

Optimización basados en caos para la búsqueda de objetivos

Karla I. Fernandez-Ramirez, Arturo Baltazar

CINVESTAV Unidad-Salttillo, Coahuila, México
k.ivonne.fernandez.rmz@gmail.com,
arturo.baltazar@cinvestav.edu.mx

Resumen. En este trabajo se discute un algoritmo propuesto para la búsqueda óptima de un objetivo basado en la teoría del caos. La propuesta se basa en dos etapas; la primera etapa realiza una búsqueda en toda el área de trabajo mediante secuencias caóticas para encontrar el mayor número posible de mínimos o máximos locales. Existen diferentes sistemas que generan secuencias caóticas, en este trabajo se utilizaron los mapas: logístico y de Kent. Los resultados obtenidos muestran un mejor desempeño con el mapa de Kent. La segunda etapa utiliza el último valor óptimo detectado en la primera etapa como punto inicial para comenzar la búsqueda del óptimo global utilizando un camino caótico. El algoritmo fue probado en diferentes funciones conteniendo múltiples mínimos locales. Se realizó una combinación de secuencias aleatorias y caóticas, observando que el mejor desempeño es utilizando un camino caótico en la segunda etapa. Los resultados muestran que el esquema propuesto puede ubicar correctamente la posición del óptimo global con un tiempo de cálculo menor que los otros métodos de optimización estudiados.

Palabras clave: optimización, caos, búsqueda.

On Chaos Optimization for Target Search

Abstract. In this paper we discuss a proposed algorithm for the optimal target search based on chaos theory. The proposal is based on two stages; the first stage performs a search in the entire work area using chaotic sequences to find as many minimums or maximums local as possible. There are different systems that generate chaotic sequences, in this work the maps were used: logistic and Kent. The obtained results show a better performance with the Kent map. The second stage uses the last optimum value detected in the first stage as an initial point to begin the search for the global optimum using a chaotic sequence. The algorithm was tested in different functions containing multiple local minimums. A combination of random and chaotic sequences was performed, observing that the best performance is using a chaotic path in the second stage. The results show that the proposed scheme can correctly position the global optimum position with a short calculation time than other deterministic, stochastic and chaotic optimization methods.

Keywords: optimization, chaos, search.

1. Introducción

La teoría de cómo encontrar una estrategia para la búsqueda de un objetivo o una fuente de perturbación, fija o en movimiento, ha sido estudiada por científicos durante más de 50 años. En [1], se presenta la teoría de búsqueda desarrollada durante la Segunda Guerra Mundial para la Marina de los Estados Unidos. En esa teoría, el escaneado sistemático se utiliza comúnmente como una técnica de búsqueda [2]. Se han encontrado aplicaciones manuales o automáticas de esta técnica, por ejemplo, en astronomía, en la industria, medicina, biología y exploración minera.

Se puede ver que, si la resolución del paso de búsqueda es lo suficientemente pequeña y se conoce el espacio de trabajo, es probable que se alcance una búsqueda completa del área de trabajo. Sin embargo, tal esquema puede consumir mucho tiempo y potencia computacional. El problema de buscar un objetivo estacionario o no estacionario por un robot móvil también ha ganado considerable atención por parte de la comunidad científica para su posible aplicación en tareas de búsqueda como en operaciones militares, de vigilancia o de rescate [3]. La dinámica del Caos ha sido recientemente reportada como una herramienta para explorar zonas de vigilancia con una cobertura completa y rápida del espacio de trabajo [4] utilizando sistemas robóticos. El problema de búsqueda puede interpretarse como un problema de optimización en el que se debe encontrar el máximo o mínimo en un mapa de distribución.

Los algoritmos de optimización basados en caos (COA, por sus siglas en inglés) se basan en dos etapas, la primera se utiliza secuencias caóticas para producir una variable de búsqueda global. La segunda etapa realiza una búsqueda refinada considerando que la localización inicial de búsqueda se encuentra cerca del óptimo global. El COA se ha estudiado utilizando secuencias numéricas generadas por un mapa caótico [5]. En [6] demostraron que las propiedades estadísticas de los mapas caóticos afectan la eficiencia de los algoritmos en la búsqueda global, mientras que en [7] muestran que el éxito del algoritmo depende de la segunda etapa.

En este trabajo, desarrollamos un esquema de búsqueda de objetivos que utiliza la teoría del caos para su optimización. Los objetivos son los siguientes; primero se estudiará el efecto de utilizar una distribución caótica y una distribución aleatoria uniformemente distribuida en la primera etapa; segundo, se propone una modificación de la segunda etapa del COA, utilizando el camino caótico utilizado en [8]. El algoritmo propuesto se probará numéricamente usando funciones analíticas con más de un mínimo local. El objetivo es estudiar la robustez y rapidez del algoritmo propuesto para localizar el óptimo global.

2. Optimización aleatoria y caótica

Los algoritmos de optimización estocástica no dependen de una propiedad matemática estricta, como continuidad, diferenciabilidad o una descripción matemática exacta de las funciones objetivo y las condiciones de restricción, y por lo tanto sus procedimientos de optimización son fáciles de implementar [6]. Sin embargo, estos

algoritmos generalmente muestran una convergencia prematura y una capacidad de explotación débil, lo que lleva a un óptimo local en lugar de un óptimo global y, por lo tanto, una convergencia lenta. Alternativamente, el caos se ha propuesto como un algoritmo de optimización debido a sus propiedades topológicas y estadísticas. El caos ocurre en muchos sistemas no lineales, donde su evolución en el tiempo es impredecible. Para clasificar el comportamiento de un sistema como caótico, debe tener ciertas propiedades: sensibilidad a las condiciones iniciales, pseudoaleatoriedad y ergodicidad que implica que una secuencia caótica puede visitar todo el estado del atractor extraño [9]. Estas propiedades que se encuentran en el algoritmo de optimización del caos podrían tener un mejor rendimiento que los algoritmos estocásticos [10].

La Fig. 1 muestra la secuencia de una variable aleatoria uniformemente distribuida y su respectivo histograma para los primeros 500 valores. Los resultados de esta pequeña muestra evidencian la naturaleza de esta distribución.

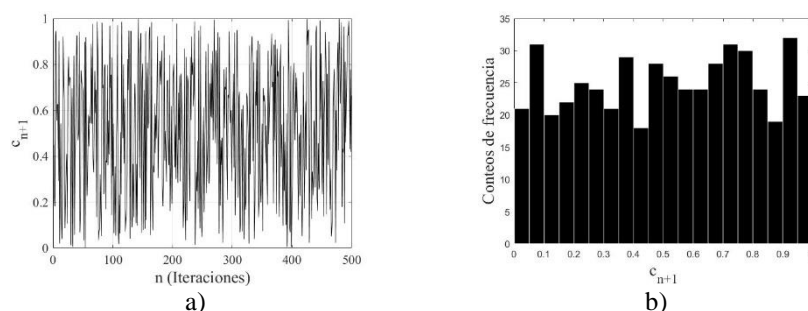


Fig. 1. a) Distribución aleatoria uniforme; b) histograma de la secuencia aleatoria.

Un mapa de una dimensión (1D) que puede generar una secuencia caótica es el mapa logístico. Este es un mapa polinomial que puede conducir a fenómenos complicados de dinámica caótica y se puede representar como se muestra en la ecuación (1):

$$c_{n+1} = \mu c_n(1 - c_n), \quad 0 < \mu \leq 4, \quad c_n \in (0,1), \quad (1)$$

donde c_n es el n -ésimo número caótico, n representa el número de iteraciones y μ es el parámetro de control. El sistema (1) tiene diferentes características dinámicas para diferentes valores de μ . Cuando el valor de $\mu = 4$, se genera la secuencia caótica entre valores de (0,1). Los resultados de la simulación utilizando $\mu = 4$ se muestran en la Fig. 2.

La función de densidad de probabilidad (PDF) es una distribución de Chebyshev [11], donde el centro de la función muestra una distribución uniforme, como se puede observar en el histograma de la secuencia caótica (Fig. 2 b).

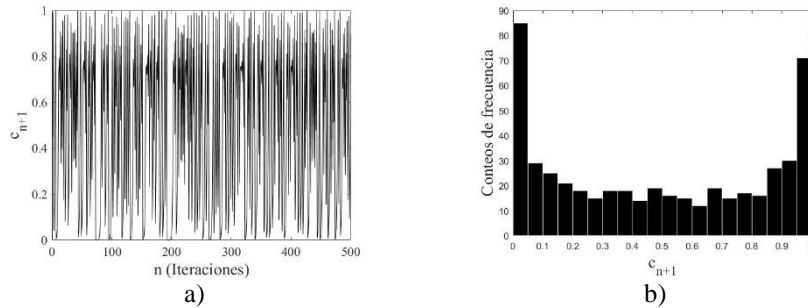


Fig. 2. a) Secuencias caóticas generadas por el mapa logístico con $\mu = 4$; b) histograma de la secuencia caótica.

Se ha reportado que el mapa logístico puede tener algunas limitaciones debido a su distribución no uniforme [11]. Una alternativa es el mapa de Kent, que se ha sido utilizado en aplicaciones como el cifrado seguro y tiene una distribución teórica uniforme PDF. El mapa definido en la ecuación (2):

$$c_{n+1} = \begin{cases} \frac{c_n}{m} & 0 < c_n \leq m \\ \frac{1-x_n}{1-m} & m < c_n \leq 1 \end{cases}, \quad (2)$$

donde m es un parámetro de control con un intervalo de $0 < m < 1$, y la secuencia está acotada entre $(0,1)$ [11]. La Fig. 3 muestra un ejemplo de la secuencias generadas cuando $m=0.8$, también se muestra el histograma correspondiente. La distribución obtenida exhibe un patrón que se asemeja a una distribución uniforme.

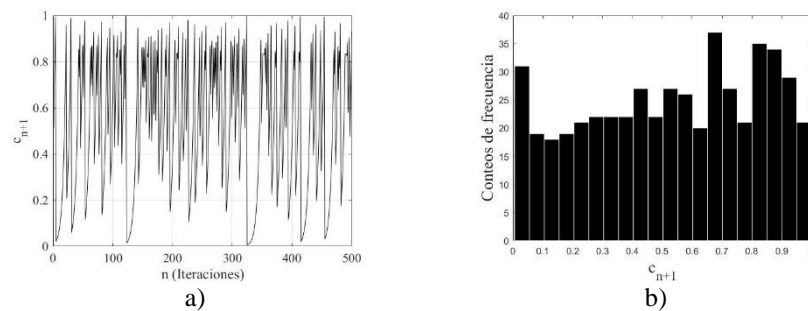


Fig. 3. a) Secuencia caótica generada por el mapa de Kent; b) histograma de la secuencia caótica.

Las secuencias caóticas en un espacio de trabajo en 2D de los dos mapas presentados se comparan con una distribución uniforme aleatoria. En las Fig. 4 y Fig. 5 muestran los resultados para 500 y 1000 iteraciones, respectivamente.

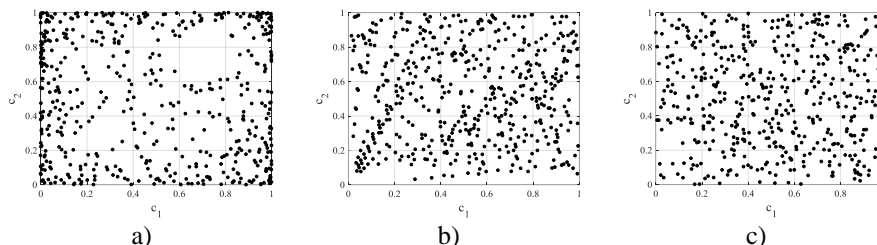


Fig. 4. Distribución en dos dimensiones con 500 iteraciones; a) mapa logístico; b) mapa de Kent; c) distribución uniforme aleatoria.

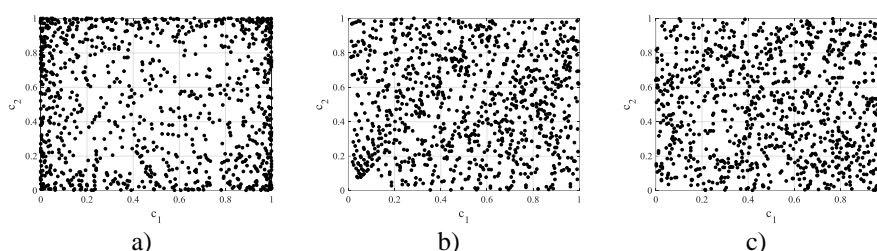


Fig. 5. Distribución en dos dimensiones con 1000 iteraciones; a) mapa logístico; b) mapa de Kent; c) distribución uniforme aleatoria.

Claramente, las distribuciones obtenidas usando mapas caóticos proporcionan una distribución casi uniforme similar a la distribución aleatoria uniforme con la ventaja de que, de acuerdo con las propiedades topológicas de la teoría del caos, todos los conjuntos abiertos en el mapa pueden ser visitados [11]. Más adelante, se probarán estos mapas con funciones seleccionadas para estudiar el desempeño para la búsqueda del máximo global.

3. Algoritmo de optimización propuesto basado en la teoría del caos

El algoritmo de optimización de caos propuesto se basa en dos etapas que se describe de manera general en la Fig. 6.

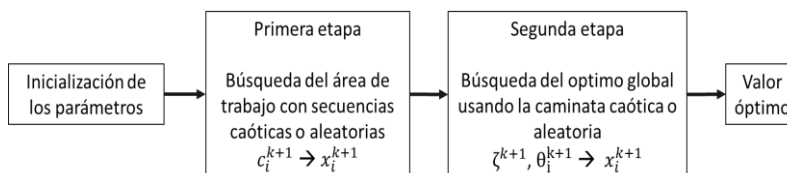


Fig. 6. Diagrama del algoritmo propuesto.

El algoritmo se describe a continuación:

1. Se define los parámetros de inicialización: número de iteraciones n , límite inferior L_i y superior U_i donde i representa las dimensiones de la función a optimizar. Se

define un valor óptimo $f^* = \infty$ (si se busca el mínimo global). Se establecen los valores iniciales c_i^k ($k = 1, \dots, n$) para generar las secuencias caóticas o aleatorias.

- Las variables caóticas c_i^k (acotadas entre (0,1)) se mapean en el rango de los límites establecidos (variables de optimización x_i^k) mediante la ecuación (3)

$$x_i^k = L_i + (U_i - L_i)c_i^k. \quad (3)$$

- La función es evaluada con las variables de optimización $f(\vec{x}^k)$. Si el valor de la función $f(\vec{x}^k) < f^*$ entonces $f^* = f(\vec{x}^k)$ y la solución óptima $\vec{x}^* = \vec{x}^k$. La siguiente variable es generada c_i^{k+1} hasta $k > n$. A continuación, se muestra el pseudocódigo de la primera etapa.

01	Inicialización $n, L_i, U_i, f^*, c_i^{k=1}$
02	Mientras $n \geq k$ entonces
03	$c_i^k = \mu c_i^{k-1} (1 - c_i^{k-1})$
04	$x_i^k = L_i + (U_i - L_i)c_i^k$
05	Si $f(\vec{x}^k) < f^*$ entonces
06	$f^* = f(\vec{x}^k)$
07	$\vec{x}^* = \vec{x}^k$
08	Fin
09	$k = k + 1$
10	Fin de la primera etapa

- La segunda etapa inicializa con los últimos valores óptimo encontrado en la primera etapa. Se define el número de iteraciones n y los valores iniciales $\theta_i^{k=1}$ y $\zeta^{k=1}$.
- En esta etapa se utiliza la ecuación de Pearson (4) definida por:

$$x_i^k = \sum_{k=1}^n \zeta^k e^{i\theta_i^k}, \quad (4)$$

donde x_i^k es la nueva posición, que se da como una suma de n vectores de dos dimensiones con un tamaño de paso (ζ^k) de la iteración k y orientación (θ_i^k). El valor de ζ^k y θ_i^k están dado por el mapa caótico. Los valores de θ_i^k son mapeados dentro del rango $(0, 2\pi)$ utilizando la ecuación (3).

- Se ajustan las variables dentro del rango, si $x_i^k < L_i$ entonces $x_i^k = L_i$; $x_i^k > U_i$ entonces $x_i^k = U_i$.
- La función es evaluada con las variables de optimización $f(\vec{x}^k)$. Si el valor de la función $f(\vec{x}^k) < f^*$ entonces $f^* = f(\vec{x}^k)$ y la solución óptima $\vec{x}^* = \vec{x}^k$. La siguiente variable es generada c_i^{k+1} hasta $k > n$.

```

01 Inicialización  $n, L_i, U_i, f^*, \theta_i^{k=1}, \zeta^{k=1}$ 
02 Mientras  $n \geq k$  entonces
03    $\theta_i^k = \mu \theta_i^{k-1} (1 - \theta_i^{k-1})$ 
04    $\zeta^k = \mu \zeta^{k-1} (1 - \zeta^{k-1})$ 
05    $x_i^k = \sum_{k=1}^N \zeta^k e^{i\theta_i^k}$ 
06   Si  $x_i^k < L_i$  entonces
07      $x_i^k = L_i$ 
08   Fin
09   Si  $x_i^k < U_i$  entonces
10      $x_i^k = U_i$ 
11   Fin
12   Si  $f(\vec{x}^k) < f^*$  entonces
13      $f^* = f(\vec{x}^k)$ 
14      $\vec{x}^* = \vec{x}^k$ 
15   Fin
16    $k = k + 1$ 
17 Fin de la segunda etapa

```

4. Resultados numéricos

Para probar la eficiencia y el comportamiento del algoritmo propuesto, se evaluó utilizando funciones conocidas: la función Rastrigin (Fig. 7 a)) y la función Rana (Fig. 7 b)). La función Rastrigin tiene varios mínimos locales y un mínimo global en la posición $\mathbf{x}^* = (\mathbf{0}, \mathbf{0})$, and $\mathbf{f}^* = \mathbf{0}$. La función Rana tiene varios mínimos locales y un mínimo global en el punto $\mathbf{x}^* = (11, 10)$, and $\mathbf{f}^* = -11$.

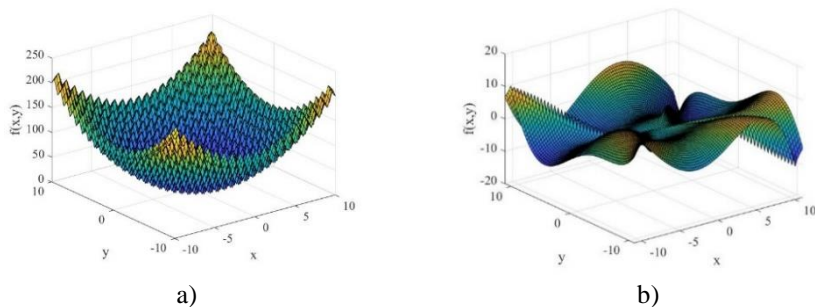


Fig. 7. a) Función Rastrigin $f(x, y) = (x^2 - 10 \cos(2\pi x) + 10) + (y^2 - 10 \cos(2\pi y) + 10)$;
 b) Función Rana $f(x, y) = x \sin \sqrt{|y + 1 - x|} \cos \sqrt{|y + 1 + x|} + (y + 1) \cos \sqrt{|y + 1 - x|} \sin \sqrt{|y + 1 + x|}$.

En la Fig. 8 muestra un ejemplo de las etapas del algoritmo propuesto en la función Rastrigin, utilizando 200 iteraciones en la primera etapa y 100 en la segunda. El valor óptimo encontrado fue 1.03, tomando un tiempo computacional de .0313 seg.

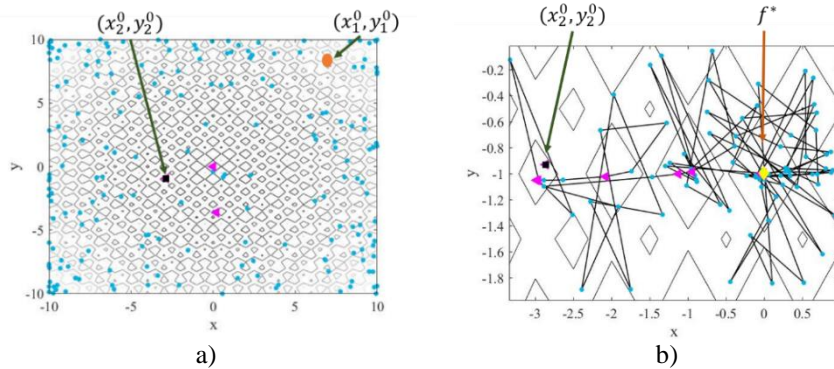


Fig. 8. Proyección de la función Rastrigin (plano x-y) donde (x_1^0, y_1^0) es el inicio de la primera etapa y (x_2^0, y_2^0) es el inicio de la segunda; a) Primera etapa; b) Segunda etapa.

La Tabla 1 muestra los resultados (tiempo computacional y valor óptimo) después de implementar el algoritmo propuesto. Los resultados muestran que los dos casos estudiados dan un valor óptimo general muy cercano al mínimo global verdadero de la función con solo una ligera mejoría para el caso 1.

Para comparar el algoritmo propuesto, se realizó una comparación con el algoritmo de gradiente descendiente y con el algoritmo de perturbación simultánea con aproximación estocástica (SPSA) partiendo de la misma posición y con el mismo número de iteraciones. Los resultados obtenidos muestran un mejor desempeño de estos dos algoritmos comparado con el caso tres del algoritmo propuesto, pero menor que los dos primeros casos. El algoritmo del gradiente puede mejorar el desempeño modificando el paso que se utilizó o usando un número mayor de iteraciones. El algoritmo del SPSA se basa en variables aleatorias, por lo que puede encontrar el valor óptimo incrementando el número de iteraciones.

Tabla 1. Resultados de la simulación de la función Rastrigin con el mapa logístico (N=300).

# Casos	Algoritmo		Valor óptimo	Tiempo (seg.)
	1er etapa	2da etapa		
1	Caos	Camino caótico	1.01	.0313
2	Aleatorio	Camino caótico	1.57	.0313
3	Caos	Camino aleatorio	9.03	.0156
4	Gradiente ($h=.025$)		6.60	.0312
5	SPSA		5.13	.0156

Los tres casos propuestos se corrieron 50 veces para adquirir las métricas estadísticas mostradas en la Tabla 2 y en la Tabla 3 para las funciones Rastrigin y Rana, respectivamente. En promedio, el algoritmo propuesto usando secuencias caóticas en las dos etapas da mejores resultados que cuando se combina con secuencias aleatorias.

En la Tabla 4 muestra los resultados de implementar el algoritmo propuesto en la función de Rastrigin utilizando el mapa de Kent para generar la secuencia caótica. Los resultados muestran que los dos casos estudiados dan un valor óptimo general muy

cercano al verdadero mínimo global de la función con solo una ligera mejoría para el caso 1.

Tabla 2. Resultados de simulación de la función Rastrigin con el mapa logístico.

# Casos	Valor óptimo			Tiempo computacional		
	Mejor	Peor	Promedio	Mejor	Peor	Promedio
1	4.03E-06	0.0009	0.0005	0.093	0.281	0.184
2	2.92E-06	2.16	0.5142	0.015	0.093	0.05
3	6.87E-06	0.0291	0.0043	0.046	0.187	0.092

Tabla 3. Resultados de simulación de la función Rana con el mapa logístico.

# Casos	Valor óptimo			Tiempo computacional		
	Mejor	Peor	Promedio	Mejor	Peor	Promedio
1	-11	-11	-11	.0156	.1093	.0409
2	-11	-11	-11	.0156	.125	.0596
3	-11	-10	-10	.0156	.125	.0415

Tabla 4. Resultados de la simulación de la función Rastrigin con el mapa Kent (N=300).

# Casos	Algoritmo		Valor óptimo	Tiempo (seg.)
	1er etapa	2da etapa		
1	Caos	Camino caótico	.0063	.1250
2	Aleatorio	Camino caótico	1.82	.0313
3	Caos	Camino aleatorio	2.12	.0313

5. Conclusiones

En el presente artículo, se propuso un algoritmo de optimización utilizando secuencias caóticas. El algoritmo se basa en dos etapas. En la primera etapa, se lleva a cabo una búsqueda de los posibles valores óptimos (máximos o mínimos). La segunda etapa comienza en el último óptimo encontrado durante la primera etapa y se realiza una búsqueda utilizando la caminata caótica.

La metodología propuesta se comparó con diferentes esquemas, combinando secuencias aleatorias y caóticas en las dos funciones. El mapa logístico y el mapa de Kent se usaron para generar las secuencias caóticas, que tienen diferentes distribuciones. Los resultados muestran que hay una ligera mejora en el tiempo computacional para el esquema propuesto. Los resultados presentados solo consideran funciones de dos dimensiones; sin embargo, como trabajo futuro se tiene contemplado atacar el problema en n -dimensiones y la implementación del esquema de optimización propuesto para la búsqueda de un objetivo.

Agradecimientos. Los autores agradecen al CONACYT por el apoyo económico para Karla I. Fernandez-Ramirez (estudiante de doctorado) y a través del proyecto CB-286907.

Referencias

1. Koopman, B.: Search and Screening: General principles with historical applications. Pergamon Press, New York (1980)
2. Gertner, I., Zeevi, Y. Y.: Scanning strategies for target detection. In: Proc. SPIE, vol. 1470, pp. 1–19. Data Structures and Target Classification (1991)
3. Benkoski, S., Monticino, M., Weisinger, J.: A survey of the search theory literature. Naval Research Logistic (39), pp. 469–494 (1991)
4. Martins-Filho, L. S., Macau, E. E.: Autonomous Robots and Agents, Springer Berlin Heidelberg, Berlín, Heidelberg (2007)
5. Bing, L., Weisun, J.: Chaos optimization method and its application. Control Theory and Application, pp. 613–615 (1997)
6. Yang, D., Liu, Z., Zhou, J.: Chaos optimization algorithms based on chaotic maps with different probability distribution and search speed for global optimization. Communications in Nonlinear Science and Numerical Simulation (19), pp. 1129–1246 (2014)
7. Velásquez, J. D.: An introduction to chaos-based algorithms for numerical optimization. Revista Avance en Sistema e Informática (8), pp. 51–60 (2011)
8. Baltazar, A., Fernandez-Ramirez, K.I., Aranda-Sanchez, J.I.: A study of chaotic searching for their application in an ultrasonic scanner. Engineering Applications of Artificial Intelligence, Unpublished (2018)
9. Yan, X. F., Chen, D. Z., Hu, S. X.: Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. Computers & Chemical Engineering (27), pp. 1393–1404 (2003)
10. Yang, D., Li, G., Cheng, G.: On the efficiency of chaos optimization algorithms for global optimization. Chaos, Solitons Fractals (34), pp. 1366–1375 (2007)
11. Baker, G., Gollud, J.: Chaotic dynamics: An introduction, 2nd edn. Cambridge University Press, New York (1996)