

# Generación Genética Iterativa de Reglas Fuzzy de Clasificación con Búsqueda Tabú

Edward Hinojosa Cárdenas

Universidad Nacional de San Agustín

Arequipa, Perú

Email: edward.hinojosa.cardenas@gmail.com

Yván J. Túpac Valdivia

Universidad Católica San Pablo

Arequipa, Perú

Email: ytupac@uscp.edu.pe

Heloisa A. Camargo

Universidade Federal de São Carlos

São Carlos, Brazil

Email: heloisa@dc.ufscar.br

**Resumen**—Este artículo estudia la utilización de búsqueda tabú para aumentar la cooperación del conjunto de reglas de un sistema de clasificación fuzzy que ha sido generado utilizando un modelo genético iterativo. El modelo genético iterativo, consiste en obtener una única regla en cada ejecución de un algoritmo genético, el cual generalmente intenta encontrar la regla más consistente y completa que será inserida en la base de reglas final, repitiendo ese proceso hasta que la inserción de la mejor regla no aumente la tasa de clasificación de la base de reglas final. En este trabajo proponemos y analizamos la influencia de utilizar búsqueda tabú, basada en reglas vecinas, para mejorar la tasa de clasificación de la base de reglas obtenida del modelo genético iterativo. Los resultados muestran una mejora significativa en la clasificación utilizando la metodología propuesta, manteniendo el número de reglas y condiciones, en diferentes bases de ejemplos.

**Palabras Clave**—Sistemas de clasificación fuzzy; base de reglas; búsqueda tabú; modelo genético iterativo; tasa de clasificación

## I. INTRODUCCIÓN

La lógica fuzzy está basada en la teoría de conjuntos fuzzy propuesta en 1965 por Zadeh[1]. Una de las principales aplicaciones de la lógica fuzzy son los Sistemas Fuzzy Basados en Reglas (SFBR) que pueden ser usados para resolver problemas reales en diferentes áreas.

Los SFBR tienen dos componente principales[2]: 1) el Mecanismos de Inferencia (MI) que ejecuta las operaciones de inferencia fuzzy para obtener un resultado dada una determinada entrada, y 2) la Base de Conocimiento (BC), este componente incluye dos sub componentes: 2a) La Base de Reglas (BR) que es el conjunto de reglas fuzzy, y 2c) La Base de Datos (BD) que es donde las funciones de pertenencia o conjuntos fuzzy son definidos para cada variable lingüística considerada en el SFBR.

Una de las más importantes y difíciles tareas en el desarrollo de SFBR es el aprendizaje o generación de la BC. Dicha tarea puede ser realizada utilizando un método manual, es decir, por expertos humanos en una determinada área[3], o por un método automático, es decir, que utilizan diferentes técnicas para extraer el conocimiento de una base de ejemplos de una determinada área[4][5]. También se puede utilizar ambos métodos para la generación de la BC[6].

La forma automática es de interés en este trabajo. Dentro de las técnicas consideradas más eficientes y altamente utilizadas

para realizar la tarea de generación de la BC son los Algoritmos Evolutivos, particularmente los Algoritmos Genéticos (AG), lo cual ha generado una línea de investigación llamada Sistemas Fuzzy Genéticos Basados en Reglas (SFGBR)[2][7].

Dentro de la taxonomía de los SFGBR, existen diferentes procesos que utilizan AG para aprender u optimizar diferentes componentes[7], por ejemplo, la optimización genética de los conjuntos fuzzy en la BD[4], la adaptación genética del mecanismo de inferencia[8], la generación genética de la BR[9], selección genética de reglas fuzzy[10], entre otras. Este trabajo se enfoca en la generación genética de la BR.

En la generación genética de la BR, cuatro modelos son generalmente utilizados[7]. 1) Pittsburgh, donde cada cromosoma representa una BR, después del proceso genético la BR final es el mejor cromosoma obtenido[11]. 2) Michigan, donde una regla es codificada en un cromosoma y la BR final es la unión de los mejores cromosomas de la última población[12]. 3) Iterativa, o Iterative Rule Learning (IRL) donde una regla es codificada en un cromosoma, pero la diferencia con el enfoque Michigan, es que una regla es generada e inserida en la BR final de forma iterativa en cada ejecución de un AG[13][14]. 4) Cooperativa-Competitiva, en donde la población entera o un subconjunto de ella codifica la BR y los cromosomas compiten y cooperan simultáneamente[15].

En este trabajo nosotros utilizamos el modelo IRL el cual generalmente se basa en la consistencia y completitud de cada regla[13] para generar la BR final aumentando la tasa de clasificación en cada ejecución de un AG mediante la inserción de una regla. Después proponemos una Búsqueda Tabú (BT)[16] para aumentar la tasa de clasificación de la BR obtenida por el modelo IRL, uno de los principales trabajos relacionados en este punto puede ser visto en[17]. Los experimentos realizados en ocho diferentes bases de ejemplos muestran una ganancia significativa de la tasa de clasificación.

Este trabajo está organizado de la siguiente manera: En la sección 2, los SFBR para clasificación, o Sistemas de Clasificación Fuzzy Basados en Reglas (SCFBR), son definidos y los componentes principales brevemente detallados. En la sección 3 los conceptos y características de la BT son detallados. En la sección 4 nosotros explicamos la metodología propuesta, basada en el modelo IRL y BT. Los resultados obtenidos en ocho bases de ejemplos utilizadas son mostrados en la sección 5. Las conclusiones de este trabajo son presentadas

en la sección 6.

## II. SISTEMAS DE CLASIFICACIÓN FUZZY BASADOS EN REGLAS

Clasificar es una importante tarea en diferentes áreas, por ejemplo medicina[18], ingeniería[19], manufactura[20], entre otras. Clasificación principalmente consiste en dado un conjunto de ejemplos  $E = e_1, e_2, \dots, e_p$ , se tiene como objetivo asignar una clase  $C_j$  desde un conjunto de clases  $C = C_1, C_2, \dots, C_m$  para cada ejemplo  $e_q$ . Cada  $e_q$  está definido por un número  $n$  de atributos  $e_q = a_{q1}, a_{q2}, \dots, a_{qn}$ .

Los SCFBR resuelven problemas de clasificación, los cuales generalmente utilizan la siguiente estructura en cada regla:

$R_i$ : SI  $J_1$  ES  $A_{1l_1}$  Y  $J_2$  ES  $A_{2l_2}$  Y ... Y  $J_n$  ES  $A_{nl_n}$   
ENTONCES Clase =  $C_k$

donde:

$R_i$ : Índice de la reglas fuzzy  $i$

$J_1, \dots, J_n$ : Atributos de los ejemplos

$A_{1l_1}, \dots, A_{nl_n}$ : Términos lingüísticos para cada atributo

$C_k$ : Clase de la regla  $R_i$

Los SCFBR utilizan un conjunto de reglas similares a la regla  $R_i$  para clasificar, definido como BR. Son utilizados también los conjuntos fuzzy de la BD y el MI para determinar la clase de un determinado ejemplo. Existen dos métodos comunes que utiliza el MI para determinar un resultado. El primer método es definido como "método de la regla vencedora", o método de raciocinio fuzzy clásico, en el cual la clase de la regla con mayor grado de compatibilidad con el ejemplo, es la clase del ejemplo. El segundo método es definido como el método de raciocinio fuzzy general, en el cual el mayor conjunto de reglas con la misma clase, compatibles con el ejemplo, determina la clase del ejemplo.

## III. BÚSQUEDA TABÚ

La BT surge en un intento de dotar de inteligencia a los algoritmos de búsqueda local. Fred Glover[16], su primer definidor, define a la BT como una guía en un procedimiento de búsqueda local para explorar el espacio de soluciones más allá del óptimo local. La BT toma de la Inteligencia Artificial (IA) el concepto de memoria y lo implementa mediante estructuras simples con el objetivo de dirigir la búsqueda teniendo en cuenta la historia de ésta, es decir, el procedimiento trata de extraer información de lo sucedido y actuar en consecuencia. En este sentido puede decirse que existe cierto aprendizaje y que la búsqueda es inteligente[21].

Según Fred Glover[16] la BT permite moverse a una solución aunque no sea tan buena como la actual, de modo que se pueda escapar de óptimos locales y continuar estratégicamente la búsqueda de soluciones aún mejores.

Para entender mejor el concepto de BT, a continuación se describen los componentes básicos de un algoritmo de BT presentado en[22] y una introducción a lo utilizado en este trabajo.

### A. Componentes básicos de un algoritmo de Búsqueda Tabú

La BT tiene como objetivo optimizar una función  $f: X \rightarrow A$ ; donde  $X$  representa el dominio o el conjunto de soluciones factibles ( $x$ ) determinado por las restricciones; y  $A$  representa el contradominio o el conjunto de valores que puede tomar la función  $f$ .

1) *Vecindad*: Se le denomina vecindad a la lista de candidatos que se tiene para realizar el siguiente movimiento. La lista de candidatos está conformada por soluciones adyacentes que pueden ser alcanzadas desde la solución actual. En este trabajo la vecindad esta dada por una BR que se diferencia de la BR principal en una regla vecina. Dos reglas se consideran vecinas si sólo difieren en un término lingüístico de una variable de entrada, y dichos términos son consecutivos.

2) *Restricciones Tabú*: Para no caer en soluciones ya generadas se restringe la búsqueda al clasificar ciertos movimientos como prohibidos o tabú. En este trabajo son las BR analizadas.

3) *Lista Tabú*: Esta lista está conformada por el inverso de cada movimientos realizado, posee un tamaño específico  $t$ . Cada uno de sus elementos esta penalizado de tal manera que no se permite realizar un movimiento que se encuentre en esta lista. En este trabajo es la lista de BR analizadas.

4) *Críticos de Aspiración*: En ocasiones un movimiento tabú nos permite llegar a una solución mejor que cualquiera visitada hasta ahora. En este trabajo no es utilizado este criterio.

5) *Solución Inicial*: Es el conjunto de valores que toman las variables de decisión a partir de los cuales se construye el algoritmo. La forma de generarlas depende del problema que se presente, en ocasiones se recomienda el uso de una heurística para generarla. En este trabajo se utiliza la BR generada por el modelo genético iterativo (IRL) para clasificación. Llamamos a dicha BR como BR principal.

6) *Movimientos*: Se realizan en diversas formas dependiendo de la estructura del problema a resolver. En este trabajo es modificar una regla, de la BR principal, por una regla vecina (modificada en una condición vecina).

7) *Criterio de Parada*: Condición que debe ser cumplida para que el método finalice la búsqueda. También este debe adaptarse el problema específico, los criterios más comunes que se establecen son:

- Un número específico de iteraciones para alcanzar por lo menos dos mejoras.
- Un número máximo de iteraciones.

En este trabajo se hace una búsqueda completa de todas las BR vecinas, en caso no se encuentre una BR que aumente la tasa de clasificación de la BR principal la búsqueda finaliza.

8) *Valor de Movimiento*: Valor asociado a cada intercambio, representa la diferencia entre el valor de la función objetivo antes y después de realizar el movimiento propuesto. Este trabajo no utiliza un valor de movimiento.

9) *Mejor solución*: Esta solución va cambiando de valor a lo largo de las iteraciones realizadas en un método y su función es guardar el valor de las variables de decisión que hasta el momento maximiza o minimiza la función objetivo. En este

trabajo la BR vecina que mejore la tasa de clasificación de la BR principal y de las demás BR vecinas, será considerada como BR principal en la siguiente iteración.

#### IV. GENERACIÓN GENÉTICA ITERATIVA DE REGLAS FUZZY DE CLASIFICACIÓN CON BÚSQUEDA TABÚ

La metodología propuesta sigue dos pasos: 1) La generación, basada en el modelo genético iterativo, de la BR para un SCFBR, y 2) Buscar un BR vecina que aumente la tasa de clasificación de la BR, obtenida en el paso anterior, basada en una BT. A continuación se explican detalladamente cada uno de los pasos mencionados anteriormente.

##### A. Generación Genética Iterativa de la Base de Reglas

El modelo genético iterativo para la generación de la BR en un SCFBR está basado en la descomposición de un problema en sub problemas más simples, llamado comúnmente "divide y vencerás". Ello permite una importante reducción en el espacio de búsqueda que implica a su vez la reducción del tiempo requerido en el aprendizaje[23]. Dentro de los algoritmos más conocidos que siguen dicho modelo son: SIA[24], MOGUL[25], SLAVE[13] o NSLV[23].

Este trabajo se basa principalmente en el método iterativo NSLV[23], que mejora el método iterativo SLAVE[13]. Entre las principales diferencias entre esos dos métodos, es que NSLV permite obtener una regla completa, es decir, condiciones y conclusión, en cada iteración de un AG; y también que el método NSLV no es necesario determinar el orden de las clases a aprender que puede ocasionar un sesgo en el aprendizaje.

Entre las principales ventajas del modelo genético iterativo tenemos las dos siguientes: reducción del espacio de búsqueda del AG; y reducción del tamaño del cromosoma que ayuda a mejorar el desempeño del AG[7].

La Fig. 1 detalla el modelo iterativo utilizado:

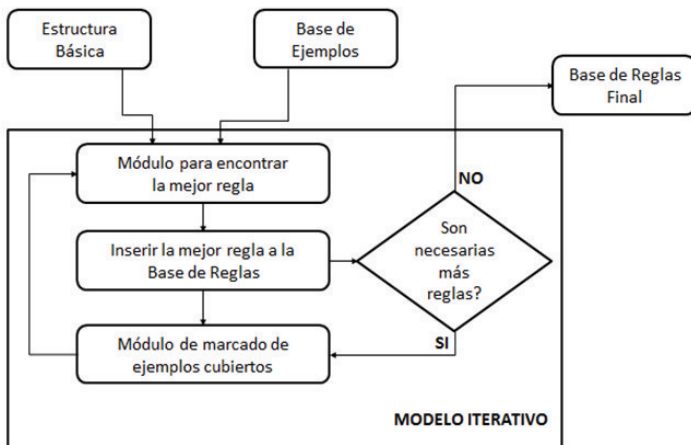


Fig. 1. Modelo Iterativo - Adaptado[23]

Primero es definida la estructura de las reglas, utilizamos la estructura definida en la sección II. El módulo para encontrar la mejor regla utiliza un AG y la base de ejemplos. Cada regla

debe ser codificada en un cromosoma, utilizamos la codificación decimal utilizada en [14]. Un ejemplo de codificación de reglas sin condiciones don't care y con condiciones don't care son presentados en las Fig. 2 y Fig. 3 respectivamente. En la Fig. 3 la segunda condición  $J_2$  es don't care, la cual es codificada con 0.

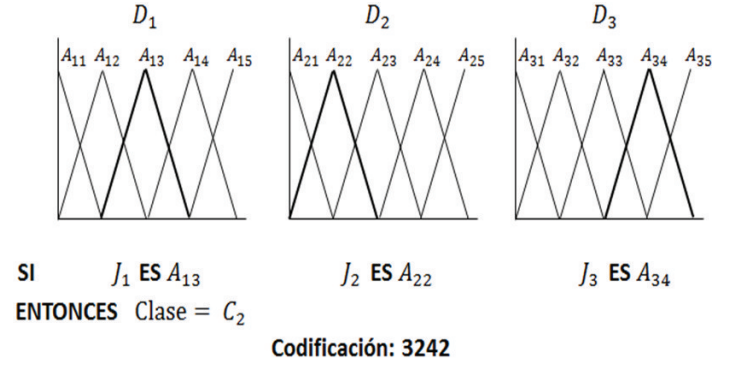


Fig. 2. Codificación de Reglas

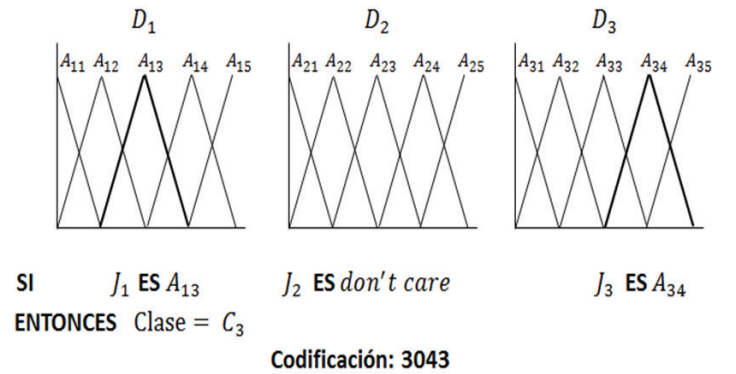


Fig. 3. Codificación de Reglas con condiciones don't care

La población inicial del AG es generada utilizando una cantidad  $T$  de ejemplos de entrenamiento, generando la regla que tenga mayor grado de compatibilidad con cada uno de los  $T$  ejemplos seleccionados aleatoriamente. Es utilizada una probabilidad  $z$  para considerar condiciones don't care, codificadas por el valor 0 como visto en la Fig. 3. Son generadas por lo menos dos reglas para cada clase del problema en la población inicial.

La función de aptitud para cada regla está basada en la suma de todos los grados de compatibilidad de los ejemplos que tienen la misma clase que la regla menos la suma de todos los grados de compatibilidad de los ejemplos que tienen diferente clase que la regla. La función de aptitud es definida en la siguiente ecuación:

$$f(R_q) = \sum_{e_p \in \text{Clase } C_k} \mu_{A_q}(e_p) - \sum_{e_p \notin \text{Clase } C_k} \mu_{A_q}(e_p) \quad (1)$$

Como operadores genéticos se utilizó selección por torneo, basado en el valor de la ecuación 1, cruzamiento uniforme y



Cromosoma 1	2	4	0	1
Cromosoma 2	1	0	3	2
Máscara	1	0	0	1
Descendiente 2	2	0	3	1
Descendiente 2	1	4	0	2

Fig. 4. Cruzamiento Uniforme

Cromosoma 1	2	4	0	1
Mutado 1	1	4	0	1

Fig. 5. Mutación Simple

mutación simple. Los dos últimos operadores genéticos son mostrados en la Fig. 4 y Fig. 5 respectivamente.

Se mantiene un elitismo de por lo menos dos reglas para cada clase del problema en cada población. Después de un número  $r$  de iteraciones la mejor regla es inserida en la BR final.

Cuando la mejor regla es inserida, el método puede seguir dos caminos: 1) Si la regla aumenta la tasa de clasificación de la BR final, entonces la regla se mantiene en la BR final y es ejecutado el módulo de marcación de ejemplos, el cual marcará lo ejemplos que tenga un grado de compatibilidad mayor a  $W$  con la mejor regla obtenida, los ejemplos marcados no afectaran en la suma de grados de compatibilidad de los ejemplos que tienen la misma clase que la regla (vea la ecuación 4.1) en las próximas iteraciones del AG. Esas dos características, que la mejor regla aumente la tasa de clasificación y la marcación de ejemplos son las formas de cooperación de las reglas en la BR final. 2) Si la regla no aumenta la tasa de clasificación de la BR final, esa regla es suprimida y el proceso de generación de la BR termina.

Una vez el proceso de generación termina, es utilizado el método propuesto basado en BT para aumentar la tasa de clasificación de la BR, aumentado así la cooperación de las reglas en base a la vecindad. Dicho método es detallado en la siguiente subsección.

### B. Búsqueda Tabú para aumentar la tasa de clasificación

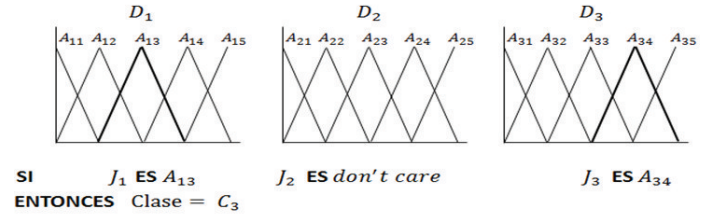
En este punto son utilizados los componentes descritos en la subsección A de la sección II. Nosotros utilizamos como primera BR principal, la BR final generada por el modelo genético iterativo explicado en la sección anterior.

Antes de definir lo que es una BR vecina, primero definiremos los que es una regla vecina. Nosotros consideramos una regla vecina a otra, si y sólo si, difieren en un término lingüístico de una única variable de entrada y dichos términos son consecutivos.

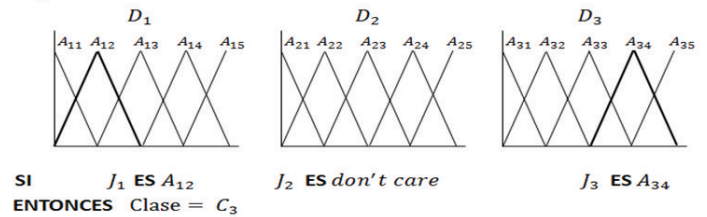
La Fig. 6 muestra una regla principal con todas sus reglas vecinas, considerando conjuntos triangulares y uniformemente distribuidos. Las condiciones don't care y la clase no varían en

reglas vecinas. La Regla Vecina 1 varía, de la Regla Principal, la primera condición  $J_1$  del conjunto  $A_{13}$  para su consecutivo  $A_{12}$ . La Regla Vecina 2 varía, de la Regla Principal, la primera condición  $J_1$  del conjunto  $A_{13}$  para su consecutivo  $A_{14}$ . La Regla Vecina 3 varía, de la Regla Principal, la tercera condición  $J_3$  del conjunto  $A_{34}$  para su consecutivo  $A_{33}$ . La Regla Vecina 4 varía, de la Regla Principal, la tercera condición  $J_3$  del conjunto  $A_{34}$  para su consecutivo  $A_{35}$ .

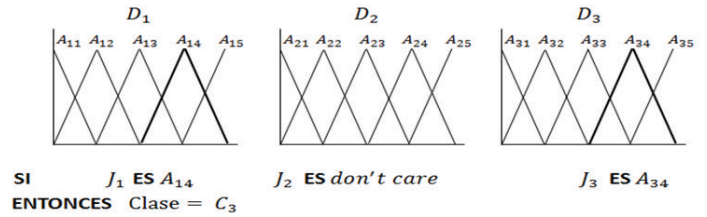
#### Regla Principal:



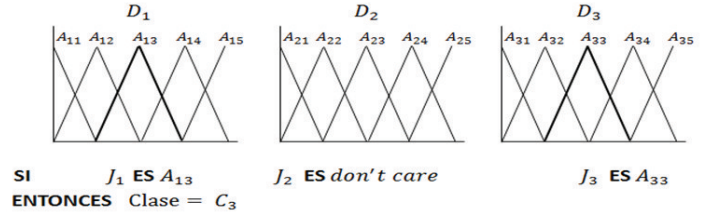
#### Regla Vecina 1:



#### Regla Vecina 2:



#### Regla Vecina 3:



#### Regla Vecina 4:

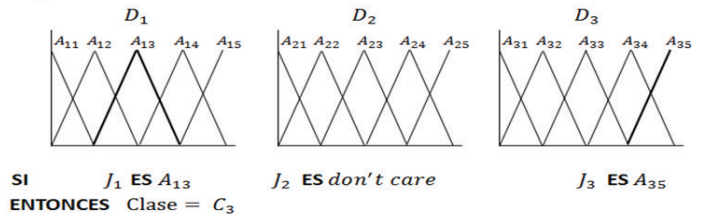


Fig. 6. Ejemplo de una Regla Principal y sus Reglas Vecinas

Una vez definido lo que es una regla vecina, definiremos lo que es una BR vecina. Nosotros consideramos dos BR vecinas si y solo si, varían en una regla y esas reglas son vecinas. La Fig. 7 muestra una BR principal con tres reglas y una BR vecina la cual que varía en la segunda regla. Podemos notar que ambas, la segunda regla de la BR principal y la segunda regla de la BR vecina, son vecinas. Dicha regla varía

la condición  $J_2$  del conjunto  $A_{22}$  para su consecutivo  $A_{23}$ . Las BR vecinas tienen la misma cantidad de condiciones y cantidad de reglas que la BR principal.

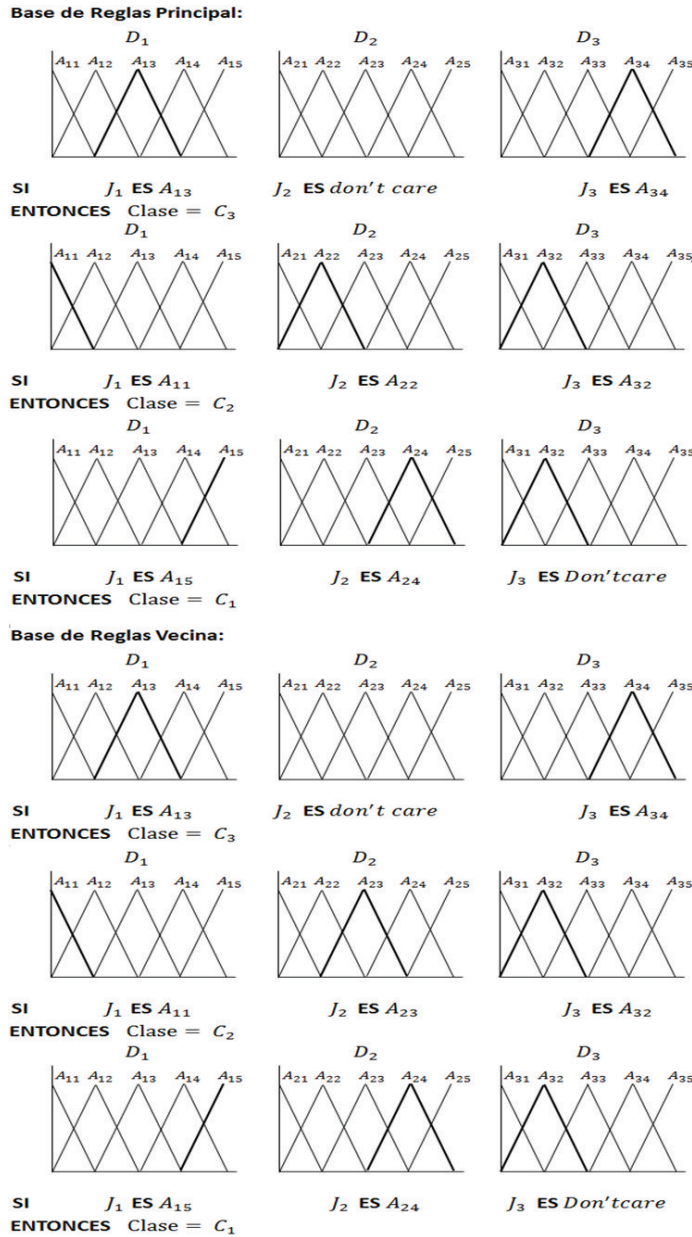


Fig. 7. Ejemplo de una BR Principal y una BR Vecina

Considerando la vecindad de las BR, la optimización de la BR parte con una BR principal proveniente del modelo genético iterativo, entonces se crean todas la BR vecinas y se evalúa la tasa de clasificación de cada una de ellas. La BR vecina que tenga la mayor tasa de clasificación y sea mayor a la tasa de clasificación de la BR principal, se convierte en la nueva BR principal y el proceso es repetido. La BR principal y todas las BR analizadas pasan a la lista tabú para que no sean nuevamente analizadas. El proceso de optimización termina cuando ninguna de las BR vecinas supera la tasa de clasificación de la BR principal.

La metodología propuesta es mostrada en la Fig. 8 en la

cual al modelo genético iterativo es adicionado un módulo para buscar la mejor BR vecina usando BT.

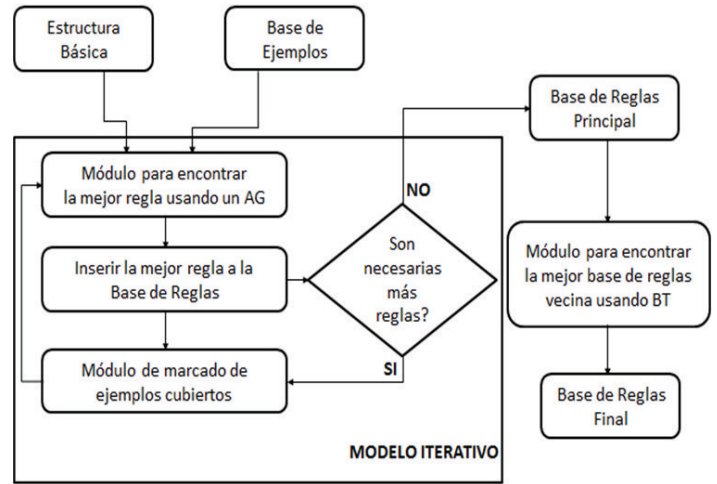


Fig. 8. Metodología propuesta para la generación genética iterativa de BR fuzzy con BT

En la siguiente sección se muestran los experimentos y resultados de la metodología propuesta en diferentes bases de ejemplos.

## V. EXPERIMENTOS

Se realizaron experimentos con ocho diferentes bases de ejemplos obtenidas del repositorio UCI de bases de ejemplos para aprendizaje[26], las cuales son detalladas en la TABLA I. La primera columna representa la base de ejemplos; la segunda, tercera y cuarta columna representan la cantidad de ejemplos (# Ejemplos), la cantidad de atributos (# Atributos) y la cantidad de clases (# Clases) de cada base de ejemplos respectivamente.

TABLA I  
BASES DE EJEMPLOS PARA LOS EXPERIMENTOS

Base de Ejemplos	# Ejemplos	# Atributos	# Clases
wine	178	13	3
bupa	345	6	2
pima	768	8	2
glass	214	9	6
new-thyroid	215	5	3
haberman	306	3	2
cleveland	297	13	5
wisconsin	683	9	2

Los parámetros definidos en la sección anterior para el modelo genético iterativo son presentados en la TABLA II. Los parámetros para la BT fueron brevemente mencionados en la sección III.

Los experimentos realizados con la metodología propuesta basada en el modelo genético iterativo de la BR y la búsqueda de la mejor BR vecina usando BT son presentados en la TABLA III. En la primera columna se muestra cada una de las Bases de Ejemplos (BE). En la segunda y tercera columna se muestra la tasa de error del Modelo Genético Iterativo (IRL) para ejemplos de entrenamiento y ejemplos de test

TABLA III  
RESULTADOS OBTENIDOS

BE	IRL-Entre.	IRL-Test	BT-Entre.	BT-Test	% BT-Entre.	% BT-Test
wine	0.0705 (0.0234)	0.0281 (0.0281)	0.0136 (0.0102)	0.0699 (0.0238)	50.00	16.00
bupa	0.3823 (0.0313)	0.3709 (0.0655)	0.0138 (0.0102)	0.0717 (0.0501)	54.00	48.14
pima	0.2455 (0.0132)	0.2368 (0.0370)	0.0093 (0.0105)	0.0299 (0.0206)	73.00	36.99
glass	0.3073 (0.0356)	0.3346 (0.0811)	0.0281 (0.0196)	0.0753 (0.0281)	72.00	47.22
new-thyroid	0.1927 (0.0318)	0.1900 (0.0654)	0.0662 (0.0189)	0.1167 (0.0484)	73.00	63.01
haberman	0.2320 (0.0097)	0.2522 (0.0536)	0.0068 (0.0038)	0.0323 (0.0000)	58.00	05.17
cleveland	0.4912 (0.0607)	0.5178 (0.0668)	0.0084 (0.0077)	0.0331 (0.0001)	25.00	20.00
wisconsin	0.0381 (0.0077)	0.0291 (0.0193)	0.0062 (0.0051)	0.0262 (0.0156)	61.00	31.15

TABLA II  
PARÁMETROS PARA EL MODELO GENÉTICO ITERATIVO

Parametros	Valor
Tamaño de la Población ( $T$ )	100
Cantidad de Iteraciones ( $r$ )	1000
Probabilidad de don't care ( $z$ )	0.1
Probabilidad de Cruzamiento ( $Cr$ )	0.7
Probabilidad de Mutación ( $Mu$ )	0.1
Grado de Compatibilidad para Marcación ( $w$ )	0.4

respectivamente. En la cuarta y quinta columna se muestra la mejora de la tasa de error después de utilizar BT para encontrar la mejor BR vecina (BT) en ejemplos de entrenamiento y ejemplos de test respectivamente. En las columnas sexta y séptima se muestra el porcentaje de BR que aumentaron su tasa de clasificación utilizando BT, no todas las BR generadas por el IRL tuvieron una BR vecina que aumentara su tasa de clasificación, tanto para ejemplos de entrenamiento y ejemplos de teste respectivamente. Se ejecutaron diez validaciones cruzadas con diez divisiones, usando una BD uniformemente distribuidas con cinco conjuntos triangulares. La desviación estándar es mostrada en paréntesis.

Los resultados muestran que se pueda obtener una ganancia en la tasa de clasificación de una BR generada por un modelo genético iterativo usando BT basada en la vecindad de las BR. Dicha ganancia se consigue sin aumentar el número de condiciones ni el número de reglas. Las conclusiones de este trabajo son definidas en la siguiente sección.

## VI. CONCLUSIONES

Nosotros proponemos el uso de búsqueda tabú para aumentar la tasa de clasificación y cooperación de la reglas en una BR generada por un modelo genético iterativo, la cual es la principal contribución de este trabajo.

La metodología propuesta está basada en dos pasos, el primer paso es la generación genética iterativa de la BR y el segundo paso es la búsqueda de la mejor BR vecina usando búsqueda tabú.

Los experimentos han demostrado que se puede mejorar la tasa de clasificación usando la metodología propuesta manteniendo el número de condiciones y el número de reglas.

## REFERENCIAS

[1] L. Zadeh, "Fuzzy sets," in *Information and Control*, 1965, pp. 338–353.  
[2] F. Herrera, "Genetic fuzzy systems: Status, critical considerations and future directions," *International Journal of Computational Intelligence Research*, vol. 1, no. 1, pp. 59–67, 2005.

[3] Y. Gao and W. Zhang, "A genetic-based method for training fuzzy systems," *IEEE International Fuzzy Systems Conference*, p. 123, 2001.  
[4] S. B. M. M. T. E. R. Hosseini, S.D. Qanadli and J. Dehmeshki, "An automatic approach for learning and tuning gaussian interval type-2 fuzzy membership functions applied to lung cad classification system," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 224–234, 2012.  
[5] J. Luengo and F. Herrera, "An extraction method for the characterization of the fuzzy rule based classification systems' behavior using data complexity measures: A case of study with fh-gbml," *IEEE International Conference on Fuzzy Systems (FUZZ)*, no. 2, pp. 1–8, 2010.  
[6] M. P. E. B. H. Bustince, G. Artola and H. Tizhoosh, "Sistema neurodifuso intervalo-valorado aplicado a la segmentación de imágenes de ultrasonidos," *XIV Congreso Español Sobre Tecnologías y Lógica Fuzzy*, pp. 307–312, 2008.  
[7] F. Herrera, "Genetic fuzzy systems: taxonomy, current research trends and prospects," *Evolutionary Intelligence*, vol. 1, no. 1, pp. 27–46, 2008.  
[8] Z. Z. Honghui and L. Yongqiang, "Application of an adaptive network-based fuzzy inference system using genetic algorithm for short term load forecasting," *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, vol. 2, pp. 314–317, 2012.  
[9] A. G. E. Leyva and R. Perez, "A genetic learning of fuzzy relational rules," *IEEE International Conference on Fuzzy Systems*, pp. 1–8, 2010.  
[10] R. A. J. Alcalá-Fdez and F. Herrera, "A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 857–872, 2011.  
[11] S. Smith, "A learning system based on genetic adaptive algorithms," in *PhD thesis - University of Pittsburgh*, 1980.  
[12] J. Holland and J. Reitman, "Cognitive systems based on adaptive algorithms," in *ACM SIGART Bulletin*, 1978.  
[13] A. Gonzalez. and R. Perez, "Slave: A genetic learning system based on the iterative approach," *IEEE Transaction on Fuzzy Systems*, vol. 7, no. 2, pp. 176–191, 1999.  
[14] M. Abadeh and J. Habibi, "Computer intrusion detection using an iterative fuzzy rule learning approach," *IEEE International Fuzzy Systems Conference*, pp. 1–6, 2007.  
[15] D. Greene and S. Smith, "Competition-based induction of decision models from examples," in *Machine Learning*, 1993.  
[16] F. F. Glover and M. Laguna, "Tabu search," in *Kluwer Academic Publishers*, 1997.  
[17] A. Bagis, "Fuzzy rule base design using tabu search algorithm for nonlinear system modeling," *ISA Transactions*, vol. 47, no. 1, pp. 32–44, 2008.  
[18] A. Z. Shabgahi and M. S. Abadeh, "Cancer tumor detection by gene expression data exploration using a genetic fuzzy system," *Developments in E-systems Engineering (DeSE)*, pp. 141–145, 2011.  
[19] Q. Zhang and M. Mahfouf, "Fuzzy modelling using a new compact fuzzy system: A special application to the prediction of the mechanical properties of alloy steels," *IEEE International Conference on Fuzzy Systems*, pp. 1041–1048, 2011.  
[20] A. H. F. M. A. I. Mahdavi and M. Bagherpour, "Applying fuzzy rule based to flexible routing problem in a flexible manufacturing system," *IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 2358–2364, 2009.  
[21] L. Moreno and F. Díaz, "Análisis comparativo entre dos algoritmos heurísticos para resolver el problema de planeación de tareas con restricción de recursos," in *Facultad de Minas - Universidad Nacional de Colombia-Medellín*, 2006.

- [22] A. Riojas, “Búsqueda tabú : conceptos, algoritmo y aplicación al problema de las n-reinas,” in *Universidad Nacional Mayor de San Marcos - Facultad de Ciencias Matemáticas. EAP. de Investigación Operativa*, 2005.
- [23] A. Gonzalez and R. Perez, “Una mejora del modelo genético iterativo de slave,” *XIV Congreso Español sobre Tecnologías y Lógica Fuzzy ? ESTYLF*, pp. 155–161, 2008.
- [24] G. Venturini, “Sia: a supervised inductive algorithm with genetic search for learning attributes based concepts,” *Machine Learning: ECML-93*, pp. 280–296, 1993.
- [25] M. d. J. O. Cordón and F. Herrera, “Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods,” *Int. Journal of Intelligent System*, vol. 13, pp. 1025–1053, 1998.
- [26] A. Frank and A. Asuncion, “Uci machine learning repository,” *Irvine, CA: University of California, School of Information and Computer Science*, 1998.