

MÉTODO HÍBRIDO DE SINTONIZACIÓN PARA ALGORITMOS GENÉTICOS A PARTIR DE REGRESIÓN POR VECTORES DE SOPORTE.

Híbrid method of tuning for genetic algorithms by support vector regression

RESUMEN

Se presenta una metodología para la sintonización de los parámetros de control de un algoritmo genético, enfocada a la reducción del tiempo de cómputo. Se utiliza un meta-algoritmo genético en combinación con una superficie de regresión por vectores de soporte. Esta última, además de reducir la demanda computacional del meta-algoritmo, proporciona un modelo de la interacción de los parámetros de control del algoritmo genético. Las pruebas y resultados presentados incluyen la sintonización de los tres parámetros de control fundamentales de un algoritmo genético, además de un parámetro adicional resultante de incluir operadores especiales.

PALABRAS CLAVES: Algoritmos genéticos, meta-algoritmo, regresión por vectores de soporte, tiempo de cómputo.

ABSTRACT

A methodology for simultaneous tuning of the control parameters of genetic algorithms is presented. This methodology is focused on reducing the number of generations the algorithm takes to find an optimal solution. The inherent computational costs due to the meta-algorithm are reduced by incorporating a support vector regression that models the interactions among genetic algorithm parameters. The results include tuning of the three main parameters of a GA plus an extra parameter resulting from the application of special operators.

KEYWORDS: *Computational cost, genetic algorithms, meta-algorithm, support vector regression.*

1. INTRODUCCIÓN

Los Algoritmos Genéticos (*Genetic Algorithms - GA*) son métodos computacionales basados en los procesos de selección y evolución natural, que han sido satisfactoriamente aplicados sobre una gran cantidad de problemas en diferentes áreas de investigación (una completa bibliografía sobre GA puede encontrarse en [1]). Sin embargo, para obtener máxima eficiencia se requiere el ajuste de sus parámetros de control. Esta tarea consume gran cantidad de tiempo y demanda a menudo la intervención subjetiva de un experto, por lo cual la mayoría de investigadores opta por utilizar valores reportados en trabajos previos, que no necesariamente proporcionan un buen desempeño en otros problemas [2]. Un método de auto-sintonización denominado meta-algoritmo, ha demostrado ser una forma acertada de resolver este problema; éste consiste en correr un GA de alto nivel, con el propósito de encontrar un conjunto de parámetros óptimos de un GA de bajo nivel. No obstante, la ejecución del meta-algoritmo sigue siendo computacionalmente costosa y no entrega un modelo de la influencia de los parámetros de control sobre el desempeño del GA [3], lo cual puede ser conveniente

JORGE A. JARAMILLO-GARZÓN

Ingeniero Electrónico, M.Sc.
G. Control y Procesamiento Digital de Señales
Universidad Nacional de Colombia
sede Manizales.
jjaramillo@gmail.com

JULIÁN D. ARIAS-LONDOÑO

Ingeniero Electrónico, M.Sc.
G. Control y Procesamiento Digital de Señales
Universidad Nacional de Colombia
sede Manizales.
jdariasl@unal.edu.co

GERMÁN CASTELLANOS-DOMÍNGUEZ

Ingeniero de Radiocomunicaciones,
Ph.D.
Profesor.
G. Control y Procesamiento Digital de Señales.
Universidad Nacional de Colombia
sede Manizales.
cgcastellanosd@unal.edu.co

para obtener una mayor claridad sobre el problema particular de optimización.

Los métodos de regresión por vectores de soporte (*support vector regression - SVR*) son una alternativa robusta para el cálculo de un modelo de los parámetros de control del GA y cuentan con fundamentos teóricos que permiten un análisis detallado. Por lo tanto, se presenta en este trabajo una metodología de optimización de los parámetros de control del GA, usando para ello un meta-algoritmo, en combinación con SVR para reemplazar el algoritmo de bajo nivel, reduciendo de esta manera el tiempo de cómputo necesario para la determinación del conjunto óptimo de parámetros. La SVR también permite modelar las diferentes relaciones entre los parámetros de control del GA.

2. ALGORITMOS GENÉTICOS

El primer modelo de GA propuesto, llamado *Algoritmo Genético Simple (SGA)*, es un método para moverse entre poblaciones de *cromosomas*, usando una especie de *selección natural* y los operadores de *cruce* y *mutación*, inspirados en la genética. A partir de éste, han sido desarrolladas diversas variaciones que incluyen otros

operadores, los cuales introducen parámetros de control adicionales.

2.1 Codificación de los individuos y construcción de la población inicial

Cada posible solución debe codificarse en una estructura, denominada *cromosoma*. En [4] se demostró a través de la teoría de esquemas que la mejor codificación son las cadenas binarias.

Es necesario generar una *población inicial* P_0 de r cromosomas \mathbf{c} :

$$P_0 = \{c_1, c_2, \dots, c_r\} \quad (1)$$

Esta población es generada de una forma aleatoria. El parámetro r , denominado *longitud de la población inicial*, es el primer parámetro de control fundamental.

2.2 Selección de los mejores individuos

Es necesario asignar a cada individuo un valor de *calidad* (*fitness*), a partir del cual se hará la selección. En el SGA, la calidad está definida por:

$$f(\mathbf{c}_i) = \frac{g(\mathbf{c}_i)}{\bar{g}}, i = 1, 2, \dots, r \quad (2)$$

donde $g(\mathbf{c}_i)$ es la evaluación del i -ésimo individuo y \bar{g} es el promedio de las evaluaciones de toda la población. Sin embargo, existen dos problemas asociados a esta asignación: la *convergencia prematura* y el *terminado lento*. En ambos casos, el problema puede resolverse por medio de un escalamiento de la función objetivo [5]. Diversos métodos de escalamiento de la calidad han sido desarrollados, y se ha encontrado experimentalmente que la *selección por ordenamiento* (*ranking*) es uno de los mejores [6]. Este es un método en el que los individuos son ordenados de acuerdo a su calidad y el valor esperado de cada individuo depende de su posición. Particularmente, se utilizará el *ordenamiento exponencial*, en el cual, después de ordenar los individuos desde uno (peor calificado) hasta r (mejor calificado), el valor de calidad es asignado según:

$$f(\mathbf{c}_i) = s^{r-i} \quad (3)$$

donde $s \sim 1$ es un parámetro de control.

Con base en este valor de calidad se escogen los más aptos a través de un muestreo estocástico, en particular se utiliza el modelo de *Muestreo Estocástico sobre el restante* (*Remainder Stochastic Sampling*) [2].

2.3 Recombinación y mutación

La recombinación es aplicada con probabilidad χ a cromosomas apareados al azar. El SGA utiliza específicamente la recombinación de un solo punto. Después de la recombinación, se aplica el operador de

mutación con una probabilidad μ sobre cada bit de la población, que consiste en la *negación* del bit. Estas probabilidades (χ y μ), constituyen parámetros de control.

Las etapas de selección, recombinación y mutación, son repetidas iterativamente hasta que el 95% de la población haya convergido a un individuo solución c_{opt} [7].

4. SUPERFICIES DE REGRESIÓN POR VECTORES DE SOPORTE

Las superficies de regresión por vectores de soporte (SVR), son una poderosa técnica para el análisis predictivo de datos [10]. Supóngase que se tienen los datos de entrenamiento $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$ donde $\mathbf{x}_i \in \mathbb{R}^N$. En la regresión ϵ SV [11], la meta es encontrar una función que tenga como máximo una desviación ϵ de los objetivos y_i y al mismo tiempo sea lo más plana posible. En el caso de una función lineal, se tiene:

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (4)$$

donde $\langle \cdot, \cdot \rangle$ denota el producto interno. Que la función sea lo más plana posible, significa que se quiere buscar el menor parámetro \mathbf{w} posible. Una forma de asegurar esto es por medio de la minimización de la norma $\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$. Este problema se puede reescribir como un problema de optimización convexa:

$$\begin{aligned} & \text{Minimizar } \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{Sujeto a } \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon \end{cases} \end{aligned} \quad (5)$$

La suposición tácita que se ha hecho es que dicha función g existe. Sin embargo, algunas veces este no es el caso, o se quiere tener la capacidad de permitir algunos errores. Se pueden introducir entonces, variables de flexibilidad ξ_i y ξ_i^* , llegando a la formulación establecida por [11]:

$$\begin{aligned} & \text{Minimizar } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m (\xi_i + \xi_i^*) \\ & \text{Sujeto a } \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon + \xi_i \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (6)$$

La constante C determina la compensación entre qué tan plana es g y el porcentaje de error sobre el cual las desviaciones mayores a ϵ son permitidas.

Finalmente, cuando el problema es no lineal, es necesario introducir un operador $\Phi(\mathbf{x})$ que mapee los puntos a un espacio de dimensión superior. Esto puede hacerse implícitamente, reemplazando los productos punto de las ecuaciones (4) y (6) por una función *kernel*, que

equivalga al producto punto de los vectores transformados, sin usar la transformación explícitamente:

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle \quad (7)$$

Estas funciones se encuentran caracterizadas por las condiciones de Mercer [10].

5. MÉTODO HÍBRIDO DE SINTONIZACIÓN: MODELO PROPUESTO

En este trabajo la sintonización se enfoca a la reducción del tiempo de cómputo, sujeta a una mínima degradación en la calidad de la solución encontrada. Para esto, se realizó un método híbrido con meta-algoritmo genético que incorpora una regresión por vectores de soporte para reducir su alta demanda computacional.

5.1 Función de evaluación para el algoritmo genético de alto nivel

La función de evaluación debe estar definida de tal manera que, las combinaciones de parámetros que gasten un menor número de generaciones en converger, reciban una mejor evaluación. Sin embargo, no se debe descuidar la calidad de la solución encontrada, de forma que el problema se convierte en un problema de optimización sujeto a restricciones, de la forma:

$$\begin{aligned} &\text{Minimizar } t(\gamma^\ell) && (8) \\ &\text{Sujeto a } g^\ell(\mathbf{c}_{opt}^\ell) \geq \eta \end{aligned}$$

donde $t(\gamma^\ell)$ representa el tiempo de convergencia del algoritmo usando la configuración γ^ℓ , $g^\ell(\mathbf{c}_{opt}^\ell)$ es la evaluación del mejor cromosoma encontrado por el algoritmo de bajo nivel y η es un parámetro que indica qué tan precisa debe ser la solución encontrada. Este parámetro puede ser expresado como un porcentaje de la mejor solución encontrada en el conjunto de entrenamiento:

$$\eta = \alpha \max_i \{g^\ell(\mathbf{c}_{opt_i}^\ell)\}, \quad i = 1, 2, \dots, m \quad (9)$$

donde $\mathbf{c}_{opt_i}^\ell$ es el mejor individuo encontrado por el GA de bajo nivel, para el i -ésimo conjunto de parámetros de control usados en el entrenamiento y $\alpha \in [0, 1]$.

Una de las formas más comunes para tratar con problemas de optimización con restricciones, es la utilización de penalizaciones estáticas. En este trabajo se presenta una función similar a la propuesta en [12], cuya principal ventaja es la carencia de coeficientes de penalización que deban ser ajustados:

$$f(\mathbf{x}) = (1 - H(-h_i(\mathbf{x})))g(\mathbf{x}) + \sqrt{\sum_{i=1}^K H(-h_i(\mathbf{x}))h_i(\mathbf{x})^2} \quad (10)$$

donde $h_i, i = 1, \dots, K$ son las restricciones y $H(\mathbf{x})$ es la función de Heaviside. Esta función cumple con las propiedades enunciadas por [13]: “una solución viable siempre tiene una evaluación más alta que una solución inviable; a las soluciones viables, se les asigna una evaluación dependiente únicamente de la función $g(\mathbf{x})$; a una solución no viable con un valor pequeño de violación de las restricciones, se le asigna un valor de evaluación más alto que a uno con un valor alto de violación; a las soluciones no viables se les asigna una evaluación dependiente solamente de la medida de violación de las restricciones. Esto elimina la necesidad de coeficientes de penalización independientes.”

5.2 Algoritmo genético de alto nivel

Teniendo en cuenta que diferentes combinaciones de parámetros pueden entregar evaluaciones de desempeño similares, la función a optimizar es de carácter multimodal. Por esta razón, se decidió que el GA de alto nivel tenga la habilidad de crear nichos y especies. Estas técnicas permiten mantener una distribución uniforme de la población alrededor de áreas altamente calificadas (nichos) y evitar la convergencia de toda la población a un solo óptimo. El método más utilizado para generar nichos en la población de un GA es el de *Calidad Compartida (Fitness Sharing)* [14]. Sin embargo, este método acarrea varios problemas como la dificultad de la determinación del radio para los nichos sin un conocimiento *a priori* sobre el espacio de búsqueda, o la necesidad de que los nichos deban ser equidistantes.

Un método reciente para la identificación dinámica de nichos, es el propuesto en [15]. En este trabajo se propone una modificación del algoritmo, de la siguiente forma:

1. La función es muestreada con un número suficientemente alto de puntos, de forma que refleje su topografía.
2. El parámetro con la calidad más alta en la población, es escogido como el centroide del primer nicho hiperesférico.
3. Se ordenan las muestras a partir de la muestra marcada como centroide, agregando cada vez la muestra más cercana a la que se acaba de poner en la lista.
4. Se define el radio del nicho, buscando una muestra en la secuencia, cuya calidad sea superior a la de la muestra inmediatamente anterior.
5. Se escoge la muestra con calidad más alta entre la población que aún no pertenece a ningún nicho, y se repiten los pasos 2 y 3 hasta que todas las muestras pertenezcan a algún nicho.
6. Se modifican los nichos que hayan quedado traslapados.

Finalmente, utilizando la información obtenida a partir de la identificación dinámica de nichos, la calidad de cada individuo es asignada según un esquema de calidad compartida propuesto en [15]:

$$f'_i = \frac{f_i}{N_j} \quad i = 1, 2, \dots, r \quad (11)$$

donde f'_i es la nueva calidad asignada a la i -ésima muestra de la población, f_i es la calidad de la muestra, N_j es la cantidad de muestras en el nicho al que pertenece la muestra y r es el tamaño de la población. Según esta definición, cada muestra en el mismo nicho tendrá una penalización igual, mientras las muestras que no han quedado dentro de ningún nicho serán penalizadas con el promedio de las penalizaciones en todos los nichos.

5.3 Selección del modelo para la SVR

La selección del modelo consiste en la elección de los parámetros ε (ancho del tubo), C (penalización de los errores) y el kernel $k(x, y)$. Se ha escogido la forma de sintonización analítica propuesta recientemente en [16]:

$$C = \max \left| \bar{y} + 3\sigma_y \right|, \left| \bar{y} - 3\sigma_y \right| \quad (12)$$

$$\varepsilon = 3\sigma_{ns} \sqrt{\frac{\ln n}{n}}$$

donde \bar{y} y σ_y son respectivamente la media y la desviación estándar de los objetivos de entrenamiento, σ_{ns} es la desviación estándar del ruido de los datos y n es el número de muestras de entrenamiento.

La estimación de σ_{ns} se puede realizar a través de una *regresión de k-vecinos más cercanos*, utilizando la ecuación también propuesta en [16]:

$$\hat{\sigma}_{ns}^2 = \frac{n^{1/5} k}{n^{1/5} k - 1} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (13)$$

donde k es el orden de la regresión de k-vecinos más cercanos y $y_i, i = 1, \dots, n$ son las estimaciones realizadas por dicha regresión.

En cuanto al kernel, se selección el RBF gaussiano, por sus conocidas buenas propiedades [17]. Este kernel está dado por:

$$K(x, y) = \exp(-\gamma \|x - y\|^2) \quad (14)$$

para el cual es necesario determinar el parámetro γ . Para este fin, fue utilizada la metodología propuesta en [20], donde se demuestra por comparación con la *teoría de espacios escalados* que la variación de la capacidad de generalización de la SVR, es una función semiconvexa

dependiente de γ . Por lo tanto, γ puede ser encontrada mediante la minimización del error de generalización, utilizando algún método para funciones semiconvexas. El error de generalización puede ser estimado a través del límite $\xi\alpha$ enunciado en [18].

6. EXPERIMENTOS Y RESULTADOS

En primer lugar es necesario definir cuáles son los problemas de optimización sobre los cuales se prueba la metodología propuesta. Para este fin se escogió un conjunto de *funciones de evaluación comparativa (benchmarking)* presentado en [19]. En la Tabla 1 se muestran las ocho funciones a optimizar, que contienen características continuas y discontinuas, convexas y no convexas, unimodales y multimodales, de baja y alta dimensionalidad, deterministas y estocásticas. En la Figura 1 se encuentra la representación gráfica de cada una de estas funciones, para lo cual se redujo su dimensión a dos.

En segundo lugar, es necesario definir los parámetros de control que se van a optimizar en el GA de bajo nivel. Como se describió en la sección 2, se buscar optimizar los tres parámetros de control fundamentales: Tamaño de la población inicial r , probabilidad de recombinación χ y probabilidad de mutación μ . Además un parámetro resultante del operador de ordenamiento exponencial, que representa el valor esperado de copias para el segundo mejor individuo de cada generación.

Estos parámetros se codificaron en un cromosoma de la siguiente manera:

- 1 Tamaño de la población inicial: Desde 10 hasta 160 individuos, con incrementos de 10 (4 bits).
- 2 Probabilidad de recombinación: Entre 0.3 y 1, con incrementos de 0.1 (3 bits).
- 3 Probabilidad de mutación: 10^{-i} , para $i=1, \dots, 8$ (3 bits).
- 4 Parámetro del ordenamiento exponencial: Entre 0.9 y 0.99 con incrementos de 0.006 (4 bits).

Función	Límites
$f_1 = \sum_{i=1}^2 x_i^2$	$ x_i \leq 5.12$
$f_2 = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$	$ x_i \leq 2.04$
$f_3 = \sum_{i=1}^5 \text{int.}(x_i)$	$ x_i \leq 5.12$
$f_4 = \sum_{i=1}^{30} (ix_i^4 + \text{Gauss}(0,1))$	$ x_i \leq 1.28$
$f_5 = 0.002 + \sum_{j=1}^{25} \left(\frac{1}{j} + \sum_{i=1}^{25} 2(x_i - a_{ij})^6 \right)$	$ x_i \leq 65.5$
$f_6 = 41898.3 + \sum_{i=1}^{10} (-x_i \sin \sqrt{ x_i })$	$ x_i \leq 500$
$f_7 = 200 + \sum_{i=1}^{20} (x_i^2 - 10 \cos(2\pi x_i))$	$ x_i \leq 5.12$
$f_8 = 1 + \sum_{i=1}^{10} \left(\frac{x_i^2}{4000} - \prod_{i=1}^{10} \left(\cos \left(\frac{x_i}{\sqrt{i}} \right) \right) \right)$	$ x_i \leq 600$

Tabla 1. Funciones de evaluación comparativa.

Se forma entonces un espacio de búsqueda discreto de 2^{14} combinaciones (más de dieciséis mil), de las cuales se tomaron dos mil para construir cada superficie de regresión.

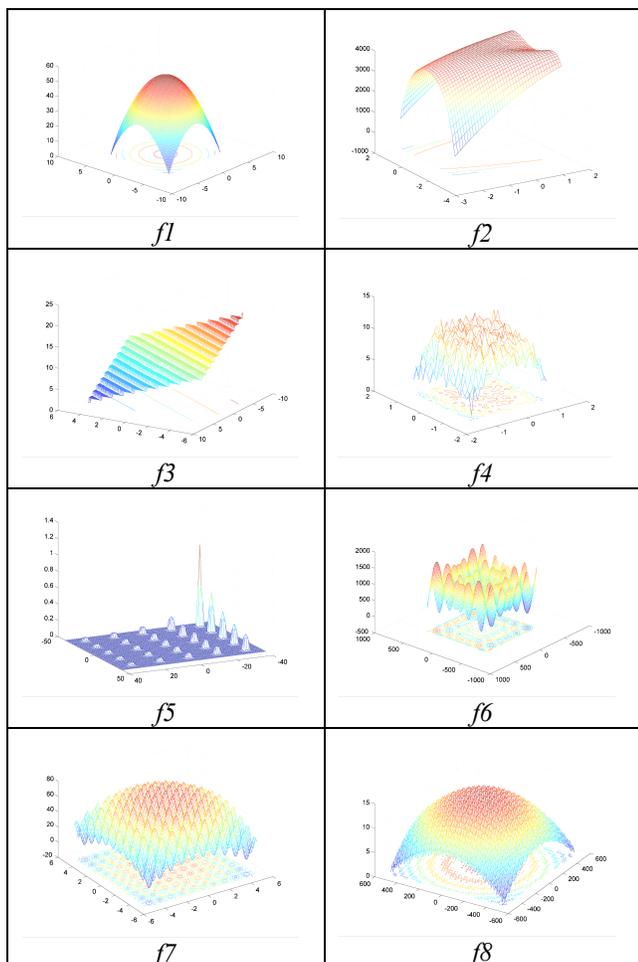


Figura 1. Funciones de evaluación comparativa.

El GA de alto nivel utiliza la función de evaluación está definida como en la ecuación (10), con $\alpha=0.99$, es decir, se busca minimizar el tiempo de convergencia del algoritmo, sujeto a que las soluciones encontradas no tengan un desempeño inferior al 99% de la mejor evaluación encontrada en el conjunto de entrenamiento. Sin embargo, dado que esta función es discontinua, sería difícil para la SVR modelar el desempeño del GA de bajo nivel, por lo tanto se decidió construir dos superficies: Una para modelar el número de generaciones y otra para modelar la evaluación obtenida, y con las estimaciones entregadas por estas superficies, evaluar la función (10). Se comparó contra un GA sintonizado con los parámetros propuestos por [7] y contra la metodología propuesta por [9]. Cada prueba se repitió 10 veces para eliminar las componentes aleatorias de los algoritmos. En las tablas 2 y 3, se muestran los promedios del número de generaciones y el porcentaje de error respecto al máximo cometidos por cada metodología.

	Parámetros recomendados	Meta-GA Red Neuronal	Meta-GA SVR ($\alpha = 0.99$)
f1	42.8	15.3	6.6
f2	48.9	10	6.2
f3	96.1	57.7	83.9
f4	>100	>100	77.4
f5	62.6	30.6	33.9
f6	70.5	15.6	47.2
f7	>100	>100	44.3
f8	94.5	47.3	32.5

Tabla 2. Número de generaciones obtenido con cada metodología.

	Parámetros recomendados	Meta-GA Red Neuronal	Meta-GA SVR ($\alpha = 0.99$)
f1	$(2.28)e-4$	$(8.39)e-4$	$(6.90)e-2$
f2	$(3.27)e-3$	$(1.41)e-3$	$(6.99)e-3$
f3	1.15	3.84	1.15
f4	1.64	2.55	$(5.33)e-1$
f5	19.94	14.20	1.71
f6	7.61	8.47	7.03
f7	21.01	4.14	2.98
f8	9.28	$6.91e^{-2}$	$(6.21)e-1$

Tabla 3. Porcentaje de error respecto al máximo verdadero.

De las tablas 2 y 3, es posible observar que la metodología propuesta, mejora el tiempo de cómputo en la mayoría de funciones, conservando unos porcentajes de error muy similares e incluso inferiores a los de la metodología propuesta en [9].

A pesar de que en algunas funciones la metodología de [9] parece haber alcanzado un buen resultado en un menor número de generaciones, como en el caso de $f3$ y $f5$, es necesario observar que los porcentajes de error en esas mismas funciones, son de 3.84 y 14.2 respectivamente, mientras que para la metodología propuesta los errores son de apenas 1.15 y 1.71. Esto es debido principalmente a la restricción impuesta sobre la función de evaluación. Recuérdese de la ecuación (10), que el haber seleccionado $\alpha=0.99$ significa que se le está pidiendo al GA de alto nivel encontrar la configuración de los parámetros de control que minimice el número de generaciones requerido para obtener una solución con un error no inferior al 1%. Aunque los errores fueron superiores al 1%, esto es debido a que el algoritmo de entrenamiento no conoce el máximo verdadero sino la mayor evaluación dentro del conjunto de entrenamiento, sin embargo, los errores se aproximan al valor pedido.

7. CONCLUSIONES

Se desarrolló una metodología para la optimización de los parámetros de control de un algoritmo genético, enfocada a la reducción del número de generaciones empleado para encontrar una solución. Su uso un meta-algoritmo que incorpora una regresión por vectores de soporte lo cual representa un costo computacional mucho menor que el meta-algoritmo simple.

La optimización de los parámetros de control es planteada como un problema de optimización sujeto a restricciones. La función planteada incorpora un parámetro que permite al usuario tener un control sobre el porcentaje de error que le será permitido al algoritmo genético.

Los elementos que componen la metodología cuentan con un sustento suficiente para la sintonización de sus propios parámetros, de tal manera que el usuario no esté en la necesidad de ajustar ningún valor adicional para conseguir el mejor rendimiento.

8. AGRADECIMIENTOS

Este trabajo se realiza en el marco del proyecto “Detección de los niveles de compromiso de resonancia en niños con labio y/o paladar hendido”, financiado por la Dirección de Investigaciones DIMA, de la Universidad Nacional de Colombia Sede Manizales.

9. BIBLIOGRAFÍA

- [1] J. Alander., Indexed bibliography of genetic algorithms implementations, 1999.
- [2] D. Whitley., An overview of evolutionary algorithms: practical issues and common pitfalls. *Information and software technology*, 43(14), pp. 817-831, 2001.
- [3] M. Orozco., Optimización de los parámetros de control del algoritmo genético simple usando regresión de vectores soporte. *In IX simposio sobre tratamiento de señales, imágenes y visión artificial*. Manizales, Colombia, 2004.
- [4] J. H. Holland., *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
- [5] D.E. Goldberg., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Massachusetts, 1989.
- [6] B. T. Zhang y J. J. Kim., Comparison of selection methods for evolutionary optimization, 2000.
- [7] K. A. DeJong., An analysis of the behavior of a class of genetic adaptive system. PhD thesis, University of Michigan, 1975.
- [8] J. J. Grefenstette., Optimizations of control parameters for genetics algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, 16(1), pp. 122-128, 1986.
- [9] S. F. Smith y V. A. Cicirello., Modeling GA performance for control parameters optimization. *In Proceedings of the genetic and evolutionary Computation Conference (GECCO-2000)*, 99, 235-242, Las Vegas, Nevada, USA, 2000.
- [10] A. Smola y B. Schoelkopf. A tutorial on support vector regression, 1998.
- [11] V. Vapnik. *Statistical learning theory.*, Wiley, New York, 1998.
- [12] F. Hoffmeister y J. Sprave., Problem-independent of constraints by use of metric penalty functions. *In Evolutionary Programming*, pp. 289-294, 1996.
- [13] A. I. Oyman, K. Deb y H. G. Beyer., An alternative constraint handling method for evolution strategies. *In 1999 Congress on Evolutionary Computation*, pp. 612-619, Piscataway, NJ, 1999.
- [14] D. E. Goldberg y J. Richardson, Genetic Algorithms with sharing for multimodal function optimization. *In Genetic Algorithms and their applications: Proc. of the second Int. Conf. on Genetic Algorithms*, pp. 41-49, Hillsdale, NJ, 1987.
- [15] C. Y. Lin y W.-H. Wu., Niche identification techniques in multimodal genetic search with sharing scheme., *Advances in Engineering Software*, 33(11-12), pp. 779-791, 2002.
- [16] V. Cherkassky and Y. Ma., Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1), pp. 113-126, Jan. 2004.
- [17] W. Wang, Z. Xu, W. Lu y X. Zhang., Determination of the spread parameter in the gaussian kernel for classification and regression, *Neurocomputing*, 55(3-4), pp. 643-663, 2003.
- [18] J. Liu y Y. Tan., Estimating the leave-one-out error for support vector regression. *In International conference on Neural Networks and Brain*, 1, pp. 208-213, 2005.
- [19] K. Digalakis y K. Margaritis., An experimental study of benchmarking functions for evolutionary algorithms. *International Journal of Computer Mathematics*, 79(4), pp. 403-416, April 2002.