

Un Algoritmo Genético para Aprendizaje de un Sistema Basado en Reglas Difusas tipo Takagi Sugeno

F. Herrera¹, P. Villar²

Resumen— Este trabajo presenta un método de aprendizaje de la base de conocimiento de un Sistema Basado en Reglas Difusas tipo Takagi Sugeno mediante ejemplos. Para el aprendizaje de las particiones difusas utilizadas en los antecedentes se utiliza un Algoritmo Genético que incluye tanto el número de etiquetas de cada partición como la semántica de dichas etiquetas. Para el aprendizaje de las reglas se utiliza un mecanismo simple de cubrimiento en los antecedentes y los coeficientes de los consecuentes se aprenden mediante mínimos cuadrados

Palabras clave— Aprendizaje automático, sistemas basados en reglas difusas TS, algoritmos genéticos,

I. INTRODUCCIÓN

El modelado difuso lingüístico es una aplicación importante de los Sistemas Basados en Reglas Difusas (SBRDs). En este tipo de modelos, la interpretabilidad es un requisito fundamental, junto con la precisión. En algunos casos, se necesita algo más de precisión de la que pueden proporcionar a costa de perder algo de interpretabilidad. Los SBRDs de Tipo Takagi-Sugeno (TS) presentan este comportamiento ya que los antecedentes de las reglas difusas toman valores de un conjunto de términos con una interpretación en el mundo real (variables lingüísticas) mientras que los consecuentes son una combinación lineal de los valores de entrada (sin interpretación sencilla).

La Base de Conocimiento (BC) de un SBRD TS está formada por dos componentes: 1) la Base de Reglas (BR), constituida por el conjunto de reglas difusas, y 2) la Base de Datos (BD), que incluye las funciones de pertenencia de cada uno de los términos lingüísticos (o etiquetas) de las particiones difusas asociadas a las variables de entrada. La BC de un SBRD TS depende del problema que se intente resolver, por tanto, se hace necesario algún mecanismo de derivación automática de dicha BC. Dentro de la literatura especializada existen métodos de aprendizaje de la BR a partir de una BD previamente definida. En la mayoría de esos métodos, se consideran particiones difusas uniformes con el mismo número de etiquetas para todas las variables. Sin embargo, ya se ha demostrado para SBRD lingüísticos (donde los consecuentes también son etiquetas lingüísticas), que la elección de la BD tiene una decisiva influencia en el comportamiento del SBRD resultante[6]

¹Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S. Ing. Informática, Avda. Andalucía, 18, 18071 Granada (España) herrera@decsai.ugr.es

² Departamento de Lenguajes y Sistemas Informáticos, E.T.S. Ing. Informática, Avda. Andalucía, 18, 18071 Granada (España) pvillar@ugr.es

Teniendo en cuenta la fuerte interrelación entre los dos componentes de la BC, sería muy aconsejable un mayor grado de cooperación entre las tareas de aprendizaje de la BR y la BD, que permitiese obtener modelos con un buen equilibrio entre interpretabilidad y precisión. Basándonos en esta idea, se propone un método de aprendizaje de la BC que mantiene una población de posibles definiciones de la BD al mismo tiempo que busca un conjunto compacto de reglas (BR). Más concretamente, para la derivación de las reglas difusas utilizaremos un sencillo mecanismo de cubrimiento de ejemplos para escoger los antecedentes, mientras que los coeficientes que aparecen en el consecuente se obtendrán mediante regresión por mínimos cuadrados. Para la definición de la BD se empleará un Algoritmo Genético que incluye sus principales componentes para cada variable: dominio, número de etiquetas y partición difusa, siguiendo la propuesta presentada en [5] para el caso de SBRDs de tipo Mamdani. Describiremos primero la estructura de un SBRD TS. A continuación, se presenta el método propuesto y finalmente, se muestran unos resultados experimentales y algunas conclusiones.

II. SISTEMAS BASADOS EN REGLAS DIFUSAS TS

En [11], Takagi y Sugeno propusieron una herramienta matemática para obtener un modelo difuso para un sistema. Dicho modelo está basado en reglas difusas que presentan la siguiente estructura:

$$R_i : \text{If } X_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } X_n \text{ is } A_{in} \\ \text{Then } Y = p_{i1} \cdot X_1 + \dots + p_{in} \cdot X_n + p_{i0},$$

donde las X_i son las variables de entrada del sistema mientras que Y es la variable de salida de dicho sistema, que determina una relación lineal entre la entrada y la salida por medio de los coeficientes p_{ij} . La salida de un SBRD con una base de conocimiento con m reglas TS es la media ponderada de cada una de las reglas individuales, y_i , con $i = 1, \dots, m$:

$$\frac{\sum_{i=1}^m h_i \cdot y_i}{\sum_{i=1}^m h_i},$$

siendo $h_i = T(A_1(x_1), \dots, A_n(x_n))$ el grado de emparejamiento entre el antecedente de la i -ésima regla y la entrada actual al sistema (x_1, \dots, x_n) , y siendo T una T-norma.

Cada una de esas relaciones parciales se combina mediante una agregación, teniendo en cuenta su dominancia en su área de aplicación respectiva y el conflicto existente en las áreas solapadas [11]. De esta manera, los

SBRDs TS presentan una serie de características interesantes como pueden ser su localidad y la existencia de herramientas matemáticas para el diseño del sistema.

III. MÉTODO DE APRENDIZAJE DE LA BASE DE CONOCIMIENTO DE UN SBRD TS

En este apartado describiremos nuestra propuesta para el aprendizaje automático de la BC de un SBRD TS, compuesta por dos mecanismos:

- Un proceso de aprendizaje de la BD mediante un Algoritmo Genético (AG) que nos va a definir:
 - Dominio de cada variable de entrada que interviene en el modelo. El dominio inicial de cada variable se puede alargar ligeramente por los dos extremos.
 - Número de etiquetas de cada variable de entrada.
 - Semántica de cada etiqueta considerando particiones difusas no uniformes. Para construir dichas particiones se emplea una función de escala no lineal que define áreas con mayor o menor “sensibilidad”.
- el algoritmo de aprendizaje de reglas, que deriva la BR tomando como punto de partida la BD previamente obtenida en el paso anterior. De esta manera, el método híbrido propuesto obtiene una definición completa de la BC mediante la acción cooperativa de ambos procesos.

Ya que todos los componentes de la BD irán evolucionando mediante un AG, sería aconsejable reducir la dimensión del espacio de búsqueda. Por tanto, el uso de una función de escala no lineal debería estar condicionado a escoger una función parametrizada con un reducido número de parámetros. Para este trabajo, hemos considerado la función propuesta en [9], que solamente tiene un parámetro denominado a ($a \in \mathbb{R}$). La función es:

$$f : [-1, 1] \rightarrow [-1, 1]$$

$$f(x) = \text{sign}(x) \cdot |x|^a, \quad \text{con } a > 0$$

El resultado final es un valor entre $[-1, 1]$. La acción del parámetro a en la partición difusa depende de su valor: sensibilidad uniforme ($a = 1$), mayor sensibilidad en los valores centrales ($a > 1$), o mayor sensibilidad en los valores extremos ($a < 1$). En este trabajo se han considerado funciones de pertenencia triangulares, por tanto, la función de escala no lineal sólo se aplica a los tres puntos que definen la función de pertenencia. Es importante destacar que esa función produce efectos simétricos alrededor del punto central del dominio. La Figura 1 muestra esas tres posibilidades.

Esta función de escala es válida para variables simétricas ya que produce los mismos efectos en la partición difusa de forma simétrica sobre el centro de dicha partición. Para permitir que el área de mayor sensibilidad de la partición se desplace sobre un único extremo de la misma, se añade un nuevo parámetro (S) como se describe en [5]. Dicho parámetro no aumenta significativamente la complejidad del espacio de búsqueda ya que únicamente puede tomar dos valores ($\{0, 1\}$), que permiten desplazar el área de mayor sensibilidad hacia un

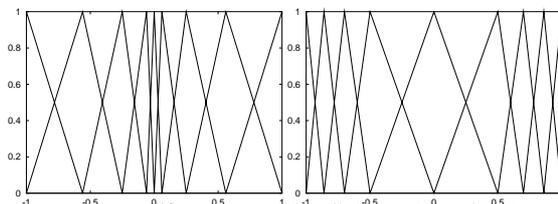
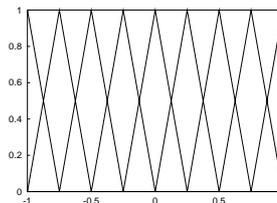


Fig. 1. Particiones difusas con $a = 1$ (arriba), $a > 1$ (izquierda) y $a < 1$ (derecha)

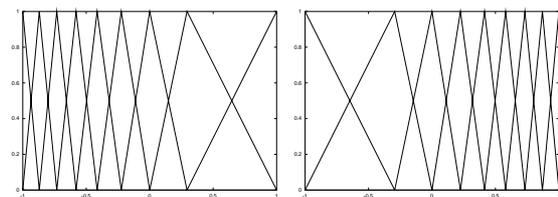


Fig. 2. Particiones difusas con $S = 1$ (izquierda con $a > 1$ y derecha con $a < 1$)

extremo u otro si $S = 1$ tal y como se puede ver en la Figura 2.

Es importante resaltar que cualquiera de las anteriores transformaciones de las particiones difusas siguen obteniendo lo que se conoce como una partición difusa “fuerte”, en la que la suma de los grados de pertenencia en todas las etiquetas de cada punto del dominio de la variable es uno.

A continuación, se describirá el proceso evolutivo que monitoriza el aprendizaje de la BC. Los AGs[10] son técnicas de optimización y búsqueda basadas en una formalización del proceso evolutivo de los seres vivos. Cuando se trabaja con AGs, los aspectos importantes son cómo codificar cada solución (en este caso, la BD de un SBRD TS), como evaluar esa solución y como crear nuevas soluciones:

A. Codificación de la BD.

Para un modelo con N variables de entrada, cada individuo o cromosoma está compuesto de cuatro partes:

- Número de etiquetas (C_1): codificados en un vector de valores enteros de longitud N . Para este trabajo, el rango considerado es $E \in \{2, \dots, 7\}$.
- Parámetros de desplazamiento a uno de los extremos (el parámetro S comentado anteriormente): (C_2): codificados en un vector de enteros de longitud N , que puede tomar los valores $S \in \{0, 1\}$.
- Parámetros de sensibilidad (C_3): codificados en un vector de valores reales de longitud N , donde se almacena el valor del parámetro (a) de la función de escala no lineal para cada variable. El rango considerado es el intervalo $(0, 10)$.
- Dominios (C_4): codificados en un vector de $N \times 2$ valores reales que almacenan el dominio de cada va-

riable ($[v_{min}, v_{max}]$). Si el dominio inicial de una variable es $[v_{inf}^i, v_{sup}^i]$, y d es la dimensión del intervalo ($d = v_{sup}^i - v_{inf}^i$), el rango para el límite inferior del dominio es $[v_{inf}^i - (1/4 * d), v_{inf}^i]$, y el rango para el límite superior es $[v_{sup}^i, v_{sup}^i + (1/4 * d)]$.

Una representación gráfica del cromosoma sería:

$$C_1 = (E_1, \dots, E_N)$$

$$C_2 = (S_1, \dots, S_N)$$

$$C_3 = (a_1, \dots, a_N)$$

$$C_4 = (v_{min}^1, v_{max}^1, \dots, v_{min}^N, v_{max}^N)$$

$$C = C_1 C_2 C_3 C_4$$

B. Población inicial

La población inicial está compuesta de cuatro bloques con el mismo número de cromosomas en cada uno de ellos. El proceso de generación de la misma se describe a continuación:

- En el primer bloque, cada cromosoma tiene el mismo número de etiquetas para todas las variables del problema y se consideran funciones de pertenencia distribuidas uniformemente a lo largo del universo de discurso de la variable, es decir, $a = 1$ y $S = 0$.
- En el segundo bloque, dentro de cada cromosoma puede haber distintos valores, escogidos aleatoriamente, para el número de etiquetas de las variables. Las funciones de pertenencia se distribuyen uniformemente al igual que en el primer bloque.
- En el tercer bloque tiene cromosomas son número de etiquetas aleatorio en las variables y se varía de forma aleatoria el dominio de cada una de las variables, al igual que el parámetro a , cuyos valores van alternando las dos posibilidades de desplazamiento de las etiquetas hacia el centro o los extremos ($a < 1$ y $a > 1$). El parámetro S sigue siendo igual a cero.
- En el último bloque todos los componentes del cromosoma de escogen de forma aleatoria considerando sus correspondientes intervalos de variación.

C. Evaluación de los cromosomas.

El cálculo de la función de evaluación consta de tres pasos:

- Construir las particiones difusas de cada variable de entrada, utilizando la información contenida en el cromosoma. Esta tarea consta de varias fases:
 - el dominio de cada variable (C_4) es normalizado al intervalo $[-1, 1]$.
 - Se crea una partición difusa uniforme considerando su número de etiquetas (C_1).
 - La función de escala no lineal, con su “forma” (C_2) y su parámetro de sensibilidad (C_3) se aplica a los tres puntos que definen cada etiqueta.
 - La partición es trasladada de nuevo a su dominio original (C_4).
- Generar la BR considerando la BD obtenida en el paso previo. Se compone de dos procesos:

– El primer paso es seleccionar los antecedentes de las reglas. Se recorren todos los ejemplos del conjunto de entrenamiento y para cada uno de ellos se incluye la combinación de antecedentes que mejor cubre dicho ejemplo. cuanto mayor sea el número de etiquetas, mayor ser el número de combinaciones de antecedentes seleccionado. Cada una de estas combinaciones dará lugar a una regla.

– Para cada regla obtenida en el paso anterior, se obtienen los coeficientes que componen su consecuente mediante mínimos cuadrados.

• Calcular el error cuadrático medio (ECM) sobre el conjunto de datos de entrenamiento usando la BC obtenida (BD + BR). Hay que resaltar que el método de mínimos cuadrados obtiene la configuración óptima para minimizar el error sobre los datos de entrenamiento, lo que suele provocar un gran sobreaprendizaje (ECM sobre el conjunto de datos de prueba muy elevado). Para mejorar la capacidad de generalización del SBRD resultante, intentando evitar dicho sobreaprendizaje, penalizaremos los SBRDs con un número de reglas (NR) excesivo. Así, la función de evaluación es:

$$F = \omega_1 \cdot ECM + \omega_2 \cdot NR$$

siendo ω_1 y ω_2 porcentajes de peso de cada componente ($\omega_1 + \omega_2 = 1$). Para los experimentos, hemos considerado $\omega_2 = 0.75$.

D. Operadores genéticos

El mecanismo de selección empleado ha sido el Muestreo Universal Estocástico, propuesto por Baker en [2], incorporando una selección elitista. En lo referente a los restantes operadores genéticos, se ha considerado la especial estructura de los cromosomas (con cuatro niveles de información distintos, pero algunos fuertemente relacionados), para obtener una definición adecuada que haga el mejor uso posible de la representación escogida. Los operadores considerados son los siguientes:

D.1 Cruce

Utilizaremos dos operadores diferentes según las características de los dos cromosomas padre implicados en el cruce:

- *Cruce cuando los dos padres tienen igual número de etiquetas para cada variable:* Si los dos cromosomas tienen los mismos valores en C_1 (cada variable tiene el mismo número de etiquetas en ambos padres), podemos suponer que la búsqueda genética ha localizado una zona prometedora del espacio que conviene explotar. Una buena elección para esta tarea es el operador de cruce max-min-aritmético (MMA)[8] sobre los componentes del cromosoma con codificación real, es decir, los parámetros a (C_3) y los universos de discurso de las variables de entrada (C_4), manteniendo el valor de C_1 en ambos descendientes. Si el parámetro S_i es el mismo en ambos cromosomas, éste se mantiene en la descendencia, en caso de que sean diferentes, se prueban las dos combinaciones y se selecciona la mejor de ellas. Dicho cruce MMA funciona de la siguiente manera:

Si $C_v^t = (c_1, \dots, c_k, \dots, c_H)$ y $C_w^t = (c'_1, \dots, c'_k, \dots, c'_H)$ se van a cruzar, se generan los cuatro descendientes siguientes (con $d \in [0, 1]$):

$$\begin{aligned} C_1^{t+1} &= dC_w^t + (1-d)C_v^t \\ C_2^{t+1} &= dC_v^t + (1-d)C_w^t \\ C_3^{t+1} &\text{ con } c_{3k}^{t+1} = \min\{c_k, c'_k\} \\ C_4^{t+1} &\text{ con } c_{4k}^{t+1} = \max\{c_k, c'_k\} \end{aligned}$$

De esos cuatro, se escogen los dos mejores para sustituir a los padres. Si el parámetro S_i es distinto en ambos cromosomas padre, se generaran ocho descendientes, los cuatro anteriores con $S = 0$ y los mismos con $S = 1$.

- *Cruce cuando los padres tienen distinto número de etiquetas en alguna variable:* En este segundo caso, parece interesante utilizar la información contenida en ambos padres para explorar el espacio de búsqueda y tratar de localizar nuevas zonas prometedoras. El funcionamiento del operador de cruce es sencillo, se selecciona aleatoriamente un punto de corte p ($p \in \{2, \dots, N\}$, siendo N el número de variables), y se cruzan las cuatro partes C_1 , C_2 , C_3 y C_4 de los dos cromosomas de acuerdo al operador clásico de cruce.

Una representación gráfica del funcionamiento de este cruce se muestra a continuación (siendo R_i el dominio de la variable i): sean

$$\begin{aligned} C_i &= (E_1, \dots, E_p, E_{p+1}, \dots, E_N, S_1, \dots, S_p, S_{p+1}, \dots, S_N, \\ & a_1, \dots, a_p, a_{p+1}, \dots, a_N, R_1, \dots, R_p, R_{p+1}, \dots, R_N) \end{aligned}$$

$$\begin{aligned} C'_i &= (E'_1, \dots, E'_p, E'_{p+1}, \dots, E'_N, S'_1, \dots, S'_p, S'_{p+1}, \dots, S'_N, \\ & a'_1, \dots, a'_p, a'_{p+1}, \dots, a'_N, R'_1, \dots, R'_p, R'_{p+1}, \dots, R'_N) \end{aligned}$$

dos individuos que se van a cruzar en p , los dos descendientes resultantes son:

$$\begin{aligned} C_i &= (E_1, \dots, E_p, E'_{p+1}, \dots, E'_N, S_1, \dots, S_p, S'_{p+1}, \dots, S'_N, \\ & a_1, \dots, a_p, a'_{p+1}, \dots, a'_N, R_1, \dots, R_p, R'_{p+1}, \dots, R'_N) \end{aligned}$$

$$\begin{aligned} C'_i &= (E'_1, \dots, E'_p, E_{p+1}, \dots, E_N, S'_1, \dots, S'_p, S_{p+1}, \dots, S_N, \\ & a'_1, \dots, a'_p, a_{p+1}, \dots, a_N, R'_1, \dots, R'_p, R_{p+1}, \dots, R_N) \end{aligned}$$

D.2 Mutación

Debido a la distinta naturaleza de los valores almacenados en las cuatro partes de los cromosomas, se utilizarán tres operadores diferentes de mutación, uno para los valores enteros (número de etiquetas), otro para los valores reales (parámetro de “sensibilidad” y universos de discurso) y otro para los parámetros S :

- *Mutación en C_1 :* El operador de mutación seleccionado es el propuesto por Thrift en [12]. Si el gen a mutar es de C_1 , se aumenta o disminuye en una unidad su granularidad asociada (la decisión se toma aleatoriamente). Cuando el valor que se pretende cambiar es el mínimo (2) o el máximo (7), se realiza el único cambio posible.

- *Mutación en C_3 y C_4 :* Como ambas partes del cromosoma emplean una codificación real, utilizaremos el operador de mutación no uniforme de Michalewicz [10]. Este operador implementa una búsqueda uniforme en el espacio al principio de la ejecución del AG y una muy localizada en las últimas generaciones. A continuación, describimos su modo de trabajo:

Dado un cromosoma de la población $P(t)$, $C_v^t = (c_1, \dots, c_k, \dots, c_H)$, y uno de sus genes, c_k , $k \in 1, \dots, H$, definido en $[c_{ki}, c_{kd}]$, seleccionado para ser mutado, el cromosoma obtenido tras la mutación presenta la forma $C_v^{t+1} = (c_1, \dots, c'_k, \dots, c_H)$, con

$$c'_k = \begin{cases} c_k + \Delta(t, c_{kd} - c_k), & \text{si } p = 0 \\ c_k - \Delta(t, c_k - c_{ki}), & \text{si } p = 1 \end{cases}$$

donde p es un número aleatorio generado en $\{0, 1\}$ y la función $\Delta(t, y)$ devuelve un valor en el intervalo $[0, y]$, de modo que la probabilidad de que $\Delta(t, y)$ sea cercana a 0 aumenta cuando lo hace el contador de generaciones t :

$$\Delta(t, y) = y(1 - r^{(1 - \frac{1}{T})^b})$$

donde, a su vez, r es un número aleatorio generado en $[0, 1]$, T es el número de generaciones durante las que se ejecutará el AG y b es un parámetro escogido por el usuario que determina el grado de dependencia existente con respecto al número de generaciones. Esta propiedad da lugar a que el operador lleve a cabo una búsqueda uniforme en el espacio cuando t es pequeña, es decir, en las primeras iteraciones, y una mucho más localizada en generaciones posteriores.

- *Mutación en C_2 :* En este caso, se utiliza el operador de mutación binaria clásico ya que este parámetro solo admite dos valores.

IV. RESULTADOS EXPERIMENTALES

Para probar nuestro método hemos utilizado un problema real de modelado de la longitud de la línea de baja tensión en núcleos rurales asturianos [4], que consiste en emplear un sistema de predicción que sea capaz de determinar la relación existente entre la longitud de la línea de baja tensión tendida en una población (variable de salida) y otras características de ésta, tales como su radio o el número de usuarios que viven en ella, que serán nuestras dos variables de entrada. Para resolverlo, dispondremos de un conjunto de datos proporcionado por la

Hidroeléctrica del Cantábrico que contiene información sobre distintas características de 495 núcleos rurales situados en la provincia de Asturias.

Para los experimentos, se ha realizado una validación cruzada con 5 particiones distintas. Es decir, el conjunto de datos se divide en 5 subconjuntos de igual tamaño (aproximadamente) y el método es ejecutado 5 veces, dejando en cada una de ellas uno de esos subconjuntos como conjunto de datos de prueba y utilizando los otros cuatro como conjunto de datos de entrenamiento¹. Para cada una de las cinco particiones de datos, se ha ejecutado el AG cuatro veces con diferentes valores para la semilla de inicio de la secuencia de números aleatorios. Los parámetros del AG utilizados en los experimentos se muestran en la tabla I.

TABLA I
VALORES DE LOS PARÁMETROS GENÉTICOS

Parámetro	Valor
Tamaño de la población	81
Probabilidad de cruce	0.6
Probabilidad de mutación	0.1
Parámetro b (Mutación de Michalewicz)	0.5
Parámetro d (Cruce max-min-aritmético)	0.35
Número de generaciones	30

Los resultados obtenidos por el método propuesto (etiquetado como **AG+MC**) se han comparado con otros mecanismos:

- Uno de ellos utiliza un algoritmo simple de cubrimiento de ejemplos que trabaja con SBRDs de tipo Mamdani, el método de Wang y Mendel [13]. Posteriormente, se desechan los consecuentes y se utiliza una estrategia de evolución $E_{(\mu,\lambda)}$ para aprender los consecuentes de las reglas obtenidas. En este caso se han considerado, para todas las variables de entrada, particiones difusas uniformes con el mismo número de etiquetas en todas ellas (5 etiquetas, al ser el valor con el que se ha obtenido los mejores resultados). A continuación, se ajustan las particiones difusas mediante un AG como se describe en [3]. Este método está etiquetado en la tabla como **WM-TS-ajuste**.
- El método etiquetado como **LEL-TSK**, propuesto en [1], y que consta de dos etapas principales. En la primera de ellas se desarrolla un proceso local de identificación de prototipos utilizando una Estrategia de Evolución (1+1), mientras que la segunda es una etapa de post-procesamiento donde se utiliza un AG para seleccionar las reglas con mejor cooperación y otro AG para un ajuste de las funciones de pertenencia que componen las particiones difusas de las variables de entrada.
- el método **ANFIS**, propuesto por Jang en [7], considerando funciones de pertenencia triangulares en lugar de las gaussianas del método original. Este método representa el sistema difuso con una estructura neuronal y se realiza en dos etapas. En la primera de ellas se

¹Los conjuntos de datos utilizados se pueden obtener en <http://decsai.ugr.es/~casillas/FMLib/>

mantienen fijos los antecedentes y se ajustan los consecuentes por mínimos cuadrados, mientras que en el segundo paso se mantienen fijos los consecuentes y se ajustan los antecedentes mediante gradiente descendente

Los resultados se muestran en la tabla II. En cada columna aparecen los valores medios de las ejecuciones realizadas para cada una de las cinco particiones. Los cuatro valores que se presentan para cada método son:

- **#E**: Media del número de etiquetas de cada variable de entrada.
- **#R**: Media del número de reglas obtenidas en el modelo.
- **ECM_{entr}**: Error cuadrático medio sobre el conjunto de datos de entrenamiento.
- **ECM_{test}**: Error cuadrático medio sobre el conjunto de datos de prueba.

TABLA II
RESULTADOS OBTENIDOS

Method	#E	#R	ECM _{entr}	ECM _{test}
AG+MC	2.2	3	166308	205497
WM-TSK-ajuste	5	12.4	139915	261397
LEL-TSK	5	32	127206	206558
ANFIS	-	19.6	112942	120732480

Como se puede observar en la tabla, el método propuesto obtiene buenos resultados en capacidad de generalización ya que apenas presenta sobreaprendizaje aunque eso suponga algo menos de precisión en el ECM sobre el conjunto de entrenamiento. Además, el error sobre el conjunto de prueba mejora bastante al obtenido por ANFIS, que presenta mucho sobreaprendizaje al tratarse de un problema real complejo fuertemente no lineal. respecto a LEL-TSK, el error de prueba es sólo ligeramente mejor, pero nuestra propuesta obtiene dichos resultados con un modelo mucho más simple e interpretable (3 reglas de media frente a 32).

El principal inconveniente del mecanismo propuesto es la dependencia del parámetro ω_2 , que acaba determinando, de forma indirecta, la complejidad del modelo (número de etiquetas por partición y, por tanto, número máximo de reglas), de forma que se puede optar por modelos más precisos reduciendo el valor de dicho parámetro, y por tanto, la influencia del número de reglas en la función de evaluación. Por otro lado, la optimización por mínimos cuadrados consume mucho tiempo y puede suponer una desventaja, especialmente en problemas complejos (con muchas variables). En cualquier caso, el método ha obtenido buenas soluciones con un reducido número de generaciones.

V. CONCLUSIONES

En este trabajo, se ha propuesto un método híbrido para el aprendizaje de la BC de un SBRD TS. Dicho método ha obtenido unos buenos resultados, especialmente en capacidad de generalización y en simplicidad, obteniendo modelos interpretables (en la parte de los antecedentes, al tratarse de reglas TS) que obtienen parti-

ciones difusas “fuertes” para las variables de entrada que aparecen en dichos antecedentes.

En trabajos futuros intentaremos eliminar la influencia del parámetro ω_2 , que obliga a realizar varias pruebas hasta obtener el balance precisión-simplicidad deseado. Para ello utilizaremos un AG multiobjetivo que considere los dos objetivos que aparecen en la función de evaluación (error cuadrático y número de reglas) para obtener modelos con distinto balance en una única ejecución del método.

REFERENCIAS

- [1] R. Alcalá, J. Alcalá-Fdez, J. Casillas, O. Cordón, F. Herrera., Local identification of prototypes for genetic learning of accurate TSK fuzzy rule-based systems, *International Journal of Intelligent Systems* In press.
- [2] Baker, J.E., Reducing bias and inefficiency in the selection algorithm, *Proceedings of the Second International Conference on Genetic Algorithms (ICGA'87)*, Hillsdale, 1987, pp. 14-21.
- [3] Cordón, O., Herrera, F. A two stage evolutionary process for designing TSK fuzzy rule-based systems *IEEE Transactions on Systems, Man, and Cybernetics. Part B. Cybernetics*, volume 29:6, pages 703-715, 1999.
- [4] Cordón, O., Herrera, F., Sánchez, A., Solving electrical distribution problems using hybrid evolutionary data analysis techniques, *Applied Intelligence* 10, 1999, pp. 5-24.
- [5] O. Cordón, F. Herrera, L. Magdalena, P. Villar, A genetic learning process for the scaling factors, granularity and contexts of the fuzzy rule-based system data base, *Information Sciences* 136, 2001, pp. 85-107.
- [6] O. Cordón, F. Herrera, P. Villar, Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing, *International Journal of Approximate Reasoning* 25, 2000, pp. 187-215.
- [7] J.R. Jang. ANFIS: Adaptive network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, volume 23:3, pages 665-684, 1993.
- [8] F. Herrera, M. Lozano and J.L. Verdegay, “Fuzzy connectives based crossover operators to model genetic algorithms population diversity”, *Fuzzy Sets and Systems*, vol. 92, no. 1, pp. 21-30, 1997.
- [9] L. Magdalena, Adapting the gain of an FLC with genetic algorithms, *International Journal of Approximate Reasoning* 17(4), 1997, pp. 327-349.
- [10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (Springer-Verlag, 1996).
- [11] T. Takagi, M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, volume 15:1, pages 116-132, 1985.
- [12] P. Thrift, Fuzzy logic synthesis with genetic algorithms. *Proceedings of Fourth International Conference on Genetic Algorithms (ICGA'91)*, 1991, pp. 509-513.
- [13] Wang, L.X., Mendel, J.M., Generating fuzzy rules by learning from examples, *IEEE Transactions on Systems, Man, and Cybernetics* 22, 1992, pp. 1414-1427.