

Recibido 12 Marz 2026

ReCIBE, Año 15 No. 2, Junio 2026

Aceptado 24 May 2026

**Mejora del rendimiento de los algoritmos metaheurísticos
mediante poblaciones estructuradas y teoría de juegos
evolutiva**

**Improving the performance of metaheuristic algorithms through
structured populations and evolutionary game theory**

Hector Joaquin Escobar Cuevas'

hector.escobar@uas.edu.mx

Jesus Ivan Aramburo Gutierrez'

ivan_1_killer@hotmail.com

Alma Yadira Quiñonez Carrillo'

yadiraqui@uas.edu.mx

Joaquin Pantaleón Escobar Moreno'

joaquin.escobar@uas.edu.mx

/ Universidad Autónoma De Sinaloa, México

Resumen

La diversidad desempeña un papel fundamental en los algoritmos metaheurísticos, ya que contribuye a prevenir la convergencia prematura, favorece el equilibrio entre exploración y explotación y reduce la probabilidad de quedar atrapado en óptimos locales. Muchos algoritmos metaheurísticos tradicionales emplean una sola estrategia para generar nuevas soluciones, lo que puede limitar la diversidad dentro de la población. En contraste, el uso de múltiples estrategias permite generar distintos comportamientos de búsqueda y producir un conjunto más variado de soluciones candidatas, mejorando así la exploración del espacio de búsqueda. La teoría de juegos evolutiva introduce mecanismos de adaptación en los que las estrategias de los agentes evolucionan mediante procesos de competencia, fortaleciendo aquellas más efectivas y descartando las menos eficientes. Por su parte, las poblaciones estructuradas, a diferencia de las no estructuradas, preservan una mayor diversidad estratégica gracias a interacciones locales, donde cada individuo compite únicamente con un subconjunto de la población. En este trabajo se propone un nuevo método metaheurístico basado en teoría de juegos evolutiva aplicado a poblaciones estructuradas. Inicialmente, las soluciones se generan cerca de regiones prometedoras por medio del algoritmo Metropolis–Hastings. Posteriormente, a cada persona se le fija una táctica de investigación específica y la población se divide en varios clústeres. Dentro de cada clúster, las estrategias evolucionan mediante competencia intraclúster para mejorar la eficiencia de búsqueda. La propuesta se evaluó utilizando 30 funciones de prueba y se comparó con diversos algoritmos metaheurísticos. Los resultados muestran mejoras en la calidad de las soluciones y en la velocidad de convergencia.

Palabras clave: metaheurística; teoría de juegos; optimización; competencia; Metropolis–Hastings

Summary: Diversity plays a fundamental role in metaheuristic algorithms, as it helps prevent premature convergence, maintains a balance between exploration and exploitation, and reduces the likelihood of being trapped in local optima. Many traditional metaheuristic algorithms rely on a single strategy to generate new solutions, which can limit the diversity of the population. In contrast, incorporating multiple strategies enables different search behaviors and produces a broader set of candidate solutions, thereby improving the exploration of the search space. Evolutionary game theory introduces adaptive mechanisms in which agents modify their strategies through competitive interactions, reinforcing successful strategies while discarding less effective ones. Structured populations, unlike unstructured ones, help preserve strategic diversity through localized competition, where each individual interacts only with a subset of the population rather than with all individuals. In this work, a novel metaheuristic method based on evolutionary game theory applied to structured populations is proposed. Initially, individuals are positioned near promising regions using the Metropolis–Hastings algorithm. Subsequently, each individual is assigned a specific search strategy, and the population is divided into several clusters. Within these clusters, strategies evolve through intra-cluster competition to improve search efficiency and solution quality. The proposed approach was evaluated using 30 benchmark functions and compared with several well-known metaheuristic algorithms. The results demonstrate improvements in both solution quality and convergence speed.

Keywords: metaheuristics; game theory; optimization; competition; Metropolis–Hastings

1. Introducción

En los últimos años se ha incrementado el desarrollo de algoritmos metaheurísticos como alternativa a los métodos clásicos de optimización, los cuales buscan maximizar o minimizar una función objetivo bajo un conjunto de restricciones mediante la generación e interacción de una población de soluciones candidatas (Abdel-Basset et al., 2018; Yang, 2010b). Estos enfoques combinan componentes aleatorios y deterministas inspirados en fenómenos naturales y suelen clasificarse en algoritmos bioinspirados y algoritmos basados en la física (Almufti et al., 2019; Chopard y Tomassini, 2018; Dokeroglu et al., 2019). Los métodos bioinspirados emulan principios biológicos y abarcan técnicas como el algoritmo genético (GA) (Holland, 1984), optimización por enjambre de partículas (PSO) (Kennedy et al., 1995), optimización social de arañas (SSO) (Cuevas et al., 2013), evolución diferencial (DE) (Storn y Price, 1995), optimización del lobo gris (GWO) (Mirjalili et al., 2014) y colonia artificial de abejas (ABC) (Karaboga, 2005), mientras que los métodos basados en la física incluyen algoritmos como la búsqueda gravitacional (GSA) (Rashedi et al., 2009), optimización de campo electromagnético (EFO) (Abedinpourshotorban et al., 2016), algoritmo seno-coseno (CSA) (Mirjalili, 2016), CMA-ES (Hansen y Ostermeier, 1996), recocido simulado (SA) (Kirkpatrick et al., 1983), búsqueda de estados de la materia (SMS) (Cuevas et al., 2014), búsqueda de armonía (HS) (Geem et al., 2001), optimización de ballenas (WOA) (Mirjalili y Lewis, 2016), algoritmo de murciélagos (BA) (Yang, 2010a), búsqueda de cucos (CS) (Yang y Deb, 2009) y búsqueda de cuervos (CSA) (Askarzadeh, 2016).

Estos algoritmos han sido confirmados por su eficacia para satisfacer diversos problemas de optimización en múltiples aplicaciones (Afzal et al., 2023; Giri et al., 2022; Kaur et al., 2023; Vaziri et al., 2023). Sin embargo, debido a la complejidad y diversidad de los problemas, ninguna metaheurística es universalmente óptima (Wolpert y Macready, 1997). En este contexto, la diversidad dentro de la población de soluciones juega un papel fundamental, ya que permite equilibrar la exploración y la explotación del espacio de búsqueda, evitar la convergencia prematura y mejorar la capacidad de escapar de óptimos locales (Osuna-Enciso et al., 2022). En muchos enfoques tradicionales, el uso de una única táctica de búsqueda puede restringir la adaptabilidad del algoritmo frente a paisajes de optimización complicados. La incorporación de variadas estrategias permite generar procedimientos de búsqueda más variados, favoreciendo una dinámica progresiva más flexible y robusta. Esta mezcla estratégica contribuye a mantener un equilibrio dinámico durante el proceso de optimización y mejora la capacidad de adaptación ante problemas multimodales.

Por otra parte, la teoría de juegos evolutiva (TJE) (Weibull, 1997) brinda un marco conceptual adecuado para modelar la adaptación estratégica dentro de poblaciones. En este contexto, las estrategias evolucionan en función del desempeño referente de los individuos, promoviendo la propagación de comportamientos exitosos y reduciendo la influencia de aquellos menos eficaces. Cuando se acoge un enfoque estructurado, la interacción se restringe a subconjuntos de la población, lo que ayuda a la coexistencia de múltiples estrategias y fortalece la diversidad global del sistema. De esta forma, el proceso de inicialización desempeña un papel decisivo en el rendimiento del algoritmo, ya que una distribución apropiada de las soluciones iniciales mejora la cobertura del espacio de exploración desde las primeras iteraciones. La combinación de mecanismos de inicialización eficientes, estructuración poblacional y adaptación estratégica permite establecer un entorno evolutivo más controlado y eficaz.

En este trabajo se propone un nuevo método metaheurístico fundamentado en la teoría de juegos evolutiva aplicada a poblaciones estructuradas. El enfoque incorpora un esquema de inicialización orientado hacia regiones prometedoras del espacio de

búsqueda, la asignación de estrategias individuales y la segmentación de la población en subgrupos con interacción localizada. La dinámica estratégica dentro de cada subgrupo preserva la diversidad poblacional, favorece la exploración y fortalece el proceso de refinamiento, aumentando la probabilidad de converger hacia soluciones globales de alta calidad.

Finalmente, el resto del artículo se organiza de la siguiente manera. En la Sección 2 se presentan los conceptos preliminares necesarios para comprender la metodología propuesta, incluyendo la técnica de inicialización Metropolis–Hastings, el algoritmo de agrupamiento K-means y los fundamentos de la teoría de juegos evolutiva. La Sección 3 describe en detalle el enfoque propuesto, incluyendo el proceso de inicialización, la actualización de posiciones, la formación de grupos poblacionales y el mecanismo de adaptación de estrategias. En la Sección 4 se presentan los resultados experimentales obtenidos mediante la evaluación del método en un conjunto de funciones de referencia y su comparación con diversos algoritmos metaheurísticos ampliamente utilizados. Finalmente, la Sección 5 expone las conclusiones del trabajo y las posibles líneas de investigación futura.

2. Conceptos preliminares

Esta sección ofrece una visión general de los principios subyacentes de la técnica de inicialización Metropolis-Hastings, el algoritmo K-means y la teoría de juegos, que son necesarios para comprender la proposición investigada. La técnica de inicialización Metropolis-Hastings es una metodología de nueva generación propuesta para superar ese subdesarrollo. Mejora el rendimiento y la velocidad de convergencia, al tiempo que la solución también mejora aún más. El algoritmo Kmeans permite la partición eficaz de los datos en clústeres, que pueden utilizarse en diversas tareas relacionadas con la compresión de datos, la segmentación de imágenes y la identificación de patrones dentro de grandes conjuntos de datos. La teoría de juegos constituye un marco analítico para el estudio de interacciones estratégicas entre individuos, permitiendo modelar escenarios en los que los resultados dependen de las decisiones adoptadas por múltiples agentes.

2.1 Técnica de inicialización Metropolis-Hastings

El proceso de optimización es iniciado con la generación de un conjunto inicial de soluciones candidatas distribuidas a lo largo del espacio de búsqueda, promoviendo la diversidad poblacional respecto a la función de coste. Este proceso es la inicialización de la población. El objetivo de la inicialización es aportar al algoritmo un punto de partida con diferentes soluciones para comenzar el proceso de evolución de las posiciones individuales y mejorar su rendimiento.

El método de inicialización Metropolis-Hastings (MH) (Cuevas et al., 2022) aplica un muestreo de la función de coste $f(x)$ para la generación de un conjunto de soluciones candidatas para iniciar. Este método tiene como objetivo investigar la función de coste mediante la generación de soluciones que se pretenden usar en un inicio, y puede utilizarse para muestrear distribuciones con un gran número de dimensiones. En el enfoque MH, se crea una muestra en el espacio de búsqueda aplicando un mecanismo de probabilidad de aceptación; esta muestra puede ser aceptada o rechazada. El criterio de aceptación viene determinado por la relación existente entre el rendimiento de la muestra actual y las muestras candidatas en términos de la función de coste.

El proceso de muestreo se inicia a partir de un muestreo de una función de probabilidad $f(x)$, donde x corresponde a un valor del espacio d-dimensional. Este valor se utiliza en un proceso de cadena de Markov desde el estado actual x_k en la iteración k -ésima hasta

un nuevo estado x_{k+1} implementado en dos pasos. La producción de una nueva muestra y a partir de una perturbación de la muestra actual x_k proporcionada por un número aleatorio distribuido normalmente $N(0,1)$. La muestra generada y se decide además si puede utilizarse como nuevo estado x_{k+1} mediante un criterio con una probabilidad α que se define de la siguiente manera:

$$\alpha(y, x_k) = \min \left\{ \frac{f(y)}{f(x_k)}, 1 \right\} \quad (1)$$

Mediante esta regla probabilística, se genera un número aleatorio distribuido uniformemente $U(0,1)$ y se compara con la probabilidad α . Si se cumple la regla $r < \alpha$, el candidato y se acepta como el nuevo estado x_{k+1} de la población. De lo contrario, se descarta y se selecciona la muestra x_k como nuevo estado.

2.2 Algoritmo K-Means

Debido a las características de los enfoques estructurados en EGT, un paso crucial es separar a los agentes de la población en diferentes grupos en los que cada sujeto participará en las competencias. En el esquema propuesto, se selecciona el algoritmo K-means [31] para realizar esta acción, que se explica en esta sección. El algoritmo K-means es un conocido algoritmo de agrupamiento en el que un conjunto de n vectores $x_1, x_2, \dots, x_n$ se divide en m grupos $C_1, C_2, \dots, C_m$. En este enfoque, cada clúster C_i se caracteriza por su centroide μ_i , que se calcula mediante la minimización de una función objetivo. Suponiendo la distancia euclidiana como criterio de similitud, la función objetivo se define de la siguiente manera:

$$J = \sum_{i=1}^m J_i = \sum_{i=1}^m \sum_{k, x_k \in C_i} |x_k - \mu_i|^2 \quad (2)$$

donde J_i representa el costo asociado al grupo i . De acuerdo con la ecuación (1), el valor de J_i está determinado por la distribución espacial de los vectores dentro del clúster J_i y por la ubicación de su centroide μ_i . La asignación de los elementos a cada grupo se establece mediante el parámetro u_{ij} , el cual indica la relación entre el vector y el clúster correspondiente. Bajo estas condiciones, el valor de u_{ij} puede expresarse de la siguiente forma:

$$u_{ij} = \begin{cases} 1, & \text{en caso contrario} \\ 0, & \text{si } x_j \notin C_i \end{cases} \quad (3)$$

Por lo tanto, el problema de agrupamiento puede resolverse encontrando los valores de u_{ij} y μ_i que minimizan la función de costo J . El algoritmo K-means realiza esta minimización mediante un proceso iterativo J compuesto por dos etapas. En la primera etapa, la función J se minimiza con respecto a u_{ij} , manteniendo fijas las posiciones de los centroides μ_i . Posteriormente, en la segunda etapa, la minimización se realiza respecto a μ_i , considerando constantes los valores de u_{ij} . Estas dos fases se repiten de manera alternada hasta que el algoritmo alcanza la convergencia. Los valores de u_{ij} se obtienen durante el primer paso del algoritmo. Debido a que la función objetivo es lineal con respecto a u_{ij} y cada valor u_{ij} implica la asignación del vector a un único clúster C_{ij} , es posible determinar cada u_{ij} de forma independiente. En este contexto, $u_{ij} = 1$ cuando el centroide μ_i es el

2 más cercano al vector , es decir, cuando se cumple la distancia mínima $\|x_j - \mu_i\|$. Por lo tanto, se obtiene la siguiente expresión:

$$u_{ij} = \begin{cases} 1, & i = \operatorname{argmin} \|x_j - \mu_i\| \\ 0, & \text{en caso contrario} \end{cases} \quad (4)$$

En la segunda parte, se establecen los valores de μ_i conservando constantes los valores de u_{ij} conseguidos anteriormente. Dado que la función objetivo J es una función cuadrática en términos de μ_i , se consigue minimizar diferenciando J y dejando su valor en 0. Teniendo lo anteriormente mencionado en cuenta, los valores de μ_i se establecen de la siguiente manera:

$$\mu_i = \frac{\sum_{j=1}^n u_{ij} x_j}{\sum_{j=1}^n u_{ij}} \quad (5)$$

2.3 Teoría de juegos evolutiva

En la teoría de juegos evolutiva (EGT) (Weibull, 1997), la cuestión fundamental es el proceso de adaptación de las estrategias, en el que los individuos adaptan sus métodos hacia otros más favorables. La diferencia de rendimiento entre las estrategias impulsa la adaptación (Gintis, 2000). Cuando una estrategia funciona peor que otra, los individuos cambian su comportamiento y se produce un cambio gradual de la población hacia las estrategias más ventajosas, lo que significa la supervivencia del más apto (Grüne-Yanoff, 2011).

Durante la adaptación, se seleccionan aleatoriamente dos agentes para compararlos. Si la estrategia de i es menos eficaz que la de j , la estrategia de i se modifica para parecerse más a la estrategia de j . Esta variación es de naturaleza probabilística y está relacionada con la diferencia de recompensa entre estrategias. Con el tiempo, las estrategias exitosas se extienden y se vuelven dominantes, y las estrategias inferiores se deterioran.

Existen dos paradigmas principales: enfoques estructurados y no estructurados (Stella y Bauso, 2017). En los modelos estructurados, los individuos se organizan en grupos y la interacción se limita a esa estructura, pudiendo coexistir diferentes estrategias simultáneamente. En los modelos no estructurados, toda la población es una unidad bien mezclada y suele conducir al predominio de una única estrategia. Debido a sus características, en este trabajo se adopta un enfoque estructurado.

3. El enfoque propuesto

La diversidad de las soluciones candidatas se garantiza dentro de los algoritmos metaheurísticos. Esto evita caer en la trampa de los óptimos locales y evita la convergencia temprana hacia soluciones peores. Equilibra la compensación entre exploración y explotación intrínseca a la resolución de problemas complejos, lo que hace que la diversidad sea esencial para el éxito de la optimización. En metaheurística, una sola estrategia suele explorar con poca diversidad. La ejecución de múltiples estrategias mejora la diversidad y da lugar a un mejor resultado. Esta diversidad contribuye a mantener un mejor equilibrio entre exploración y explotación, lo que permite una búsqueda más eficiente del óptimo global. En consecuencia, la utilización de múltiples estrategias aumenta la probabilidad de alcanzar una solución óptima. Por esta razón, el

empleo de diferentes enfoques dentro del algoritmo mejora su robustez y eficiencia. En este contexto, el presente trabajo introduce un nuevo método metaheurístico basado en un IJE estructurado que permite la coexistencia de varias estrategias. El objetivo del método propuesto es encontrar la solución global de un problema de optimización que puede formularse de la siguiente manera:

$$\begin{aligned} \max/\min: \quad & f(p) \quad p = (p_1, \dots, p_d) \in R^d \\ \text{Sujeto a:} \quad & p \in X \end{aligned} \quad (6)$$

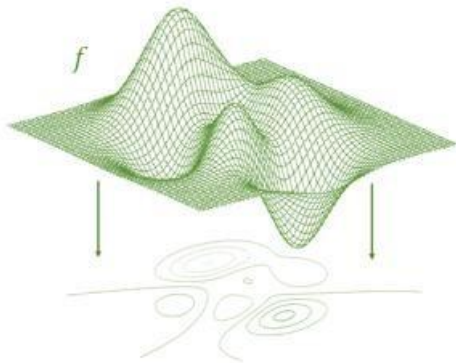
donde $f: R^d \rightarrow R$ es una función no lineal de dimensión $d = (p_1 \dots p_d) \in R^d$ que representa los vectores de dimensión d correspondientes a los elementos de la población, y X representa el espacio de búsqueda ($l_i \leq p_i \leq u_i, i = 1, \dots, d$) definido por los límites inferior (l_i) y superior (u_i). La explicación del algoritmo propuesto se ha dividido en cuatro partes principales: inicialización, cálculos de la nueva posición, determinación del grupo de población y alteración de la estrategia.

3.1 Inicialización

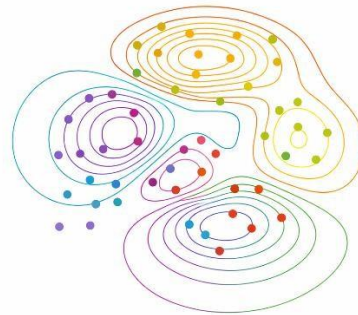
En este paso del algoritmo, se llevan a cabo dos procesos importantes para una población de tamaño n . El primer proceso consiste en posicionar a cada agente con una posición inicial dada por

$\{p_1^0, \dots, p_n^0\}$ que proporciona el punto de partida en la configuración del algoritmo. En el segundo proceso, se asigna una desviación estándar a cada agente, denotada como $(\sigma_1, \dots, \sigma_n)$. La estrategia asignada a cada agente i es la combinación de la posición y la desviación estándar correspondiente, y define cómo el agente explora su nueva posición mediante la fórmula: $S(i) = p_i + N(0, \sigma_i)$. No solo se trata de un enfoque dual de posición y varianza que define la estrategia de cada agente, sino que también sienta las bases sobre las que se desarrollarán las acciones y decisiones posteriores del algoritmo. El conjunto de posiciones iniciales se obtiene aplicando el método correspondiente, mediante el cual las soluciones iniciales se posicionan estratégicamente alrededor de las regiones significativas de la función objetivo f .

El vector de desviaciones estándar $p_1^0, \dots, p_n^0$ se genera a partir de d elemento, $(\sigma_{\{1,i\}}, \dots, \sigma_{\{d,i\}})$ cada uno correspondiente a una variable de decisión. A cada elemento se le asigna aleatoriamente un valor en una distribución uniforme entre 0 y 1, lo que mantiene la desviación estándar diversificada y hace que el enfoque del algoritmo sea variable y adaptable. Tras estos dos procesos, a cada agente de la población se le asigna un comportamiento o estrategia aleatoria denotada como $S(i)$, que implica un punto de partida p_i^0 y una desviación estándar σ_i .



(a)



(b)

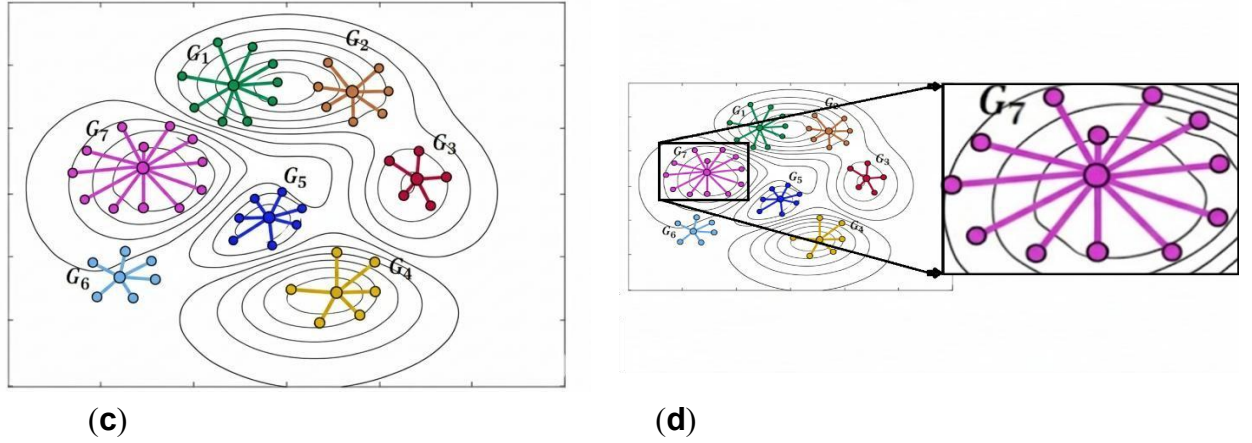


Figura 1. Ilustración del proceso de segmentación de la población. (a) Representación de la función objetivo acompañada de su mapa de contorno; (b) distribución inicial de 50 individuos; (c) resultado de aplicar el algoritmo K-means considerando siete grupos $\{G_1, \dots, G_7\}$; y (d) ampliación de los individuos pertenecientes al clúster G_7 .

3.2 Cálculo de nuevas posiciones y determinación de los grupos de población

El algoritmo de optimización propuesto actualiza de manera sistemática las soluciones candidatas dentro del espacio de búsqueda mediante la exploración de distintas posiciones. Este proceso se realiza de forma iterativa en cada generación de la búsqueda, favoreciendo la convergencia hacia aquellas regiones con mayor potencial para contener la solución global. Para generar una nueva posición, en cada iteración se modifica ligeramente la solución actual p^k , introduciendo pequeñas perturbaciones aleatorias que se obtienen a partir de una distribución de probabilidad normal $N(0, \sigma_i)$, de acuerdo con el modelo:

$$p^{k+1} = p^k + R \cdot N(0, \sigma_i) \cdot S(i) \quad (7)$$

Con el propósito de implementar un enfoque estructurado dentro de la competencia EGT, una vez determinadas las nuevas posiciones, el algoritmo procede a organizar la población en distintos grupos. Esta organización permite estructurar de manera clara la interacción entre los individuos dentro del proceso de búsqueda. Además, facilita la identificación de patrones de comportamiento dentro de la población durante la evolución del algoritmo. Estos grupos permiten estructurar la población y definir conjuntos de elementos que influyen de manera particular en la dinámica de búsqueda de cada individuo.

Para dividir adecuadamente toda la población según sus posiciones en el espacio de búsqueda, se aplica el método K-means al conjunto de n elementos debido a su simplicidad y bajo coste computacional. Con esta aplicación, se definen los centroides de cada uno de los m grupos $\mu_1, \dots, \mu_m$. Cada agente de la población se asigna al centroide más cercano en función de la distancia euclídea, permitiendo la integración de m grupos $G_1, \dots, G_m$, clasificando los individuos según su similitud con los centroides. Cada grupo G_i está claramente asociado con un conjunto único de elementos que pertenecen exclusivamente a sus respectivos grupos, creando divisiones claras dentro de la población.

Para realizar esta división de la población en función de sus posiciones dentro del espacio de búsqueda, se emplea el método K-means para la conformación de los clústeres, debido a su simplicidad computacional. En esta etapa se determinan los centroides

correspondientes a cada uno de los grupos $\mu_1, \dots, \mu_m$. Cada agente de la población es entonces asignado al centroide más cercano según la distancia entre sus posiciones $G_1, \dots, G_m$, lo que permite que los individuos se agrupen en función de su similitud con dichos centroides. De esta manera, cada grupo G_i queda asociado a un conjunto específico de elementos que pertenecen exclusivamente a su respectivo clúster, generando una segmentación clara dentro de la población.

Esta correspondencia exclusiva permite que cada grupo funcione como una entidad diferenciada, con características y miembros propios. Dicha organización resulta fundamental para el funcionamiento del algoritmo, ya que facilita la aplicación de estrategias particulares dirigidas a cada grupo, promoviendo así un proceso de resolución más estructurado y eficiente, lo que contribuye a mejorar el desempeño global del algoritmo.

3.3 Modificación de estrategias y adaptación de la desviación estándar y posición

Una vez formados los clústeres mediante K-means, el algoritmo realiza una actualización intragrupal de estrategias. Cada individuo compara su estrategia únicamente con miembros de su mismo grupo, garantizando una evaluación contextual y estructurada. Si la estrategia $S(i)$ de un individuo i es inferior a la estrategia $S(j)$ de otro individuo j dentro del mismo clúster, entonces i puede imitar progresivamente a j . La decisión de cambio se realiza de forma probabilística, dependiendo de la diferencia en sus valores de aptitud. Cada estrategia está definida por dos componentes: posición p_i y desviación estándar σ_i . Cuando se detecta inferioridad, estos parámetros se ajustan para aproximarse a los del individuo con mejor desempeño. La probabilidad de actualización se define como:

$$Pr = \frac{fit(EC) - fit(p^i)}{MF} \quad (8)$$

donde MF representa el peor valor de la función objetivo encontrado hasta el momento. A mayor diferencia de desempeño, mayor probabilidad de adaptación. Si la prueba probabilística es exitosa (según Pr), el individuo i adapta su estrategia imitando al individuo j dentro del mismo clúster. A su vez, esta implementación permitirá al algoritmo buscar alrededor de la posición del individuo exitoso, y, además, ayudará a encontrar soluciones nuevas.

4. Resultados experimentales

En esta subsección se analiza el desempeño del método SEGTA en comparación con ocho algoritmos ampliamente reconocidos en la literatura: ABC, BA, CMAES, DE, HS, PSO, CS y SCA. Con el objetivo de evaluar el comportamiento de la metodología propuesta, se emplea un conjunto de 30 funciones de referencia, específicamente $f_1(x)$ a $f_{30}(x)$, cuyo detalle puede consultarse en el apéndice A. De acuerdo con los resultados presentados en la Tabla 1, el método SEGTA muestra un desempeño superior al obtenido por los algoritmos comparados en 23 funciones de prueba. En particular, se observan mejores resultados en $f_1(x)$, $f_3(x)$, $f_7(x)$, $f_8(x)$, $f_{10}(x)$, $f_{11}(x)$, $f_{13}(x)$, $f_{16}(x)$, $f_{18}(x)$ y $f_{20}(x)$ a $f_{30}(x)$, lo que evidencia la competitividad del enfoque propuesto frente a métodos bien establecidos.

Asimismo, se observa que en la función $f_4(x)$ tanto PSO como SEGTA alcanzan el valor mínimo global. Resultados similares se presentan en $f_5(x)$ y $f_6(x)$, donde el algoritmo DE obtiene resultados comparables a los de SEGTA. Del mismo modo, en las funciones $f_9(x)$, el rendimiento de ABC, CMAES, DE, PSO, CS y UEGTA es equivalente en términos de la solución mínima obtenida. Además, en la función $f_{17}(x)$, el algoritmo CMAES muestra resultados comparables al compararse con SEGTA. No obstante, es importante señalar que CMAES únicamente alcanza este comportamiento en dos funciones de referencia. A partir de estos resultados, puede inferirse que el método propuesto mantiene un desempeño consistente y estable, proporcionando soluciones confiables. Asimismo, los hallazgos obtenidos indican que el esquema desarrollado presenta un alto nivel de robustez y competitividad frente a otros enfoques similares.

Función		ABC	BA	CMAES	DE	HS	PSO	CS	CSA	SEGTA
$f_1(X)$	AB	2.16×10^0	1.54×10^1	5.62×10^{-10}	1.24×10^{-7}	1.22×10^1	9.48×10^{-5}	6.86×10^0	1.72×10^1	0.00×10^0
	MD	2.16×10^0	1.53×10^1	5.51×10^{-10}	3.71×10^{-8}	1.20×10^1	2.54×10^{-6}	6.69×10^0	1.74×10^1	0.00×10^0
	SD	4.64×10^{-1}	1.05×10^0	1.67×10^{-10}	3.57×10^{-7}	5.46×10^{-1}	4.86×10^{-4}	1.11×10^0	5.41×10^{-1}	0.00×10^0
$f_2(X)$	AB	5.67×10^2	4.87×10^0	6.67×10^{-1}	6.87×10^{-1}	1.99×10^4	1.07×10^2	8.88×10^0	1.49×10^5	2.49×10^{-1}
	MD	5.02×10^2	6.71×10^{-1}	6.67×10^{-1}	6.67×10^{-1}	1.87×10^4	4.17×10^1	8.81×10^0	1.57×10^5	2.49×10^{-1}
	SD	2.82×10^2	1.07×10^1	2.98×10^{-8}	1.13×10^{-1}	5.11×10^3	1.36×10^2	1.89×10^0	3.61×10^4	6.17×10^{-4}
$f_3(X)$	AB	1.04×10^0	1.54×10^2	7.40×10^{-18}	3.12×10^{-3}	4.32×10^1	1.00×10^{-2}	1.15×10^0	1.67×10^2	0.00×10^0
	MD	1.04×10^0	1.34×10^2	0.00×10^0	4.34×10^{-14}	4.42×10^1	3.70×10^{-3}	1.16×10^0	1.70×10^2	0.00×10^0
	SD	1.83×10^{-2}	6.09×10^1	4.05×10^{-17}	5.25×10^{-3}	6.55×10^0	1.30×10^{-2}	4.11×10^{-2}	2.54×101	0.00×10^0
$f_4(X)$	AB	2.86×10^{-3}	4.92×10^{-17}	6.22×10^{-59}	1.38×10^{-9}	1.08×10^{-2}	0.00×10^0	2.40×10^{-8}	1.42×10^{-1}	0.00×10^0
	MD	2.11×10^{-3}	4.93×10^{-17}	3.15×10^{-59}	2.01×10^{-14}	1.08×10^{-2}	0.00×10^0	1.97×10^{-8}	1.33×10^{-1}	0.00×10^0
	SD	2.88×10^{-3}	1.86×10^{-17}	9.35×10^{-59}	6.78×10^{-9}	4.04×10^{-3}	0.00×10^0	1.60×10^{-8}	4.26×10^{-2}	0.00×10^0
$f_5(X)$	AB	-9.59×10^3	-3.16×10^7	-9.25×10^1	2.00×10^9	3.27×10^1	3.41×10^1	1.81×10^9	2.15×10^7	2.00×10^9
	MD	-1.53×10^3	-1.05×10^6	-4.16×10^1	2.00×10^9	3.29×10^1	9.00×10^0	2.00×10^9	1.73×10^5	2.00×10^9
	SD	3.24×10^4	1.38×10^8	1.51×10^2	0.00×10^9	1.15×10^1	1.13×10^2	3.80×10^9	6.88×10^7	0.00×10^9
$f_6(X)$	AB	-4.32×10^3	-2.24×10^7	-8.51×10^1	2.00×10^9	4.81×10^1	1.24×10^1	2.33×10^9	3.29×10^7	2.00×10^9
	MD	-1.19×10^3	-2.16×10^6	-2.33×10^1	2.00×10^9	4.37×10^1	9.00×10^0	2.00×10^9	3.53×10^5	2.00×10^9
	SD	7.49×10^3	7.59×10^7	1.37×10^2	1.55×10^{-15}	2.01×10^1	1.83×10^1	4.30×10^0	1.40×10^8	0.00×10^9
$f_7(X)$	AB	1.38×10^7	1.18×10^7	7.22×10^1	8.79×10^1	6.57×10^6	8.28×10^1	1.36×10^5	4.32×10^7	7.11×10^1
	MD	1.27×10^7	1.41×10^7	7.24×10^1	7.37×10^1	6.75×10^6	8.29×10^1	1.42×10^5	4.52×10^7	7.11×10^1
	SD	3.69×10^6	7.65×10^6	1.09×10^0	7.59×10^1	1.18×10^6	3.28×10^0	2.89×10^4	8.04×10^6	3.45×10^{-3}
$f_8(X)$	AB	1.90×10^6	9.38×10^6	1.05×10^2	1.06×10^2	8.55×10^6	1.02×10^2	3.05×10^4	7.58×10^7	4.65×10^1
	MD	1.83×10^6	6.14×10^6	1.05×10^2	1.05×10^2	8.69×10^6	1.02×10^2	3.11×10^4	7.09×10^7	4.65×10^1
	SD	5.46×10^5	1.10×10^7	1.16×10^0	3.51×10^0	2.29×10^6	4.71×10^0	8.62×10^3	1.99×10^7	8.33×10^{-2}
$f_9(X)$	AB	3.00×10^1	5.63×10^1	3.00×10^1	3.00×10^1	3.45×10^1	3.00×10^1	3.00×10^1	5.92×10^1	3.00×10^1
	MD	3.00×10^1	5.60×10^1	3.00×10^1	3.00×10^1	3.45×10^1	3.00×10^1	3.00×10^1	5.90×10^1	3.00×10^1
	SD	0.00×10^0	4.33×10^0	0.00×10^0	0.00×10^0	1.36×10^0	0.00×10^0	1.83×10^{-1}	1.58×10^0	0.00×10^0
$f_{10}(X)$	AB	5.42×10^2	8.48×10^{-3}	1.51×10^{-3}	7.52×10^{-4}	4.63×10^2	2.62×10^2	1.29×10^0	2.07×10^3	7.70×10^{-96}
	MD	4.94×10^2	6.78×10^{-3}	1.29×10^{-3}	7.06×10^{-4}	4.52×10^2	2.05×10^2	1.28×10^0	2.20×10^3	3.02×10^{-249}
	SD	2.21×10^2	4.60×10^{-3}	9.48×10^{-4}	4.13×10^{-4}	9.30×10^1	2.90×10^2	4.25×10^{-1}	3.78×10^2	4.22×10^{-95}
$f_{11}(X)$	AB	1.19×10^1	2.22×10^1	9.05×10^0	9.76×10^0	1.46×10^1	1.07×101	1.06×10^1	2.42×10^1	8.87×10^0
	MD	1.20×10^1	2.04×10^1	9.11×10^0	9.76×10^0	1.45×10^1	1.07×101	1.06×10^1	2.38×10^1	8.94×10^0
	SD	7.80×10^{-1}	7.65×10^0	2.93×10^{-1}	5.31×10^{-1}	6.82×10^{-1}	5.88×10^{-1}	3.41×10^{-1}	3.00×10^0	4.67×10^{-1}
$f_{12}(X)$	AB	2.14×10^2	4.59×10^1	1.34×10^{-7}	5.83×10^{-5}	1.67×10^3	5.13×10^{-5}	4.93×10^1	1.41×10^4	2.94×10^{-2}
	MD	1.87×10^2	4.25×10^1	1.18×10^{-7}	2.34×10^{-5}	1.64×10^3	1.20×10^{-5}	4.94×10^1	1.35×10^4	2.69×10^{-2}
	SD	9.41×10^1	1.62×10^1	6.25×10^{-8}	9.00×10^{-5}	4.37×10^2	8.97×10^{-5}	3.85×10^0	3.78×10^3	2.16×10^{-2}
$f_{13}(X)$	AB	2.31×10^2	9.37×10^1	1.06×10^2	1.52×10^2	9.34×10^1	4.84×10^1	1.09×10^2	2.60×10^2	0.00×10^0
	MD	2.31×10^2	8.76×10^1	1.60×10^2	1.47×10^2	9.53×10^1	4.78×10^1	1.07×10^2	2.61×10^2	0.00×10^0
	SD	1.37×10^1	4.21×10^1	7.77×10^1	2.37×10^1	9.73×10^0	1.89×10^1	1.50×10^1	1.22×10^1	0.00×10^0
$f_{14}(X)$	AB	1.54×10^3	5.46×10^1	1.89×10^1	3.04×10^1	1.58×10^4	2.96×10^4	1.92×10^2	2.32×105	9.57×10^{-1}
	MD	1.45×10^3	2.94×10^1	1.08×10^1	2.56×10^1	1.61×10^4	2.53×10^3	1.94×10^2	2.30×105	1.52×10^{-5}
	SD	5.41×10^2	3.99×10^1	2.51×10^1	1.71×10^1	3.38×10^3	3.75×10^4	3.92×10^1	5.10×10^4	5.24×10^0
$f_{15}(X)$	AB	5.59×10^1	4.44×10^1	4.21×10^{-7}	1.20×10^1	4.04×10^1	6.81×10^0	1.10×10^1	5.12×10^1	5.67×10^{-71}
	MD	5.58×10^1	4.51×10^1	3.88×10^{-7}	1.19×10^1	4.03×10^1	7.12×10^0	1.09×10^1	5.14×10^1	1.84×10^{-128}
	SD	4.41×10^0	6.67×10^0	1.58×10^{-7}	5.37×10^0	2.26×10^0	2.31×10^0	1.20×10^0	2.40×10^0	3.03×10^{-70}
$f_{16}(X)$	AB	1.99×10^{25}	2.14×10^{32}	2.05×10^{-7}	2.09×10^{-5}	1.56×10^2	2.29×10^2	1.00×10^1	6.2×10^{29}	3.09×10^{-32}
	MD	2.76×10^{23}	5.97×10^{26}	2.02×10^{-7}	5.91×10^{-6}	1.63×10^2	2.14×10^2	1.00×10^1	1.72×10^{29}	1.09×10^{-82}
	SD	8.15×10^{25}	8.30×10^{32}	6.77×10^{-8}	6.57×10^{-5}	2.94×10^1	9.79×10^1	0.00×10^0	1.03×10^{30}	1.69×10^{-31}
$f_{17}(X)$	AB	9.00×10^0	1.94×10^4	0.00×10^0	1.33×10^{-1}	4.79×10^3	3.33×10^{-2}	3.38×10^1	1.87×10^4	0.00×10^0
	MD	9.00×10^0	1.95×10^4	0.00×10^0	0.00×10^0	4.71×10^3	0.00×10^0	3.40×10^1	1.85×10^4	0.00×10^0
	SD	2.63×10^0	6.37×10^3	0.0						

$f_{25}(X)$	AB	5.73×10^{-1}	1.73×10^{-1}	1.41×10^{-18}	4.03×10^{-15}	5.88×10^2	6.33×10^1	2.02×10^0	2.47×10^3	2.28×10^{-69}
	MD	5.12×10^{-1}	1.77×10^{-3}	1.16×10^{-18}	1.52×10^{-15}	5.87×10^2	3.94×10^{-11}	1.94×10^0	2.63×10^3	5.12×10^{-217}
	SD	2.27×10^{-1}	5.86×10^1	8.43×10^{-19}	8.45×10^{-15}	6.35×10^1	1.10×10^2	4.67×10^{-1}	4.05×10^2	1.25×10^{-68}
$f_{26}(X)$	AB	6.46×10^{-4}	1.02×10^{-8}	6.08×10^{-10}	6.69×10^{-12}	3.03×10^{-5}	1.07×10^{-23}	3.99×10^{-11}	8.21×10^{-3}	1.02×10^{-32}
	MD	4.18×10^{-4}	1.01×10^{-8}	4.14×10^{-10}	5.52×10^{-18}	3.03×10^{-5}	9.47×10^{-29}	3.94×10^{-11}	7.48×10^{-3}	1.56×10^{-48}
	SD	6.92×10^{-4}	3.84×10^{-9}	5.48×10^{-10}	3.57×10^{-11}	1.41×10^{-5}	5.82×10^{-23}	2.36×10^{-11}	4.28×10^{-3}	5.61×10^{-32}
$f_{27}(X)$	AB	7.03×10^2	3.02×10^{-4}	3.60×10^{-9}	4.45×10^{-8}	4.46×10^3	1.34×10^3	1.53×10^3	1.97×10^4	2.24×10^{-46}
	MD	1.25×10^2	2.83×10^4	3.66×10^{-9}	3.97×10^{-8}	4.46×10^3	4.28×10^{-7}	1.35×10^3	2.02×10^4	3.97×10^{-83}
	SD	2.50×10^3	2.10×10^4	1.20×10^{-9}	2.67×10^{-8}	7.40×10^2	3.47×10^3	7.46×10^2	3.07×10^3	1.23×10^{-45}
$f_{28}(X)$	AB	2.11×10^2	5.39×10^2	3.26×10^1	3.63×10^1	3.44×10^2	6.60×10^1	1.61×10^2	7.33×10^2	2.90×10^1
	MD	2.14×10^2	5.47×10^2	2.90×10^1	3.69×10^1	3.46×10^2	6.35×10^1	1.60×10^2	7.39×10^2	2.90×10^1
	SD	2.12×10^1	1.28×10^2	6.51×10^0	7.08×10^0	2.77×10^1	1.81×10^1	1.31×10^1	6.32×10^1	5.83×10^{-5}
$f_{29}(X)$	AB	2.72×10^5	3.27×10^7	3.20×10^1	3.46×10^1	6.61×10^6	9.52×10^1	3.12×10^2	6.78×10^7	3.20×10^1
	MD	2.11×10^5	1.44×10^7	3.20×10^1	3.32×10^1	6.63×10^6	8.72×10^1	3.06×10^2	7.05×10^7	3.20×10^1
	SD	2.46×10^5	4.23×10^7	5.64×10^{-10}	4.28×10^0	1.98×10^6	3.66×10^1	4.74×10^{-1}	2.24×10^7	3.73×10^{-6}
$f_{30}(X)$	AB	3.47×10^2	9.24×10^2	3.03×10^1	3.24×10^1	3.44×10^2	7.46×10^1	3.54×10^2	9.18×10^2	2.90×10^1
	MD	3.34×10^2	9.31×10^2	2.90×10^1	2.90×10^1	3.49×10^{-2}	5.48×10^1	3.42×10^2	9.14×10^2	2.90×10^1
	SD	7.40×10^1	3.06×10^2	3.98×10^0	6.44×10^0	3.14×10^1	6.80×10^1	5.84×10^1	1.13×10^2	0.00×10^0

Tabla 1. Comparación en términos de calidad de soluciones.

5. Conclusiones

Este artículo presenta un algoritmo metaheurístico inspirado en la teoría de juegos evolutiva (TJE) aplicado a poblaciones constituidas. El enfoque Empieza estableciendo a los sujetos en regiones adecuadas del espacio de búsqueda utilizando el algoritmo Metropolis-Hastings, determinando a cada persona una estrategia de búsqueda única. A continuación, la población se divide en subpoblaciones utilizando el algoritmo K-means, lo que permite la adaptación localizada de la estrategia. Esta estructura mantiene la diversidad al permitir que las estrategias dominen dentro de los grupos, al tiempo que preserva la variedad en toda la población. La implementación de diferentes estrategias fomenta diversos comportamientos exploratorios, lo que mejora la capacidad del algoritmo para explorar a fondo el espacio de búsqueda. En comparación con ocho metaheurísticas conocidas en 30 funciones de referencia, el método propuesto demostró una convergencia, una calidad de solución y una adaptabilidad superiores en varias dimensiones. Estos prometedores resultados sugieren que el método puede ser versátil y eficaz para resolver una amplia gama de problemas de optimización del mundo real.

El trabajo futuro debería ampliar y probar aún más su aplicabilidad en diversos ámbitos problemáticos para comprender mejor su rendimiento y adecuación. Como trabajo futuro, surgen dos prometedoras líneas de investigación a partir de abordar las limitaciones de combinar TJE con metaheurística. La primera línea propone el desarrollo de mecanismos de agrupación adaptativos que puedan reestructurar dinámicamente los clústeres basándose en la diversidad individual y las métricas de rendimiento a lo largo del tiempo. Estos mecanismos tienen como objetivo mejorar el equilibrio entre la exploración y la explotación, especialmente en espacios de búsqueda de alta dimensión, adaptando continuamente la estructura de agrupación en respuesta a los cambios en el panorama de búsqueda. La segunda dirección sugiere la integración de dinámicas de competencia multinivel, en las que la intensidad de la competencia varía entre las escalas global, regional y local. Este enfoque jerárquico permitiría a los individuos interactuar más allá de sus clústeres inmediatos, lo que fomentaría comparaciones estratégicas más amplias. Al permitir interacciones más diversas, esta estructura podría mejorar la exploración global y minimizar el riesgo de estancamiento, lo que la hace muy adecuada para entornos problemáticos complejos.

	Nombre	Función	S	Mínimo
$f_1(X)$	Ackley	$-20 \exp \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos \cos(2\pi x_i) \right) + 20 + \exp(1)$	$[-30,30]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_2(X)$	Dixon	$(x_1 - 1)^2 + \sum_{i=1}^n i(2x_i^2 - x_{i-1})^2$	$[-10,10]^n$	$f(x^*) = 0;$ $x_i^* = 2^{-\frac{2^{i-2}}{2^i}}, i = 1, \dots, n$
$f_3(X)$	Griewank	$\sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600,600]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_4(X)$	Infinity	$\sum_{i=1}^n x_i^6 \sin(x_i + 2)$	$[-1,1]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_5(X)$	Mishra 1	$(1 + x_n)^n;$ $x_n = n - \sum_{i=1}^{n-1} x_i$	$[0,1]^n$	$f(x^*) = 2;$ $x^* = (1, \dots, 1)$
$f_6(X)$	Mishra 2	$(1 + x_n)^{x_n}; x_n = n - \sum_{i=1}^{n-1} \frac{x_i + x_{i+1}}{2}$	$[0,1]^n$	$f(x^*) = 2;$ $x^* = (1, 1, \dots, 1)$
$f_7(X)$	Penalty 1	$\frac{\pi}{n} X \left\{ 10(\pi \phi_1) + \sum_{i=1}^{n-1} (\phi_i - 1)^2 [1 + 10(\pi \phi_{i+1})] + (\phi_n - 1)^2 \right\} +$ $\sum_{i=1}^n u(x_i, a, k, m) \phi_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) =$ $k(x_i - a)^m \text{ if } -x_i \leq a \text{ if } -a \leq x_i k(-x_i - a)^m \text{ if } x_i < a$ $a = 10, k = 100, m = 4$	$[-50,50]^n$	$f(x^*) = 0;$ $x^* = (-1, \dots, -1)$
$f_8(X)$	Penalty 2	$0.1 X \left\{ (3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\}$ $+ \sum_{i=1}^n u(x_i, a, k, m)$ $u(x_i, a, k, m) = k(x_i - a)^m \text{ if } x_i > a \text{ if } -a \leq x_i k(-x_i - a)^m \text{ if } x_i < a$ $a = 5, k = 100, m = 4$	$[-50,50]^n$	$f(x^*) = 0;$ $x^* = (1, \dots, 1)$
$f_9(X)$	Plateau	$30 + \sum_{i=1}^n x_i $	$[-5.12, 5.12]^n$	$f(x^*) = 30;$ $x^* = (0, \dots, 0)$
$f_{10}(X)$	Powell	$\sum_{i=1}^{\frac{n}{4}} [(x_{4i-3})^2 + 5(x_{i-1} + x_{4i})^2 + (x_{4i-2} + 2x_{4i-1})^4 + 10(x_{4i-3} + 2x_{4i-1})^4]$	$[-4,5]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{11}(X)$	Quartic	$\sum_{i=1}^n i x_i^4 + \text{rand}(0,1)$	$[-1.28, 1.28]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{12}(X)$	Quintic	$\sum_{i=1}^n x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4 $	$[-10,10]^n$	$f(x^*) = 0;$ $x^* = (-1, \dots, -1)$
$f_{13}(X)$	Rastrigin	$10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$[-5.12, 5.12]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{14}(X)$	Rosenbrock	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-5,10]^n$	$f(x^*) = 0;$ $x^* = (1, \dots, 1)$
$f_{15}(X)$	Schwefel21	$418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$[-100,100]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{16}(X)$	Schwefel22	$\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100,100]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{17}(X)$	Step	$\sum_{i=1}^n (x_i + 0.5)^2$	$[-100,100]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{18}(X)$	Stybtang	$\frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	$[-5,5]^n$	$f(x^*) = -39.1659n;$ $x^* = (-2.90, \dots, -2.90)$
$f_{19}(X)$	Trid	$\sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	$[-n^2, n^2]^n$	$f(x^*) = -\frac{n(n+4)(n-1)}{6}$ $x^* = [i(n+1-i)]$ for $i = 1, \dots, n$

$f_{20}(X)$	Vincent	$-\sum_{i=1}^n \sin(10 \log x_i)$	$[0.25, 10]^n$	$f(x^*) = -n;$ $x^* = (7.70, \dots, 7.70)$
$f_{21}(X)$	Zakharov	$\sum_{i=1}^n x_i^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^2 + \left(\sum_{i=1}^n 0.5 i x_i \right)^4$	$[-5, 10]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{22}(X)$	Rothyp	$\sum_{i=1}^n \sum_{j=1}^i x_j^2$	$[-65.536, 65.536]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{23}(X)$	Schwefel2	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	$[-100, 100]^n$
$f_{24}(X)$	Sphere	$\sum_{i=1}^n x_i^2$	$[-5, 5]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{25}(X)$	Sum 2	$\sum_{i=1}^d i x_i^2$	$[-10, 10]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{26}(X)$	Sum of different powers	$\sum_{i=1}^n x_i ^{i+1}$	$[-1, 1]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{27}(X)$	Rastrigin+ Schwefel2+ Sphere	$10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)] + \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 + \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	$f(x^*) = 0;$ $x^* = (0, \dots, 0)$
$f_{28}(X)$	Griewank+ Rastrigin+ Rosenbrock	$\sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 + 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$ $+ \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-100, 100]^n$	$f(x^*) = n - 1;$ $x^* = (0, \dots, 0)$
$f_{29}(X)$	Ackley+ Penalty2+ Rosenbrock+ Schwefel2	$f(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp \exp(1)$ $+ (0.1 \left\{ \sin \sin(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 (1 + (3\pi x_{i+1} + 1)) + (x_n - 1)^2 (1 + (2\pi x_n)) \right\})$ $+ \sum_{i=1}^n u(x_i, 5.100, 4) + \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	$f(x^*) = (1.1n) - 1;$ $x^* = (0, \dots, 0)$
$f_{30}(X)$	Ackley+ Griewank+ Rastrigin+ Rosenbrock+ Schwefel22	$-20 \exp \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos \cos(2\pi x_i) \right) + 20 + \exp \exp(1)$ $+ \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 + 10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$ $+ \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] + \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100, 100]^n$	$f(x^*) = n - 1;$ $x^* = (0, \dots, 0)$

6. Referencias

- Abdel-Basset, M., Abdel-Fatah, L., y Sangaiah, A. K. (2018). Metaheuristic algorithms: A comprehensive review. En *Computational intelligence for multimedia big data on the cloud with engineering applications* (pp. 185–231). Academic Press.
- Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z., y Jawawi, D. N. A. (2016). Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, 26, 8–22.
- Afzal, A., Buradi, A., Jilte, R., Shaik, S., Kaladgi, A. R., Arıcı, M., Lee, C. T., y Nizetić, S. (2023). Optimizing the thermal performance of solar energy devices using meta-heuristic algorithms: A critical review. *Renewable and Sustainable Energy Reviews*, 173, 112903.
- Ahmed, M., Seraj, R., y Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9, 1295.
- Almufti, S. M., Marques, R. B., y Saeed, V. A. (2019). Taxonomy of bio-inspired optimization algorithms. *Journal of Advanced Computer Science and Technology*, 8, 23.
- Askarzadeh, A. (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, 169, 1–12.
- Chopard, B., y Tomassini, M. (2018). *An introduction to metaheuristics for optimization*. Springer.
- Cuevas, E., Cienfuegos, M., Zaldívar, D., y Pérez-Cisneros, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, 40, 6374–6384.
- Cuevas, E., Echavarría, A., y Ramírez-Ortegón, M. A. (2014). An optimization algorithm inspired by the states of matter that improves the balance between exploration and exploitation. *Applied Intelligence*, 40, 256–272.
- Cuevas, E., Escobar, H., Sarkar, R., y Eid, H. F. (2022). A new population initialization approach based on Metropolis–Hastings (MH) method. *Applied Intelligence*, 53, 16575–16593.
- Dokeroglu, T., Sevinc, E., Kucukyilmaz, T., y Cosar, A. (2019). A survey on new generation metaheuristic algorithms. *Computers & Industrial Engineering*, 137, 106040.
- Geem, Z. W., Kim, J. H., y Loganathan, G. V. (2001). A new heuristic optimization algorithm: Harmony search. *Simulation*, 76, 60–68.
- Gintis, H. (2000). *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton University Press.
- Giri, A. R., Chen, T., Rajendran, V. P., y Khamis, A. (2022). A metaheuristic approach to emergency vehicle dispatch and routing. En *Proceedings of the 2022 IEEE International Conference on Smart Mobility (SM)* (pp. 27–31). IEEE.
- Grüne-Yanoff, T. (2011). Evolutionary game theory, interpersonal comparisons and natural selection: A dilemma. *Biology & Philosophy*, 26, 637–654.
- Hansen, N., y Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. En *Proceedings of the IEEE International Conference on Evolutionary Computation*. IEEE.
- Holland, J. H. (1984). Genetic algorithms and adaptation. En *Adaptive control of ill-defined systems*. Springer.
- Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Erciyes University.
- Kaur, S., Kumar, Y., Koul, A., y Kamboj, S. K. (2023). A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: Open issues and challenges. *Archives of Computational Methods in Engineering*, 30, 1863–1895.
- Kennedy, J., Eberhart, R., y Gov, B. (1995). Particle swarm optimization. En *Encyclopedia of machine learning* (pp. 760–766). Springer.
- Kirkpatrick, S., Gelatt, C. D., y Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
- Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120–133.
- Mirjalili, S., y Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, 95, 51–67.
- Mirjalili, S., Mirjalili, S. M., y Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.
- Osuna-Enciso, V., Cuevas, E., y Castañeda, B. M. (2022). A diversity metric for population-based metaheuristic algorithms. *Information Sciences*, 586, 192–208.

- Rashedi, E., Nezamabadi-Pour, H., y Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179, 2232–2248.
- Stella, L., y Bauso, D. (2017). Evolutionary game dynamics for collective decision making in structured and unstructured environments. *IFAC-PapersOnLine*, 50, 11914–11919.
- Storn, R., y Price, K. (1995). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Australasian Plant Pathology*, 38, 284–287.
- Vaziri, E., Dehdar, F., y Abdoli, M. R. (2023). Feasibility study of using meta-heuristic algorithms on optimizing of the integrated risk in banking system. *International Journal of Finance, Management and Accounting*, 8, 143–158.
- Weibull, W. (1997). *Evolutionary game theory*. MIT Press.
- Wolpert, D. H., y Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1, 67–82.
- Yang, X.-S. (2010a). A new metaheuristic bat-inspired algorithm. En *Nature inspired cooperative strategies for optimization (NISCO 2010)* (*Studies in Computational Intelligence*, Vol. 284, pp. 65–74). Springer.
- Yang, X.-S. (2010b). *Engineering optimization: An introduction with metaheuristic applications*. Wiley.
- Yang, X.-S., y Deb, S. (2009). Cuckoo search via Lévy flights. En *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 210–214). IEEE.



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