

ESTRATEGIAS EVOLUTIVAS COMO UNA OPCIÓN PARA LA OPTIMIZACIÓN DE FUNCIONES NO LINEALES CON RESTRICCIONES

Diana M. Ortiz*
Juan David Velásquez H**
Patricia Jaramillo***

Recibido: 28/07/2009

Aceptado: 11/05/2011

RESUMEN

Estrategias de evolución es una técnica bio-inspirada, eficiente y robusta para resolver problemas de optimización donde el espacio de soluciones es no restringido. Sin embargo, esta suposición es irreal en muchos casos porque el espacio de soluciones es limitado por fronteras complejas en la forma de restricciones tanto lineales como no lineales. En este artículo de investigación, se propone una modificación al algoritmo original de estrategias de evolución para optimizar problemas donde el espacio de soluciones es limitado usando restricciones complejas. El método propuesto es basado en el uso de una función de penalización la cual es cero dentro de la región factible, e igual al máximo valor dentro de la región factible cuando se considera un punto que es no factible. La aproximación propuesta es probada usando seis problemas de prueba bien conocidos. En todos los casos, esta aproximación encontró un punto óptimo igual o menor que los valores reportados en la literatura.

Palabras clave: estrategias evolutivas, optimización, minimización, funciones de penalización.

* Universidad Nacional de Colombia. Escuela de Sistemas. Grupo de Computación Aplicada. Carrera 80 No. 65-223, Bloque M8A, Oficina 201, Medellín, Colombia. Tel.: +57 +4 4255350. E-mail: dmortiz@unal.edu.co

** Autor para correspondencia. Universidad Nacional de Colombia. Escuela de Sistemas. Grupo de Computación Aplicada. Carrera 80 No. 65-223, Bloque M8A, Oficina 206, Medellín, Colombia. Tel.: +57 +4 4255370. E-mail: jdvelasq@unal.edu.co

*** Universidad Nacional de Colombia. Escuela de Sistemas. Grupo de Computación Aplicada. Carrera 80 No. 65-223, Bloque M8A, Oficina 213, Medellín, Colombia. Tel.: +57 +4 4255222. E-mail: gpjarami@unal.edu.co

EVOLUTION STRATEGIES AS AN OPTION FOR OPTIMIZING NON-LINEAR FUNCTIONS WITH RESTRICTIONS

ABSTRACT

Evolution Strategies is a bio-inspired, robust, and efficient technique for solving optimization problems where the solution space is unrestricted. However, this assumption is unreal in many cases because the solution space is limited by complex boundaries in the form of linear and non-linear restrictions. In this paper, a modification of the original algorithm of Evolution Strategies for optimizing problems where the solution space is bounded using complex restrictions is proposed. The proposed method is based on the use of a penalization function which is zero inside of the feasible region and equal to the maximum value inside of the feasible region when an unfeasible point is considered. The proposed approach is proved using six benchmark problems. In all cases, our approach found an optimal point equal or lower than the values reported in the literature.

Key words: evolution strategies, optimization, minimization, penalization functions.

INTRODUCCIÓN

Es indudable la importancia de contar con metodologías eficientes para la solución de problemas de optimización, de tal forma que no es sorprendente que se haya desarrollado un número importante de metodologías durante las últimas décadas [1]. En el caso general, donde se consideran variables continuas, el problema de optimización puede ser planteado como:

$$\min f(\vec{x}), \vec{x} \in R^n \quad (1)$$

Sujeto a unas restricciones, que pueden ser lineales o no lineales:

$$h_i(\vec{x}) = 0, \quad i=1, \dots, m \quad (2)$$

$$g_i(\vec{x}) \leq 0, \quad i=1, \dots, n \quad (3)$$

donde:

- $f(\vec{x})$ representa una función objetivo escalar multivariable.
- $h_i(\vec{x}) = 0$ representa un conjunto de igualdades, lineales y no lineales, que deben satisfacerse.
- $g_i(\vec{x}) \leq 0$ representan un conjunto de desigualdades, lineales y no lineales, que deben satisfacerse.

En el caso en que las variables x son continuas y tanto la función objetivo como las restricciones son lineales, el modelo se reduce a un problema de programación lineal que puede ser resuelto eficientemente por diferentes métodos [2]. Cuando no existen restricciones, y la función $f()$ es no lineal, el modelo representa un problema de optimización global continua, cuya solución se ve dificultada cuando la función $f()$ no es diferenciable o presenta una superficie compleja con múltiples puntos de óptima [3]; en este último caso, la búsqueda del mínimo global podría hacerse mediante técnicas determinísticas [4] o mediante técnicas que vinculan elementos estocásticos en el proceso de búsqueda, tales como la búsqueda lineal adaptativa [5], el temple simulado [6, 7], la búsqueda aleatoria dinámica [8], o la búsqueda tabú [9].

Entre las técnicas estocásticas, los algoritmos

evolutivos forman un conjunto de técnicas que basan su funcionamiento en modelar procesos evolutivos fundamentados en la supervivencia de los individuos más aptos en la población [10]. Son utilizados con frecuencia en problemas del mundo real donde la complejidad y la dimensionalidad son considerables, y las técnicas basadas en gradientes son de poca utilidad. Entre sus mayores ventajas se encuentran su elegancia conceptual, su amplia aplicabilidad, y su superioridad a las técnicas tradicionales en muchos problemas reales. Al basarse en el principio de evolución, son robustos a cambios dinámicos y proporcionan el potencial para incorporar conocimiento sobre el dominio y para hibridarse con otras técnicas de optimización [11]. La calidad de la solución depende principalmente de la transformación de la función objetivo para que pueda ser analizada dentro de la región factible y llegar a la región de la solución óptima (individuo más apto).

Las estrategias evolutivas (EE) hacen parte de los algoritmos evolutivos, por lo que utilizan el mismo concepto básico; a diferencia de los algoritmos genéticos, la solución es codificada directamente por sus valores reales; su operador principal es la mutación, y el secundario el cruce; la selección es determinada con base en la aptitud; y sus parámetros son auto-adaptativos [10]. En su desarrollo original, permiten realizar la búsqueda del óptimo global del problema definido en (1) cuando no existen restricciones. Por ejemplo, se sabe que la estimación de los parámetros de modelos de redes neuronales es un problema difícil debido a la complejidad de la superficie de error, y la existencia de múltiples puntos de óptima local, de ahí que algoritmos estocásticos como EE puedan ser ventajosos en la estimación de dichos modelos. Particularmente, en [12] se compara EE con el algoritmo tradicional de *backpropagation*, mientras que en [13] se compara las EE con el algoritmo basado en gradientes RPROP.

Tal como ya se indicó anteriormente, la incorporación de restricciones, definidas en (2) y (3),

dificulta la búsqueda de la solución de (1), por lo que se han propuesto diferentes algoritmos para ello. Dichas técnicas incluyen: funciones de penalización, representaciones y operadores especiales, algoritmos de reparación, separación de restricciones y objetivos, y métodos híbridos. Una amplia discusión es presentada en [14, 15]. El objetivo de este artículo es presentar una modificación a la metodología de EE que permite el manejo de restricciones generales, tales como las definidas en (2) y (3). La modificación consiste en el uso de la función de penalización propuesta en [16], la cual hace que en un problema de minimización, la función objetivo tome un valor igual al máximo de la región factible más el cuadrado de la violación de las restricciones. El efecto de esta forma de penalización es que cualquier punto dentro de la región factible tenga un valor inferior a cualquier punto por fuera, forzando a que el algoritmo de búsqueda no pueda explorar regiones no factibles.

Para cumplir con el objetivo propuesto, en la próxima sección se describe de forma sucinta el algoritmo de EE; en la siguiente sección, se describe, de forma detallada, el método de penalización y su integración al algoritmo de EE; posteriormente, se analizan los resultados obtenidos al optimizar seis funciones de prueba con el algoritmo propuesto.

1 ESTRATEGIAS EVOLUTIVAS

Esta técnica de optimización basada en el principio de evolución de Darwin, fue originalmente desarrollada por Rechenberg [17, 18] y llevada a su forma actual por Schwefel [19, 20]. En EE [10], cada individuo a_k está formado por:

- Un conjunto de variables objeto: $x_i, i=1, \dots, n$
- Un conjunto de mutaciones: $\sigma_i, i=1, \dots, n$
- Un conjunto de ángulos de rotación: $\alpha_j, j=1, \dots, n(n-1)/2$

La optimización es realizada sobre una población conformada por μ individuos siguiendo el algoritmo descrito en la figura 1.

```

t := 0;
Inicializar:  $\mathbf{P}(0) := \{a_1(0), \dots, a_\mu(0)\}$ ;
Evaluar:  $\Phi(\mathbf{P}(0)) := \{f(a_1(0)), \dots, f(a_\mu(0))\}$ ;
while(  $\iota(\mathbf{P}(t)) \neq \text{true}$  ) do
    Recombinar:  $\mathbf{P}'(t) := r(\mathbf{P}(t))$ ;
    Mutar:  $\mathbf{P}''(t) = m(\mathbf{P}'(t))$ ;
    Evaluar:  $\Phi(\mathbf{P}''(t)) := \{f(a_1(t)), \dots, f(a_\lambda(t))\}$ ;
    Seleccionar  $\mathbf{P}(t+1) = s_{(\mu+\lambda)}(\mathbf{P}''(t))$ ;
    t := t + 1;
end do

```

Figura 1. Estrategias de evolución

Fuente: adaptado de [10]

La población de μ individuos es inicializada de la siguiente forma: las variables objeto x_i toman un valor aleatorio inicial; las variables de estrategia σ_i toman un valor de 3.0 según la sugerencia dada por Bäck [10]; los ángulos de rotación α_j se inician en un valor aleatorio entre $-\pi$ y π . La población inicial es evaluada calculando el valor de la función objetivo $f()$ para cada individuo de la población inicial. El proceso de evolución se realiza dentro del ciclo *while*. Una población de λ hijos $\mathbf{P}'(t)$ es obtenida al aplicar el operador de recombinación $r()$ a la población actual de padres $\mathbf{P}(t)$; la recombinación puede darse en tres formas: por clonación del padre, por recombinación sexual entre dos padres (local), y por recombinación sexual entre toda la población (global). La recombinación opera sobre las variables objeto, las variables de estrategia y sobre los ángulos de rotación. Así, cada componente (variable objetivo, variable de estrategia o ángulo de rotación) del nuevo individuo, c_i en la población de hijos, puede ser calculada, dependiendo del tipo de recombinación:

- Sin recombinación:

$$c_i = c_{S,i} \quad (5)$$

- Discreta:

$$c_i = c_{S,i} \text{ or } c_{T,i} \quad (6)$$

- Panmítica o global discreta:

$$c_i = c_{S,i} \text{ or } c_{T,i} \quad (7)$$

- Intermedia:

$$c_i = c_{S,i} + (c_{T,i} - c_{S,i}) / 2 \quad (8)$$

- Panmítica o global intermedia:

$$c_i = c_{S,i} + (c_{T,i} - c_{S,i}) / 2 \quad (9)$$

- Intermedia generalizada:

$$c_i = c_{S,i} + \chi \times (c_{T,i} - c_{S,i}) \quad (10)$$

- Reproducción global o panmítica intermedia:

$$c_i = c_{S,i} + \chi_i \times (c_{T,i} - c_{S,i}) \quad (11)$$

Los subíndices S y T representan a dos padres seleccionados al azar de la población de μ individuos. Todos los padres tienen la misma probabilidad de ser seleccionados. El valor χ_i representa un número aleatorio uniforme entre cero y la unidad.

Una vez se ha obtenido la población de λ hijos, se procede a mutar cada uno de ellos; esto se hace de la siguiente forma:

$$\sigma'_i = \sigma_i \exp(\tau' \cdot N[0, 1] + \tau \cdot N_i[0, 1]) \quad (12)$$

$$x'_i = x_i + \sigma'_i \cdot N_i[0, 1] \quad (13)$$

$$\alpha'_{i,j} = \alpha_{i,j} + \beta \cdot N_j[0, 1] \quad (14)$$

Donde $N[0, 1]$ representa un número aleatorio normal con media cero y desviación estándar unitaria, y los símbolos τ' , τ y β representan constantes.

Una vez se han mutado los individuos usando (12), (13) y (14), se procede a escoger la nueva población con la que se iniciará el ciclo nuevamente. En este caso en particular, se realiza una selección del tipo +, en la cual se hace un ordenamiento del conjunto de padres unido con el conjunto de hijos, y se selecciona la nueva población de esta unión. El proceso se continúa hasta que se cumpla con un criterio de terminación representado por $t(\mathbf{P}(t))$.

Las EE poseen la propiedad de auto-adaptación, lo cual implica que no sólo evolucionan las variables del problema, sino también los parámetros mismos del algoritmo (variables de estrategia)

2 INCORPORACIÓN DE LAS RESTRICCIONES

El algoritmo presentado en la sección anterior corresponde a la versión tradicional de EE, y permite

la búsqueda del óptimo global para una función no lineal. Una de sus desventajas es que no permite la optimización de problemas más generales, tal como el definido por (1), (2) y (3). En esta sección se describe el mecanismo de penalización desarrollado por [14] y su integración dentro del algoritmo presentado en la figura 1.

Hoffmeister y Sprave [16] proponen introducir una penalización dentro de la función objetivo $f()$ que permita medir el grado en que las restricciones descritas por (2) y (3) son violadas. Así, el algoritmo descrito en (1) debe operar sobre una nueva función $F()$ definida como:

$$F(x) = \begin{cases} f(x) & x \in \Omega \\ M_c + d_c & x \notin \Omega \end{cases} \quad (15)$$

Donde: Ω representa la región factible; M_c es el máximo conocido en la región factible; y d_c es una medida de penalización de las restricciones que no se cumplen. En esta implementación, d_c se define como:

$$d_c = \max\{H_1 \times |R_1|, \dots, H_m \times |R_m|\} \quad (16)$$

Donde H_i es una variable que toma el valor de uno si la i -ésima restricción ha sido violada, y cero en caso contrario; y R_i es el valor que toma $g_i(x)$ o $h_i(x)$ en el punto x .

La versión modificada de EE para el manejo de restricciones es presentada en la figura 2. La principal diferencia con el algoritmo presentado en la figura 1 corresponde a la evaluación de la función objetivo modificada (15); para su cálculo se requiere evaluar las restricciones del problema y estimar la penalización d_c para cada individuo. El principal inconveniente radica en la estimación del valor máximo (M_c) de la función objetivo $f()$ dentro de la región factible, ya que este es desconocido y debe ser obtenido como un resultado del proceso de evolución. Así, en cada generación, M_c es calculado como el máximo entre el valor actual de M_c y la función objetivo $f()$ evaluada para cada uno de los λ hijos; si un nuevo máximo es encontrado, la función $F()$ debe ser reestimada para la población de padres $\mathbf{P}(t)$.

```

t := 0;
Inicializar: P(0) := {a1(0),...,aμ(0)};
Evaluar: f(P(0)) := {f(a1(0)),...,f(aμ(0))};
Evaluar: Mc := max{f(P(0))};
Evaluar: dc(P(0)) := {dc(a1(0)),...,dc(aμ(0))};
Evaluar: Φ(P(0)) := {F(a1(0)),...,F(aμ(0))};
while(!P(t) ≠ true ) do
  Recombinar: P'(t) = r(P(t));
  Mutar: P''(t) = m(P'(t));
  Evaluar: Mc' := max{Φ(P''(t)), Mc}
  Evaluar: dc(P''(t)) := {dc(a1(t)),...,dc(aλ(t))};
  Evaluar: F(P''(t)) := {F(a1(t)),...,F(aλ(t))};
  Seleccionar P(t+1) = s(μ+λ)(P(t), P''(t));
  t := t + 1;
end do
    
```

Figura 2. Estrategias de evolución modificada para el manejo de restricciones

Fuente: elaboración propia

3 CASOS DE ESTUDIO Y ANÁLISIS

Para validar el algoritmo propuesto, se hizo la estimación del mínimo para seis funciones sujetas a restricciones, las cuales han sido ampliamente utilizadas en la literatura para la evaluación de

algoritmos de optimización.

Los problemas presentados, que son descritos en la tabla 1, incluyen variedad en la cantidad de restricciones con ecuaciones e inecuaciones tanto lineales como no lineales, de variables, de puntos de inicio tanto factibles como no factibles y límites superior e inferior para las diferentes variables; además de poseer funciones-objetivo no lineales. El problema 1 posee dos restricciones: una inecuación no lineal y una ecuación lineal [4], y su punto de inicio es no factible; el problema 2 no posee restricciones [21]; el problema 3 fue diseñado con un punto estacionario no óptimo en $f(\bar{x}) \approx 8$, lo que puede causar una convergencia prematura [22]; el problema 5 posee 5 variables, está sujeto a 6 inecuaciones no lineales y 10 límites sobre las variables independientes. Nótese que en la función objetivo los coeficientes de x_2x_4 son cero, es decir, x_2x_4 no están incluidos en la definición de $f(\bar{x})$; y el problema 6 está sujeto a dos restricciones: una ecuación no lineal y una inecuación no lineal [4].

Tabla 1. Soluciones obtenidas usando el algoritmo propuesto.

Problema		Punto de Inicio	Resultados		
				Reportado	EE
1.	Minimizar: $f(\bar{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$ Sujeto a: $h_1(\bar{x}) = x_1 - 2x_2 + 1 = 0$ $g_1(\bar{x}) = \frac{x_1^2}{4} + x_2^2 - 1 \leq 0$	No Factible $\bar{x}^{(0)} = [2.0 \ 2.0]^T$ $f(\bar{x}^{(0)}) = 1$	$f(\bar{x}^*)$	1.3930	1.3900
			x_1^*	0.8230	0.8243
			x_2^*	0.9110	0.9117
2.	Minimizar: $f(\bar{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	$\bar{x}^{(0)} = [-1.2 \ 1.0]^T$ $f(\bar{x}^{(0)}) = 24.20$	$f(\bar{x}^*)$	0.0000	0.0000
			x_1^*	1.0000	1.0000
			x_2^*	1.0000	1.0000
3.	Minimizar: $f(\bar{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$ $+ 90(x_4 - x_3^2)^2 + (1 - x_3)^2$ $+ 10.1[(x_2 - 1)^2 + (x_4 - 1)^2]$ $+ 19.8(x_2 - 1)(x_4 - 1)$ Sujeto a: $-10 \leq x_i \leq 10 \quad i = 1, 2, 3, 4$	Factible $\bar{x}^{(0)} = \begin{bmatrix} -3 \\ -1 \\ -3 \\ -1 \end{bmatrix}$ $f(\bar{x}^{(0)}) = 192192$	$f(\bar{x}^*)$	0.0000	0.0000
			x_1^*	1.0000	1.0000
			x_2^*	1.0000	1.0000
			x_3^*	1.0000	1.0000
			x_4^*	1.0000	1.0000

Problema		Punto de Inicio	Resultados		
				Reportado	EE
4.	Minimizar $f(\vec{x}) = (x_1 - 2)^2 + (x_2 - 1)^2$ Sujeto a: $g_1(\vec{x}) = x_1^2 - x_2^2 \leq 0$ $g_2(\vec{x}) = x_1 + x_2 - 2 \leq 0$	No Factible $\vec{x}^{(0)} = [2.0 \ 2.0]^T$ $f(\vec{x}^{(0)}) = 1$	$f(\vec{x}^*)$	1.0000	0.9987
			x_1^*	1.0000	1.0007
			x_2^*	1.0000	1.0003
5.	Minimizar: $f(\vec{x}) = 5.3578547 x_3^2 + 0.8356891 x_1 x_5$ $+ 37.293239 x_1 - 40792.141$ Sujeto a: $0 \leq 85.334407 + 0.0056858 x_2 x_5$ $+ 0.0006262 x_1 x_4$ $- 0.0022053 x_3 x_5 \leq 92$ $90 \leq 80.51249 + 0.0071317 x_2 x_5$ $+ 0.0029955 x_1 x_2$ $+ 0.0021813 x_3^2 \leq 110$ $20 \leq 9.300961 + 0.0047026 x_3 x_5$ $+ 0.0012547 x_1 x_3$ $+ 0.0019085 x_3 x_4 \leq 110$ $78 \leq x_1 \leq 102$ $33 \leq x_2 \leq 45$ $28 \leq x_3 \leq 45$ $27 \leq x_4 \leq 45$ $27 \leq x_5 \leq 45$	Factible $\vec{x}^{(0)} = \begin{bmatrix} 78.62 \\ 33.44 \\ 31.07 \\ 44.18 \\ 35.22 \end{bmatrix}$ $f(\vec{x}^{(0)}) = -30665.50$		-30665.5000	-30667.0000
				78.0000	77.9990
				33.0000	32.9990
				29.9950	29.9899
				45.0000	45.0010
				36.7760	36.7820
6.	Minimizar: $f(\vec{x}) = 1000 - x_1^2 - 2x_2^2 - x_3^2$ $- x_1 x_2 - x_1 x_3$ Sujeto a: $h_1(\vec{x}) = x_1 - 2x_2 + 1 = 0$ $g_1(\vec{x}) = \frac{x_1^2}{4} + x_2^2 - 1 \leq 0$	Factible $\vec{x}^{(0)} = \begin{bmatrix} 78.62 \\ 33.44 \\ 31.07 \\ 44.18 \\ 35.22 \end{bmatrix}$ $f(\vec{x}^{(0)}) = -30665.50$	$f(\vec{x}^*)$	961.715	955.4206
			x_1^*	3.512	-0.7189
			x_2^*	0.217	4.8216
			x_3^*	3.552	-0.7189

Fuente: elaboración propia.

Los problemas de la tabla 1, fueron optimizados con los siguientes parámetros:

- Generaciones: 1000
- Tamaño de la población de hijos (λ): 500
- Tamaño de la población de padres (μ): 50
- Tipo de selección: +
- Tipo de recombinación para x , σ y α : panmítica.

En la tabla 1, se presentan los resultados obtenidos para los casos de estudio propuestos; la evidencia empírica aportada en esta investigación muestra que el algoritmo propuesto permite encontrar el mínimo para cada caso de estudio. Más aún, nuestra implementación logra obtener puntos de óptima inferiores a los reportados en la literatura para los problemas 1, 4, 5 y 6. Para los problemas 2 y 3, los puntos obtenidos son iguales a los alcanzados por otras técnicas. En el problema 3, el algoritmo propuesto supera el punto estacionario no óptimo, y llega al óptimo global.

4 CONCLUSIONES

En este artículo se presenta una modificación al algoritmo de estrategias de evolución para minimizar problemas sujetos a restricciones generales, basada en una función de penalización. Este mecanismo asegura que la función objetivo en cualquier punto externo a la región factible tiene un valor superior a cualquier punto en su interior. Los seis problemas de optimización utilizados en este artículo fueron resueltos satisfactoriamente por el algoritmo propuesto, y en algunos de los casos se encontraron soluciones en las cuales el valor de la función objetivo es inferior al reportado en la literatura.

Como trabajo futuro se plantea hacer una evaluación del comportamiento del algoritmo propuesto en la estimación de los parámetros de modelos de redes neuronales artificiales.

REFERENCIAS

- [1] P. M. Pardalos, y M. G. C. Resende, ed, "Handbook of Applied Optimization," New York: Oxford University Press, 2002, 1077 p.
- [2] M. Bazaraa *et al.*, *Nonlinear Programming: Theory & Applications*, Hoboken, NJ: Wiley, 1994, 872 p.
- [3] L. T. Biegler, y I. E. Grossmann, "Retrospective on Optimization," *Computers & Chemical Engineering*, vol. 28, no. 8, pp. 1169-1192, 2004.
- [4] J. Bracken, y G. P. McCormick, *Selected Applications of Nonlinear Programming*, New York: Jhon Wiley & Sons, Inc, 1968, 122 p.
- [5] H. A. Bremermann, "A method for unconstrained global optimization," *Mathematical Biosciences*, no. 1970, pp. 1-15, 1970.
- [6] S. Kirkpatrick *et al.*, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, 1983.
- [7] W. L. Goffe *et al.*, "Global Optimization of Statistical Functions with Simulated Annealing," *Journal of Econometrics*, vol. 60, pp. 65-99, 1994.
- [8] C. Hamzacebi, y F. Kutay, "Continuous functions minimization by dynamic random search technique," *Applied Mathematical Modelling*, vol. 31, pp. 2189-2198, 2007.
- [9] A. R. Hedar, y M. Fukushima, "Tabu search directed by direct search methods for nonlinear global optimization," *European Journal of Operations Research*, vol. 170, no. 2, pp. 329-349, 2006.
- [10] T. Bäck, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*, New York: Oxford University Press, 1996, 328 p.
- [11] C. A. Coello Coello, "Constraint-handling using an evolutionary multiobjective optimization technique," *Civil Engineering and Environmental Systems*, vol. 17, no. 4, pp. 319-346, 2000.
- [12] M. Mandischer, "Comparison of evolution strategies and backpropagation for neural network training," *Neurocomputing*, vol. 42, no. 1-4, pp. 87-117, 2002.
- [13] D. Ortiz *et al.*, "Una Comparación entre Estrategias Evolutivas y RPROP para la Estimación de Redes Neuronales," *Avances en Sistemas e Informatica*, vol. 4, no. 2, pp. 135-144, 2007.
- [14] C. A. Coello *et al.*, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," *Advanced Engineering Informatics*, vol. 16, no. 3, pp. 193-203, 2002.
- [15] Z. Michalewicz, y M. Schoenauer, "Evolutionary Algorithms for Constrained Parameter Optimization Problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32, 1996.
- [16] F. Hoffmeister, y J. Sprave, "Problem-independent handling of constraints by use of metric penalty functions,"

- en Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP'96), San Diego: pp. 289-294, 1996.
- [17] I. Rechenberg, *Cybernetic solution path of an experimental problem*, Reporte Technical Report 1122, Royal Aircraft Establishment, Library translation, Farnborough, Hants., UK, 1965.
- [18] I. Rechenberg, *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Stuttgart: Frommann-Holzboog, 1973, p.
- [19] H.-P. Schwefel, "Kybernetische Evolution als Strategie der experimentellen Forschung in der Stromungstechnik," tesis de Ingeniero, Technische University Berlin, Berlin, 1965.
- [20] H.-P. Schwefel, "Evolutionsstrategie und numerische Optimierung," tesis de Doctorado, Technische Universität Berlin, Berlin, 1975.
- [21] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest and Least value of a Function," *Computer Journal*, vol. 3, pp. 175-184, 1960.
- [22] A. R. Colville, *A comparative study on nonlinear programming codes*, Reporte 320-2949, IBM New York Science Center, New York, 1968.