

## Propuesta para determinar la ruta más corta utilizando la arquitectura M-ANFIS

Eduardo Chandomí-Castellanos, Elías N. Escobar-Gómez,  
Sabino Velázquez-Trujillo, J. Alexander López-Rosas,  
Carlos Venturino De-Coss-Pérez, Yordi López-Escalante,  
Daniel Domínguez-Romero, Alejandro Flores-Méndez

Tecnológico Nacional de México,  
I.T. Tuxtla Gutiérrez,  
México

eduardo.chandomi@ittuxtlagutierrez.edu.mx,  
{elias.eg, sabino.vt, carlos.dp }@tuxtla.tecnm.mx,  
{Alexander.lr, yordi\_lopez}@outlook.com,  
daniel\_dr98@hotmail.com, Alejandrofm21@live.com

**Resumen.** Este trabajo propone un método para resolver el problema de la ruta más corta, considerando variables imprecisas. El método se basa en un esquema de lógica difusa y una arquitectura adaptable con redes neuronales. Para considerar la incertidumbre, en el modelo se toma en cuenta el efecto de tres variables: Estado de las calles, Zonas de tráfico y Precipitación pluvial (intensidad de lluvia) junto a una arquitectura adaptable con redes neuronales. Las funciones de pertenencia de estas variables se calculan de manera experimental, obteniendo tiempos de recorrido más cercanos a los reales. El modelo evalúa la incertidumbre para cada una de las aristas de la red (calles), con el objetivo de ajustar el tiempo de recorrido de las rutas establecidas. Para hacer adaptativo el modelo se emplea un sistema de inferencia difuso ANFIS (Adaptive network based Fuzzy Inference System) Mandani, que aproxima el resultado a un valor deseado, finalmente el modelo actualiza y modifica las funciones de pertenencia haciendo adaptable a nuevos escenarios. Para la validación del modelo M-ANFIS se comparan los tiempos de recorrido obtenidos con el modelo propuesto y los obtenidos con lógica difusa con los tiempos de recorrido obtenidos en un muestreo.

**Palabras clave:** Neurodifuso, lógica difusa, redes neuronales artificiales, ruta más corta, ajuste de tiempo.

### Proposal for Determining Shortest Route using M-ANFIS Architecture

**Abstract.** This work proposes a method to solve the shortest path problem with imprecise variables. The method is based on a fuzzy logic

scheme and an adaptive architecture with neural networks. For the uncertainty, the effect of three variables for each street is considered: state of the streets, traffic zones and rainfall (intensity of rain) with an adaptable neural networks architecture. The membership functions are experimentally calculated, getting times closer to the real. The model evaluates the uncertainty for each of the edges of the network (streets), with the objective to adjust set route travel time. The model uses Mandani Adaptive network based fuzzy inference system, which approximates the result to a expected value, finally the model update and modify the membership functions making it adaptable to new scenarios. For the validation is compared the travel time M-ANFIS model with fuzzy logic travel time and desired travel time.

**Keywords:** Neuro fuzzy, fuzzy logic, artificial neural network, shortest path, time adjust.

## 1. Introducción

Los costos de un producto generalmente varían dependiendo del tipo de producto que se va a comercializar, el proceso de manufactura y las condiciones del proceso de distribución, siendo el proceso de distribución quien genere costos del 30 % del costo de producción. En la actualidad, las empresas invierten en máquinas, procesos y métodos que aseguren un producto de calidad, que sea entregado al cliente en tiempo y forma. Esto genera una necesidad en las empresas por buscar alternativas que mejoren la entrega de dichos productos, siendo el primer objetivo que lleguen al lugar adecuado en el tiempo determinado [17], en buenas condiciones y a un costo competitivo [12]. Las mejores soluciones de optimización de ruta ayudan a las empresas de reparto a minimizar el tiempo de conducción y reducir el consumo de combustible.

La inteligencia artificial es una herramienta que nos ayuda a resolver problemas difíciles de modelar. Métodos heurísticos como la lógica difusa o redes neuronales son alternativas con más aplicaciones en el área de ruta más corta [14,1,9,18,5]. En la literatura existen trabajos que presentan redes neuronales para resolver problemas de transporte [7] y planeación de ruta de robots [19,15], entre otros. La unión de estas dos técnicas dan como resultados métodos que adaptan sus parámetros, llamados Neurodifusos [2,11,16,10]. El objetivo principal de este trabajo es aplicar una red Neurodifusa donde las funciones de pertenencia se adaptan y se ajustan a los nuevos escenarios dando como resultado la optimización de la ruta de reparto. Generando para ello nuevas alternativas de recorrido asegurando las entregas oportunas, es decir entregar la mercancía en el horario que se establece. Adicionalmente obtener beneficios como la reducción de gasto de combustible, disminución del mantenimiento del vehículo ocasionado por el desgaste y el recorrido de rutas en mal estado, en igual manera como empresa manejar la política y garantizar la entrega oportuna.

## 2. Definiciones

### 2.1. Ruta más corta

El problema de la ruta más corta es quizás el más simple de todos los problemas de flujo de red. Para este problema, deseamos encontrar una ruta de costo mínimo (o longitud) desde un nodo fuente específico "s" a otro nodo receptor especificado "t", asumiendo que cada arco  $(i, j) \in A$  tiene un costo asociado (o longitud)  $c_{ij}$ . Algunas de las aplicaciones del problema de la ruta más corta son determinar una ruta entre dos nodos especificados de una red que tenga una longitud mínima, o una ruta que tarde menos en atravesar, o una ruta que tenga la máxima confiabilidad.

### 2.2. Neurodifuso

El Sistema de Inferencia Neurodifuso Adaptativo ANFIS (por sus siglas en inglés, *Artificial Neural Fuzzy Inference System*), es una combinación de dos métodos de computación blanda: las Redes Neuronales Artificiales (RNA) y Lógica Difusa (LD) [8]. La Lógica Difusa tiene la capacidad de cambiar los aspectos cualitativos del conocimiento humano y las ideas en el proceso de análisis cuantitativo. Por otro lado, las RNA's, tiene una mayor capacidad en el proceso de aprendizaje, para adaptarse a su entorno.

Las RNA's se utilizan para ajustar automáticamente las funciones de pertenencia partir de datos o para enriquecerlo mediante aprendizaje a partir de ejemplos. Esto reduce la tasa de errores en la determinación de reglas en lógica difusa. El término de "sistemas Neurodifusos" se aplica a sistemas que tengan las siguientes propiedades:

- Se entrenan con un algoritmo de aprendizaje que modifica información del mismo sistema difuso. El proceso de aprendizaje no se basa en conocimiento, sino en un conjunto de datos dado.
- El sistema Neurodifuso toma la forma de una red neuronal multicapa, donde las unidades en sus nodos utilizan operaciones T-norma o T-conorma sustituyendo las funciones de activación que normalmente emplea las redes neuronales. La primera capa representa las variables de entrada. La segunda contiene las funciones de pertenencia y las reglas difusas que representan la experiencia del experto mediante conexiones y capas ocultas. La última capa representa las variables de salida.
- Aproxima una función de  $n$  dimensiones, definida parcialmente de datos de entrenamiento. Las reglas difusas codificadas dentro del sistema representan muestras difusas y se interpretan como patrones datos de entrenamiento.

Los sistemas difusos y de red neuronal trabajan juntos en una arquitectura homogénea híbrida, interpretada como una red neuronal con parámetros difusos o como un sistema difuso con parámetros o funcionamiento distribuidos.

### 3. Método propuesto

En esta investigación se aplica un sistema Neurodifuso que combina las redes neuronales con las técnicas de inferencia difusa. La metodología de los sistemas ANFIS proporcionan alternativas de solución con métodos cualitativos y cuantitativos en la construcción de modelos.

#### 3.1. Variables de entrada

Para la identificación de factores que reducen el tiempo de reparto se realiza un estudio a operarios de vehículos de comida rápida, el estudio detecta 52 factores. En dicho el análisis se determinan aquellos cuyo comportamiento es incierto, con mayor frecuencia y con una relación importante en el proceso de reparto. El modelo propuesto consta de tres variables, que modifican el tiempo de viaje de cada arista de la red, descritos a continuación.

- Estado de la calle (E), es definida por la cantidad y tamaño de obstáculos que varía la velocidad del vehículo, es considerada difusa interpretando un nivel para cada estado de la calle.
- Zona con tráfico (Z), se consideran ciertas zonas que generan más tráfico al momento de transitar, debido a la proximidad de zonas escolares, centros comerciales, entre otros.
- Precipitación pluvial (P), la intensidad de lluvia es la cantidad de agua que se precipita en el área donde se conduce, la que provoca una disminución de visibilidad y de velocidad del vehículo; esta variable está medida en  $ml/m^2$ .

#### 3.2. Modelo difuso

El sistema difuso y las funciones de pertenencia calculadas de manera experimental, fueron presentados en el Congreso Mexicano de Inteligencia Artificial (COMIA) 2019 y publicados como número especial de la revista Research in Computing Science, ISSN: 1870-4069, indexada en Latinindex y DBLP [3].

#### 3.3. Selección del tipo de arquitectura multicapa

Los modelos ANFIS integran información incluida dentro de un conjunto de datos con el conocimiento de expertos, expresados en forma lingüística. Se utiliza una arquitectura basada en el mecanismo de inferencia tipo mamdani (M-ANFIS) [6], Ecuación 1:

$$\text{Regla } k = \text{si } P \text{ es } M_{i,j}, E \text{ es } M_{i,j} \text{ y } Z \text{ es } M_{i,j} \text{ entonces } D_k = M_{ok}(\mu; AD_{ok}). \quad (1)$$

En la Figura 1 se ilustra la arquitectura FIS de Mamdani[13]. La base de reglas en este trabajo se escribe en la Ecuación 2.

1. **si** ( $P$  es  $M_{11}$ ), ( $E$  es  $M_{21}$ ), ( $Z$  es  $M_{31}$ ), **entonces**  $D_1 = M_{o1}(\mu; AD_{o1})$ ,
  2. **si** ( $P$  es  $M_{11}$ ), ( $E$  es  $M_{21}$ ), ( $Z$  es  $M_{32}$ ), **entonces**  $D_2 = M_{o2}(\mu; AD_{o2})$ ,
  3. **si** ( $P$  es  $M_{11}$ ), ( $E$  es  $M_{21}$ ), ( $Z$  es  $M_{32}$ ), **entonces**  $D_3 = M_{o3}(\mu; AD_{o3})$ ,
  - $\vdots$ ,
  53. **si** ( $P$  es  $M_{16}$ ), ( $E$  es  $M_{22}$ ), ( $Z$  es  $M_{33}$ ), **entonces**  $D_{53} = M_{o53}(\mu; AD_{o53})$ ,
  54. **si** ( $P$  es  $M_{16}$ ), ( $E$  es  $M_{23}$ ), ( $Z$  es  $M_{31}$ ), **entonces**  $D_{54} = M_{o54}(\mu; AD_{o54})$ ,
- (2)

donde: P, E, Z, corresponde a las variables de entrada, Precipitación pluvial, Estado de la calle y Zona con tráfico respectivamente.  $M_{ij}$  representan la  $j$ -ésima función de pertenencia de la  $i$ -ésima entrada,  $D_k$  y  $M_{ok}$  la salida de la  $k$ -ésima regla, y la  $k$ -ésima FM de salida.  $AD_{ok}$  son los parámetros consecuentes que caracterizan las formas de las funciones de pertenencia de salida, llamada Ajuste Difuso.

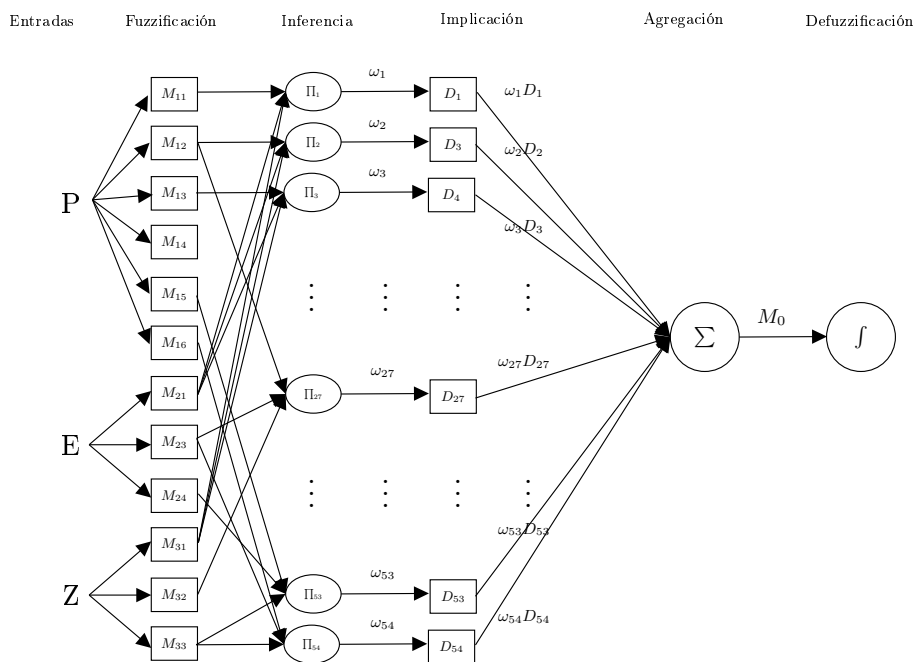


Fig. 1. propuesta de la arquitectura del sistema de inferencia difuso de Mamdani

### 3.4. Desarrollo de la arquitectura tipo Mamdani

El modelo ANFIS Mamdani consta de 5 capas (Layers), cada salida de la capa se describe a continuación:

**Capa1 (Fuzzificación):** Las entradas de esta capa P, E, Z representan las entradas reales del sistema. La salida representa el grado de pertenencia para el cual la variable de entrada satisface el término lingüístico. En este trabajo se utilizan funciones de pertenencia calculadas de manera experimental, descritos en las Ecuaciones 3-7 [3].

No realizan ningún tipo de procesamiento sobre los valores de entrada de las variables, simplemente se encargan de distribuir estos valores a la capa siguiente:

- i) Gaussianas para las entradas  $M_{i,j}$  para  $(i = 1), (j = 2, 3, 4, 5); (i = 2), (j = 2); (i = 3), (j = 2)$ :

$$\mu_{\{M_{i,j}\}}(x_i) = \begin{cases} e^{-k_j(x_i - m_j)^2}, & 0 \leq x_i \leq b_j, \\ 0, & b_j > x_i. \end{cases} \quad (3)$$

- ii) Pseudo-exponencial para la entrada  $M_{i,j}$   $(i = 2), (j = 3); (i = 3), (j = 1, 3)$ :

$$\mu_{\{M_{i,j}\}}(x_i) = \begin{cases} \frac{1}{1+k_j(x_i - n_j)^2}, & a \leq x_i \leq b_j, \\ 0, & x_i < a_j. \end{cases} \quad (4)$$

- iii) Función de pertenencia Singleton para la entrada  $M_{1,1}$ :

$$\mu_{\{M_{1,1}\}}(x_i; a_j) = \begin{cases} 0, & x_i = a_j, \\ 1, & x_i \neq a_j. \end{cases} \quad (5)$$

- iv) Gamma para la función de pertenencia,  $M_{1,6}$ :

$$\mu_{\{M_{1,6}\}}(x_i) = \begin{cases} 1 - e^{-k_j(x_i - p_j)^2}, & d_j \leq x_i \leq e_j, \\ 0, & 0 < d_j - . \end{cases} \quad (6)$$

- v) La función Sigmoidal corresponde únicamente a la entrada  $M_{2,3}$ :

$$\mu_{\{M_{2,3}\}}(x_i) = \begin{cases} \frac{1}{1+e^{-k_j(x_i - q_j)}}, & a_j \leq x_i \leq b_j, \\ 0, & x_i < b_j, \end{cases} \quad (7)$$

donde  $a_j, b_j, d_j, e_j, k_j, m_j, n_j, p_j, q_j$  son los parámetros que caracterizan a las funciones de pertenencia de cada variable de entrada.

**Capa 2 (Capa de Inferencia o Capa de Reglas):** En esta capa cada nodo le corresponde una etiqueta lingüística (bueno, regular, malo entre otros), para cada una de las variables de entrada de cada regla se calcula  $\omega_k$ , considerado como el factor de ponderación o fuerza de disparo, se determina evaluando las expresiones de pertenencia de la regla antecedente.

Los valores de entrada o valores reales se transforman en valores difusos usando las funciones de cada variable de la capa 1, Ecuación 8.

$$\begin{aligned}\omega_1 &= \mu_{M11}(P) \mu_{M21}(E) \mu_{M31}(Z), \\ \omega_2 &= \mu_{M12}(P) \mu_{M21}(E) \mu_{M31}(Z), \\ \omega_3 &= \mu_{M13}(P) \mu_{M21}(E) \mu_{M31}(Z), \\ &\vdots \\ \omega_{53} &= \mu_{M15}(P) \mu_{M23}(E) \mu_{M32}(Z), \\ \omega_{54} &= \mu_{M16}(P) \mu_{M23}(E) \mu_{M33}(Z).\end{aligned}\tag{8}$$

**Capa 3 (Implicación o Antecedente):** Un consecuente es un conjunto difuso representado por una función de pertenencia  $Z = D_i$ . Existen métodos conocidos, dos de los más usados son: el método "y" (mínimo), que trunca el conjunto difuso de salida, y el PROD (producto), el cual escala al conjunto difuso de salida.

El grado de verdad de la premisa de cada regla es computada y aplicada al consecuente de cada regla. Este resultado es un subconjunto difuso que será asignado a cada variable de salida correspondiente. Se aplica el método de Mamdani o método de inferencia MIN-MAX. Para cada regla se hace el MIN entre el nivel de disparo y la función de pertenencia de la variable de salida indicada en el consecuente. En esta capa, la implicación de cada salida FM se calcula con la Ecuación 9:

$$M_{imp,k} = \omega_k D_k, \quad k = 1, 2, 3, \dots, 53, 54,\tag{9}$$

donde,  $M_{imp,k}$  representa la implicación de la función de pertenencia de salida.

**Capa 4 (Agregación o Consecuente):** En esta capa, la agregación se realiza para producir una salida general,  $M_o(u)$ , utilizando el operador de unión o máximo, Ecuación 10:

$$M_o(u) = \sum_{k=1}^{54} M_{imp,k}.\tag{10}$$

Sustituyendo la Ecuación 9 en 10, se obtiene 11:

$$M_o(u) = \sum_{k=1}^{54} \omega_k D_k,\tag{11}$$

donde,  $D_k$  representa la salida "entonces", el cual describe una función de pertenencia, reemplazando la función de pertenencia pseudo-exponencial en la Ecuación 11 se obtiene 12:

$$M_o(u) = \sum_{k=1}^{54} \omega_k e^{-k(x-m)^2}.\tag{12}$$

**Capa 5 (Combinación o Defuzificación):** La salida es calculada utilizando el método de centroide o centro de gravedad (COG) que retorna el centro del área bajo la curva, descrita en la Ecuación 13:

$$o = \frac{\int M_o(u) u du}{\int M_o(u) du}. \tag{13}$$

### 3.5. Actualización de pesos

La determinación de los parámetros del sistema de inferencia difusa es de vital importancia. Se utiliza un algoritmo de optimización para determinar estos parámetros, se realiza de tal forma que se minimiza la medida de error entre el destino y la salida real.

Durante el proceso de aprendizaje del FIS, los parámetros de la premisa y consecuentes se sintonizan hasta que se alcanza la respuesta deseada del FIS.

Los ANFIS utilizan sistemas de aprendizaje híbridos, dichos métodos son del gradiente descendiente y el estimador de mínimos cuadrados.

Más específicamente, la regla realiza la minimización del error al cuadrado, entre la entrada  $u$  y la salida  $d$  con el objetivo de ajustar el vector de peso  $w = [\theta, w_1, w_2, \dots, w_n]^T$  de la red. En resumen, el objetivo es obtener un peso  $w^*$  óptimo para que el error al cuadrado  $\{E(w^*)\}$  de todo el conjunto de muestras sea lo más bajo posible. En [4] se afirma la notación matemática, considerando una configuración de peso óptima, Ecuación 14:

$$E(w^*) \leq E(w), \quad \forall w \in \mathfrak{R}^{n+1}. \tag{14}$$

La función del error al cuadrado relacionado con las  $p$  muestras de entrenamiento se define en la Ecuación 15:

$$E(w) = \frac{1}{2} \sum_{k=1}^P (d^k - u)^2. \tag{15}$$

Sustituyendo el resultado de la Ecuación 15 requerida para obtener la salida de una neurona ADALINE, Ecuaciones 16 y 17:

$$E(w) = \frac{1}{2} \sum_{k=1}^P \left( d^k - \left( \sum_{i=1}^n w_i x_i^k - \theta \right) \right)^2, \tag{16}$$

$$E(w) = \frac{1}{2} \sum_{k=1}^P (d^k - (w^k x_i^k - \theta))^2. \tag{17}$$

Así, la expresión de la Ecuación 17 calcula el error cuadrático medio para las  $p$  muestras de entrenamiento proporcionadas para el proceso de entrenamiento del ADALINE. El siguiente paso consiste en la aplicación del operador de gradiente en el error cuadrático medio con respecto al vector  $w$ , con el fin de buscar un valor óptimo para la función de error cuadrático, es decir, Ecuación 18:



$$\nabla E(w) = \frac{\partial E(w)}{\partial w}. \quad (18)$$

Sustituyendo 17 en 18 se obtienen 19 y 20.

$$\nabla E(w) = \sum_{k=1}^P (d^k - (w^T x_i^k - \theta)) (-x^k), \quad (19)$$

$$E(w) = \frac{1}{2} \sum_{k=1}^P \left( d^k - \left( \sum_{i=1}^n w_i x_i^k - \theta \right) \right)^2. \quad (20)$$

Finalmente, los pasos para adaptar el vector de peso se deben ejecutar en la dirección opuesta del gradiente, el objetivo de optimización es minimizar el error cuadrático. La variación  $D_w$  para actualizar el vector de peso ADALINE viene dada por la Ecuación 21:

$$\nabla E(w) = - \sum_{k=1}^P (d^k - u) x^k. \quad (21)$$

Insertando (20) en (21) se obtiene (22):

$$\nabla w = \eta \nabla E(w). \quad (22)$$

De manera complementaria, se expresa (22) como (23):

$$w^{actual} = w^{anterior} + \eta \sum_{k=1}^P (d^k - u) x^k. \quad (23)$$

Simplificando la expresión,  $w$  se actualiza después de presentar cada  $k$ -ésima muestra de entrenamiento, Ecuación 24:

$$w^{actual} = w^{anterior} + \eta (d^k - u) x^k, \quad (24)$$

donde:

$w = [\theta, w_1, w_2, \dots, w_n]^T$ , es el vector que contiene el umbral y los pesos.  $x^k = [-1, x_1^k, x_2^k, \dots, x_n^k]^T$ , es la  $k$ -ésima muestra de entrenamiento.  $d^k$ , es la  $k$ -ésima muestra de entrenamiento.  $u$ , es el valor de salida del elemento lineal.  $\eta$ , es una constante que define la tasa de aprendizaje.

Se procede a indicar los pasos a seguir para la implementación de esta regla.

1. Se inicializan los parámetros de cada función de pertenencia con valores pequeños al azar.
2. Se selecciona el par de entrenamiento y se aplica como vector de entrada a la red.
3. Se calcula la salida de la red.
4. Se calcula el error entre la salida de la red y la salida deseada, y se acumula el valor en el error total.
5. Se repiten los pasos del 1 al 4 por cada vector en el conjunto de entrenamiento hasta que el error del conjunto sea suficientemente bajo.

#### 4. Análisis de resultados

Una red adaptable es una estructura de red con un comportamiento general de entrada-salida y está determinado por una colección de parámetros modificables. El modelo se adapta cuando los parámetros se modifican y se ajustan hasta obtener un valor lo bastante aproximado a la salida real.

Este mapeo no lineal está regulado por un conjunto de datos de entrada-salida deseados de un sistema. Este conjunto de datos se denomina conjunto de datos de entrenamiento así mismo los procedimientos que ajustan los parámetros para mejorar el rendimiento de la red se denominan reglas de aprendizaje.

Una regla de aprendizaje explica cómo se deben actualizar estos parámetros (o pesos) para minimizar una medida de error predefinida. En la Figura 2 se observa la comparación entre la salida estimada con el método M-ANFIS y la salida muestreada de una ruta seleccionada de 15 calles de la ciudad de Tuxtla Gutiérrez, Chiapas; se observa puntos donde la diferencia mínima es de 0.007 (nodo 6) y la diferencia máxima es de 0.19 (nodo 14), siendo la mayoría menor que 0.1. El modelo ajusta estos parámetros hasta obtener resultados aproximados a los deseados, minimizando el error entre ambos conjuntos, esta medida de error calcula la discrepancia entre la salida real de la red y una salida deseada (Figura 3), entre más cercano esté a cero mejor será el resultado.

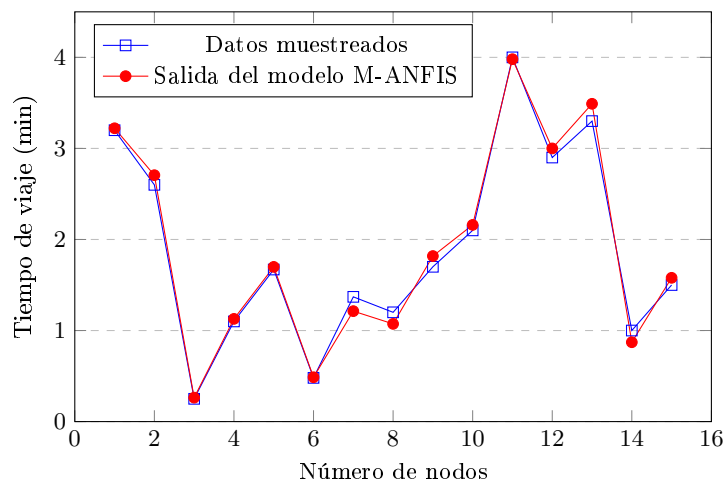


Fig. 2. Datos muestreados vs salida adaptativa

El modelo propuesto aproxima el valor de salida difusa con el valor muestreado del tiempo recorrido. El modelo aprende a partir de un vector de entrada con datos de recorrido, un vector de salida con tiempos estimados por el modelo, 1000 épocas con un mínimo de error de 0.01 y una tasa de aprendizaje de 0.001.

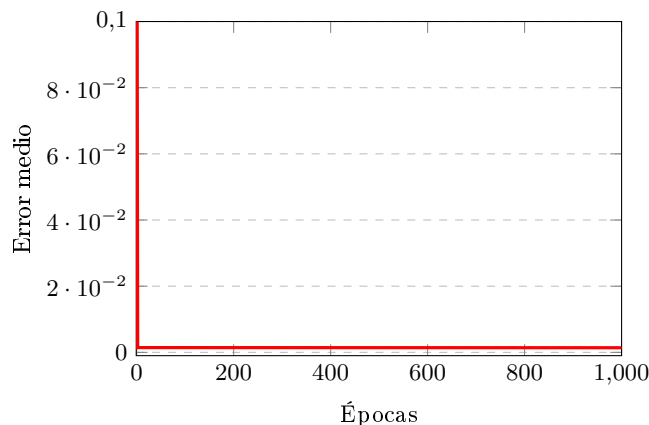


Fig. 3. Convergencia a cero del error cuadrático medio

En la Tabla 1 se muestran los tiempos estimados con lógica difusa comparadas con los tiempos estimados aplicando el modelo M-ANFIS. Los resultados de ambos modelos se comparan con los tiempos obtenidos en el muestreo. Esto con el objetivo de comparar el error porcentual absoluto medio de los modelos con Lógica Difusa y el error porcentual absoluto medio del modelo M-ANFIS. Se observa en la Tabla 2 las medias de los errores porcentuales de LD con 25.06 % y 5.14 % de M-ANFIS, las varianzas de 11 % y 0.14 % respectivamente, comprobando así que existe menos variabilidad con el modelo M-ANFIS.

Una vez calculado el error se ajustan los pesos, que son parámetros que modifican las funciones de pertenencia del modelo, a continuación se muestra la diferencia de las funciones de pertenencia antes de aprender y después, ver Figura 4(a) y 4(b). Como en el caso de estudio no hubo lluvias, no se activaron reglas de dicha variable y no se vieron modificadas en la investigación.

## 5. Conclusiones y recomendaciones

En conclusión la combinación de sistemas adaptables y su cualidad de ajuste de parámetros con el conocimiento de expertos en forma de reglas del tipo "si... entonces", dado un conjunto de pares de datos de entrada y salida, se adapta y aproxima los valores a valores cercanos a los reales. Estos sistemas son convenientes para repetir el comportamiento de un sistema no lineal cuyo modelo matemático no es obtenido fácilmente.

Las variables difusas utilizadas en este trabajo fueron elegidas por su frecuencia y su ocurrencia de aparición, en el estudio se encontraron más variables que afectan ocasionalmente el tráfico y que contienen un impacto en el tiempo de recorrido, para futuros trabajos se recomienda evaluar esos factores e integrarlos en un sistema que evalúe esas y otras variables consideradas en entornos diferentes.

**Tabla 1.** Comparación de la salida difusa con la salida adaptativa.

Variables					Valor Real (min)	Lógica Difusa		M-ANFIS	
Difusas				Salida		Diferencia	Salida	Diferencia	
P	E	Z							
1	0	1	1	3.3	2.9	0.4	3.2213	0.0787	
2	0	7	1	2.6	2.552	0.048	2.706	0.106	
3	0	1	21	0.25	0.2625	0.0125	0.263	0.013	
4	0	1	21	1.1	0.57	0.53	1.128	0.028	
5	0	1	1	1.67	1.695	0.025	1.699	0.029	
6	0	1	1	0.48	1.1078	0.6278	0.487	0.007	
7	0	1	1	1.37	0.87	0.5	1.213	0.157	
8	0	1	1	1.2	1.088	0.112	1.073	0.127	
9	0	1	1	1.7	1.856	0.156	1.818	0.118	
10	0	1	1	2.1	1.5296	0.5704	2.16	0.06	
11	0	1	42	4	3.977	0.023	3.98	0.02	
12	0	23	21	2.9	3.2038	0.3038	3	0.1	
13	0	14	1	3.3	4.8361	1.5361	3.49	0.19	
14	0	1	21	1	1.1625	0.1625	0.8715	0.1285	
15	0	1	21	1.5	1.8075	0.3075	1.579	0.079	
Total				28.47	29.4178		28.6888		

**Tabla 2.** Datos para comparar el error porcentual absoluto medio.

	Tiempo de recorrido muestreado	Tiempo con LD con LD	Porcentaje de error con LD	Tiempo con M-ANFIS	Porcentaje de error con M-ANFIS
1	3.3	2.9	12 %	3.2213	2 %
2	2.6	2.552	2 %	2.706	4 %
3	0.25	0.2625	5 %	0.263	5 %
4	1.1	0.57	48 %	1.128	3 %
5	1.67	1.695	1 %	1.699	2 %
6	0.48	1.1078	131 %	0.487	1 %
7	1.37	0.87	36 %	1.213	11 %
8	1.2	1.088	9 %	1.073	11 %
9	1.7	1.856	9 %	1.818	7 %
10	2.1	1.5296	27 %	2.16	3 %
11	4	3.977	1 %	3.98	1 %
12	2.9	3.2038	10 %	3	3 %
13	3.3	4.8361	47 %	3.49	6 %
14	1	1.1625	16 %	0.8715	13 %
15	1.5	1.8075	21 %	1.579	5 %
Media			25.06 %		5.14 %
Varianza			11.0 %		0.14 %

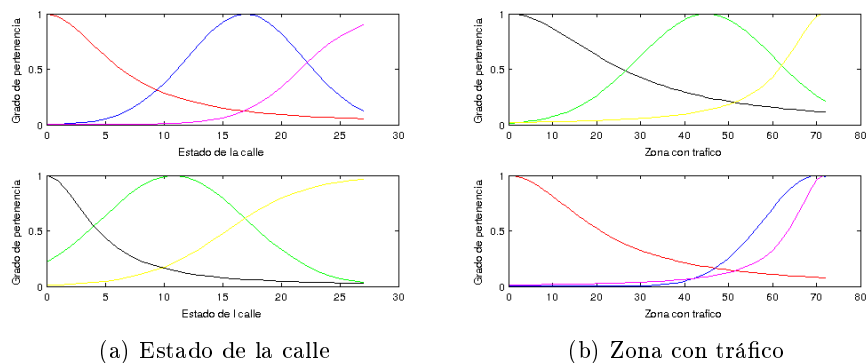


Fig. 4. Funciones de pertenencia antes y después del aprendizaje

Opcionalmente se recomienda comparar los resultados del sistema M-ANFIS con otros sistemas de adaptabilidad como lo son algoritmos genéticos y evaluar la eficiencia que alguno de los dos tendría en un sistema informático, a diferencia de las RNA's los Algoritmos Genéticos son utilizados en la resolución de problemas de búsqueda y optimización reproduciendo genéticamente una población a lo largo de una serie de generaciones.

## Referencias

1. Arnold, W., Hellendoorn, H., Seising, R., Thomas, C., Weitzel, A.: Fuzzy routing. *Fuzzy Sets and Systems* 85(2), 131-153 (1997)
2. Chai, Y., Jia, L., Zhang, Z.: Mamdani model based adaptive neural fuzzy inference system and its application. *International Journal of Computational Intelligence* 5(1), 22-29 (2009)
3. Chandomí-Castellanos, E., Escobar-Gómez, E.N., Velázquez-Trujillo, S., De León, H.R.H., Pérez-Patricio, M., Pérez, C.V.D.C., Gutiérrez, T.: Modelo para la determinación de la ruta más corta con funciones experimentales para arcos difusos. *Research in Computing Science* 148, 317-330 (2019)
4. Da Silva, I.N., Spatti, D.H., Flauzino, R.A., Liboni, L.H.B., dos Reis Alves, S.F.: *Artificial Neural Networks: A Practical Course*. Springer (2016)
5. Escobar-Gómez, E., Camas-Anzueto, J., Velázquez-Trujillo, S., Hernández-de León, H., Grajales-Coutiño, R., Chandomí-Castellanos, E., Guerra-Crespo, H.: A linear programming model with fuzzy arc for route optimization in the urban road network. *Sustainability* 11(23), 6665 (2019)
6. Guney, K., Sarikaya, N.: Comparison of mamdani and sugeno fuzzy inference system models for resonant frequency calculation of rectangular microstrip antennas. *Progress In Electromagnetics Research* 12, 81-104 (2009)
7. Ida, K., Gen, M., Li, Y.: Neural networks for solving multicriteria solid transportation problem. *Computers & industrial engineering* 31(3-4), 873-877 (1996)
8. Jang, J.S.: Anfis: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics* 23(3), 665-685 (1993)

9. Ji, X., Iwamura, K., Shao, Z.: New models for shortest path problem with fuzzy arc lengths. *Applied Mathematical Modelling* 31(2), 259–269 (2007)
10. Kim, S.W., Park, S.Y., Park, C.: Spacecraft attitude control using neuro-fuzzy approximation of the optimal controllers. *Advances in Space Research* 57(1), 137–152 (2016)
11. Kiran, T.R., Rajput, S.: An effectiveness model for an indirect evaporative cooling (iec) system: Comparison of artificial neural networks (ann), adaptive neuro-fuzzy inference system (anfis) and fuzzy inference system (fis) approach. *Applied Soft Computing* 11(4), 3525–3533 (2011)
12. Kotler, P., Armstrong, G., Brown, L.G., McEnally, M.R., Paczkowski, T.J.: *Principles of Marketing: Company Case & Video Commentaries*. Prentice Hall (1991)
13. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies* 7(1), 1–13 (1975)
14. Okada, S., Gen, M.: Fuzzy shortest path problem. *Computers & industrial engineering* 27(1-4), 465–468 (1994)
15. Orozco-Soto, S.M., Fernández, J.C.R.: Control par calculado difuso basado en pasividad para seguimiento de trayectorias de robots manipuladores. *Research in Computing Science* 91, 131–141 (2015)
16. Suparta, W., Alhasa, K.M.: Modeling of tropospheric delays using anfis. *Springer-Briefs in Meteorology* (2016)
17. Visser, J., Nemoto, T., Browne, M.: Home delivery and the impacts on urban freight transport: A review. *Procedia-social and behavioral sciences* 125, 15–27 (2014)
18. Wei, D.c.: Implementation of route selection function based on improved floyd algorithm. In: 2010 WASE International Conference on Information Engineering. vol. 2, pp. 223–227. IEEE (2010)
19. Yang, S.X., Luo, C.: A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 34(1), 718–724 (2004)