclusión de actividad tarea instrucción	y_{11}	EQUIPMENT successfully received	y_9
Power supply	y_{62}	Functionality	y_{12}
Equipment climate	y_{59}	Damaged processor board	y_{25}
Incorrect request for a change	y_{46}	Transitory Crash	y_{30}
Error running	y_{68}		

Given the two resulting clusters, we observe that C_1 has the majority of instances that caused a system crash (general failure of the system). C_2 contains very few of this type of instances. This grouping allows us to proceed to further analysis of failure behavior. The nominal values of *Closure_code* presented in table 3 represent those with more relevance to cluster determination.

As a result of K-means clustering we obtained the following clustered instances: cluster C_1 has 4729 (43%) and cluster C_2 contains 6343 (57%) instances. Cluster C_1 is identified mainly with *Infrastructure Company*, and cluster C_2 with *support / validation*. It is interesting to observe that the values, y_{30} which is *Transitory Crash*, which refers to a total or general failure of the DSD system, can just be found in C_2 opening the possibility of relating these failures with the other features.

In order to verify our clustering results, we applied the Expectation Maximization (EM) algorithm. We also applied cross validation in order to obtain better results. The classification results with the whole data set are presented in Fig. 3.

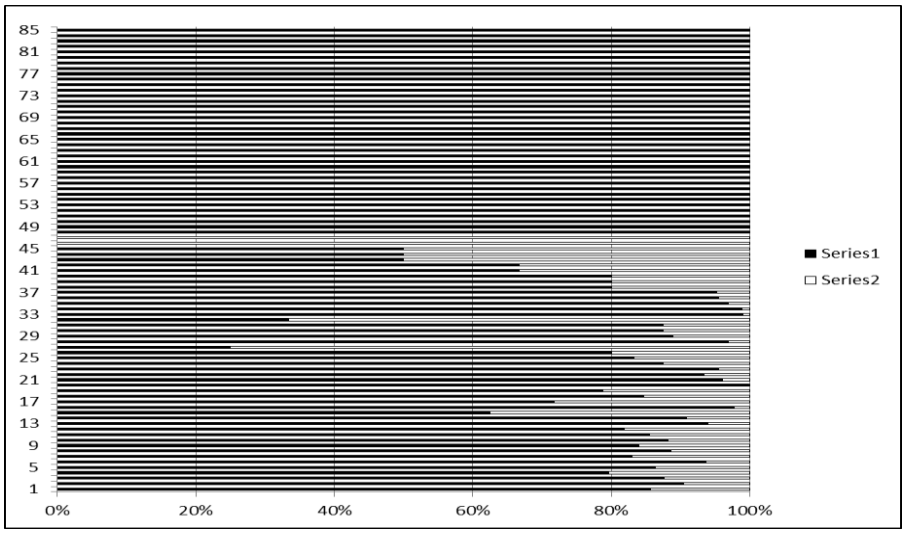


Fig. 3. Clusters with EM

The K-means and EM process used the same features for each cluster, but EM gives a better explanation of the clusters percent. The cluster classifications were 92% and 8% for two clusters.

The following plot graphics has shown the visual representation of cluster 1 data set. We can observe how the cluster items are related to each other; Fig. 4 shows the relation between *Closure_Code* and *Classification*, and Fig. 5 shows the relation of *Closure_code* with the *Service Call Id*. Fig. 6 shows the relation of cluster 2 with the *Service Call Id*, all behavioral differences are shown in clusters. The graphics were developed with K-means.

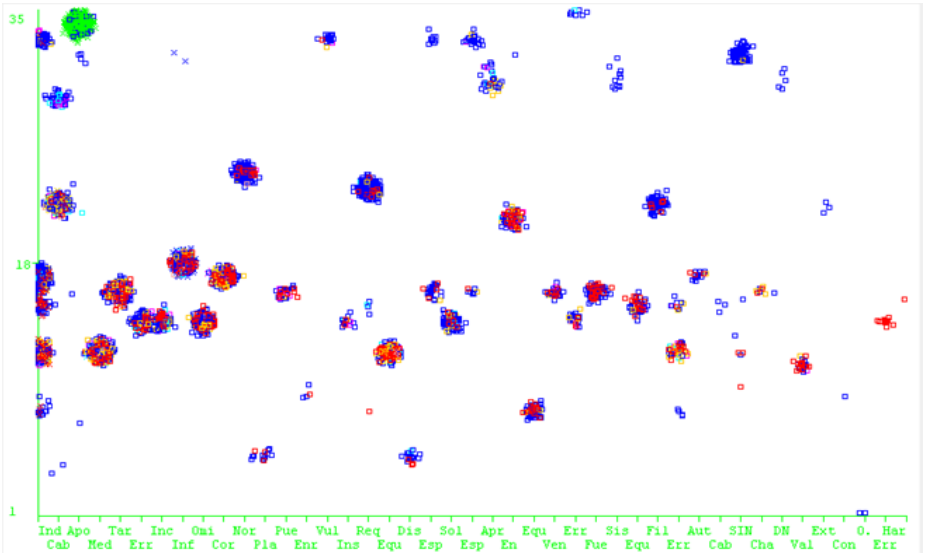


Fig. 4. Closure Code vs Classification in cluster C_1

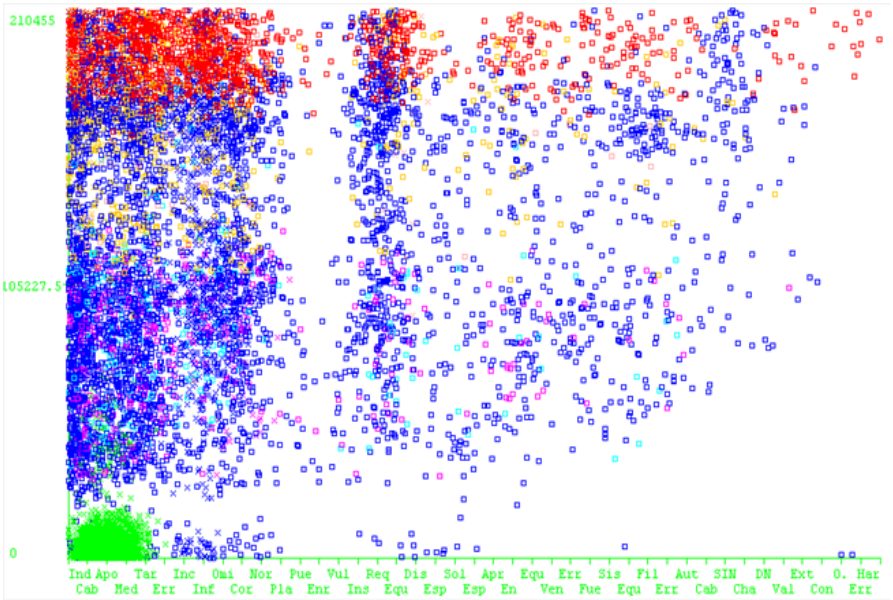


Fig. 5. Closure_code vs Service Call Id in cluster C_1

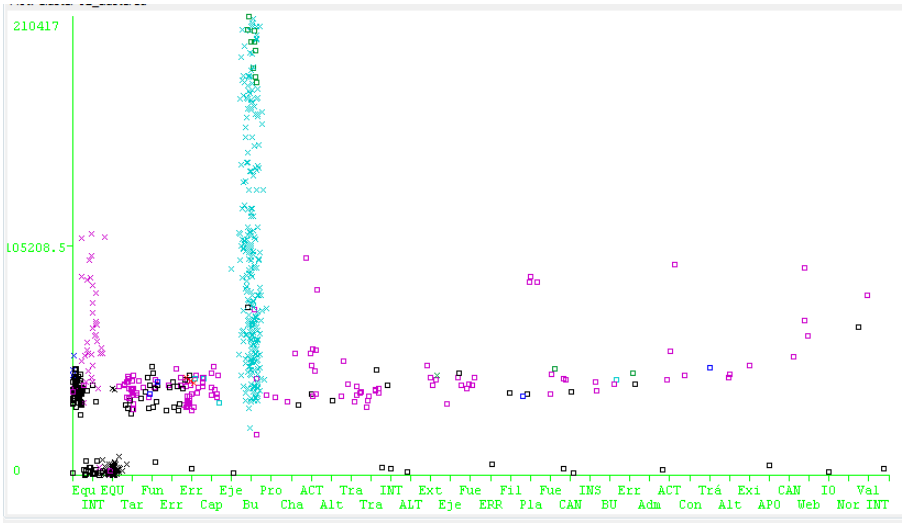


Fig. 6. Closure_code vs Service Call Id in cluster C_2

In fig. 5 and 6 we can observe different behaviors in the relation of the features. *Closure_code* vs *Service Call Id* for each cluster.

3.3 Classification Process

The fourth step is performing a classification first in the whole data set, and afterwards in each of the identified cluster groups in order to find the rules that can describe failure patterns that allow us to detect and predict faults. We used the dataset with seven relevant features described in table 2 in addition to *Closure_code* as the supervised class feature to apply the following algorithms: C4.5, J48, Random Forest and Table Decision.

The results obtained with four applied classification algorithms, C4.5, J48, Random Forest and Decision Table, are very similar. The percentages of correctly classified instances are in the range of 78 to 79% which is a fair classification. They also show a root mean squared error around 0.06. Results are presented in table 4.

Table 4. Classification algorithms results with the whole data set.

Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary								
Correctly Classified Instances	8774	79.24%	8790	79.39%	8676	78.36%	8671	78.31%
Incorrectly Classified Instances	2298	20.76%	2282	20.61%	2396	21.64%	2401	21.69%
Mean absolute error	0.006		0.006		0.006		0.01	
Root mean squared error	0.061		0.058		0.059		0.063	

Table 5 shows the results obtained from the classification performed only on the dataset of cluster C_1 using the same classifiers: C4.5, J48, Random Forest, and

Decision Table. We can observe 80% of correctly classified instances and a root mean squared error of 0.08 which is considered low.

Table 5. Classification algorithms results cluster C_1

Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary Cluster 01								
Correctly Classified Instances	8437	80.84%	8368	80.18%	8277	79.30%	8353	80.03%
Incorrectly Classified Instances	2000	19.16%	2069	19.82%	2160	20.70%	2084	19.97%
Mean absolute error	0.011		0.011		0.011		0.016	
Root mean squared error	0.079		0.082		0.082		0.083	

Another classification on *Closure_code* feature was performed in the same way with the dataset of cluster C_2 using the same classifiers: C4.5, J48, Random Forest, and Decision Table. We obtained 81% of correctly classified instances and a root mean squared error of 0.08 which is also considered low. Results are shown in table 6.

Table 6. Classification algorithms results C_2

Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary Cluster 02								
Correctly Classified Instances	516	81.26%	524	82.52%	515	81.10%	516	81.26%
Incorrectly Classified Instances	119	18.74%	111	17.48%	120	18.90%	119	18.74%
Mean absolute error	0.012		0.011		0.012		0.025	
Root mean squared error	0.082		0.082		0.083		0.097	

4 Discussion of Results

In this section, we interpret and discuss the results described in section 3. We described a nested model for predicting failures in the components involved in a DSD system. Firstly we presented a clustering process. Processes K-means and EM process have differences in percentage of the two clusters; however K-mean is more representative in the separation of variables, since K-mean separates almost 50% instances in each cluster. We can also observe different behavior in each cluster comparing the relation of two features in figures 4, 5 and 6. With the K-means clustering process we obtained two interesting groups. Cluster C_1 represent the instances that produce a general failure of the DSD system, and cluster C_2 represent failure that don't produce a total failure of the system.

In the second phase we applied four different classification processes in order to find which component is failing described in the feature *Closure_code*. Classifiers show fair performance for each cluster: for cluster 1 with the classifier J48 it shows a 80.18% of instances classified correctly with a square error of 0.082, and cluster 2

present 82.52% (Tables 5 and 6) of instances classified correctly and the error square is 0.082.

We found that the seven relevant features obtained with the PCA analysis can predict which component of the DSD system will probably produce a failure with a percentage and error determined by the model.

5 Conclusions

We presented a nested model for failure prediction in a DSD system. Our approach firstly separated the instances in two groups that represent the instances that produced general failure of the DSD system, and those which produce other types of failure. Later on, we determined how to predict which part of the system is possibly failing by using seven relevant features. If the classification process indicates which part is possibly failing, we can predict if the system will produce a total failure or not.

We observed that our proposed methodology obtained significant results in DSD system failure detection and prediction. The nested model included in our research is a novel approach in this field.

We found seven significant features for classification on *Closure_code* class that determine which part is failing. The clustering process generated two groups that allow us to predict if a general failure is going to occur, and the classification process permits us to forecast which component is probably going to present a failure.

As a future work, the prediction model can be improved with other clustering and classification algorithms, and we will prove these nested models with real data.

We are also trying to forecast failures considering time intervals, using time series methods.

References

1. Salfner, F., Lenk,M., Malek, M.: A survey of online failure prediction methods. ACM Comput. Surv. vol. 42, no. 3, (2010) DOI=10.1145/1670679.1670680
2. Tessier, R., Burleson, W.: Reconfigurable Computing for Digital Signal Processing: A Survey. Journal of VSLI signal processing systems for signal, image and video technology, vol. 28, no. 1-2, pp. 7–27 (May 2001).
3. Salfner Felix: Event-based Failure Prediction: an Extended Hidden Markov Model Approach. Dissertation. Humboldt Universitat zu Berlin (Feb. 2008)
4. Andrzejak, A., Silva, L.: Deterministic models of software aging and optimal rejuvenation schedules. In: 10th IEEE/IFIP International Symposium on Integrated Network Management (IM'07) pp. 159–168 (2007)
5. Blischke, W. R., Murthy, D. N. P.: Reliability: Modeling, Prediction, and Optimization. Probability and Statistics. John Wiley and Sons (2000)
6. Candea, G., Kiciman, E., Kawamoto, S., Fox, A.: Autonomous recovery in componentized internet applications. Cluster Computing 9(2):175–190 (2006)
7. Csenki, A.: Bayes predictive analysis of a fundamental software reliability model. IEEE Transactions on Reliability 39(2):177–183 (1990)

8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3:1157–1182, Special Issue on Variable and Feature Selection (2003)
9. Muthumani, N. et al.: A Survey on Failure Prediction Methods. International Journal of Engineering Science and Technology (IJEST) 3(2):1400–1404 (2011)
10. Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., Sahoo, R.: BlueGene/L, Failure Analysis and Prediction Models. In: IEEE Proceedings of the International Conference on dependable Systems and Networks (DSN 2006), pp. 425–434 (2006)
11. Ting-Ting, Y., Lin, D., Siewiorek, P.: Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis. IEEE Transactions on reliability 39(4): 419–432 (1990)
12. Breuer, M. A.: Testing for intermittent faults in digital circuits. IEEE Trans. Compurers, vol. C-22, pp. 241–246 (1973)
13. Dolev, D. et al.: Fault-tolerant Distributed Systems in Hardware. Hebrew University of Jerusalem, Distributed Computing Column, Stefan Schmid TU Berlin & T-Labs, Germany (2015)
14. Kerkhoff, H. G., Ebrahimi, H.: Intermittent Resistive Faults in Digital CMOS Circuits. University of Twente, Centre for Telematics and Information Technology (CTIT) (2015)

A Secure Compression Scheme for Real-time Applications Using 2D-WT and Cellular Automata

M. T. Ramírez-Torres, J. S. Murguía, M. Mejía Carlos, and J. A. Aboytes-González

Universidad Autónoma de San Luis Potosí,
Coordinación Académica Región Altiplano Oeste e IICO, San Luis Potosí,
Mexico

tulio.torres@alumnos.uaslp.edu.mx,
ondeleto@uaslp.mx,
marcela.mejia@uaslp.mx,
j.a.a.g.85@hotmail.com
http://salinas.uaslp.mx

Abstract. In this work is presented a numerical implementation of a system that combines a compression scheme with an improved encryption procedure, which is applied to digital images. For the compression stage is considered the two-dimensional Haar wavelet transform, where an energy criterion is contemplated. On the other hand, the encryption scheme regarded is based on the synchronization of the cellular automaton rule 90, this system presents a good performance to encrypt images and it is resistant to cryptanalysis attacks such as the Chosen/Known-plaintext attack. The numerical conjunction of these procedures could be an appealing option for real-time applications such as video communications, video-surveillance among others.

Keywords: Cellular automata, encryption system, two-dimensional wavelet transform, compression

1 Introduction

Nowadays, there exists a great interest in the protection and manipulation of the data. Due to the great advances in technology, each time is required to have better and more efficient algorithms for confidential and secure data handling. This information may vary depending on the application area and in many cases is necessary processing it in real time. For example, now is very common in different countries that the police install surveillance video cameras on the cities. If the information transmitted from these cameras is not encrypted the confidentiality of data is exposed in the links allowing access to third parties without being detected. But if an encryption processes is added, the latency of the data transmission could increase and the information will not be available on time.

To overcome the eavesdropping problem several encryption systems have been proposed, such as AES (Advanced Encryption Standard), IDEA (International Data Encryption Algorithm), RSA (Rivest, Shamir y Adleman) among others. These systems are generally used in text and binary data, but they are not suitable for the encryption of multimedia data due to their massive volumes, high adjacent correlation and sometimes the multimedia data require real-time interactions (displaying, bit rate conversion, etc.) [2]. Hence many encryption systems with different approaches have been developed for image encryption area [5,8].

On the other hand, different compression schemes which can either be lossless or lossy, work by squeezing redundancy out of data, reducing substantially the initial size of the analysed signals. In this issue, the wavelet transform has proved to be a powerful tool to efficiently process signals that involve large amounts of information. In particular, it has been noticed that this transform is a flexible mathematical tool employed in a great variety of applications and its numerical implementation is often easy to perform [3].

The encryption system considered in this implementation is based on the synchronization of the cellular automaton rule 90 [9]. This cryptosystem is named ESCA (for short) and it has been validated and implemented for image encryption in [6,7]. The ESCA system can be considered secure for images encryption but latency time is too high to incorporate it in real-time applications. In this work we implement a joint encryption and compression procedure to image information. The compression scheme is based on the Haar wavelet transform with an energy approach. The results show that this scheme could be an efficient solution to protect information with a low latency. The structure of this paper is organized as follows. Section 2 discusses briefly the two-dimensional Haar wavelet transform as a tool to compress images. The encryption system and the image encryption algorithm are described in Section 3. The numerical implementation of the joint scheme and results of this proposal is discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2 Two-Dimensional Wavelet Transform

The discrete wavelet transform has a huge number of applications in different areas, and many signals have a bi-dimensional nature like images. For this case, the wavelet transform has also a discrete version to process them. The wavelet transform in two dimensions considers a two-dimensional scaling function, $\Phi(x, y)$, and three two-dimensional wavelets, $\Psi^H(x, y)$, $\Psi^V(x, y)$, and $\Psi^D(x, y)$, where the superscript index indicates the information of the signal at the horizontal (H), vertical (V), and diagonal (D) directions. Each function corresponds to the product of a one-dimensional scaling function φ and corresponding wavelet ψ such that the product does not produce a one-dimensional result, i. e.,

$$\Phi(x,y) = \varphi(x)\varphi(y), \tag{1}$$
$$\Psi^H(x,y) = \psi(x)\varphi(y), \tag{2}$$
$$\Psi^V(x,y) = \varphi(x)\psi(y), \tag{3}$$
$$\Psi^D(x,y) = \psi(x)\psi(y)\ . \tag{4}$$

Given separable two-dimensional scaling and wavelet functions, we define the scale and translation versions as:

$$\Phi_{j;m,n}(x,y) = 2^{j/2}\Phi(2^j x - m, 2^j y - n), \tag{5}$$
$$\Psi^d_{j;m,n}(x,y) = 2^{j/2}\Psi^d(2^j x - m, 2^j y - n), \tag{6}$$

where $j, m, n \in Z$ and the superscript index d assumes the values H, V and D to identify the directional wavelets given in (2)-(4).

In the same spirit as in the case of the DWT in one dimension, and considering that (5)-(6) constitute an orthonormal basis for $L^2(\mathbb{R}^2)$, the expansion of a function $f(x,y)$ of finite energy is then

$$\begin{aligned} f(x,y) = \frac{1}{\sqrt{UV}} \sum_m \sum_n \mathbf{a}_{j_0;m,n}\Phi_{j_0;m,n}(x,y) \\ + \frac{1}{\sqrt{UV}} \sum_{d=H,V,D} \sum_{j=j_0} \sum_m \sum_n \mathbf{d}^d_{j;m,n}\Psi^d_{j;m,n}(x,y), \end{aligned} \tag{7}$$

where the scaling $\mathbf{a}_{j;m,n}$ and wavelet $\mathbf{d}^d_{j;m,n}$ coefficients are defined as

$$\begin{aligned} \mathbf{a}_{j;m,n} &= \int\int f(x,y), \Phi_{j;m,n}(x,y)dxdy, \\ \mathbf{d}^d_{j;m,n} &= \int\int f(x,y), \Psi^d_{j;m,n}(x,y)dxdy\ . \end{aligned} \tag{8}$$

Equation (7) represents the synthesis equation, whereas (8) is the analysis equation. Both equations constitute the two-dimensional discrete wavelet transform (2D-DWT). From now on, unless otherwise stated, we refer a two-dimensional function or signal $f(x,y)$ as an image function $\mathbf{I}(x,y)$ with dimensions $U \times V$, since the 2D-DWT is generally used to image analysis.

To compute numerically the two-dimensional wavelet transform, we follow the Mallat's algorithm for two-dimensional functions [3]. With this algorithm, the multiresolution decomposition of a two-dimensional function or an image is represented by a series of approximations and details of sub-images, which become increasingly coarse. In general, a 2D-DWT can be considered as a separable filter bank of row and column directions that decomposes one resolution

level of an image into four sub-images. A one stage of this procedure is shown in Figure 1, where h and g correspond to a lowpass and highpass filter, respectively, and they are followed for the operation of downsampling by two. After applying the first wavelet level transform, we have four sub-images, and if the original image function $\mathbf{I}(x, y)$ has dimensions $U \times V$, then each sub-image have $U/2$ rows and $V/2$ columns. The approximation sub-image is obtained by computing approximations along rows of the signal $\mathbf{I}(x, y)$ followed by computing approximations along columns. This sub-image is an averaged version of the image $\mathbf{I}(x, y)$ with half resolution and with statistical properties that are similar to those of the original signal $\mathbf{I}(x, y)$. In the same way, in the horizontal sub-image, we first compute the approximations along the rows of the image $\mathbf{I}(x, y)$ followed by computing the details along the columns. As a result, the horizontal edges of $\mathbf{I}(x, y)$ will be always detected by the details along the columns. Since this sub-image analyses the horizontal information, it is clear why it is denoted as the horizontal sub-image. In the multiresolution decomposition the same wavelet transformation is applied but only to the approximation sub-image obtaining again four sub-images, but now with dimensions of $U/2^k$ rows and $V/2^k$ columns, where $k = 1, \ldots, \min(\log_2(U), \log_2(V))$ is the wavelet level. The two-dimensional Haar wavelet transform is considered in this paper, because with this wavelet function the algorithm is memory efficient and reversible.

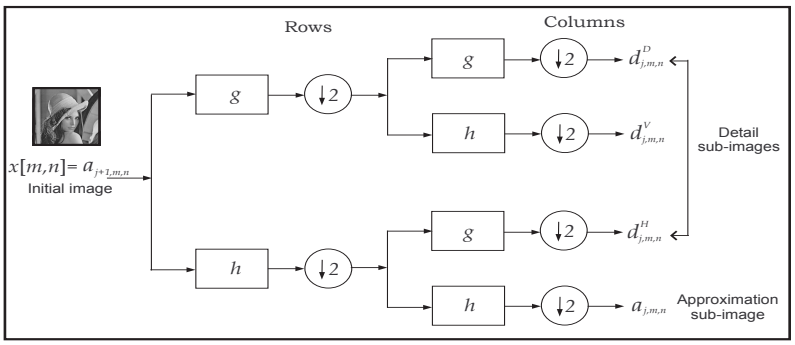


Fig. 1. One stage in a multiresolution image decomposition.

Wavelet compression scheme. The compression procedure that we employ is shown in Figure 2. First of all, the Haar wavelet transform is applied to the initial image. Next, the transformed image is submitted to an elimination process of the transformed coefficients which lie below a threshold value. In fact, the key step here is to choose a threshold through an energy criterion. We look at the normalized cumulative energy applied to the ordered transformed coefficients. To select the threshold value we consider the magnitude of the coefficient for which a proposed energy percentage is obtained. With an established threshold

value ε any coefficient in the wavelet transformed data whose magnitude is less than ε will be reset to zero. Hence the amount of obtained compression can be controlled by varying the threshold parameter ε.

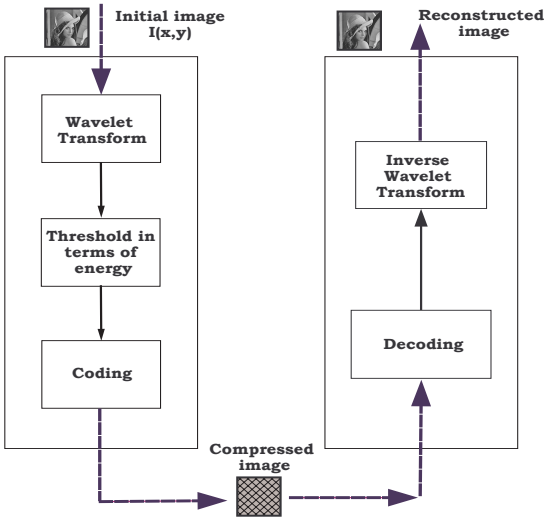


Fig. 2. Basic wavelet compression procedure with an energy approach.

3 Encryption System ESCA

In this work is considered the encryption scheme used in [9], where the synchronization phenomenon of cellular automata has been applied to design two families of permutations Ψ and Φ, and an asymptotically perfect pseudo-random number generator. This cryptosystem is flexible and reconfigurable for different bit-lengths. Figure 3 illustrates a general block diagram of the encryption system ESCA.

The ESCA system comprises the sets M, C and K of binary words, M and C correspond to the plaintexts and ciphertexts respectively, the length of these words is $J = 2^j$, for $j = 1, 2, 3....$ Also it is possible to concatenate several blocks of these lengths. The set K corresponds to the enciphering keys of length $N = 2^n - 1$ for $n = 1, 2, 3...$, for a complete encryption $n > j$ such that N must be larger than J. The two indexed families of permutations $\Psi = \{\psi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ and $\Phi = \{\phi_{\mathbf{k}} : \mathbf{k} \epsilon K\}$ are called encryption and decryption functions respectively. Basically, the cryptosystem transforms a plaintext sequence $\mathbf{m}$ into a ciphertext sequence $\mathbf{c}$, i.e. for every $\mathbf{k} \epsilon K$ one has $\mathbf{c} = \psi_{\mathbf{k}}(\mathbf{m})$, whereas to disclose from the sequence of cipher-blocks, one uses the decryption function $\mathbf{m} = \phi_{\mathbf{k}}(\psi_{\mathbf{k}}(\mathbf{m}))$.

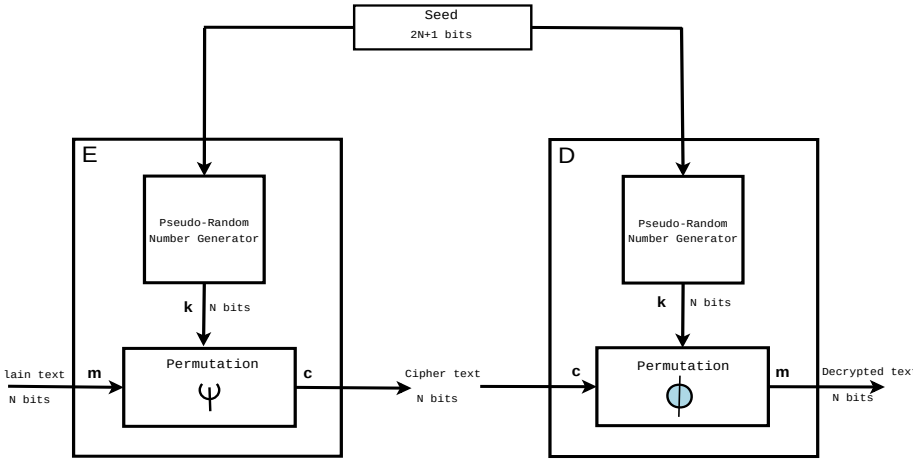


Fig. 3. The encryption scheme ESCA with its main components: the indexed families of permutations and the pseudorandom generator keys.

Since the complete encryption scheme is a symmetric algorithm, the encryption and decryption processes use the same enciphering key $\mathbf{k}$.

In [4], the authors present an ergodic and mixing transformation of binary sequences in terms of a cellular automaton, which is the main element of a pseudo-random number generator (PRNG). The PRNG in its basic form, follows the algorithm shown in Figure 4. At first, the key generator requires two seeds, $\mathbf{x} = \mathbf{x}_0^{k+1}$, of N bits, and $\mathbf{y} = \mathbf{x}_0^k$, of $(N+1)$ bits, which are the input of function $\mathbf{k} = h(\mathbf{x}, \mathbf{y})$. The seeds are $\mathbf{x} = \{x_1, x_2, x_3, \ldots, x_N\}$ and $\mathbf{y} = \{y_1, y_2, y_3, \ldots, y_{N+1}\}$, and the first number generated of N bits is the sequence output of function h, $\mathbf{k} = x_0^1 = \{k_1, k_2, k_3, \ldots, k_N\}$. Now this sequence is feeding back to the input, which becomes the next value of $\mathbf{x}$, and the previous value of $\mathbf{x}$ becomes the initial bits of the new $\mathbf{y}$, where the missing bit is the least significant bit (LSB) of the previous $\mathbf{y}$, which becomes the most significant bit (MSB) of this sequence, and the same procedure is iterated repeatedly.

In [6], it was added a pre-processing to make this scheme resistant to Chosen / Known-plaintext attacks. This process is similar to PRNG, where the function $\widehat{\mathbf{m}} = h(\mathbf{m}, \mathbf{z})$ is used to transform the blocks $\mathbf{m}$ into an unintelligible form denominated $\widehat{\mathbf{m}}$. The Figure 5 shows a block diagram of this process, the inputs are a block $\mathbf{m}$ of J bits and a seed $\mathbf{z}$ of $J+1$ bits. At the output is obtained a block $\widehat{\mathbf{m}}$ of J bits, that will be encrypted later with the permutation Ψ, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$. Also in the Figure 5 is shown the feedback, this is different from the key generation, the next $\mathbf{z}$ is obtained from the joint of $\widehat{\mathbf{m}}$ and the LSB of previous $\mathbf{z}$ as the MSB. This change allows to process images with a high adjacent correlation.

The encryption of an image with the ESCA system proceeds as follows

1. Load the plain-image $\mathbf{I}$ of size $V \times U$.

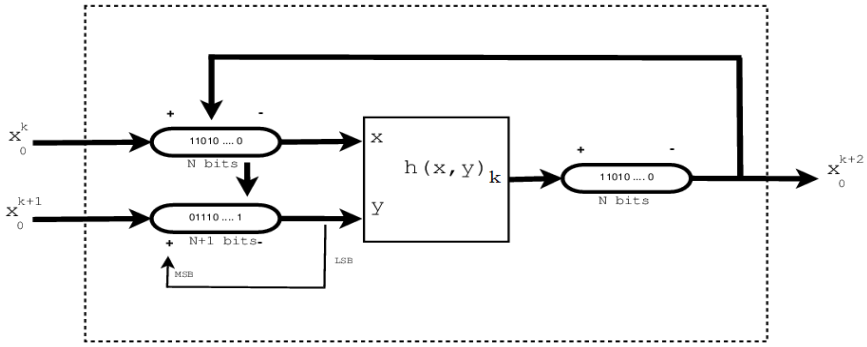


Fig. 4. Basic form of the pseudo-random number generator.

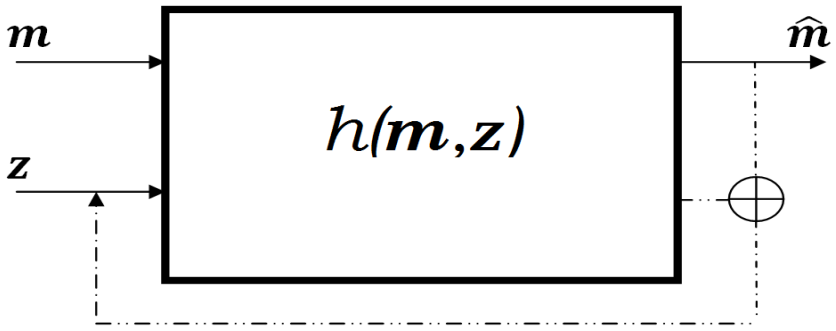


Fig. 5. Pre-processing to obtain $\widehat{\mathbf{m}}$.

2. By scanning the image I row by row, arrange its respective pixels as a sequence or a vector, and convert each pixel value to their corresponding binary value.
3. Establish the length of the encryption key, it must be larger than plaintext bit-length.
4. Compute the modified plaintext sequence $\widehat{\mathbf{m}}$ using the pre-processing.
5. Encrypt each modified block, $\mathbf{c} = \psi_{\mathbf{k}}(\widehat{\mathbf{m}})$ with a different $\mathbf{k}$ each one.
6. By reshaping the set of ciphered sequences of the previous step into an $V \times U$ image, obtain the ciphered image.

This algorithm has proved to be secure against statistical attacks and model threats as Chosen/Known plaintext attacks [6] [7].

4 Numerical Implementation and Results

To implement numerically the joint scheme, it was considered the LabVIEW graphical programming language, a trademark of National Instruments [1]. In Figure 6 is depicted this scheme, which comprises two stages. The top block, Module A, carries out the compression and encryption of images, as was described above, whereas the bottom block, Module B, performs the reverse process to obtain a reconstructed image. It is worth to say that there are two signals after the compression stage, the wavelet coefficients that survived to the value of the threshold, which are the values to encrypt, and a binary vector indicating the positions of such coefficients. Hence the encrypted image and the binary vector are available to be transmitted through a public channel, but it will depend on the application if it is required to convey. Of course, in the Module B the received image is decrypted, and the decompression procedure takes place to the decrypted image with the binary position vector obtaining a reconstructed image.

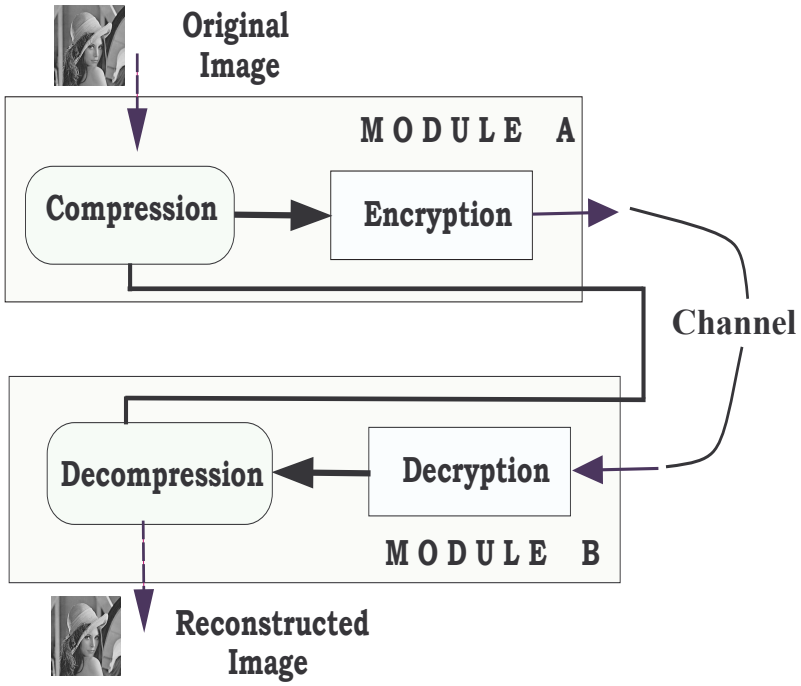


Fig. 6. Image compression encryption scheme.

In such numerical implementation the system was tested with 512×512 RGB and grayscale images, Lena, mandrill and peppers. For the RGB versions, the compression stage should be applied for each color channel. To measure the

degradation of the reconstructed images it was used Peak Signal-to-Noise Ratio (PSNR) as a quality metric. The individual results of the wavelet compression procedure for the Lena image can be observed in Figure 7, it illustrates some reconstructed images for different energy criteria and its PSNR. In Table 1 is shown the compression rate of the RGB images with a 50% of energy. Based on these results is clear that the Haar wavelet transform helps to achieve good compression rates, and the PSNR helps to determine degradation on the reconstructed images.

Fig. 7. a) The source image. Reconstructed images when a b) 90% (37.32 dB), c) 75% (30.32 dB), and d) 50% (23.70 dB) of energy is considered in the compression stage.

Table 1. Compression rate achieved with 50% of energy and 4 levels. PSNR of the reconstructed images from these settings.

Image	**Compression rate**	**PSNR**
Lena	24.68:1	24.68 dB
mandrill	23.93:1	18.60 dB
peppers	38.34:1	34.38 dB

The ESCA system encrypts the survived coefficients, for RGB images before to encrypt the survived coefficients from each channel, they are concatenated again. Figure 8 illustrates the results obtained in the stages contained in Module A for a source Lena image. In this case, an energy criterion of 90% was considered to compress the source image.

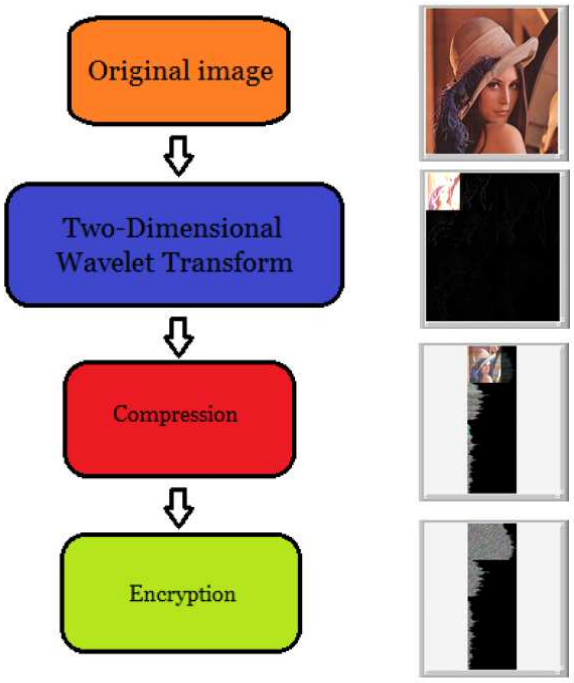


Fig. 8. Results of the application of Module A to a source Lena image.

The results of this scheme show that is possible to reduce latency time until 80% in comparison with only encryption scheme. The Table 2 contains a comparative between latency time (in milliseconds) of the encryption process without compression stage and our proposed scheme, using RGB images with a resolution of 128×128 pixels, considering a 75% of energy and 4 levels of 2D-WT. The Table 2 also shows the PSNR of the recovered images and the compression ratio reached.

Finally, this scheme was implemented in a video application using a conventional webcam with a resolution 128×128 pixels and 30 frames per second. In this program the user can select the number of levels of the two-dimensional wavelet transform and the energy percent to preserve. In the same front panel is shown the compressed and encrypted images as result of Module A, and the recovered images as output of the Module B. Also the program calculates the PSNR of the recovered video, in this way the user can select the parameters

Table 2. Comparison of latency times between different schemes.

	Encryption process (ms)	Our proposed scheme (ms)	PSNR (dB)	Compression rate
Lena	72	28	24.07	11:1
mandrill	72	29	21.54	10:1
peppers	72	26	23.72	12.5:1

according to the quality of the recovered image and the latency based on the flow of the recovered video. The Figure 9 shows the front panel of the video application.

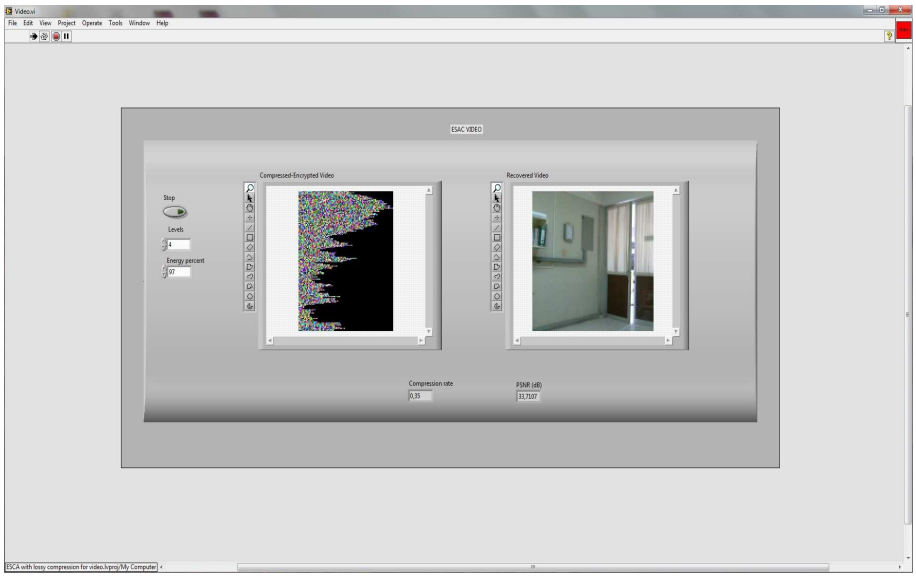


Fig. 9. Front panel of the video application with the proposed scheme, considering 97% of energy, 4 levels and the recovered images have a is 33.71 dB.

5 Conclusions

In this work we presented the numerical implementation of a system that combines a compression procedure with an improved encryption system, which was applied to images. The compression procedure is based on an energy criterion of the Haar wavelet coefficients, and the obtained results provide us good compression rates, because a high energy concentration was presented in a few wavelet

coefficients of the transformed image. On the other hand, the improvement of the encryption system presented a high security and a great flexibility to encrypt image information. In fact, this allowed that some cryptanalysis attacks were outperformed, and now with the compression stage it presented a good performance in time efficiency issues. With the obtained results of this proposal, we think that it can be a useful tool in the current multimedia applications.

Acknowledgments. J. A Aboytes-González is doctoral fellow of CONACYT (México) in the program of "Ciencias Aplicadas" at IICO-UASLP. The authors also want to thank to FAI of UASLP for the economical support to this work.

References

1. The labview environment. url http://www.ni.com/labview/
2. Lian, S.: Multimedia content encryption: techniques and applications. CRC press (2008)
3. Mallat, S.: A wavelet tour of signal processing. Academic press (1999)
4. Mejıa, M., Urıas, J.: An asymptotically perfect pseudorandom generator. Discrete and Continuous Dynamical Sys 7, 115–126 (2001)
5. Pareek, N.K., Patidar, V., Sud, K.K.: Image encryption using chaotic logistic map. Image and Vision Computing 24(9), 926–934 (2006)
6. Ramírez-Torres, M., Murguía, J., Carlos, M.M.: Image encryption with an improved cryptosystem based on a matrix approach. International Journal of Modern Physics C 25(10), 1450054 (2014)
7. Ramirez-Torres, M., Murguia, J., Mejia-Carlos, M.: Fpga implementation of a reconfigurable image encryption system. In: ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on. pp. 1–4. IEEE (2014)
8. Suresh, V., Madhavan, C.V.: Image encryption with space-filling curves. Defence Science Journal 62(1), 46–50 (2012)
9. Urıas, J., Salazar, G., Ugalde, E.: Synchronization of cellular automaton pairs. Chaos: An Interdisciplinary Journal of Nonlinear Science 8(4), 814–818 (1998)

Electronic System of an Intelligent Machine: the Case of an Assistive Bed Device

E. Vázquez-Santacruz, C. Morales-Cruz, and M. Gamboa-Zúñiga

Centro de Investigación y de Estudios Avanzados del IPN,
Coordinación General de Tecnologías de la Información y las Comunicaciones,
México D.F., Mexico

{efvazquez,cmorales,mgamboaz}@cinvestav.mx

Abstract This work presents the electronic design of an intelligent device that includes a monitor system for automatic movements of a robotic hospital bed based on posture classification and identification. This feature was carried out in response to the necessities defined by the application of a diagnostic identification methodology. This method was successfully applied to a public Mexican hospital and the issue identified was the mobility of elderly people and physically challenged individuals. The movement of these patients can be performed routinely or sporadically during their stay in a hospital. For patients who require a particular routine application of this action, the system includes an intelligent monitor system. This intelligent system allows medical experts to program the movements of the robotic bed considering the posture of the patients and the time in bed. This paper shows the hardware and software design of the electronic system and the physicals results.

Keywords: Assistive bed, robotic bed, posture classifciation

1 Introduction

In recent years, the Artificial Intelligent Systems (AIS) have been used in several applications such as industrial control, robot control, traffic surveillance, remote sensing, and speech recognition to mention a few. In particular, the insertion of AIS to medical environments has been a challenging task due to the high-risk decisions in the diagnosis, monitoring, and care of patients. However, in the rehabilitation of patients with limited or restricted mobility, as is the case of geriatric patients, AIS have been used to control the positioning of robotic hospital beds. This approach prevents the appearance of ulcers because of the bed pressure on the tissue. The system also takes into account the activity monitoring and bed-rails control [9,4,12].

Most of the commercial systems for automatic control of hospital bed positions are based on the detection of patient posture. This information is obtained using presence sensors, digital cameras, thermal cameras and mattress pressure sensors [5]. On the other hand, modern hospital beds can adopt twelve positions

ranging from the home position to the sit to stand position but do not have a mechanism to prevent accidents when the change of configurations takes place. Some of the most frequent accidents, when the bed is moving, are the downfall of patients, bad posture when the bed is moving, and injuries caused by improper use of motion controls. In this research, we show the electronic design of an AIS, proposed to reduce the risks of operating a hospital bed with multiple positions. The primary objective of our AIS is to prevent accidents when the bed is moving; this is done by detecting the posture of patients using a mattress pressure sensor. We identify the position of patients by performing an analysis and classification of the pressure distributions using an initial training set of correct postures for all bed posts.

Electronic design is essential to acquire, manage and transport the control and power signals to all the system. The implementation of a module to process inputs and outputs has been widely developed [2,13,14]. As an example, the work of Bustamante Malla [1], designed and implemented a control card and data acquisition with a resolution of 12 bits and it is managed by a computer (PC). Ordoñez [10] implements a device to obtain the voltage-current measurements from a solar system, and it is based on a micro-controller connected via RS-232 to a computer for processing the information. This development has also been implemented to solve the bigger requirements of particular systems, where it is needed to read and manage the information of an array of sensors. In [6] the data acquisition of 96 high-resolution underwater sensors was carried out. Another goal is to achieve the velocity requirements of some systems, for example, image processing in [8]. This field is still developing and growing up since the requirements of new systems are increasing.

2 System Requirements

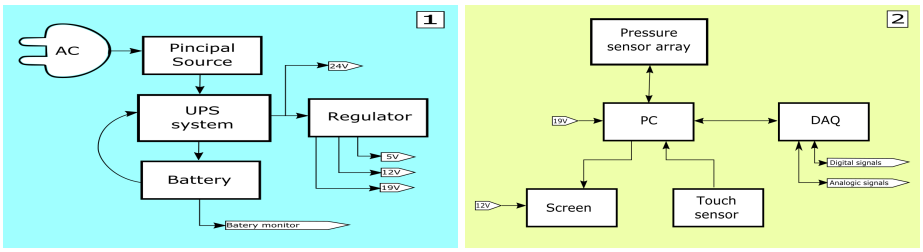
Electronic requirements to achieve are: source from 110 VCA at 60 Hz which is part of the facilities in all Mexican hospitals, autonomy of at least 5 hours, ability to handle electrical breakouts and the energy supply for the actuators to move the system. The system to be designed must also be capable of getting the different type of data from all the sensors on the physical system. The electronic design also has to handle the information to be sent back to control the behavior of the machine. It must read all the devices at the same rate and maintain the information in the buffer until the manager program asks for the data. This is proposed to avoid data loss when the CPU is performing another action, for example transferring data online or attending interaction with the user. The complete system must be able to deal with 65 digital input/output signals, 28 analog inputs, two analog outputs and 3 USB ports. Digital pins will be used to manage the data acquisition and communication from a 20 buttons keypad array, read with nine digital lines and distributed into distinct areas to deliver comfort to the final user. It will use 16 motors with the capability of changing its direction at any time or PWM control. It also uses various limit switches to determine the bound of some movements, two presence detectors, and a teach

pendant with eight buttons, driven with four lines. Finally, the system includes some extra lines for future improvements.

Analog inputs contemplate the management of 17 feedback signals from motors, 2 inclinometers, the battery monitoring pin, 4 load cells to know the pressure at the support points of the system, two thermometers wires and 3 extra lines to allow future changes. Finally, analog outputs will be used to manage the RGB LED's and provide quickly visible information about the status of the entire system. Moreover, finally the USB ports to acquire the information from 2 arrays of pressure sensors and the last one will be used to read the information from the touchscreen of the graphical user interface.

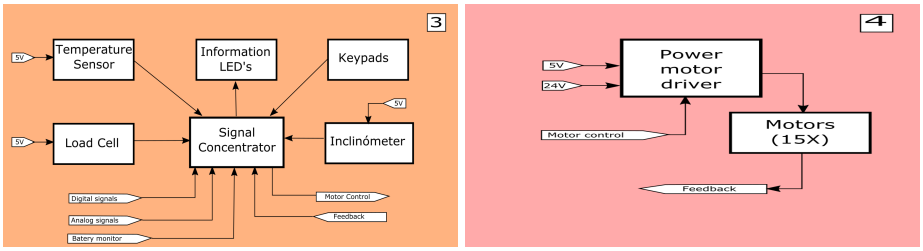
3 Electronic Design

The architecture of the system is shown in Figure 1 and 2. In these pictures can be seen the functional division of each element that compose the system. This division also helps to design carefully and individually the logic sections of the robot. Another positive feature of this conceptual and physical disposal is the advantage of providing straightforward repair and technical support for the modular design give the opportunity to change the malfunctioning card only.



(a) (b)

Fig. 1. System architecture



(a) (b)

Fig. 2. System architecture

3.1 Power Sources

The system will be connected to a 110 AC voltage network, and the elements inside must fulfill the supply requirements of each internal component. Therefore, it was implemented some linear and switching voltage regulators to provide the proper value. If the AC voltage supply fails the system must remain working, and this device handles this case by implementing a UPS system. In Figure 1(a) can be seen the block diagram of this module.

3.2 CPU and User Interface

Module two contains some commercial products that were previously selected to work together, and they are shown in Figure 1(b). The CPU was selected by comparing the ARM and Microcontroller architectures. Finally, a PC was chosen, considering the advantages of increased capacity and processing speed; it allows programming a graphical interface with a high-level language and easy programming of complex algorithms. In this work the BOXDC521HYE Intel NUC was implemented using a Linux distribution like the operating system, since this approach has shown important advantages among others [14]. The pressure sensor array has its data acquisition, but the processing of the information must be carried out on the PC, and then all the data must be moved via two USB ports. On the other hand, the touch sensor located on the screen to interact with the user also has to be connected via a USB port. These connections are handled directly by the PC. The last element in this module is a commercial NI USB-6212 to move data to its final stop, the PC. This element manages the digital and analog signals to be sent in one direction or another.

3.3 Signal Concentrator

This module can be seen in Figure 2(a) and handle all the digital or analog information used in the robot from all the sensors. Given that the characteristics of the DAQ-6212 are lower than the requirements of the system, this module aims to manage the signals by multiplexing the digital pins. Additional functions of this module are the organization of all the wired connections and isolation to provide a safer configuration. In some cases, this module also accomplishes the signal conditioning to provide the necessary characteristics to be read or written to its target device.

3.4 Power Drivers for Actuators

The design of this module contains the electronic required to interact the 24 volts power supply of the motor by changing the state of the TTL connections from the signal concentrator module. This card also contains an electronic configuration to avoid undesired activation of the motors as well as to prevent configurations that can damage the card. This module is displayed in Figure 2(b).

4 Supervisory Control

The system acquisition is governed by a program dedicated to administrating all the information read from the physical system. This program is running on the CPU selected. Figure 3 shows the flow diagram of this specific program to interact with the DAQ.

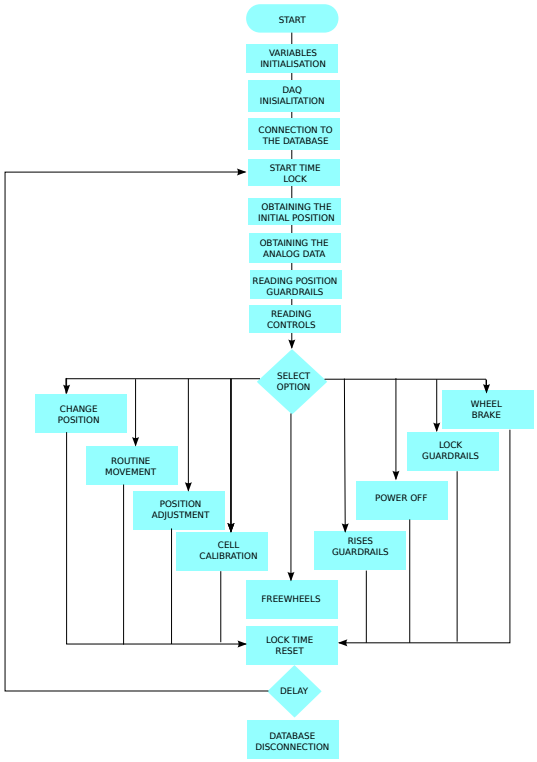


Fig. 3. Software behavior

Thus, the database of features (see Fig. 4) is constructed such as it contains three set of equal size of the three basic positions and an appropriate number of its variants. The database of features is then used to construct the model of the classifier to make predictions of pressure distributions that do not belong to the database.

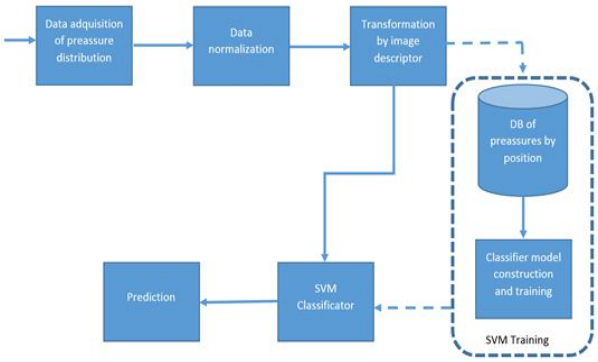


Fig. 4. Main blocks of the IS for posture recognition.

5 Experimental Implementation

This section shows the experimental implementation of the described system. In Figure 5 can be seen a diagram that contains the four modules explained before and all the electrical connections corresponding to the complete system. It also plots an idea of the number of elements to be read with the data acquisition system developed in this work.

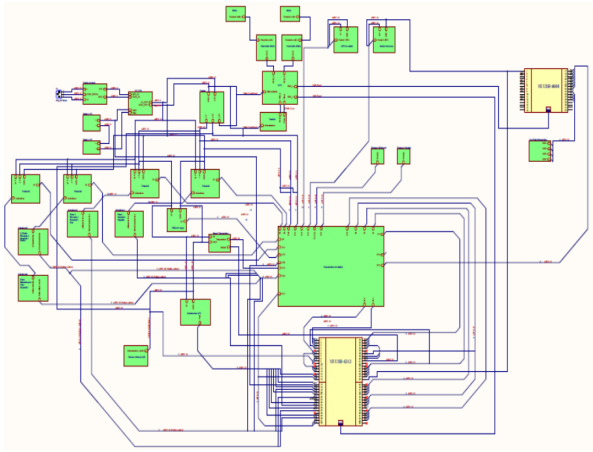
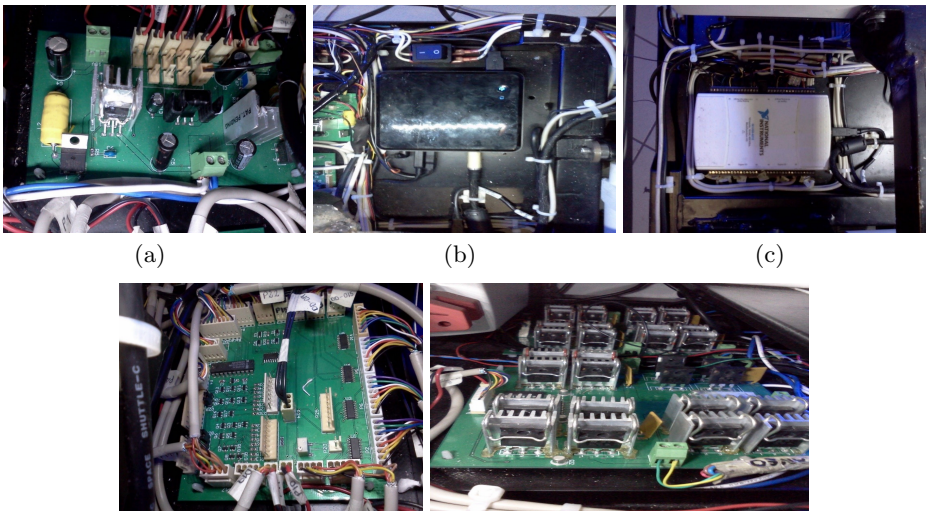


Fig. 5. Electrical diagram of the system

Figure 6 shows some pictures of the implemented system. In Figure 6(a) can be seen the physical card that corresponds to the module 1, this PCB will provide the correct voltage to each of the devices in the robot. Figures 6(b)

and 6(c) display two of the components of the second module, these are the PC (NUC) and the USB-6212 respectively. The signal concentrator card can be seen in Figure 6(d) and correspond to the module 3 explained before. Finally, the power driver card is shown in Figure 6(e).

It is important to mention that this development accomplishes the requirements of the environment where this robot will be working. To achieve this goal some previous prototypes were implemented to check functionality and in the end the final design was carried out.



(a) (b) (c)

(d) (e)

Fig. 6. Experimental implementation of the system

6 Intelligent System for Automatic Control of Bed Positions

A robotic hospital bed can adopt several positions depending on the needs of a particular patient, also can be programmed to perform a series of movements over a period. The figure 7 shows the transition diagram of our robotic hospital bed of the most used positions by medical specialists. The initial position is the *home position* to which all other positions can reach, except the sit-to-stand position. The transitions of the robotic bed are performed using mechanical actuators, and the time it takes to go from one position to another depends on the weight of the patient and the current position. When a transition is performed may happen that the patient falls or can be hurt by being in a bad posture, even when a specialist is operating the robotic bed.

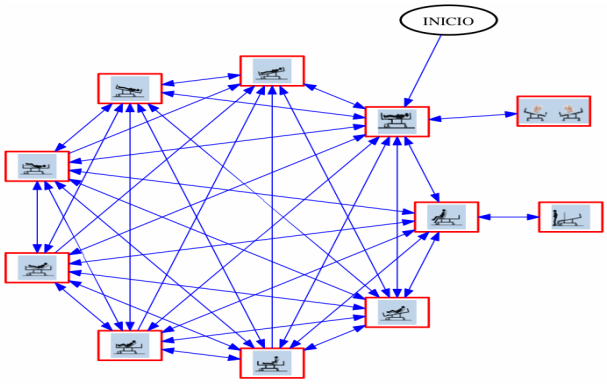


Fig. 7. Pression levels obtained of one person.

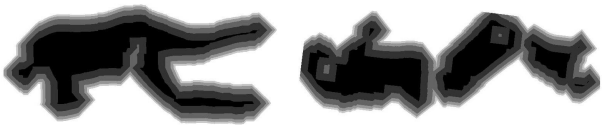
The AIS will be able to detect if the patient is in a correct position to perform the requested transition, and will send a visible alert to prevent possible downfalls. Figure 3 shows the main stages of the AIS for the posture recognition. In the initial phase, the pressure distributions are obtained from the pressure sensor array. The second and third stages present an analysis and pre-processing is performed. Also, a feature extraction using Histogram of Oriented Gradients (HOG) [3] and Scale Invariant Feature Transform (SIFT) [7] descriptors are applied over the pressure distributions that are considered as grayscale images. In the fourth stage, a database of features is constructed and in the last two phases we build a model for feature classification and prediction. We compare the results of three classifiers such as Support Vector Machines (SVM), Decision Trees (DT), and Bayes-Naives Networks (BNN).

To simplify the posture recognition we consider these basic postures: the *right lateral decubitus*, *supine* and the *left lateral decubitus* positions (see Table 1), and since that the *prone* position is almost the same as the supine position, its detection is achieved by an analysis of the pressure distribution. The Table 1 shows the three basic correct positions displayed as grayscale images, obtained from simulated data of the pressure sensor array.

Thus, the database of features (see figure 4) is constructed such as it contains three set of equal size of the three basic positions and an appropriate number of its variants. The database of features is then used to construct the model of the classifier to make predictions of pressure distributions that do not belong to the database.

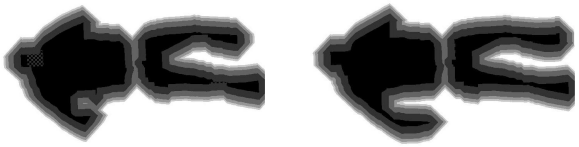
Finally, the AIS for posture recognition can be used to control the actuators of the robotic hospital bed in a semi or automatic way, and can prevent accidents when the bed is moving slowly. Thus, when a bad posture is detected, then the AIS send a signal to the actuators either to stop the transition movement or to return to the previous position. This intelligent device allows the hospital medical team to improve attending in caring for people with motor disabilities.

Table 1. Simulated basic posture positions.



a) Left lateral decubitus LLD (**class 1**)

b) Right lateral decubitus RLD (**class 2**)



c) Supine (**class 3**)

d) Prone (**class 4**)

7 Methodology for Posture Recognition

The proposed methodology takes the raw data of sensors and transforms it into HOG image. This representation as HOG image will be used as input in one SVM classificator, the final output is the classificator prediction about the position. Our system receives, as input, one array of 448 elements, with pression levels (0-4096 units of pression). Each of wich represents one sensor on the surface where a person is lying down; these sensors are distributed in 32 rows and 14 columns.

Figure 4 shows the main blocks of the proposed methodology for posture recognition. In the initial stage, the pressure distribution is obtained from the pressure sensor array. The data obtained from the sensors is in the range of 0 to 4096 levels of pression, where 0 is the maximum pression possible, and 4096 is null pression. Then, in the second block we transform raw data taking three considerations. The first considers the pression applied by the human body (considering weight between 40 to 150 Kg) which is between 2500 to 4096 in the scale of pressures (see Figure 8). Then we can cut the range only to human body requirements and gain definition. Second consideration is an array scalation from 2500-4096 scale to 0-255 scale for process the array as gray scale image in the next block. The final consideration of the second block is applying a scale algorithm for images to obtain interpolated image of raw pressure data.

In third stages, we use a feature extraction using HOG descriptor and applying it to the pressure distributions that are considered as gray scale images. In the stage delimited with a broken line, we make the SVM model, the first three stages are repeated several times with one human body in different positions. We consider the positions described in Table I and we make a data base with this. Then we use this data base for make a SVM model to know the position.

When SVM model is ready, we can use it to monitor person movements. The fourth stage implies to take this SVM model and to use it as input the first three stages output; then the fifth stages have an accurate prediction.

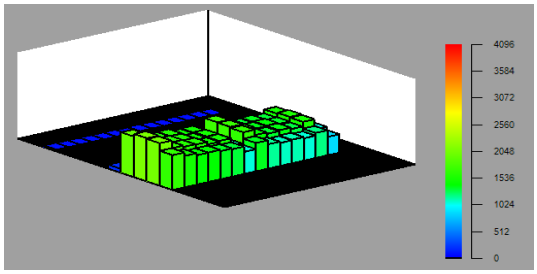


Fig. 8. Pression levels obtained of one person.

8 Interface

Our development also implies a design and manufacturing respectively to graphical interfaces that display the positions of the patient (Figure 10). As well as an exclusive work for the aesthetic look of the device that allows safe, comfortable, reliable and clean. These details are very important in the context of usability to give confidence to the user.

In constant operation of the bed, measured position is presented in the graphical interface, and it must display the real position of the physical system. Therefore, the user must have real information to observe and handle the variables in the machine.

Additionally, this development takes into account the implementation of a function call [11], to isolate the design and constructibility of the Graphical User Interface (GUI). Therefore, this work is done considering the flexibility of the entire system.

In Figure 9, the primary screen of the graphical user interface is shown, which displays patient data found in the bed. The system takes data from the position in which the patient as well as the pressure and temperature continuously to have a better management system is.

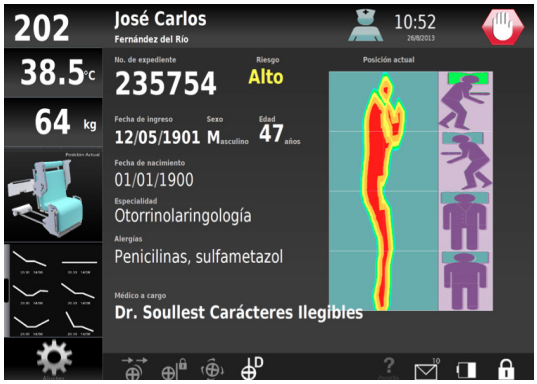


Fig. 9. Interface.

In one of the screens, the twelve possible configurations that can take the bed as well as a short description of the position are shown. Within the interface is the option to set the position or series of positions that will receive the patient at a given time. This is done to continue with routines that the patient has already predefined or update routines depending on the patient's improvement. In the section go to position a preview of the desired configuration is done to take into account which is the position to which anger the system if the desired position the start button is pressed to go to the settings or begin programmed sequence before. The interface has blockages or stops to the bed which depend on the position in which the current system is and who is to come, this depends mainly on the position where the patient is.

Fig. 10. Positions in the interface.

The Figure 11 contains an interactive way to manage the positions of the

system with easy handle method. The use of the touch screen enable the user to perform its task using specific areas in the screen. On this screen limits movement with every part of your system.

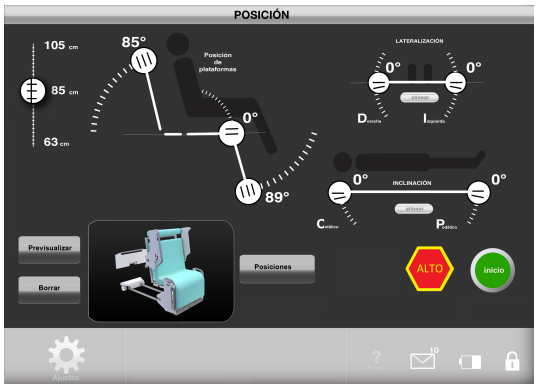


Fig. 11. Movements in the interface.

9 Conclusion

In this work a successful implementation of a data acquisition system was carried out, and the resultant system was applied to manage the behavior of an assistive robot. The design an implementation of an Artificial Inteligent System was carried out. This approach was based on an NI-DAQ-6212 device and all the required electronics to achieve the desired objective using a system with lower capabilities. The design and implementation of a GUI taking into account the actual requirements for applications as well as a cognitive approach to making the product more efficient when the user is interacting with it. This GUI was also applied to command the behavior of a mechanical machine, therefore making a Mechatronic system. The whole system was built and tested with successful results at a prototype level.

Acknowledgment. The authors would like to thank for the support from CINVESTAV.

References

1. Bustamante, M., de Jesús, R.: Diseño y construcción de un prototipo de adquisición de datos para variaciones de voltaje, corriente y temperatura en función del tiempo, utilizando comunicación ethernet para el Laboratorio de Física de la Facultad de Ciencias de la Escuela Politécnica Nacional. Ph.D. thesis, QUITO/EPN/2011 (2011)

2. Chung-Ching, W., Booth, A., Yen-Min, C., Botlo, M.: A graphical environment for daq simulations. Nuclear Science, IEEE Transactions on 41(1), 169–173 (Feb 1994)
3. Dalal, N., Triggs., B.: Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) 1(5), 886–893 (2005)
4. DeVocht, J.W., Wilder, D.G., Bandstra, E.R., Spratt, K.F.: Biomechanical evaluation of four different mattresses. Applied Ergonomics 37(3), 297–304 (2006)
5. Grimm, R., Bauer, S., Sukkau, J., Hornegger, J., Greiner, G.: Markerless estimation of patient orientation, posture and pose using range and pressure imaging. International journal of computer assisted radiology and surgery 7(6), 921–929 (2012)
6. Hamid, U., Qamar, R., Shahzad, M.: Pc based data acquisition and signal processing for underwater sensor arrays. In: Applied Sciences and Technology (IBCAST), 2013 10th International Bhurban Conference on. pp. 321–327 (Jan 2013)
7. Lowe., D.G.: Object recognition from local scale-invariant features. In Proceedings of the International Conference on Computer Vision 2(5), 1150 (1999)
8. Nakazawa, M., Ohi, J., Furumiya, T., Tsuda, T., Furuta, M., Sato, M., Kitamura, K.: Pet data acquisition (daq) system having scalability for the number of detector. In: Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2012 IEEE. pp. 2475–2478 (Oct 2012)
9. Nicol, K., Rusteberg, D.: Pressure distribution on mattresses. Journal of Biomechanics 26(12), 1479 (2015)
10. Ordóñez, M., Bartolomeo, M., Barrera, D.: Diseño de dispositivo para la caracterización de módulos fotovoltaicos
11. Sun-Myung, H., Ja-Yun, J.: Advanced software engineering and its applications. In: A Development Method of GUI in Military System Software. pp. 249–251 (2008)
12. Townsend, D., Holtzman, M., Goubran, R., Frize, M., Knoefel, F.: Relative thresholding with under-mattress pressure sensors to detect central apnea. Instrumentation and Measurement, IEEE Transactions on 60(10), 3281–3289 (Oct 2011)
13. Unel, G., Ambrosini, G., Beck, H.P., Cetin, S., Conka, T., Crone, G., Fernanedes, A., Francis, D., Joos, M., Lehmann, G., Lopez, J., Mailov, A., Mapelli, L., Mornacchi, G., Niculescu, M., Petersen, J., Tremblet, L., Veneziano, S., Wildish, T., Yasu, Y.: Using linux pcs in daq applications. In: Real Time Conference, 1999. Santa Fe 1999. 11th IEEE NPSS. pp. 73–77 (1999)
14. Unel, G., Ambrosini, G., Beck, H., Cetin, S., Conka, T., Crone, G., Fernandes, A., Francis, D., Joos, M., Lehmann, G., Lopez, J., Mailov, A., Mapelli, L., Mornacchi, G., Niculescu, M., Petersen, J., Tremblet, L., Veneziano, S., Wildish, T., Yasu, Y.: Using linux pcs in daq applications. Nuclear Science, IEEE Transactions on 47(2), 109–113 (Apr 2000)

Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Octubre de 2015
Printing 500 / Edición 500 ejemplares