

Recibido 09 Feb. 2024

ReCIBE, Año 13 No. 1, May. 2024

Aceptado 13 Feb. 2024

Metropolis-Hastings (MH): Una Perspectiva Innovadora en la Inicialización de Poblaciones

Metropolis-Hastings (MH): An Innovative Perspective on Population Initialization

Oscar Francisco Barba Toscano^{1*}

oscar.barba7889@alumnos.udg.mx

Eric Ricardo Lopez Marin¹

Hector Joaquin Escobar Cuevas¹

Erik Valdemar Cuevas Jimenez¹

Miguel Angel Alejandro Islas Toski¹

¹Universidad de Guadalajara

*Autor para Correspondencia

Resumen

En este artículo, se propone un nuevo método de inicialización de poblaciones para algoritmos metaheurísticos. En este enfoque, el conjunto inicial de soluciones iniciales se obtiene a través del muestreo de la función objetivo aplicando la técnica de Metropolis-Hastings (MH). Bajo este método, el conjunto inicial de soluciones adopta un valor cercano a los valores prominentes de la función objetivo a optimizar. A diferencia de la mayoría de los métodos de inicialización que únicamente consideran una distribución espacial, en el método, los puntos iniciales representan regiones promisorias del espacio de búsqueda, las cuales merecen ser explotadas para identificar la solución óptima global de una manera más rápida. brindando al algoritmo una convergencia más rápida y mejorando la calidad de las soluciones obtenidas. Con el objetivo de demostrar el rendimiento del método de inicialización a algoritmos metaheurísticos, éste ha sido embebido en el algoritmo de Differential Evolution (DE) clásico, y el sistema completo ha sido puesto a prueba en un conjunto representativo de funciones de benchmark extraído de diferentes conjuntos de datos. Los resultados experimentales demuestran una mejora en la rapidez de convergencia y un incremento en la calidad de las soluciones por parte del enfoque propuesto, a comparación de otros métodos similares.

Palabras clave: Métodos de inicialización, algoritmos metaheurísticos, optimización

Abstract

In this article, a new population initialization method for metaheuristic algorithms is proposed. In this approach, the initial set of solutions is obtained by sampling the objective function using the Metropolis-Hastings (MH) technique. Under this method, the initial set of solutions adopts values close to the prominent values of the objective function to be optimized. Unlike most initialization methods that only consider a spatial distribution, this method's initial points represent promising regions of the search space, which deserve to be exploited to identify the global optimal solution more quickly. This provides the algorithm with faster convergence and improves the quality of the solutions obtained. To demonstrate the performance of the initialization method in metaheuristic algorithms, it has been embedded in the classic Differential Evolution (DE) algorithm, and the complete system has been tested on a representative set of benchmark functions extracted from different datasets. Experimental results show an improvement in convergence speed and an increase in the quality of the solutions by the proposed approach compared to other similar methods.

Keywords: Initialization methods, metaheuristic algorithms, optimization

1. Introducción

Los esquemas de optimización clásicos corresponden diferentes métodos para resolver problemas de optimización no lineales (X. Yang, 2010). No obstante, su uso se ve claramente limitado dada la necesidad de que la función objetivo presente varias restricciones, tales como que la función debe ser doblemente derivable y esta debe ser unimodal (Jahn Johannes, 2007). Los problemas de optimización reales formulan escenarios que sobrepasan estas restricciones. Alternativamente a los métodos clásicos, es posible resolver problemas de optimización a través de los métodos metaheurísticos.

Los algoritmos metaheurísticos comprenden optimizadores basados en poblaciones basadas en principios estocásticos, estos emplean un conjunto de agentes de búsqueda que funcionan como soluciones iniciales. Guiados por los valores de la función objetivo, las variables de decisión de las soluciones iniciales se adaptan aplicando un grupo particular de operadores basados en reglas heurísticas para generar patrones de búsqueda e identificar las mejores soluciones. En general, los algoritmos metaheurísticos obtienen mejores resultados que los obtenidos por los métodos clásicos en términos de rapidez, robustez y precisión (Cuevas & Rodríguez, 2020). Los algoritmos metaheurísticos no requieren detalles funcionales de la formulación de optimización para su operación. En su lugar, emplean reglas heurísticas para la generación de patrones de búsqueda. Tales reglas corresponden a modelos abstractos supuestamente extraídos de mecanismos naturales o sociales (Maciel C. et al., 2020).

Recientemente, se han propuesto diferentes métodos metaheurísticos en la literatura. Algunos ejemplos son el Genetic Algorithm (GA), presentado por Holland (Holland, 1984), Differential Evolution (DE) propuesto por Price y Storn (Storn & Price, 1997), Artificial Immune System (AIS) propuesto por De Castro et al. (de Castro & Timmis, n.d.), Evolutionary Algorithm (EA) propuesto por Fogel et al. (Fogel D., 2009), no obstante, se pueden encontrar otro tipo de algoritmos en la literatura, tales como aquellos que imitan procesos físicos y fenómenos del comportamiento animal, tales como el Gravitational Search Algorithm (GSA) propuesto por Rashedi et al. (Rashedi et al., 2011), Simulated Annealing (SA) propuesto por Kirkpatrick et al. (Kirkpatrick et al., 1983), Electromagnetism-like Algorithm propuesto por Ilker et al. (Birbil & Fang, 2003), Charged System Search (CSS) propuesto por Kaveh et al. (Kaveh & Talatahari, 2010), Social Spider Optimization (SSO) propuesto por Cuevas et al. (Cuevas et al., 2013), Particle Swarm Optimization (PSO) propuesto por Kennedy y Eberhart (Kennedy & Eberhart, n.d.), Artificial Bee Colony (ABC) propuesto por Karaboga (Karaboga, 2005), y Cuckoo Search (CS) propuesto por Yang y Deb (X.-S. Yang & Deb, 2010).

De entre todos los métodos metaheurísticos reportados en la literatura, el algoritmo de Differential Evolution (DE) originalmente publicado por Price y Storn (Storn & Price, 1997) representa uno de los algoritmos más populares para resolver problemas de optimización con un alto grado de complejidad. Unas de sus principales características son su simplicidad y su alta eficacia, resultando en una estrategia de búsqueda de fácil implementación y con una gran capacidad para optimizar problemas sobre la no linealidad y la multimodalidad, lo que ha inspirado a aplicarse a múltiples problemas de aplicaciones reales (Kononova et al., 2021; Ochoa et al., 2020, 2021). En las últimas dos décadas han surgido múltiples variantes con el objetivo de mejorar aún más el rendimiento del algoritmo original. Algunos de los ejemplos más notorios de estas variantes son el algoritmo JADE (Jingqiao Zhang & Sanderson, 2009) el cual utiliza un enfoque de mutación diferente denominado "current-to-best". Esta operación de mutación asocia la solución actual, el mejor elemento de la población y dos soluciones aleatorias para generar un vector mutante. Otros ejemplos interesantes son el algoritmo SADE (Qin & Suganthan, n.d.) y el jDE (Brest et al., 2006). En estos algoritmos, se modifican diferentes parámetros durante el proceso de optimización a partir de la evaluación de su rendimiento. Por otra parte, los algoritmos SHADE (Tanabe & Fukunaga, 2013) y el LSHADE (Tanabe & Fukunaga, 2014) incorporan una memoria histórica de los valores de los parámetros durante el proceso de búsqueda, la cual se emplea para generar nuevos parámetros para controlar la generación de individuos. A pesar de que las nuevas versiones del algoritmo DE presentan un mejor

rendimiento a comparación del algoritmo original, este se ha seleccionado como un método base para mantener la compatibilidad con otros estudios presentes en la literatura.

Aunque los enfoques metaheurísticos producen resultados interesantes, no hay un algoritmo ideal capaz de resolver competitivamente todos los problemas. En general, todos los métodos metaheurísticos presentan inconvenientes comunes tales como una convergencia prematura, y un tiempo computacional muy largo, particularmente cuando el espacio de búsqueda es bastante complicado de explorar. Mejorar la precisión y evitar la convergencia prematura son dos de las problemáticas más importantes en la comunidad de la optimización metaheurística. Por esta razón, se ha invertido una gran cantidad de investigaciones en estos temas. La mayoría de los trabajos se concentran en el desarrollo de nuevos operadores, ajuste de parámetros y la combinación de diferentes estrategias. La inicialización es una operación importante en los métodos metaheurísticos. Se refiere al proceso de generar el primer conjunto de soluciones iniciales. Si no se dispone de suficiente información acerca del espacio de búsqueda, la técnica más común para generar la población inicial es generar soluciones aleatorias dentro de un rango válido. Aunque una inicialización de soluciones iniciales adecuada puede mejorar la precisión, aumentar la rapidez de convergencia y evitar una convergencia prematura, se han conducido pocos esfuerzos en esta dirección.

Algunos de los pocos trabajos incluyen el método presentado por Maaranen et al. (Wen et al., 2016). Bajo este enfoque, la población inicial se produce a partir de una secuencia de números cuasi aleatorios, en lugar de solo números pseudoaleatorios. Otro trabajo interesante es el presentado por Rahnamayan et al. (Rahnamayan et al., 2007), donde se presenta un método de inicialización basado en la oposición, donde se consideran dos poblaciones diferentes, una generada de manera aleatoria, y la otra considerando el opuesto espacial para cada elemento de la primera población. Wang et al (Wang et al., 2009) presentaron una técnica de inicialización de población basada en una búsqueda de transformación del espacio. Similar a (Geyer, 1992), este método considera dos poblaciones iniciales. La primera es generada completamente de manera aleatoria, y la segunda corresponde a las posiciones de la primera población, pero evaluadas en un espacio transformado. Pan et al. (Pan et al., 2014) propusieron un método de inicialización que utiliza el concepto de la aleatoriedad adaptativa para producir soluciones iniciales uniformemente distribuidas dentro del espacio de búsqueda, sobre el cual, se añade una nueva solución a la población inicial solamente si no está muy cercano a cualquier otro individuo ya añadido a la población. Todos estos métodos de inicialización comparten un elemento en común. Estos intentan distribuir uniformemente las soluciones iniciales dentro del espacio de búsqueda con el objetivo de cubrirlo completamente. La dispersión de tales soluciones es meramente espacial sin considerar la calidad en términos de la función objetivo. Bajo estas condiciones, hasta la mejor distribución inicial de las soluciones producidas por estos enfoques no podría garantizar una mejora en la calidad de las soluciones generadas, ni un incremento de la rapidez de convergencia.

La generación de soluciones en la fase inicial de un algoritmo metaheurístico se asocia con el proceso de exploración. Su objetivo es seleccionar localidades que merezcan ser exploradas en lugar de otras que deben ser descartadas debido a su baja calidad. En general, los operadores de los métodos metaheurísticos tienden a mover las soluciones existentes hacia los mejores elementos. La mayoría de los métodos de inicialización producen soluciones en base a dos enfoques: Usar principios aleatorios o caóticos (Pan et al., 2014; Rahnamayan et al., 2007; Wen et al., 2016) o generar soluciones equidistantes en el espacio de búsqueda. Las técnicas de inicialización basadas en principios aleatorios o caóticos con frecuencia producen soluciones cercanas entre sí. Bajo tales condiciones, se mantienen grandes secciones del espacio de búsqueda sin explorar. Por otra parte, los enfoques que producen soluciones equidistantes tienden a generar una población muy dispersa. Estas soluciones iniciales seguidas son colocadas lejos de las regiones promisorias. Por consiguiente, el algoritmo metaheurístico puede requerir un mayor número de iteraciones para alcanzar el óptimo global, afectando negativamente las propiedades de convergencia. Bajo estas condiciones, una técnica de inicialización capaz de explorar adecuadamente el espacio de búsqueda presentando principalmente soluciones de alta calidad puede mejorar el desempeño de un algoritmo metaheurístico.

Por otra parte, el método Metropolis-Hastings (MH) (Chib & Greenberg, 1995) puede generar muestras a partir de una distribución de probabilidad compleja $P(x)$ usando una función $f(x)$ relacionada con $P(x)$. Este produce una secuencia de puntos muestrales tales que un gran valor de $f(x)$ corresponden a una alta probabilidad de que x sea seleccionada. Bajo este enfoque del MH, tras cada iteración, se selecciona un candidato para el muestreo del siguiente elemento del candidato inicial. Entonces, a través de un mecanismo probabilístico, la muestra inicial se acepta (es utilizada para la siguiente muestra) o rechaza (se descarta y se vuelve a utilizar la muestra actual). La aceptación o rechazo de la solución propuesta depende de la relación que mantienen los valores de la muestra actual y la propuesta en términos de la función $f(\cdot)$. Por ende, la idea principal del MH es producir un muestro frecuente de secciones de x con un valor $f(x)$ mayor, evitando el muestro de zonas con un valor $f(x)$ menor.

En este artículo, se propone un nuevo método de inicialización de poblaciones para algoritmos metaheurísticos. En el enfoque, el conjunto inicial de soluciones iniciales se obtiene a través del muestro de la función objetivo $J(x)$ aplicando la técnica de Metropolis-Hastings. Por ende, el conjunto inicial de soluciones obtenidas por el algoritmo MH representa los puntos muestrales x con un alto valor de $J(x)$. A diferencia de la mayoría de los métodos de inicialización que consideran principios aleatorios o una distribución espacial, en el algoritmo, los puntos iniciales representan las regiones promisorias del espacio de búsqueda, las cuales merecen ser explotadas para identificar el óptimo global. Por ende, dado que las regiones con soluciones potenciales ya han sido identificadas, se requerirá un menor número de iteraciones para alcanzar el óptimo global. Bajo estas condiciones, la convergencia del algoritmo se acelera a la vez que las soluciones de mejor calidad son identificadas. Con el objetivo de demostrar las capacidades de este método de inicialización, se ha embebido en el algoritmo de DE clásico. Para evaluar el desempeño, el sistema completo ha sido probado en un conjunto representativo de funciones de benchmark extraído de diferentes conjuntos de datos, tales como el CEC 2017. Los resultados experimentales demuestran una mejora en la rapidez de convergencia y un incremento en la calidad de las soluciones por parte del enfoque propuesto, a comparación de otros métodos similares.

Este artículo está organizado de la siguiente manera: En la sección 2, se describe el método Metropolis-Hastings; En la sección 3 se presentan las características principales del algoritmo de DE. En la sección 4 se introduce al método de inicialización propuesto. En la sección 5 se presentan los resultados experimentales y un análisis comparativo, y en la sección 6, se discuten las conclusiones.

2. Método Metropolis-Hastings (MH)

El algoritmo Metropolis-Hastings (MH) (Chib & Greenberg, 1995) corresponde a un enfoque Monte Carlo de Cadenas de Márkov (Monte Carlo Markov Chain, o MCMC) para generar un conjunto de elementos aleatorios a partir de una distribución $f(x)$ la cual es complicado muestrear de manera directa. El método MH es comúnmente empleado para muestrear distribuciones multidimensionales, particularmente si es de un número grande de dimensiones. Bajo el esquema de MH, en cada iteración, se selecciona un punto candidato para el siguiente elemento muestreado a partir del punto muestral actual. Posteriormente, a través de un mecanismo probabilístico, la solución propuesta puede ser aceptada (usada como la siguiente muestra) o rechazada (descartada, y la muestra actual se vuelve a utilizar). La aceptación o rechazo de la solución propuesta depende de la relación que mantienen los valores de la muestra actual y la solución propuesta en términos de la función $f(\cdot)$. Asumiendo que se requiere muestrear a partir de una distribución de probabilidad $f(x)$ donde x corresponde a un valor en el espacio de d dimensiones. Entonces, una transición de las Cadenas de Márkov del estado x^k en el instante k al nuevo estado x^{k+1} se implementa en dos fases: producción de una solución propuesta y probabilidad de aceptación. Una solución propuesta y se genera a partir de una densidad de transición. Por lo general, y es generada como una perturbación del estado actual x^k con un valor aleatorio producido por una distribución normal $N(0, 1)$. Una vez que la solución propuesta y se ha generado, se usa un mecanismo probabilístico para decidir si y se acepta como el nuevo estado x^{k+1} . La probabilidad de aceptación α de la solución propuesta y

depende de la relación que mantienen los valores de la $f(\mathbf{x}^k)$ actual y de la $f(\mathbf{y})$ candidata en términos de la función $f(\cdot)$. Por ende, la probabilidad α de que \mathbf{y} sea aceptada como el nuevo estado \mathbf{x}^{k+1} se define de acuerdo con la Ecuación (1).

$$\alpha(\mathbf{x}^k, \mathbf{y}) = \min \left\{ \frac{f(\mathbf{y})}{f(\mathbf{x}^k)}, 1 \right\} \quad (1)$$

Bajo esta regla probabilística, se genera un número aleatorio r con una distribución uniforme $U(0, 1)$. Si se cumple la condición $r < \alpha$, la solución propuesta \mathbf{y} se acepta como el nuevo estado \mathbf{x}^{k+1} (o la muestra válida). En caso contrario, \mathbf{y} se descarta y se conserva \mathbf{x}^k como el nuevo estado \mathbf{x}^{k+1} .

En MH, un punto \mathbf{x} que presente un alto valor de $f(\mathbf{x})$ tiene una alta probabilidad de ser seleccionada. Por ende, en su proceso de muestreo, se produce una secuencia de puntos donde los elementos con un valor significativo en términos de $f(\mathbf{x})$ se muestrean con mayor frecuencia. Por ende, la idea principal del MH es producir un muestreo frecuente de secciones de \mathbf{x} con un valor altamente significativo de $f(\mathbf{x})$, evitando el muestreo de áreas con un valor bajo de $f(\mathbf{x})$. Una explicación teórica y demostración de este efecto pueden encontrarse en (Chaveau & Vandekerkhove, 2002)

Para visualizar los resultados del muestreo del método MH, se presenta un ejemplo numérico. En el ejemplo, se muestrea por el enfoque MH la función $f(x)$ definida en la Ecuación (2).

$$f(x) = \frac{1}{0.4\sqrt{2\pi}} e^{-\frac{x^2}{0.32}} + \frac{1}{0.2\sqrt{2\pi}} e^{-\frac{(x+2.5)^2}{0.08}} + \frac{1}{0.2\sqrt{2\pi}} e^{-\frac{(x-2.5)^2}{0.08}} \quad (2)$$

Considerando el intervalo $-4 \leq x \leq 4$, la función $f(x)$ tiene dos máximos globales ubicados en $x = -2.5$ y $x = 2.5$. De la misma forma, $f(x)$ solo un óptimo local en $x = 0$. La Figura 1 se muestran los histogramas de las muestras producidas por el método MH extraídos de la distribución objetivo $f(x)$. Estos resultados consideran histogramas de 50 bins para (a) 10, (b) 100, (c) 1000 y (d) 10000 muestras. De la Figura , es claro que, aunque el número de muestras solo es de 10 (Figura 1(a)), algunas muestras de $f(x)$ están asociadas con los valores máximos. Mientras que el número de muestras obtenidas por el MH aumenta, las muestras asociadas con los valores máximos de $f(x)$ son seleccionadas con mayor frecuencia, como se puede ver en las Figuras 1c, 1d, y 1e.

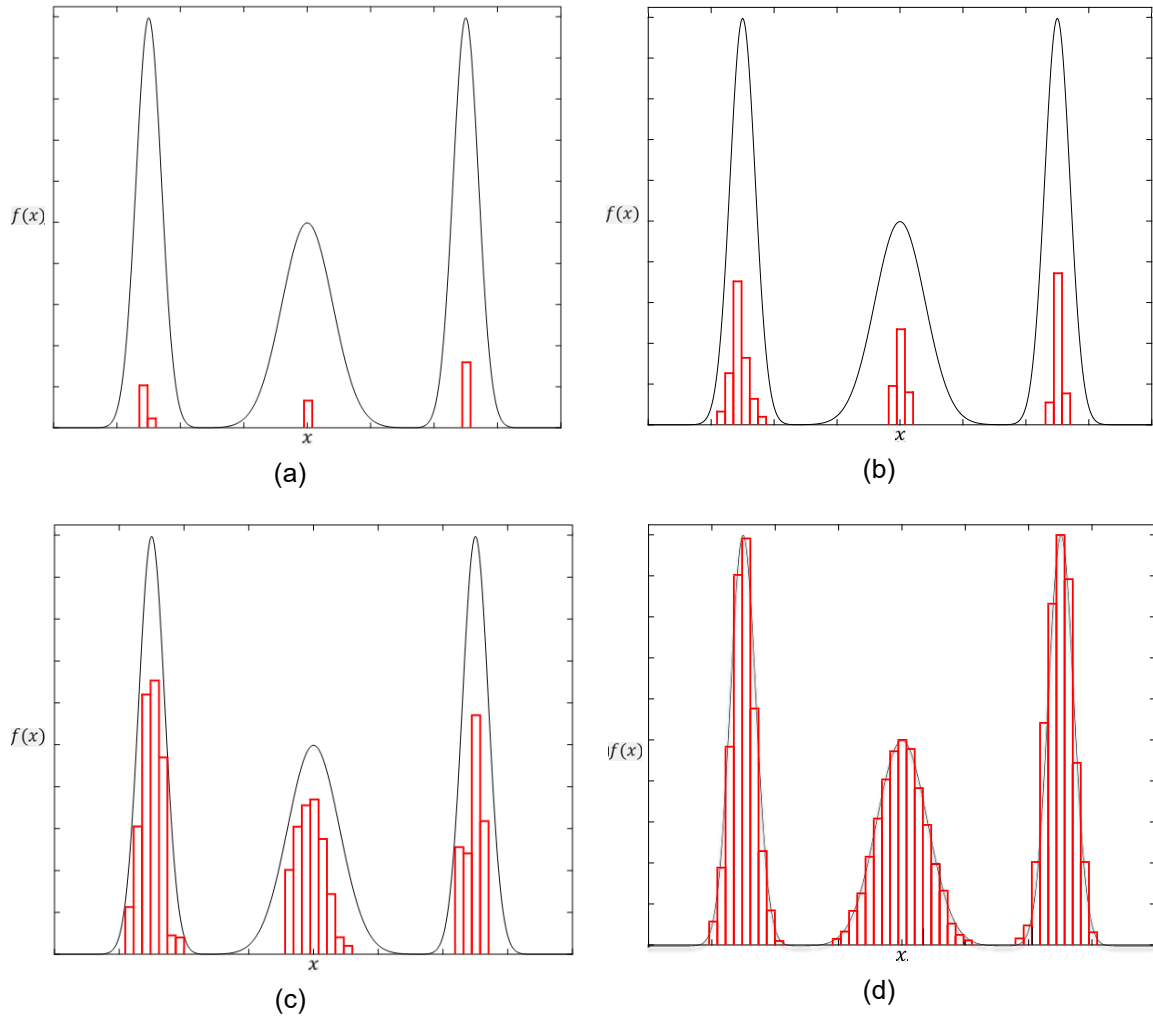


Figura 1 Histogramas de muestras producidas por el método MH extraído de la distribución objetivo $f(x)$ de la Ecuación ~~¡Error! No se encuentra el origen de la referencia.~~. Para (a) 10, (b) 100, (c) 1000, y (d) 10000 muestras

3. Algoritmo DE clásico

El algoritmo de Differential Evolution (DE) propuesto por Price y Storn (Storn & Price, 1997), es un enfoque de optimización implementado para resolver problemas complejos. El DE se ha convertido en uno de los algoritmos más usados en la literatura debido a su estructura simple, de fácil implementación y bajo costo computacional. Varios estudios y análisis comprueban que el DE presenta un mejor rendimiento y buenas propiedades de convergencia a comparación de otros métodos metaheurísticos. El algoritmo DE se ha ideado para identificar la solución global de una formulación de optimización no lineal formulada de acuerdo con la Ecuación (3).

$$\begin{aligned}
 & \text{Minimizar } J(\mathbf{x}), \mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d & (3) \\
 & \text{Sujeto a } \mathbf{x} \in \mathbf{X}
 \end{aligned}$$

Donde $J(\mathbf{x}): \mathbb{R}^d \rightarrow \mathbb{R}$ representa la función objetivo a optimizar, mientras que $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^d | P_{j,lb} \leq x_j \leq P_{j,ub}\}$ es un espacio de búsqueda limitado restringido por los límites inferior ($P_{j,lb}$)

y superior ($P_{j,ub}$). El DE es un algoritmo poblacional que considera una estrategia de búsqueda estocástica para la optimización global. La estructura de un DE consiste en 4 operaciones: Inicialización, mutación, cruce y selección

3.1 Inicialización

El algoritmo DE inicia definiendo el espacio de búsqueda del problema para obtener los valores de los límites inferior ($P_{j,lb}$) y superior ($P_{j,ub}$) de cada j -ésima variable de decisión. Entonces, una población inicial de n agentes es generada. Cada agente i es un vector d -dimensional donde a cada variable de decisión se le asigna un número aleatorio uniformemente distribuido $U(0, 1)$ entre los límites superiores e inferiores de acuerdo con la Ecuación (4):

$$x_{i,j,t} = P_{j,lb} + U(0, 1) \cdot (P_{j,ub} - P_{j,lb}), \quad (4)$$

donde $j = 1, 2, \dots, d$, $i = 1, 2, \dots, n$ y t representa la generación actual y $x_{i,j,t}$ representa el valor para la j -ésima variable de decisión del vector agente \mathbf{x}_i en la generación t .

3.2 Mutación

La primera operación en el DE es la mutación. Bajo este proceso, una nueva solución $\mathbf{V}_{i,t}$ conocida como vector mutante se genera como la combinación de diferentes soluciones de la población. Existen diferentes operadores de mutación dependiendo del modo en el que se asocian las soluciones. Los operadores de mutación más populares se definen de acuerdo con las Ecuaciones (5), (6), (7) y (8).

$$\text{DE/rand/1: } \mathbf{V}_{i,t} = \mathbf{x}_{r_1,t} + F \cdot (\mathbf{x}_{r_2,t} - \mathbf{x}_{r_3,t}) \quad (5)$$

$$\text{DE/rand/2: } \mathbf{V}_{i,t} = \mathbf{x}_{r_1,t} + F \cdot (\mathbf{x}_{r_2,t} - \mathbf{x}_{r_3,t}) + (\mathbf{x}_{r_4,t} - \mathbf{x}_{r_5,t}) \quad (6)$$

$$\text{DE/best/1: } \mathbf{V}_{i,t} = \mathbf{x}_{Best,t} + F \cdot (\mathbf{x}_{r_1,t} - \mathbf{x}_{r_2,t}) \quad (7)$$

$$\text{DE/current-to-best/1: } \mathbf{V}_{i,t} = \mathbf{x}_{i,t} + F \cdot (\mathbf{x}_{Best,t} - \mathbf{x}_{i,t}) + F \cdot (\mathbf{x}_{r_1,t} - \mathbf{x}_{r_2,t}) \quad (8)$$

Donde $\mathbf{x}_{r_1,t}$, $\mathbf{x}_{r_2,t}$, $\mathbf{x}_{r_3,t}$, $\mathbf{x}_{r_4,t}$ y $\mathbf{x}_{r_5,t}$ son diferentes soluciones ($\mathbf{x}_{r_1} \neq \mathbf{x}_{r_2} \neq \mathbf{x}_{r_3} \neq \mathbf{x}_{r_4} \neq \mathbf{x}_{r_5}$) seleccionadas aleatoriamente de la población actual ($\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5} \in \{1, \dots, n\}$), $\mathbf{x}_{Best,t}$ simboliza el mejor vector de solución para toda la población, mientras que F corresponde a un vector de escalamiento de mutación el cual es un número real positivo, comúnmente entre 0 y 1.

3.3 Cruce

Una vez que se ha producido el vector $\mathbf{v}_{i,t}$, algunos de sus parámetros son combinados (cruzados) con el i -ésimo vector de la población $\mathbf{x}_{i,t}$ para obtener un vector de prueba $\mathbf{U}_{i,t}$ como se muestra en la Ecuación (9).

$$\mathbf{U}_{i,j,t} = \begin{cases} \mathbf{V}_{i,j,t}, & \text{if } \text{rand}(0,1) \leq Cr \text{ or } j = j_{rand} \\ \mathbf{x}_{i,j,t}, & \text{otherwise} \end{cases} \quad (9)$$

Donde $j_{rand} \in \{1, 2, \dots, d\}$. Cr simboliza la constante de cruce que asume un valor dentro del intervalo $0 \leq Cr \leq 1$. Bajo estas condiciones, cada elemento del vector de prueba $\mathbf{U}_{i,j,t}$ se produce por medio de aceptación del valor de $\mathbf{V}_{i,j,t}$, si la regla probabilística es verdadera, o asumiendo el valor de $\mathbf{x}_{i,j,t}$ (en caso contrario).

3.4 Selección

Finalmente, el DE considera un mecanismo de selección para elegir las soluciones con mejor calidad de acuerdo a lo descrito en la Ecuación 10. Con este mecanismo, se compara el vector de prueba $\mathbf{U}_{i,t}$ y la solución original $\mathbf{x}_{i,t}$. De este modo, si la función de costo del vector de prueba $J(\mathbf{U}_{i,t})$ es mejor que el vector objetivo $J(\mathbf{x}_{i,t})$, entonces el vector de prueba se conserva como una solución para la siguiente población. De otro modo, el vector objetivo se preserva. Este proceso se repite hasta que se alcanza un criterio de paro, como se muestra en la Ecuación (10).

$$\mathbf{x}_{i,t+1} = \begin{cases} \mathbf{U}_{i,t}, & \text{if } J(\mathbf{U}_{i,t}) < J(\mathbf{x}_{i,t}) \\ \mathbf{x}_{i,t}, & \text{otherwise} \end{cases} \quad (10)$$

4. Algoritmo de inicialización de población propuesto

En la mayoría de los problemas de optimización, no se ofrece información a priori sobre la función objetivo (calidad de soluciones). Por ende, esta información no puede ser usada para producir soluciones iniciales cercanas a los valores óptimos para que el algoritmo metaheurístico opere de manera más eficiente. Por esto, en la ausencia de información, la población inicial se produce comúnmente considerando posiciones aleatorias distribuidas en el espacio de búsqueda del problema. En el enfoque, el método de Metropolis-Hastings (MH) se ha usado para producir el conjunto inicial de soluciones de partida. Bajo este método, el conjunto inicial de soluciones adopta un valor cercano a los valores prominentes de la función objetivo $J(\mathbf{x})$.

En el **¡Error! No se encuentra el origen de la referencia.**, se muestra la implementación del enfoque de inicialización en pseudocódigo. El algoritmo usa como entrada el número de elementos n para la población inicial y los límites del espacio de búsqueda \mathbf{X} . El método inicia configurando el conjunto de soluciones iniciales como vacío $\{\emptyset\}$ y la variable de índice $i = 1$. Luego, la primera semilla \mathbf{x}_0 para aplicar la técnica MH se genera como un punto aleatorio dentro del espacio de búsqueda \mathbf{X} . Posteriormente, comienza el proceso para producir las n soluciones iniciales. En este proceso, por principio, se produce una solución propuesta y a través de la modificación de la solución candidata anterior \mathbf{x}_{i-1} con un número aleatorio normalmente distribuido $N(0,1)$. Luego, se calcula la probabilidad de aceptación α comparando las calidades de \mathbf{y} y de \mathbf{x}_{i-1} en términos de la función objetivo $J(\cdot)$. Después, un número aleatorio r uniformemente distribuido es generado. Si $r < \alpha$, entonces la nueva solución propuesta \mathbf{y} se acepta como la nueva posición inicial \mathbf{x}_i . Bajo tales condiciones, \mathbf{x}_i es añadida al conjunto de la población inicial $I = \{I \cup \mathbf{x}_i\}$ y el índice i también se incrementa. Este proceso se repite hasta que todos los n puntos iniciales se hayan generado.

-
1. Entrada n, X
 2. $I = \{\emptyset\}, i = 1$
 3. Generar una semilla de solución inicial $x_0 \in X$
 4. **While** ($i \leq n$)
 5. $y = x_{i-1} + N(0, 1)$
 6. $\alpha = \min \left\{ \frac{J(y)}{J(x_{i-1})}, 1 \right\}$
 7. Generar un número aleatorio uniforme $r \in U[0, 1]$
 8. Si ($r < \alpha$), entonces
 9. $x_i = y$
 10. $I = \{I \cup x_i\}$
 11. $i = i + 1$
 12. Fin
 13. Fin
 14. Salida I
-

Algoritmo 1 Método de inicialización Metropolis-Hastings

Con el objetivo de demostrar las capacidades del método de inicialización, este ha sido embebido en el algoritmo DE clásico.

5. Resultados experimentales

En orden para evaluar los resultados de la metodología propuesta, se han conducido un conjunto de diferentes experimentos. Los resultados experimentales han sido comparados contra los producidos por otros enfoques de inicialización, tales como la inicialización aleatoria (RI) (Fogel D., 2009; Wen et al., 2016), el basado en la oposición (OP) (Rahnamayan et al., 2007), una generalización del basado en la oposición (GOP) (Pan et al., 2014), y la aleatoriedad adaptativa (AR) (Pan et al., 2014). Estas técnicas son consideradas como algunas de los métodos de inicialización más importante en la literatura especializada. En aras de la equidad, en las comparativa, todos los métodos de inicialización han sido embebidos en un algoritmo clásico de DE descrito en la Sección 3. El DE clásico ha sido configurado con los siguientes parámetros: Se emplea el esquema de mutación DE/best/1. El tamaño de la población ha sido configurado en 50. Los ajustes de los parámetros siguen las instrucciones de su propia referencia. La probabilidad de cruce Cr es de 0.9 y el parámetro de escalamiento F es de 0.8.

La sección experimental está descrita en las siguientes subsecciones. En la primera subsección, se evalúan todos los enfoques en términos de la calidad de soluciones. En la segunda subsección, se evalúa la rapidez de convergencia y la robustez, y en la tercera subsección se discuten las gráficas de convergencia producidas.

5.1 Comparativa en términos de calidad de soluciones

En esta subsección, se evalúan los resultados considerando 20 funciones de benchmark ($f_1(x) - f_{20}(x)$), entre las que se encuentran 11 funciones multimodales, cinco funciones unimodales y 4 funciones híbridas. Las funciones multimodales presentan superficies de optimización que involucran múltiples óptimos locales; Por esta razón, estas funciones presentan mayores complicaciones para localizar sus soluciones. En este análisis, se compara el desempeño de los enfoques de inicialización en términos de la precisión presentada por las soluciones producidas. El análisis exhibe el desempeño de cada técnica para localizar el óptimo global cuando la función contiene diferentes óptimos locales. En esta experimentación, se asume que las funciones objetivo son operadas en 30 dimensiones. Para evaluar los resultados, se consideran tres indicadores diferentes: El promedio de las mejores soluciones hasta el momento (Average Best-so-far, o AB), la mediana de las mejores soluciones (Median Best-so-far, o MD) y la desviación estándar de las

mejores soluciones (SD). En el análisis, cada problema de optimización se resuelve usando cada algoritmo 30 veces. Esta experimentación produce 30 soluciones; La media de las mejores soluciones representa el Average Best-so-far (AB). De la misma manera, la mediana de los 30 resultados se calcula para generar la Median Best-so-far (MD), y la desviación estándar de los 30 datos se estima para obtener la SD de las mejores soluciones. Los indicadores AB y MD corresponden a la precisión de las soluciones, mientras que el indicador SD representa la dispersión, y, por consiguiente, la robustez del algoritmo.

Function		DE_RI	DE_OP	DE_GOP	DE_AR	DE_MH
$f(x)_1$	AB	0.031043522	3.96821E-08	5.70285E-08	0.031043535	6.43886E-09
	MD	3.01913E-08	2.89865E-08	3.24735E-08	3.85198E-08	5.38967E-09
	SD	0.170032173	2.87877E-08	1.31248E-07	0.17003217	5.03912E-09
$f(x)_2$	AB	0.697808011	0.700534565	0.748811576	0.700735233	0.513027451
	MD	0.666666667	0.666666669	0.666666667	0.666666668	0.666666666
	SD	0.097049446	0.088071999	0.355137521	0.120344378	0.07217346
$f(x)_3$	AB	0.001642553	0.00156125	0.0021304	0.001561004	0.001187696
	MD	5.4845E-14	4.10783E-14	3.60267E-14	4.27991E-14	0
	SD	0.003952627	0.003203167	0.007141942	0.003631587	0.005659823
$f(x)_4$	AB	1.87786E-07	2.31948E-08	9.47646E-09	3.24938E-09	0
	MD	3.94204E-14	2.2538E-15	4.06318E-15	2.2466E-15	0
	SD	1.02516E-06	1.25453E-07	4.47869E-08	9.79055E-09	0
$f(x)_5$	AB	0.154202856	0.127121622	0.139280775	0.111976783	0.102971194
	MD	2.22132E-15	9.46938E-16	1.45695E-15	6.79727E-16	0.044764125
	SD	0.258019536	0.240451172	0.3650242	0.411200291	0.533490041
$f(x)_6$	AB	2	2	2	2	2
	MD	2	2	2	2	2
	SD	0	0	0	0	0
$f(x)_7$	AB	2	2	2	2	2
	MD	2	2	2	2	2
	SD	0	0	0	0	0
$f(x)_8$	AB	1.86241E-14	9.07778E-15	1.34002E-13	6.06243E-17	3.69202E-15
	MD	2.16856E-25	0	0	1.57772E-30	0
	SD	7.63867E-14	4.36561E-14	6.71808E-13	0.000177981	3.30985E-16
$f(x)_9$	AB	0	0	1.1417E-123	0	0
	MD	0	0	1.3193E-145	0	0
	SD	0	0	6.2531E-123	0	0
$f(x)_{10}$	AB	1390.30741	84.69369769	73.51647998	5867.81578	72.38309382
	MD	73.68667036	73.72968645	73.55230648	73.68669066	71.03788422
	SD	7210.702813	58.0087577	2.250862628	31705.89613	3.990924703
$f(x)_{11}$	AB	107.6940319	106.1665177	723.7773275	106.4781406	105.5287341
	MD	105.5876108	106.6846538	106.3200494	107.3751166	105.9476506
	SD	15.43514499	4.58932148	3386.715099	3.255277603	3.759620619
$f(x)_{12}$	AB	30	30	30	30	30
	MD	30	30	30	30	30
	SD	0	0	0	0	0
$f(x)_{13}$	AB	0.000802734	0.000840848	0.000759216	0.000694274	0.00053463
	MD	0.000593663	0.000797071	0.000680998	0.000656195	0.000527303
	SD	0.000530664	0.000443019	0.000372237	0.000319378	0.00028228
$f(x)_{14}$	AB	4.900254181	0.132538855	0.216299179	12433.58502	0.000430466
	MD	3.44219E-05	9.70953E-06	2.81034E-06	1.3459E-06	1.40947E-10
	SD	26.44876067	0.704877724	1.163104212	68101.19368	0.002349287
$f(x)_{15}$	AB	0.00023298	0.00460782	9.94022E-05	9.47279E-05	1.87624E-05
	MD	1.385E-05	3.175E-05	4.5901E-05	3.83058E-05	6.8682E-06
	SD	0.000601531	0.024643032	0.000127479	0.000150572	3.28772E-05
$f(x)_{16}$	AB	1.56861E-13	1.76761E-13	1.13274E-13	1.7634E-13	6.87534E-16
	MD	8.77538E-14	1.29241E-13	4.36687E-14	8.48701E-14	3.66752E-16
	SD	1.675E-13	1.83873E-13	1.96176E-13	2.05313E-13	8.44144E-16
$f(x)_{17}$	AB	3.88077E-12	3.93626E-12	9.6417E-12	4.06929E-12	7.81259E-15
	MD	2.09132E-12	2.02538E-12	2.95174E-12	1.84401E-12	4.19622E-15
	SD	4.08505E-12	5.48648E-12	2.51634E-11	5.0735E-12	1.01241E-14
$f(x)_{18}$	AB	1.35112E-16	8.03645E-17	6.34852E-17	6.00462E-17	4.20512E-17
	MD	6.01344E-17	3.52286E-17	2.41263E-17	4.53603E-17	1.81663E-17
	SD	2.80371E-16	1.6582E-16	8.9323E-17	5.75724E-17	1.5816E-17
$f(x)_{19}$	AB	3.25446E-15	2.87023E-15	1.90039E-15	4.12796E-15	1.22464E-15
	MD	1.68411E-15	1.33928E-15	1.70968E-15	2.89807E-15	3.77989E-16
	SD	4.11333E-15	3.91756E-15	1.58714E-15	3.99247E-15	1.25259E-15
$f(x)_{20}$	AB	3.79728E-12	2.23714E-11	3.30447E-13	1.89544E-12	5.42588E-10
	MD	4.66838E-18	4.96695E-17	5.26045E-18	2.8914E-17	2.221E-16
	SD	2.06036E-11	9.93866E-11	1.10462E-12	7.0471E-13	2.69109E-09
$f(x)_{21}$	AB	4.68993E-08	3.95826E-08	3.37765E-08	4.07203E-08	9.13034E-10
	MD	4.01317E-08	3.301E-10	2.75639E-08	3.26865E-08	7.80406E-10
	SD	4.15991E-08	2.09778E-10	2.37579E-08	2.5464E-08	4.28355E-10

	AB	37.48206591	38.73455392	37.19411478	35.04672896	28.99844937
$f(x)_{22}$	MD	36.99627914	37.07500036	36.85873953	31.07464569	28.99844937
	SD	7.434091747	8.074446782	8.41918179	7.136344565	1.67938E-14
	AB	34.60837286	35.86930576	34.08340826	33.88021476	32.00181632
$f(x)_{23}$	MD	33.44020328	33.5699652	32.62963339	33.20329943	32.00012277
	SD	4.516139546	6.219361875	3.264919876	1.771366812	0.004461589
	AB	34.81813253	35.19495611	35.20603434	35.00389315	29
$f(x)_{24}$	MD	29.00000088	29.00000068	29.00000086	29.00000071	29
	SD	9.365721853	9.071482111	8.781823785	8.365115733	1.17852E-09

Tabla 1 Resultados de la minimización de las funciones benchmark

De acuerdo con la Tabla 1, el esquema propuesto basado en la técnica MH obtiene mejores resultados en 19 de las 24 funciones. Esto implica que el método de inicialización presenta un mejor desempeño en aproximadamente el 80% del conjunto completo de funciones. No obstante, el enfoque propuesto MH presenta un resultado similar a los otros esquemas en las funciones $f(x)_6$, $f(x)_7$, $f(x)_9$, $f(x)_{12}$. Cabe resaltar que, en estas funciones, todos los enfoques de inicialización presentan la misma precisión de soluciones. Finalmente, el método MH presenta un peor desempeño que otros enfoques de inicialización en la función $f(x)_{20}$.

5.2 Comparación en términos de la rapidez de convergencia y robustez

En esta subsección, se evalúa la rapidez de convergencia de los enfoques de inicialización comparados en el análisis. En la evaluación, hemos considerado tres índices de desempeño. El número de llamadas a función (NFC), el rango de éxito (SR) y el rango de aceleración (ARE). Estos se han considerado para ser compatibles con otros resultados publicados en la literatura (Tanabe & Fukunaga, 2014; Wen et al., 2016)

El número de llamadas a función (NFC) representa la cantidad de accesos a la función objetivo empleados durante la operación del algoritmo. El valor NFC se registra cuando el algoritmo alcanza una solución igual o más pequeña a un valor predefinido conocido como "valor a alcanzar" (VTR), el cual se configura a 10^{-6} . El valor de NFC reportado en el estudio ha sido promediado sobre 30 pruebas independientes en orden para reducir el efecto estocástico de los algoritmos. Un valor de NFC más pequeño significa que el método evaluado tiene una mejor rapidez de convergencia. El rango de éxito (SR) representa el rango de ejecuciones exitosas obtenidas por un método de optimización. Una ejecución exitosa corresponde al hecho de que el algoritmo ha sido capaz de alcanzar el valor VTR exitosamente. El SR se define en la Ecuación (11):

$$SR = \frac{\text{Número de ejecuciones exitosas}}{\text{Número de pruebas totales}} \quad (11)$$

El SR evalúa la robustez de un algoritmo; un valor alto de SR representa un mejor desempeño de un algoritmo particular. El rango de aceleración (ARE) compara el desempeño de dos métodos diferentes A y B , en términos del número de llamadas a función (NFC). $ARE_{A/B}$ se define de la Ecuación (12):

$$ARE_{A/B} = \frac{NFC_A}{NFC_B} \quad (12)$$

Donde NFC_A y NFC_B corresponden a los valores de NFC producidos por el algoritmo A y B respectivamente. Si $ARE_{A/B}$ tiene un valor mayor significa que el algoritmo B es más rápido que el método A .

En la Tabla 2 se registran los resultados numéricos de un conjunto representativo de 15 funciones. Tales funciones han sido seleccionadas debido a su complejidad de resolución. Una inspección detallada de la Tabla 1 indicada que el enfoque MH propuesto se desempeña de manera sobresaliente ante otros enfoques en 14 de las 15 funciones seleccionadas (Aproximadamente un 93% de las funciones). Esto implica que el método de inicialización propuesto MH permite una

convergencia más rápida a su valor final que sus competidores. Aunque el método DE_OP es más rápido que el método DE_MH en la función f_7 , su valor de SR es peor.

	DE		DE_OP		DE_GOP		DE_AR		DE_MH	
	NFC	SR	NFC	SR	NFC	SR	NFC	SR	NFC	SR
f_1	42388.33	1	40986.66	1	42378.33	0.9666	42501.66	1	35713.33	1
f_3	23136.66	0.8666	22143.33	0.7333	27033.33	0.6666	23368.33	0.7333	17445	0.9666
f_4	7231.66	0.9666	8656.2	1	8842.133	1	6978.33	1	50	1
f_5	17390	0.7333	15483	0.6333	21547.80	0.6	17788.33	0.6666	12370	0.8333
f_6	43668.33	1	44079	1	43755.73	1	44943.33	0.9666	43418.33	1
f_7	44223.33	1	43208.06	0.7333	44880.53	1	45133.33	1	44361.66	1
f_8	3906.66	1	4101.66	1	4288.33	1	3883.33	1	2383.33	1
f_9	340	1	286.66	1	370	1	305	1	198.33	1
f_{12}	6258.33	1	6180.20	1	5739.66	0.9333	6311.66	1	5118.33	0.9666
f_{16}	30796.66	1	30745	1	30856.66	1	30773.33	1	24025	1
f_{17}	34888.33	1	34750	1	35028.33	1	35193.33	1	27493.33	1
f_{18}	21668.33	1	22015	1	21953.33	1	21916.66	1	21138.33	1
f_{19}	26291.66	1	26541.66	1	26401.66	1	26660	1	24251.66	1
f_{20}	6188.33	1	6200	1	5848.33	1	6156.66	1	5533.33	1
f_{21}	42693.33	1	42450	1	42520	1	42731.66	1	33581.66	1

Table 2 Resultados numéricos en términos de NFC y SR para un conjunto representativo de 15 funciones

En la Tabla 3 se registran los resultados numéricos de un conjunto representativo de 15 funciones en términos del rango de aceleración (ARE). Como se puede observar, el método propuesto basado en el algoritmo MH presenta los mejores rangos. Una excepción es la función f_7 donde los métodos de inicialización DE y DE_OP obtienen los mejores resultados.

Function	$ARE_{A/B}$			
	$ARE_{DE/MH}$	$ARE_{DE_OP/MH}$	$ARE_{DE_GOP/MH}$	$ARE_{DE_AR/MH}$
f_1	1.18690498	1.14765727	1.18662498	1.1900784
f_3	1.32626349	1.26932263	1.54963218	1.33954333
f_4	144.633333	173.124	176.842667	139.566667
f_5	1.40582053	1.25165724	1.74194018	1.4380221
f_6	1.00575794	1.01521631	1.00777091	1.0351234
f_7	0.99688169	0.97399557	1.01169628	1.0173949
f_8	1.63916033	1.72097863	1.79930181	1.62937150
f_9	1.71428571	1.44537815	1.86554622	1.53781513
f_{12}	1.22272875	1.20746337	1.12139368	1.23314881
f_{16}	1.28185917	1.27970864	1.28435657	1.28088796
f_{17}	1.2689743	1.26394277	1.27406644	1.2800679
f_{18}	1.02507293	1.04147284	1.03855555	1.03682094
f_{19}	1.08411793	1.0944265	1.0886537	1.09930589
f_{20}	1.11837349	1.12048193	1.05692771	1.1126506
f_{21}	1.2713286	1.26408258	1.26616706	1.2724701

Table 3 Resultados numéricos para un conjunto representativo de 15 funciones en términos del rango de aceleración (ARE)

5.3 Análisis de gráficas de convergencia

La evaluación de la calidad de las soluciones finales no puede caracterizar completamente el desempeño de optimización de un método metaheurístico. Por ende, en esta subsección, se analizan las gráficas de convergencia en los cinco mecanismos de inicialización. El objetivo de este estudio es evaluar la velocidad con la cual un enfoque de inicialización alcanza el valor óptimo. En las comparativas, se considera un conjunto representativo de seis funciones operadas en 30

dimensiones. Para generar las gráficas de convergencia, se procesan los datos de simulación en crudo obtenidos en los diferentes experimentos. Dado que cada simulación se ejecuta 30 veces para cada método metaheurístico, los datos de convergencia de la ejecución corresponden al valor de la mediana. La Figura 2 muestra las gráficas de convergencia de las funciones (a) $f(x)_1$, (b) $f(x)_2$, (c) $f(x)_{12}$, (d) $f(x)_{13}$, (e) $f(x)_{14}$, y (f) $f(x)_{15}$. Una inspección detallada de los resultados demuestra que el esquema propuesto presenta una mejor convergencia a comparación de los otros algoritmos.

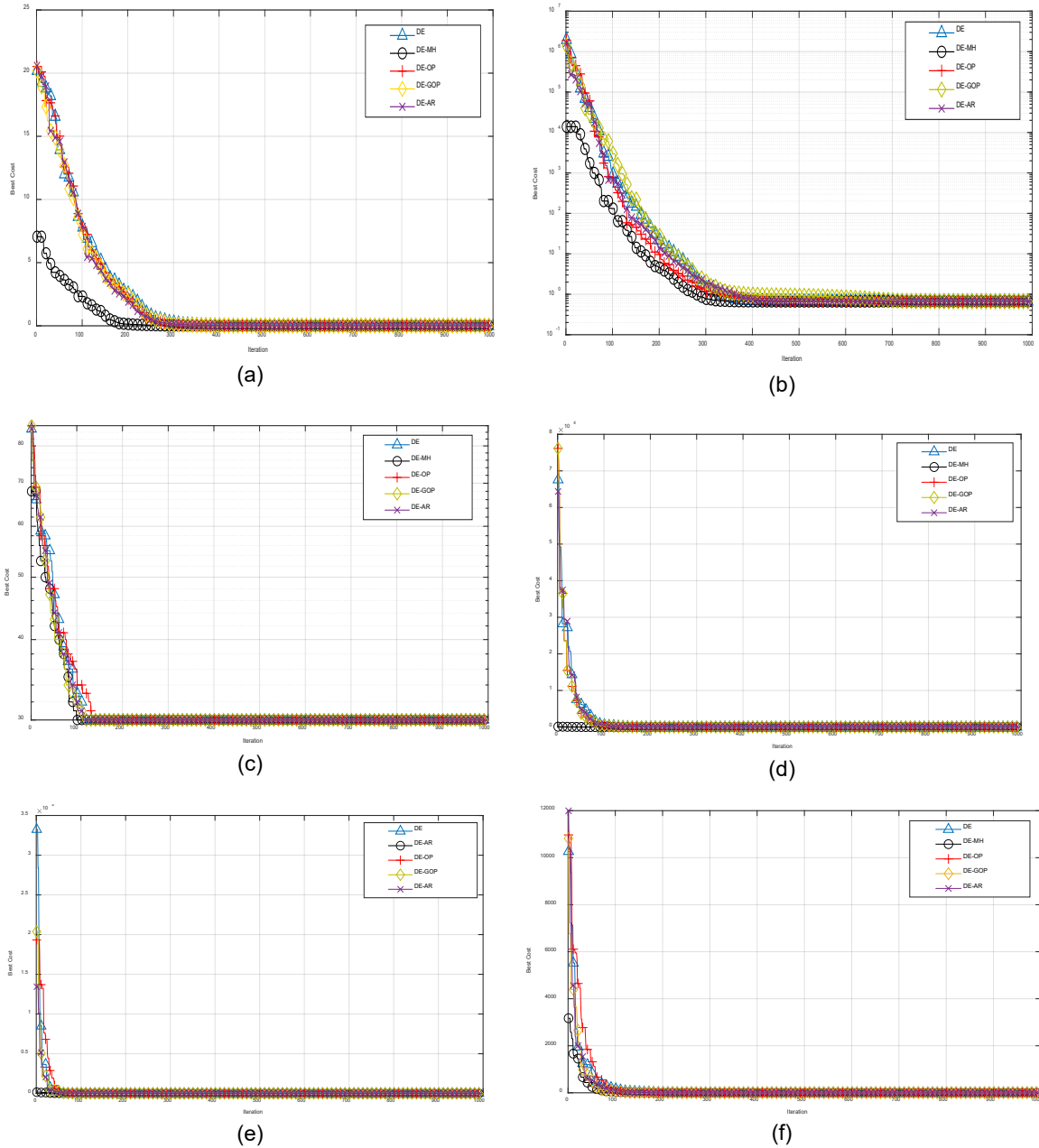


Figura 2 Gráficas de curva de convergencia para las funciones (a) $f(x)_1$, (b) $f(x)_2$, (c) $f(x)_{12}$, (d) $f(x)_{13}$, (e) $f(x)_{14}$, y (f) $f(x)_{15}$

6. Conclusiones

En este artículo, se propuso un nuevo método para inicialización de poblaciones para algoritmos metaheurísticos. En el enfoque propuesto, el conjunto inicial de soluciones iniciales se obtiene a través del muestro de la función objetivo con el uso de la técnica de Metropolis-Hastings. Bajo este método, el conjunto inicial de soluciones adopta valores cercanos a los valores prominentes de la función objetivo $J(\mathbf{x})$. A diferencia de la mayoría de los métodos de inicialización que únicamente consideran una distribución espacial, en el algoritmo, los puntos iniciales representan regiones promisorias del espacio de búsqueda, las cuales merecen ser explotadas para identificar el óptimo global. Estas características permiten que el enfoque propuesto obtenga una convergencia más rápida y mejora la calidad de las soluciones producidas.

Con el objetivo de demostrar las capacidades del método de inicialización, este se ha embebido en el algoritmo DE clásico. El rendimiento del enfoque ha sido comparado con otros mecanismos de inicialización tales como la inicialización aleatoria tradicional (RI), basado en la oposición (OP) aprendizaje basado en la oposición generalizado (GOP), y la aleatoriedad adaptativa (AR). Para evaluar el desempeño, el enfoque y otros métodos similares han sido probados en un conjunto representativo de funciones de benchmark extraídos de diferentes conjuntos de datos. Los datos experimentales demuestran que la técnica propuesta presenta una mejor rapidez de convergencia y un incremento en la calidad de soluciones que otros enfoques similares.

Referencias

- Birbil, Ş. İ., & Fang, S.-C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 25(3), 263–282. <https://doi.org/10.1023/A:1022452626305>
- Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. *IEEE Transactions on Evolutionary Computation*, 10(6), 646–657. <https://doi.org/10.1109/TEVC.2006.872133>
- Chaveau, D., & Vandekerckhove, P. (2002). Improving Convergence of the Hastings–Metropolis Algorithm with an Adaptive Proposal. *Scandinavian Journal of Statistics*, 29(1), 13–29. <https://doi.org/10.1111/1467-9469.00064>
- Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4), 327–335. <https://doi.org/10.1080/00031305.1995.10476177>
- Cuevas, E., Cienfuegos, M., Zaldívar, D., & Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, 40(16), 6374–6384. <https://doi.org/10.1016/j.eswa.2013.05.041>
- Cuevas, E., & Rodríguez, A. (2020). *Metaheuristic Computation with MATLAB®*. Chapman and Hall/CRC. <https://doi.org/10.1201/9781003006312>
- de Castro, L. N., & Timmis, J. (n.d.). An artificial immune network for multimodal function optimization. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, 699–704. <https://doi.org/10.1109/CEC.2002.1007011>
- Fogel D. (2009). Artificial Intelligence through Simulated Evolution. In *Evolutionary Computation*. IEEE. <https://doi.org/10.1109/9780470544600.ch7>
- Geyer, C. J. (1992). Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4), 473–483. <http://www.jstor.org/stable/2246094>
- Holland, J. H. (1984). Genetic Algorithms and Adaptation. In *Adaptive Control of Ill-Defined Systems* (pp. 317–333). Springer US. https://doi.org/10.1007/978-1-4684-8941-5_21

- Jahn Johannes. (2007). *Introduction to the Theory of Nonlinear Optimization*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-49379-2>
- Jingqiao Zhang, & Sanderson, A. C. (2009). JADE: Adaptive Differential Evolution With Optional External Archive. *IEEE Transactions on Evolutionary Computation*, 13(5), 945–958. <https://doi.org/10.1109/TEVC.2009.2014613>
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Technical report-tr06, Erciyes university, engineering faculty, computer
- Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213(3–4), 267–289. <https://doi.org/10.1007/s00707-009-0270-4>
- Kennedy, J., & Eberhart, R. (n.d.). Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1942–1948. <https://doi.org/10.1109/ICNN.1995.488968>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Kononova, A. V., Caraffini, F., & Bäck, T. (2021). Differential evolution outside the box. *Information Sciences*, 581, 587–604. <https://doi.org/10.1016/j.ins.2021.09.058>
- Maciel C., O., Cuevas, E., Navarro, M. A., Zaldivar, D., & Hinojosa, S. (2020). Side-Blotched Lizard Algorithm: A polymorphic population approach. *Applied Soft Computing*, 88, 106039. <https://doi.org/10.1016/j.asoc.2019.106039>
- Ochoa, P., Castillo, O., Melin, P., & Soria, J. (2021). Differential Evolution with Shadowed and General Type-2 Fuzzy Systems for Dynamic Parameter Adaptation in Optimal Design of Fuzzy Controllers. *Axioms*, 10(3), 194. <https://doi.org/10.3390/axioms10030194>
- Ochoa, P., Castillo, O., & Soria, J. (2020). High-Speed Interval Type-2 Fuzzy System for Dynamic Crossover Parameter Adaptation in Differential Evolution and Its Application to Controller Optimization. *International Journal of Fuzzy Systems*, 22(2), 414–427. <https://doi.org/10.1007/s40815-019-00723-w>
- Pan, W., Li, K., Wang, M., Wang, J., & Jiang, B. (2014). Adaptive Randomness: A New Population Initialization Method. *Mathematical Problems in Engineering*, 2014, 1–14. <https://doi.org/10.1155/2014/975916>
- Qin, A. K., & Suganthan, P. N. (n.d.). Self-adaptive Differential Evolution Algorithm for Numerical Optimization. *2005 IEEE Congress on Evolutionary Computation*, 1785–1791. <https://doi.org/10.1109/CEC.2005.1554904>
- Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. A. (2007). A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, 53(10), 1605–1614. <https://doi.org/10.1016/j.camwa.2006.07.013>
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2011). Filter modeling using gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 24(1), 117–122. <https://doi.org/10.1016/j.engappai.2010.05.007>
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>

- Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for Differential Evolution. *2013 IEEE Congress on Evolutionary Computation*, 71–78. <https://doi.org/10.1109/CEC.2013.6557555>
- Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of SHADE using linear population size reduction. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 1658–1665. <https://doi.org/10.1109/CEC.2014.6900380>
- Wang, H., Wu, Z., Liu, Y., Wang, J., Jiang, D., & Chen, L. (2009). Space transformation search. *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 537–544. <https://doi.org/10.1145/1543834.1543907>
- Wen, J., Ma, H., & Zhang, X. (2016). Optimization of the occlusion strategy in visual tracking. *Tsinghua Science and Technology*, 21(2), 221–230. <https://doi.org/10.1109/TST.2016.7442504>
- Yang, X. (2010). *Engineering Optimization*. Wiley. <https://doi.org/10.1002/9780470640425>
- Yang, X.-S., & Deb, S. (2010). *Cuckoo Search via Levy Flights*.



Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 2.5 México.