

Modelado de Sistemas Difusos Basado en el Algoritmo Genético Compacto

Miguel Melgarejo¹, Alvaro Prieto¹, Carlos Ruiz¹

¹ Laboratorio de Automática, Microelectrónica e Inteligencia Computacional,
Universidad Distrital Francisco José de Caldas
Bogotá D.C., Colombia
{mmelgarejo, aaprietom, camruizc} @udistrital.edu.co

Resumen. Este artículo presenta dos casos de estudio para el modelado de sistemas difusos basado en el Algoritmo Genético Compacto (CGA). Se estudian los casos de clasificación de la flor Iris de Fisher y la estimación de la función Sinc. Se realiza una comparación del enfoque basado en el CGA contra el enfoque basado en el Algoritmo Genético clásico (GA). Los resultados obtenidos de 100 experimentos para variaciones de los enfoques de CGA y GA, apuntan a que para los casos de estudio, el desempeño del enfoque basado en CGA es mejor dado que el error disminuye, y es más rápido con respecto al número de iteraciones necesarias para converger.

Palabras clave. Sistemas Difusos, Modelado de Sistemas Difusos, Algoritmos Evolutivos, Caminata aleatoria, Clasificador, Estimador.

1. Introducción

Dentro del Laboratorio de Automática, Micro-electrónica e Inteligencia Computacional (LAMIC) de la Universidad Distrital Francisco José de Caldas, se ha venido desarrollando una línea de trabajo en el área de inteligencia computacional. De esta experiencia pedagógica en el pregrado de Ingeniería Electrónica, se han reportado algunos trabajos orientados a la solución de problemas relacionados con sistemas de inferencia difusa (FIS, por sus siglas en inglés) [1], [2]. De estos trabajos, vale la pena destacar uno relacionado con este artículo, y es el realizado por Hostos et al [1]. En este trabajo se reporta el uso de una caja de herramientas para el modelado coevolutivo de FIS, en el problema de la predicción de una serie de tiempo.

El modelado de FIS con el uso de algoritmos evolutivos ha sido utilizado en la solución de problemas en control [2], [3], [4] y reconocimiento de patrones y clasificación [5], [6], [7], por nombrar algunos ejemplos. Con el fin de explorar alrededor del modelado de FIS con una perspectiva evolutiva diferente, se presenta en este artículo una propuesta metodológica basada en el Algoritmo Genético Compacto (CGA). El CGA pertenece a

una clase reciente de algoritmos genéticos conocidos como algoritmos genéticos construidos por medio de modelos probabilísticos (PMBGA, por sus siglas en inglés) [8].

Debido a que la mutación y la recombinación en el algoritmo genético clásico (GA) tienen una naturaleza aleatoria, su compartimiento es nocivo en el proceso evolutivo [8], [9]. Esto se debe a que estos operadores no discriminan las características que tienen una alta contribución al buen desempeño de un individuo, de aquellas que no aportan o empeoran el desempeño del individuo. Las características que aportan al buen desempeño son conocidas como bloques constructores. Es por esta razón que nacen los PMBGA, cuyo objetivo es conservar los bloques constructores de un individuo durante el proceso evolutivo. En la Figura 1 se presenta la propuesta metodológica para el modelado de FIS basada en el CGA.

Hasta la fecha, en la literatura no se reportan documentos en los que se realice el modelado evolutivo de un FIS con un CGA. Dicho algoritmo se ha implementado como algoritmo de optimización en [10], [11], [12]. La validación de dicha propuesta se realizó sobre dos casos de estudio: la clasificación de la flor Iris a partir de los datos de Fisher [13], [14] y la estimación de la función Sinc. Los resultados obtenidos muestran un impacto positivo en el proceso evolutivo cuando se utiliza el CGA como algoritmo de optimización, comparado contra el algoritmo genético clásico GA.

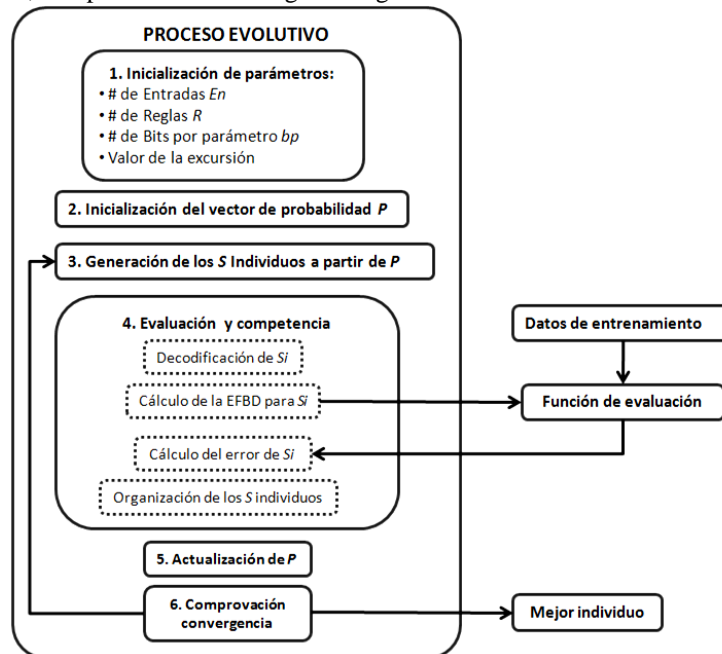


Fig 1. Propuesta para el modelado de un FIS tipo-1 con CGA

El artículo se organiza de la siguiente manera: la sección II introduce algunos principios metodológicos de la propuesta. La sección III presenta la implementación de la metodología sobre el primer caso de estudio: Clasificador de la flor de Iris. La sección IV presenta el segundo caso: El estimador de la función Sinc. Para ambos casos se muestran y discuten los resultados obtenidos de los experimentos. Finalmente, se concluye en la sección V.

2. Consideraciones metodológicas

2.1 FIS

En el bloque 4 de la Figura 1 se observa el cálculo de una Expansión de Funciones de Base Difusa (EFBD). Una EFBD es una forma de representar un FIS, en la cual un vector de entrada \mathbf{x} se relaciona con una salida puntual y de tal forma que $y = f(\mathbf{x})$ [15]. De esta manera se representa de manera compacta el proceso de inferencia en un FIS y la función resultante es un estimador universal [15], [16]. En las ecuaciones (1), (2) y (3) se muestran la representación matemática de una EFBD.

$$y(\mathbf{x}) = f(\mathbf{x}) = \sum_{l=1}^M y_l \phi_l(\mathbf{x}) \quad (1)$$

$$\phi_l(\mathbf{x}) = \frac{\prod_{i=1}^N \mu_{A_i^l}(x_i)}{\sum_{l=1}^M \prod_{i=1}^N \mu_{A_i^l}(x_i)} \quad (2)$$

$$\mu_{A_i^l}(x_i) = e^{-\frac{(x_i - c_i^l)^2}{2(\sigma_i^l)^2}} \quad (3)$$

El FIS representado por (1), (2) y (3) tiene las siguientes características: *Singleton Fuzzification*, T-norma producto, motor de inferencia producto, *defuzzification* por promedio de centros y una función de pertenencia Gaussiana. El índice l hace referencia a la l -ésima regla, siendo M el total de reglas. Por su parte el índice i hace referencia a la i -ésima entrada y N es el total de las mismas. La función de pertenencia $\mu_{A_i^l}(x_i)$ sería, entonces, única para cada entrada en cada regla. De igual manera el centro del conjunto consecuente y_l sería único en cada regla. De esta manera se garantiza una base de reglas completa y cada regla tendría N términos lingüísticos asociados.

Un FIS que se presenta a través de una EFBD se puede parametrizar y de esta manera queda totalmente definido a partir de una colección de parámetros. El número de entradas N y el número de reglas M definirían la cantidad de conjuntos antecedentes y centros de conjuntos consecuentes. Por su parte, la media c y la desviación estándar σ son los parámetros que definen la función de pertenencia Gaussiana. Teniendo en cuenta que por entrada y en cada regla hay dos parámetros del conjunto antecedente y un centro en el consecuente, entonces la cantidad de parámetros que definen el FIS queda definida en (4).

$$Op = (2 * M * N) + M \quad (4)$$

2.2 CGA

Por medio del modelo matemático conocido como paseo aleatorio [9], se puede analizar el crecimiento o caída de un gen (característica) de un individuo en un proceso evolutivo. En el transcurrir de las generaciones, este gen, lucha con sus competidores y prevalecerá dependiendo de las decisiones que el algoritmo tome, entendiendo como decisión el proceso de selección de los individuos que conforman la siguiente generación. La variable que determina el crecimiento o la caída de un gen es la población n , que representa la cantidad de individuos generados en un proceso evolutivo [9].

En un proceso evolutivo el genoma está conformado por aquellos parámetros o características que definen una solución para el problema dado. La cantidad de dichos parámetros está necesariamente ligada a la naturaleza del problema. En el proceso de selección en un algoritmo genético clásico (GA, por sus siglas en inglés) se le da relevancia al individuo con mejor desempeño sin distinguir aquellas características que aportan más a la solución. Por su parte, el CGA además de dar prevalencia en la selección al individuo con mejor desempeño, también da prevalencia a aquellos genes que son diferentes a los de su competencia y que por tanto son los que aportan, para bien o para mal, en el desempeño. En la Figura 2 se presenta el pseudocódigo del CGA.

En primera instancia, el vector de probabilidad es una representación estadística de toda la población en la cual cada posición representa la probabilidad de que dicha posición en cada uno de los S individuos sea '1'. El tamaño de este vector dependerá de la cantidad de características (genes) que representa una única solución al problema. Cuando se inicializa el algoritmo este vector tiene un valor de 0.5 en todas las posiciones con el objetivo de iniciar la búsqueda en todo el universo de soluciones disponibles.

Dependiendo del problema que se busca solucionar se escoge la función de aptitud con la que se evaluarán los individuos. A partir del valor que se obtiene de la función de aptitud para cada individuo se selecciona el que tiene el mejor desempeño, en otras palabras, aquel individuo que representa la mejor solución. A continuación se realiza la actualización del vector de probabilidad. Dicha actualización se realiza comparando bit a bit (gen a gen) cada uno del resto de individuos perdedores, $S-1$, con respecto al mejor. La actualización se realiza sobre cada posición del vector de probabilidad, y sólo se realiza si en dicha posición el valor del individuo perdedor difiere con respecto al valor del individuo ganador en la misma posición. El valor que se suma o resta para realizar la actualización es el inverso del valor n .

El criterio de convergencia del CGA está ligado al vector de probabilidad. Si el vector de probabilidad en todas las posiciones tiene valores de '0' o '1' significa que la probabilidad de que dichas posiciones tengan un valor de '1' es del 100% o 0%. Por lo tanto todos los individuos generados a partir del vector de probabilidad van a ser iguales al vector de probabilidad y este, al final del proceso evolutivo, representará la mejor solución al problema.

```

1.  Generar S individuos a partir del vector de probabilidad
    For i: = 1 to s do
      S[i] := generate(p);
2.  Evaluar y ordenar los individuos de tal manera que el mejor quede en
    la posición S [1].
3.  Competencia del individuo en S[1] con los demás individuos.
    For j: = 2 to s do
      Begin
        Winner, loser: = compete (S[1], S[j]);
        For i: 1 to L do
          If winner[i] ≠ loser[i] then
            If winner[i] = 1 then P[i]:=P[i] + 1/n
              Else p[i]:=P[i] – 1/n;
4.  Revisar si el vector de probabilidad converge
    For i: =1to L do
      If P[i]>0 and P[i] <1 then
        Return to step 2;
5.  El vector de probabilidad P representa la solución final

Parámetros del algoritmo genético compacto:
L=tamaño del individuo
n=tamaño de la población

```

Fig 2. Pseudocódigo del CGA. Tomado de [9]

Es importante en este punto recalcar que si los parámetros que conforman el genoma del proceso evolutivo no están representados de manera binaria, es necesario diseñar una codificación acorde con el problema que transforme los parámetros desde su representación original a una representación binaria.

De esta manera, se hace evidente la similitud entre la propuesta presentada en la Figura 1 y el pseudocódigo de la Figura 2. Un individuo generado dentro del CGA corresponde a una descripción binaria de los parámetros que conforman un FIS completo. Para este caso particular el genoma está conformado por las medias μ y desviaciones estándar σ de los conjuntos antecedentes, y por los centros μ y de los conjuntos consecuentes.

3 Primer caso de estudio

3.1 Problema

El problema de clasificación de la flor Iris es un problema tipo juguete bien conocido en el reconocimiento de patrones [13] [14], el cual consiste en la clasificación de tres tipos de flores Iris a partir de mediciones hechas del sépalo y pétalo de la flor. La base de datos

está compuesta por 50 instancias para cada flor, y cada instancia contiene la longitud y ancho tanto del sépalo como del pétalo. De esta manera se tienen 4 entradas al FIS.

Se propone un enfoque para el clasificador tipo controlador, como el presentado en [13], que tiene como objetivo que el FIS aproxime una función escalera con una cantidad de peldaños equivalente al número de clases existentes. Así pues, se establecen unos umbrales que al final determinan a qué clase pertenece cada instancia. Entonces, tomando como punto de referencia el trabajo desarrollado en este problema en [13], se propone que el primer escalón de la función escalera corresponda a la clase *Setosa*, el segundo a la clase *Vericolor* y el tercer escalón sea *Virginica*.

Una de las partes fundamentales de un proceso evolutivo y que es inherente al problema es la función de evaluación. La función de evaluación propuesta está compuesta por dos partes: Una función que mide el error medio cuadrático *Frmse* y otra que mide el porcentaje de instancias mal clasificadas, *Fc*. Esta función de evaluación busca que el FIS se aproxime lo mejor posible a una función escalera y además que el error de clasificación del sistema sea mínimo. En la ecuación (5) se muestra la definición de la función de evaluación *F*. *Frmse* se define en la ecuación (6)

$$F = \begin{cases} Frmse * Fc, & \text{si } Fc \neq 0 \\ Frmse, & \text{si } Fc = 0 \end{cases} \quad (5)$$

$$Frmse = \sqrt{\frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2} \quad (6)$$

El subíndice *i* de la ecuación (6) hace referencia al *i*-ésimo valor que se evaluará, y el valor real que se busca y *f* el valor que se obtiene del FIS. De esta manera, la salida del FIS que se encuentre en el intervalo (0,1) pertenece a la clase 1 *Setosa*, si está en el intervalo (1,2) pertenece a la clase 2 *Vericolor* y en el intervalo (2,3) a la clase 3 *Virginica*.

3.2 Experimentos

En las siguientes tablas se compilan los valores de los parámetros que limitan los experimentos realizados. En la Tabla 1 se presenta el número de reglas y el número de entradas para los experimentos. En primera instancia el valor del número de entradas se define por el número de características definidas en la base de datos. El número de reglas se definió teniendo en cuenta el trabajo realizado en [13].

Tabla 1 Parámetros del FIS

Parámetro	Valor
Número de reglas	3, 6
Número de entradas	4

Tabla 2. Parámetros del CGA

Parámetro	Valor
Numero de experimentos	100
Tamaño de población (N)	100,500,1000
Número de individuos generados (S)	30
Método de codificación binaria	Signo-magnitud
Cantidad de bits	8 bits
Criterio de parada	Convergencia de vector de probabilidad y repetición de error
Máximo número de generaciones	10000
Función de desempeño	Función de evaluación compuesta

Tabla 3. Parámetros del GA

Parámetro	Valor
Numero de experimentos	100
Numero de generaciones	3000
Tamaño de población (N)	30
Número de individuos intermedios	60
Probabilidad de cruce	0.7
Probabilidad de mutación	0.03
Criterio de parada	Numero de generaciones
Función de desempeño	Error medio cuadrático

3.3 Resultados

Con el fin de medir porcentualmente la relación existente entre proceso evolutivo con el CGA que obtiene los mejores resultados versus el proceso evolutivo con CGA que obtiene los segundos mejores resultados y el mejor proceso evolutivo con el GA, se define (7):

$$\frac{\text{Resultado } X}{\text{Resultado del mejor CGA}} * 100\% \quad (7)$$

Donde X representa el resultado obtenido por el segundo mejor proceso evolutivo con CGA, o el mejor resultado obtenido por el proceso evolutivo con GA.

En la tabla 4 presentada a continuación se presentan los resultados obtenidos de los experimentos realizados. Esta compila los valores de la media del error de clasificación Fc, la desviación del error Fc, media y desviación estándar del número de iteraciones para converger. Y finalmente, unas medidas de efectividad. Dichas medidas corresponden: al mejor y peor resultado de error de clasificación Fc obtenido de los 100 experimentos realizados, el porcentaje de experimentos en el cual el mejor individuo posee un error de clasificación que se encuentra por debajo de la media del error.

Tabla 4. Resultados del CGA para el clasificador de la flor Iris

T1-FLS para el clasificador de la flor Iris sintonizado con CGA						
	3 reglas			6 reglas		
	n=100	n=500	n=1000	n=100	n=500	n=1000
Media Error Fc	6.3333333	1.6666666	2.0444444	3.1777777	2.288888	2.4222222
Desviación Estándar Error Fc	9.9900187	1.5593988	1.74949443	5.0363933	1.6549909	1.5822623
Media Iteraciones	61.68	917.51	1936.99	72.3	1123.82	2401.34
Desviación Estándar Iteraciones	12.511715	135.34575	139.90166	12.86330	74.027483	180.40242
Mejor resultado	0	0	0	0	0	0
Peor resultado	42.22	6.66	13.33	35.55	6.66	8.88
% bajo media	73	39	24	74	77	84

Tabla 5. Resultado del GA para la clasificación de la flor Iris

T1-FLS para el clasificador de la flor Iris sintonizado con GA		
	3 reglas	6 reglas
Media del error Fc	5.977777778	4.71111111
Desviación Estándar del error Fc	8.270854448	7.22523463
Mejor resultado	0	0
Peor resultado	35.55555556	40
% bajo media	70	75

Los experimentos apuntan a que el mejor proceso evolutivo es el caso en el que se tiene una población n de 500 y tres reglas (CGA-1). Con respecto al segundo mejor, n de 1000 y tres reglas (CGA-2), la media del error para este proceso es 1.2266 veces (122.66%) la media del error del proceso evolutivo con CGA-1. De igual manera la desviación estándar del CGA-2 es 1.1219 veces (112.19%) la desviación de los resultados obtenidos por CGA-1. Esto implica, que de manera general, los resultados obtenidos por el CGA-2 están más dispersos que los del CGA-1. Además, la media del número de iteraciones necesarias para que el CGA-2 converja es 2.11 veces (211.11%) la media del mejor proceso. Por lo tanto, el mejor proceso es más rápido con respecto al número de iteraciones necesarias para converger.

De acuerdo con los resultados obtenidos para el GA, en la tabla 5, el mejor proceso evolutivo es el caso de un FIS con 6 reglas (GA). De esta manera, si se compara con

respecto al CGA-1, la media del error de clasificación del GA es 2.8266 veces (282.66%) la del CGA-1. La desviación estándar de los resultados obtenidos por el GA son 4.6333 veces (463.33%) la desviación de los resultados obtenidos por el CGA-1. Esto implica que los resultados de los 100 experimentos realizados para el GA están más dispersos que los obtenidos por el CGA-1. Y con respecto a la velocidad de convergencia, el número de iteraciones del GA es 3.2697 veces (326.97%) el número de iteraciones del CGA-1.

4 Segundo caso de estudio

4.1 Problema

El estudio de los FIS ha llevado a determinar que bajo ciertas consideraciones un FIS es un estimador universal [15], [16]. Los problemas de estimación apuntan a encontrar una función por medio de la cual se pueda aproximar de la mejor manera posible otra función. De esta manera, como una segunda aplicación de la metodología propuesta se escoge otro problema juguete el cual consiste en construir un FIS que estime la función seno cardinal, también conocida como función Sinc. A continuación se define matemáticamente la función en la ecuación (8).

$$F(x) = \begin{cases} 1 & \text{si } x = 0 \\ \frac{\sin x}{x} & x \neq 0 \end{cases} \quad (8)$$

El intervalo para el cual se desea hacer la aproximación de esta función es de $[-10, 10]$, muestreando la función cada 0.05, conformando así 401 muestras de la función. Entonces, al FIS se le inyectan las 401 muestras y se evalúa qué tan similar es la respuesta del FIS a la función original. Para realizar dicha evaluación se propone como función de aptitud el error medio cuadrático. La ecuación (6) se muestra la función de aptitud utilizada.

4.2 Experimentos

A continuación se compila en tablas los parámetros que limitan la experimentación. En la tabla 6 se definen la cantidad de entradas y reglas. La tabla 7 muestra los valores de los parámetros de experimentación del CGA y la tabla 8 los parámetros de experimentación del GA.

Tabla 6. Parámetros del FIS

Parámetro	Valor
Número de reglas	8, 16
Número de entradas	1

Tabla 7. Parámetros del CGA

Parámetro	Valor
Numero de experimentos	100
Tamaño de población (N)	1000
Número de individuos generados (S)	30, 20
Método de codificación binaria	Signo-magnitud
Cantidad de bits	16 bits
Criterio de parada	Convergencia de vector de probabilidad
Máximo número de generaciones	10000
Función de desempeño	Error medio cuadrático

Tabla 8. Parámetros del GA

Parámetro	Valor
Numero de experimentos	100
Numero de generaciones	3000
Tamaño de población (N)	30
Número de individuos intermedios	60
Probabilidad de cruce	0.7
Probabilidad de mutación	0.03
Criterio de parada	Numero de generaciones
Función de desempeño	Error medio cuadrático

4.3 Resultados

En la tabla 9 presentan los resultados obtenidos por el enfoque CGA para el caso de estimación de la función Sinc. En esta tabla se compilan los valores de media y desviación del error medio cuadrático, la media y desviación de las iteraciones necesarias para la convergencia del algoritmo. Finalmente se presentan algunas medidas de efectividad como lo son el porcentaje de individuos por debajo de la media y el mejor individuo obtenido. En la tabla 10 se presentan los resultados de media y desviación estándar del error medio cuadrático para el enfoque GA. Al igual que en el caso de estudio anterior, se medirá la relación existente entre los dos enfoques de manera porcentual con la ecuación (7).

Los resultados obtenidos de los experimentos realizados apuntan a que colección de parámetros para el CGA corresponde a un S de 30 y 16 reglas (CGA-1). La segunda mejor colección de parámetros corresponde a 16 reglas y S de 20 (CGA-2). Aunque la media del error de CGA-2 sea menor que la de CGA-1, 0.9775 veces (97.75%), la desviación estándar de CGA-2 es 1.04 veces (104.89%) la de CGA-1. Además, la media de iteraciones necesarias para converger de CGA-2 es 1.69 veces (169.28%) la de CGA-1. Por lo tanto CGA-1 es más rápido y los resultados están menos dispersos.

Tabla 9. Resultados del enfoque CGA para la estimación de la función Sinc

T1-FLS para SINC sintonizado con CGA				
	16 reglas		8 reglas	
	S=30	S=20	S=30	S=20
Media ECM	0.44731597	0.437263	0.78611941	0.80046639
Desviación Estándar ECM	0.09790743	0.10270467	0.21154149	0.24571734
Media Generaciones	2716.64	4598.74	1787.85	3156.88
Desviación Estándar Generaciones	484.385365	507.14072	268.123158	280.332569
Mejor resultado	0.20048283	0.19539059	0.40648368	0.25342646
% bajo media	54	51	61	64

Tabla 10. Resultados del enfoque GA para la estimación de la función Sinc

T1-FLS para SINC sintonizado con GA		
	16 reglas	8 reglas
Media	0.70456677	0.92090163
Desviación Estándar	0.17510043	0.26729528
Mejor resultado	0.3609232	0.51123592
% bajo media	55	63

Ahora, si se compara CGA-1 contra la mejor colección de parámetros del enfoque GA, 16 reglas (GA), se tiene que: la media del error de GA es 1.57 veces (157.09%) la de CGA-1. La desviación del error de GA es 1.78 veces (178.84%) la de CGA-1. Finalmente, la cantidad de iteraciones necesarias para converger de GA es de 1.10 (110.43%) veces las iteraciones que utiliza CGA-1

5 Conclusiones

Se presentaron dos casos de estudio en los que se estudia el modelado evolutivo de un FIS con un CGA. La intención del artículo es presentar una propuesta metodológica para el modelado de un FIS y evaluar las posibles ventajas de utilizar este enfoque, comparando el desempeño obtenido con un enfoque basado en un GA.

Los resultados obtenidos en los problemas presentados apuntan a que la propuesta presentada es más rápida y posee un mejor desempeño con respecto al enfoque basado en el GA. En términos generales los resultados obtenidos por el enfoque basado en el CGA son más estables debido a que están menos dispersos. Lo anterior es evidente, dado que en la mayoría de los casos la desviación estándar de los errores de clasificación y medio cuadrático para el enfoque basado en CGA era menor que el enfoque basado en GA.

En este punto surge el cuestionamiento, para un trabajo futuro, de extender esta propuesta al modelado de FIS tipo dos. ¿Cómo construir un FIS tipo dos de intervalo con un CGA? ¿Qué parámetros evolucionar? ¿En qué medida dicho enfoque puede ser mejor

que el enfoque presentado en este artículo? Estas preguntas quedan abiertas a la comunidad académica.

Agradecimientos. Los autores agradecen al Prof. Sergio Rojas, en la Universidad Distrital Francisco José de Caldas, Colombia, por la sugerencia de ideas durante en el desarrollo de este trabajo.

Referencias

- 1 H. Hostos, F. Sanabria, O. Mendez, and M. Melgarejo, "Towards a Coevolutionary Approach for Interval Type-2 Fuzzy Modeling," IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems, 2011.
- 2 J.E. Bonilla, V.H. Grisales, and M.A. Melgarejo, "Genetic tuned FPGA based PD fuzzy LUT controller," The 10th IEEE International Conference on Fuzzy Systems, 2001., vol. 3, 2001, p. 1084.
- 3 A. Satyadas and K. Krishnakumar, "An application of evolutionary fuzzy modeling to spacecraft fuzzy controller synthesis," IEEE/IAS International Conference on Industrial Automation and Control, 1995, 1995, p. 121.
- 4 K. Y. Uosaki, with Kawaguchi, "Nonlinear system identification based on evolutionary fuzzy modeling," Congress on Evolutionary Computation, 2004., vol. 1, 2004, p. 646.
- 5 T. Van Le, "Evolutionary fuzzy clustering," IEEE International Conference on Evolutionary Computation, 1995., vol. 2, 1995, p. 753.
- 6 G.J. Bo Yuan Klir and J.F. Swan-Stone, "Evolutionary fuzzy c-means clustering algorithm," IEEE International Conference on Fuzzy Systems, 1995., vol. 4, 1995, p. 2221.
- 7 A.G. Di Nuovo, M. Palesi, and V. Catania, "Multi-Objective Evolutionary Fuzzy Clustering for High-Dimensional Problems," IEEE International Fuzzy Systems Conference, 2007., 2007, p. 1.
- 8 D.E. Goldberg and K. Sastry, "Probabilistic Model Building and Competent Genetic Programming," Engineering, 2003.
- 9 G.R. Harik, F.G. Lobo, and D.E. Goldberg, "The Compact Genetic Algorithm," IEEE Transactions on Evolutionary Computation, vol. 3, Oct. 1999, pp. 287-294.
- 10 M.H. Afshar, "Application of a Compact Genetic Algorithm to Pipe Network Optimization Problems," Civil Engineering, vol. 16, 2009, pp. 264-271.
- 11 D. F. Naso, with Cupertino, "Real-Valued Compact Genetic Algorithms for Embedded Microcontroller Optimization," IEEE Transactions on Evolutionary Computation, vol. 12, 2008, p. 203.
- 12 F. Cupertino, E.L. Mininno, and D. E. Naso, "Optimization of Position Control of Induction Motors using Compact Genetic Algorithms," IEEE 32nd Annual Conference on Industrial Electronics, 2006, 2006, p. 55.
- 13 C. Peña, Evolutionary fuzzy modeling, Springer-Verlag, 2004.
- 14 C.L. Blake and C.J. Merz, UCI repository of machine learning databases, 1998.
- 15 J.M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Upper Saddle River, NJ: Prentice-Hall, 2001.
- 16 L.X. Wang and J.M. Mendel, "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Square Learning," IEEE Transactions on Neural Networks, vol. 3, 1992, pp. 807-813.