Vol. 13 Núm. 1 ISSN 2007-5448

RECIBE

Revista electrónica

DE COMPUTACIÓN, INFORMÁTICA, BIOMÉDICA Y ELECTRÓNICA

Índice

Computación e Informática

Metropolis-Hastings (MH): Una Perspectiva Innovadora en la Inicialización de Poblaciones C1
Oscar Francisco Barba Toscano, Eric Ricardo Lopez Marin, Hector Joaquin Escobar Cuevas, Erik Valdemar Cuevas Jimenez, Miguel Angel Alejandro Islas Toski

Un nuevo enfoque de optimización basado en la teoría evolutiva de juegos no estructurada C2
Marcos Mario Vasquez Franco, Nahum Juda Aguirre Rodriguez, Hector Joaquin Escobar Cuevas, Alberto Luque Chang, Erik Valdemar Cuevas Jimenez

Segmentación rápida de imágenes con múltiples características C3
Sergio Gomez-Vega, Alberto Luque Chang, Hector Joaquin Escobar Cuevas, Fernando Vega Parra

Estudio bibliográfico sobre las metodologías de big data y su aplicación en la movilidad inteligente C4
Miguel Andrés Viteri Chica, Chancay García Leonardo

Electrónica

Identificación y Clasificación de Imágenes Farmacéuticas mediante Técnicas de Visión e Inteligencia Artificial: Un Enfoque en Máquinas de Vectores de Soporte (SVM) E1
Jesus Alfonso Medrano Hermosillo, Dr. Rodriguez Mata, Dr. Djilali, Dr. Edgar Macias , Dr. Oscar Suárez, Raul Vazquez

COMPUTATION E INFORMATICA

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 Marin[1]
Hector Joaquin Escobar Cuevas[1]
Erik Valdemar Cuevas Jimenez[1]
Miguel Angel Alejandro Islas Toski[1]

[1]Universidad 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 fungen 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 seguido 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 $P(\mathbf{x})$ usando una función $f(\mathbf{x})$ relacionada con $P(\mathbf{x})$**.** Este produce una secuencia de puntos muestrales tales que un gran valor de $f(\mathbf{x})$ corresponden a una alta probabilidad de que $\mathbf{x}$ 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 $\mathbf{x}$ con un valor $f(\mathbf{x})$ mayor, evitando el muestro de zonas con un valor $f(\mathbf{x})$ 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 $J(\mathbf{x})$ aplicando la técnica de Metropolis-Hastings. Por ende, el conjunto inicial de soluciones obtenidas por el algoritmo MH representa los puntos muestrales $\mathbf{x}$ con un alto valor de $J(\mathbf{x})$. 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 $f(\mathbf{x})$ 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 (descardada, 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 $f(\mathbf{x})$ donde $\mathbf{x}$ corresponde a un valor en el espacio de d dimensiones. Entonces, una transición de las Cadenas de Márkov del estado $\mathbf{x}^k$ en el instante k al nuevo estado $\mathbf{x}^{k+1}$ se implementa en dos fases: producción de una solución propuesta y probabilidad de aceptación. Una solución propuesta $\mathbf{y}$ se genera a partir de una densidad de transición. Por lo general, $\mathbf{y}$ es generada como una perturbación del estado actual $\mathbf{x}^k$ con un valor aleatorio producido por una distribución normal $N(0,1)$. Una vez que la solución propuesta $\mathbf{y}$ se ha generado, se usa un mecanismo probabilístico para decidir si $\mathbf{y}$ se acepta como el nuevo estado $\mathbf{x}^{k+1}$. La probabilidad de aceptación α de la solución propuesta $\mathbf{y}$

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 bines 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.

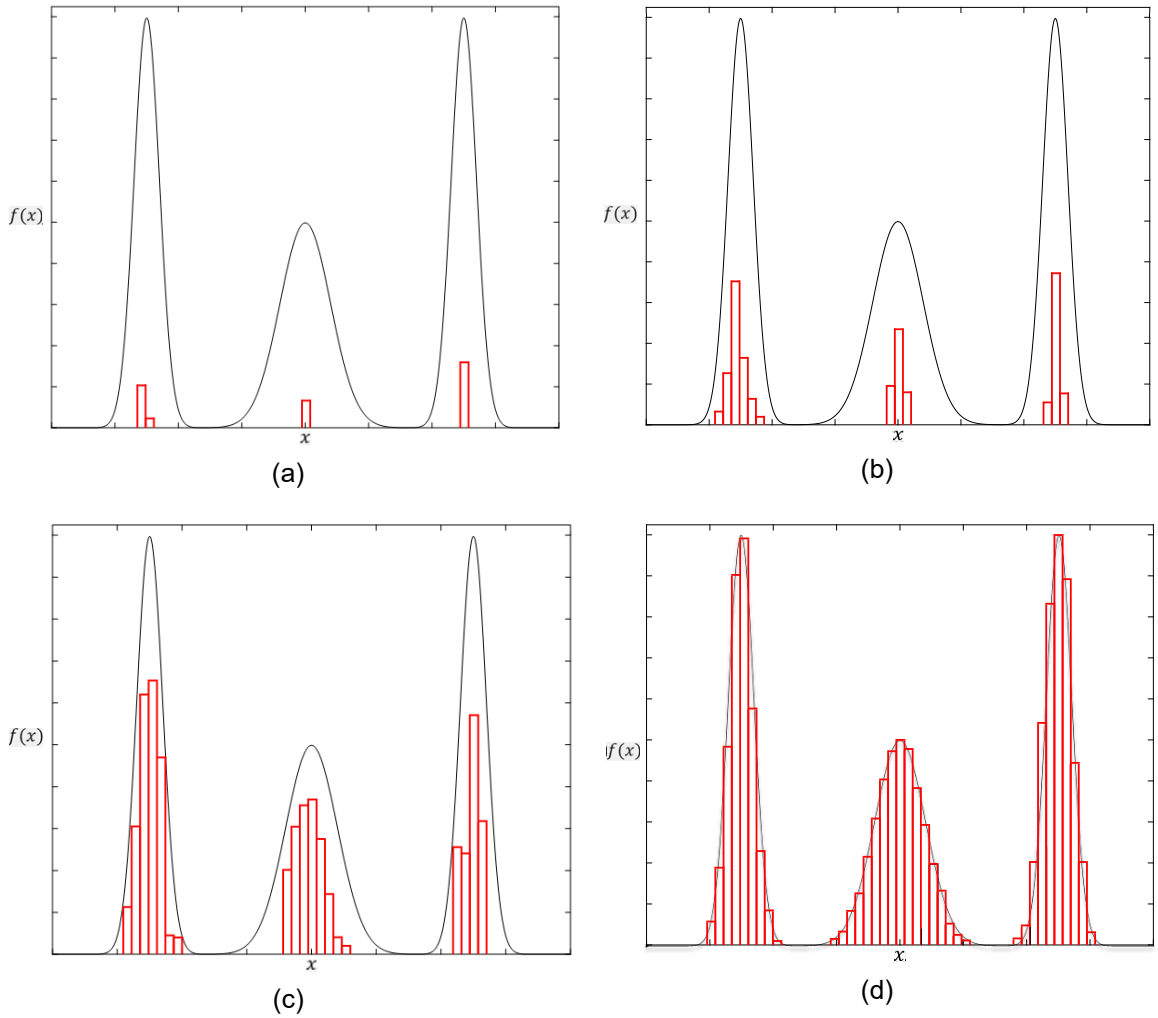


Figura 1 Histogramas de muestras producidas por el método MH extraído de la distribución objetivo f(x) de la Ecuación **¡Error! No se encuentra el origen de la referencia.**. Para (a) 10, (b) 100, (c) 1000, y (d) 10000 muestras

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).

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

Donde $J(\mathbf{x}): \mathbb{R}^d \to \mathbb{R}$ 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):

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

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 asocien las soluciones. Los operadores de mutación más populares se definen de acuerdo con las Ecuaciones (5), (6), (7) y (8).

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

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

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

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

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 $\left(\mathbf{x}_{r_1} \neq \mathbf{x}_{r_2} \neq \mathbf{x}_{r_3} \neq \mathbf{x}_{r_4} \neq \mathbf{x}_{r_5}\right)$ seleccionadas aleatoriamente de la población actual $\left(\mathbf{x}_{r_1}, \mathbf{x}_{r_2}, \mathbf{x}_{r_3}, \mathbf{x}_{r_4}, \mathbf{x}_{r_5} \in \{1, \dots, n\}\right)$, $\mathbf{x}_{Best,t}$ 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).

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

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 $x_{i,j,t}$ (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 origina $\mathbf{x}_{i,t}$. 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).

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

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 $\mathbf{y}$ 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 $n, \mathbf{X}$
2. $I = \{\emptyset\}, i = 1$
3. Generar una semilla de solución inicial $\mathbf{x}_0 \in \mathbf{X}$
4. **While** $(i \leq n)$
5. $\mathbf{y} = \mathbf{x}_{i-1}$**+** $N(0,1)$
6. $\alpha = min\left\{\frac{J(\mathbf{y})}{J(\mathbf{x}_{i-1})}, 1\right\}$
7. Generar un número aleatorio uniforme $r \in U[0,1]$
8. Si $(r < \alpha)$, entonces
9. $\mathbf{x}_i = \mathbf{y}$
10. $I = \{I \cup \mathbf{x}_i\}$
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 cásico 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 robustes, 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
	AB	0.031043522	3.96821E-08	5.70285E-08	0.031043535	**6.43886E-09**
$f(x)_1$	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**
	AB	0.697808011	0.700534565	0.748811576	0.700735233	**0.513027451**
$f(x)_2$	MD	0.666666667	0.666666669	0.666666667	0.666666668	**0.666666666**
	SD	0.097049446	0.088071999	0.355137521	0.120344378	**0.07217346**
	AB	0.001642553	0.00156125	0.0021304	0.001561004	**0.001187696**
$f(x)_3$	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
	AB	1.87786E-07	2.31948E-08	9.47646E-09	3.24938E-09	**0**
$f(x)_4$	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**
	AB	0.154202856	0.127121622	0.139280775	0.111976783	**0.102971194**
$f(x)_5$	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
	AB	2	2	2	2	2
$f(x)_6$	MD	**2**	**2**	**2**	**2**	**2**
	SD	0	0	0	0	0
	AB	2	2	2	2	2
$f(x)_7$	MD	**2**	**2**	**2**	**2**	**2**
	SD	0	0	0	0	0
	AB	1.86241E-14	9.07778E-15	1.34002E-13	6.06243E-17	**3.69202E-15**
$f(x)_8$	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
	AB	0	0	1.1417E-123	0	0
$f(x)_9$	MD	**0**	**0**	1.3193E-145	**0**	**0**
	SD	0	0	6.2531E-123	0	0
	AB	1390.30741	84.69369769	73.51647998	5867.81578	**72.38309382**
$f(x)_{10}$	MD	73.68667036	73.72968645	73.55230648	73.68669066	**71.03788422**
	SD	7210.702813	58.0087577	2.250862628	31705.89613	**3.990924703**
	AB	107.6940319	106.1665177	723.7773275	106.4781406	**105.5287341**
$f(x)_{11}$	MD	**105.5876108**	106.6846538	106.3200494	107.3751166	105.9476506
	SD	15.43514499	4.58932148	3386.715099	**3.255277603**	3.759620619
	AB	30	30	30	30	**30**
$f(x)_{12}$	MD	**30**	**30**	**30**	**30**	**30**
	SD	0	0	0	0	**0**
	AB	0.000802734	0.000840848	0.000759216	0.000694274	**0.00053463**
$f(x)_{13}$	MD	0.000593663	0.000797071	0.000680998	0.000656195	**0.000527303**
	SD	0.000530664	0.000443019	0.000372237	0.000319378	**0.00028228**
	AB	4.900254181	0.132538855	0.216299179	12433.58502	**0.000430466**
$f(x)_{14}$	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**
	AB	0.00023298	0.00460782	9.94022E-05	9.47279E-05	**1.87624E-05**
$f(x)_{15}$	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**
	AB	1.56861E-13	1.76761E-13	1.13274E-13	1.7634E-13	**6.87534E-16**
$f(x)_{16}$	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**
	AB	3.88077E-12	3.93626E-12	9.6417E-12	4.06929E-12	**7.81259E-15**
$f(x)_{17}$	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**
	AB	1.35112E-16	8.03645E-17	6.34852E-17	6.00462E-17	**4.20512E-17**
$f(x)_{18}$	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**
	AB	3.25446E-15	2.87023E-15	1.90039E-15	4.12796E-15	**1.22464E-15**
$f(x)_{19}$	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**
	AB	3.79728E-12	2.23714E-11	3.30447E-13	**1.89544E-12**	5.42588E-10
$f(x)_{20}$	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
	AB	4.68993E-08	3.95826E-08	3.37765E-08	4.07203E-08	**9.13034E-10**
$f(x)_{21}$	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 $\boldsymbol{f(x)_6}$, $\boldsymbol{f(x)_7}$, $\boldsymbol{f(x)_9}$, $\boldsymbol{f(x)_{12}}$. 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 $\boldsymbol{f(x)_{20}}$.

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):

$$SR = \frac{Número\ de\ ejecuciones\ exitosas}{Número\ de\ pruebas\ totales} \tag{11}$$

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):

$$ARE_{A/B} = \frac{NFC_A}{NFC_B} \tag{12}$$

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.

	$ARE_{A/B}$			
Function	$ARE_{DE/MH}$	$ARE_{DE_OP/MH}$	$ARE_{DE_GOP/MH}$	$ARE_{DE_AR/MH}$
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) $\boldsymbol{f(x)_1}$, (b) $\boldsymbol{f(x)_2}$, (c) $\boldsymbol{f(x)_{12}}$, (d) $\boldsymbol{f(x)_{13}}$, (e) $\boldsymbol{f(x)_{14}}$, y (f) $\boldsymbol{f(x)_{15}}$. Una inspección detallada de los resultados demuestra que el esquema propuesto presenta una mejor convergencia a comparación de los otros algoritmos.

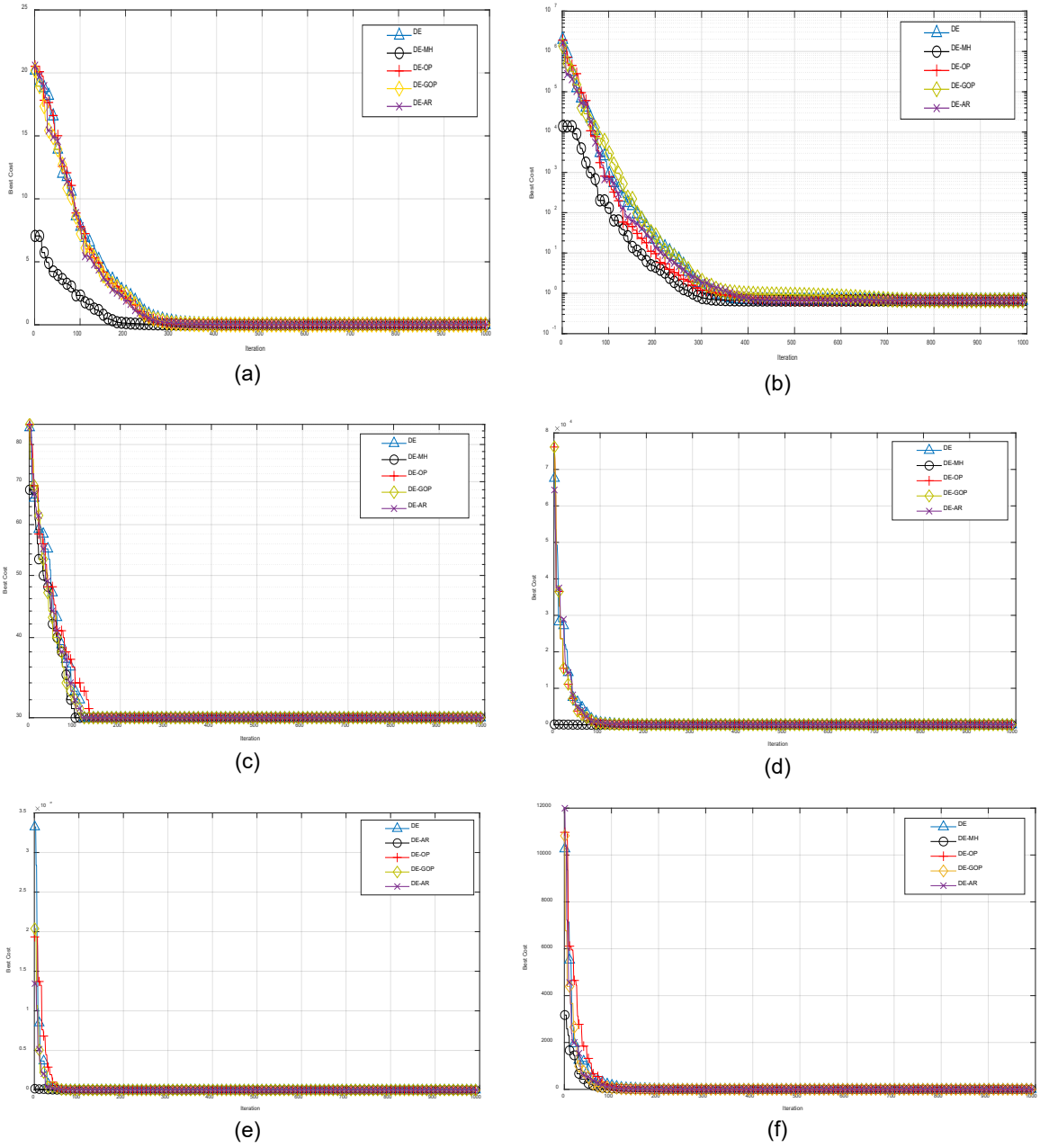


Figura 2 Gráficas de curva de convergencia para las funciones (a) fx1, (b) fx2, (c) fx12, (d) fx13, (e) fx14, y (f) fx15

6. Conclusiones

En este artículo, se propuse 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., & Vandekerkhove, 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., Zaldívar, 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
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

COMPUTATION E INFORMATICA

Recibido 08 Feb. 2024 *ReCIBE, Año 13 No. 1, May. 2024*

Aceptado 13 Feb. 2024

Un nuevo enfoque de optimización basado en la teoría de juegos evolutiva no estructurada

A New Optimization Approach Based on Unstructured Evolutionary Game Theory

Marcos Mario Vasquez Franco[1*]
marcos.vasquez7891@alumnos.udg.mx
Nahum Juda Aguirre Rodriguez[1]
Hector Joaquin Escobar Cuevas[1]
Alberto Luque Chang[1]
Erik Valdemar Cuevas Jimenez[1]

[1]Universidad de Guadalajara
[*]Autor para correspondencia

Resumen

Proponer nuevos métodos metaheurísticos es crucial para la mejora continua en el desarrollo de algoritmos y la capacidad de abordar con eficacia problemas de optimización del mundo real cada vez mas complejos. Por otro lado, la Teoría Evolutiva de Juegos analiza cómo a través de la competencia es posible modificar las estrategias de los individuos dentro de una población con el fin de extender los mecanismos exitosos y reducir o eliminar las estrategias menos exitosas. Este artículo presenta un novedoso enfoque de optimización basado en los principios de la Teoría Evolutiva de Juegos. En el método propuesto, todos los individuos se inicializan mediante la técnica Metropolis-Hasting, que sitúa las soluciones en un punto de partida más cercano a las regiones óptimas o casi óptimas del problema. Se asigna una estrategia original a cada individuo de la población. Al tener en cuenta las interacciones y la competencia entre los distintos agentes del problema de optimización, el enfoque modifica las estrategias para mejorar la eficiencia de la búsqueda y encontrar mejores soluciones. Para evaluar el rendimiento de la técnica propuesta, se compara con ocho algoritmos metaheurísticos bien conocidos utilizando 30 funciones de prueba. La metodología propuesta demostró superioridad en términos de calidad de la solución, dimensionalidad y convergencia en comparación con otros enfoques.

Palabras clave: Metaheurística, Teoría de Juegos, Optimización, competición, Metropolis-Hasting.

Abstract

Proposing new metaheuristic methods is crucial for the continuous improvement in algorithm development and the ability to effectively tackle increasingly complex real-world optimization problems. On the other hand, Evolutionary Game Theory analyzes how, through competition, it is possible to modify the strategies of individuals within a population to extend successful mechanisms and reduce or eliminate less successful strategies. This article presents a novel optimization approach based on the principles of Evolutionary Game Theory. In the proposed method, all individuals are initialized using the Metropolis-Hasting technique, which places the solutions at a starting point closer to the optimal or near-optimal regions of the problem. Each individual in the population is assigned an original strategy. By taking into account the interactions and competition among the various agents of the optimization problem, the approach modifies the strategies to improve search efficiency and find better solutions. To evaluate the performance of the proposed technique, it is compared with eight well-known metaheuristic algorithms using 30 test functions. The proposed methodology demonstrated superiority in terms of solution quality, dimensionality, and convergence compared to other approaches. **Keywords:** Metaheuristic, Game Theory, Optimization, Competition, Metropolis-Hasting.

1. Introducción

La optimización [1] se utiliza para determinar la mejor solución para una función objetivo teniendo en cuenta un conjunto dado de restricciones. El proceso de optimización se centra en maximizar o minimizar la función objetivo. Además, pueden clasificarse en optimización clásica o metaheurística [2]. La primera se basa en cálculos matemáticos sistemáticos como los gradientes. Aunque la optimización clásica puede encontrar soluciones adecuadas, tiene varias desventajas, como el estancamiento en óptimos locales, especialmente para problemas complejos no diferenciables.

Las metaheurísticas [3] son aproximaciones estocásticas que incorporan diversos mecanismos para explorar y explotar el espacio de búsqueda con el fin de encontrar una solución factible. Estos algoritmos son técnicas de uso general que no requieren el uso de gradientes. Se han utilizado diferentes taxonomías para clasificar los algoritmos metaheurísticos. El método mas común para clasificarlos se basa en la inspiración. Se pueden clasificar como bio-inspirados o basados en leyes físicas [4].

Los algoritmos metaheurísticos bio-inspirados se han caracterizado por simular principios y mecanismos biológicos. La mayoría de los algoritmos entran en esta categoría. Estas técnicas pueden agruparse en volutivas, de inteligencia de enjambre y de sistema inmune [2]. La colonia de abejas artificiales (ABC) [5], el algoritmo del murcielago (BA) [6], el algoritmo de evolución diferencial (DE) [7], el algoritmo de optimización del enjambre de partículas (PSO) [8], la búsqueda del cuco (CS) [9] y el algorítmo de búsqueda del cuervo (CSA) [10] son algunos ejemplos de algorítmos metaheurísticos bioinspirados. Por otro lado, los métodos metaheurísticos basados en leyes físicas se distinguen por el uso de principios y procesos físicos, matemáticos y químicos. La estrategia de evolución adaptativa de la matriz de covarianza (CMA-ES) [11], el algoritmo de búsqueda armónica (HS) [12], el algoritmo seno-coseno (SCA) [13], l algoritmo de recocido simulado (SA) [14], y la búsqueda de estados de la materia (SMS) [15] son ejemplos de esta categoría. Los algoritmos metaheurísticos han demostrado su superioridad en varias aplicaciones del mundo real en las que no pueden utilizarse técnicas clasicas [16-19]. Sin embargo, ningún algoritmo puede resolver todos los problemas [20]. Además, la complejidad de estos problemas ha aumentado, teniendo en cuenta la no linealidad y la naturaleza dinámica de sus entornos. Aunque se han desarrollado muchos algoritmos metaheurísticos, deben continuar los esfuerzos para crear nuevos métodos que aborden estos complejos retos.

En metaheurística, la inicialización [21] es el proceso de crear una población inicial de soluciones al principio del proceso de optimización. La población inicial sirve como punto de partida para que el algoritmo metaheurístico explore y mejore iterativamente. El obetivo de un método de inicialización es generar un conjunto diverso de soluciones candidatas que cubran una amplia gama del espacio de soluciones, proporcionando una base para que el algoritmo comience su búsqueda de soluciones óptimas o casi óptimas. En la literatura se han documentado ampliamente varios métodos de inicialización [22]. Entre ellos, destaca el método Metropolis-Hasting [23]. Este método en particular introduce una forma única de generar valores iniciales mediante el muestreo de la distribución de probabilidad de la función objetivo, centrándose en concentrase alrededor de sus valores más prometedores

Por otro lado, la Teoría Evolutiva de Juegos (EGT) [24] es una rama de la biología matemática y la economía que trata de comprender la dinámica de las interacciones estratégicas entre individuos o agentes de una población, donde el éxito de la estrategia de cada individuo se ve influido por las estrategias de los demás. Combina principios de la teoría clásica de juegos (GT) con ideas de la biolgía evolutiva para estudiar cómo pueden surgir y evolucionar distintas estrategias a lo largo del tiempo mediante un proceso de interacción. En la Teoría de Juegos tradicional, los individuos toman decisiones para maximizar su propia utilidad o beneficio sin tener en cuenta el impacto de sus acciones en la población. Sin embargo, en la EGT, los individuos se consideran parte de una

población, y sus estrategias están sujetas a cambios y adaptaciones en función de su éxito relativo en comparación con los demás [25]. En la EGT, los individuos de una población interactúan en competiciones repetidas entre sí. En estas competiciones, los individuos comparan y contrastan sus respectivas estrategias. En este proceso competitivo, los individuos evalúan el éxito relativo o la rentabilidad de sus estrategias basándose en los resultados de sus interacciones con los demás. Los resultados de estas comparaciones desempeñan un papel crucial a la hora de determinar la adaptación y la evolución de las estrategias dentro de la población. Según los principios de la EGT, el objetivo primordial es conducir la evolución de las estrategias dentro de una población hacia el éxito y la optimización del comportamiento de los individuos. Esto implica modificar las estrategias de cada individuo de tal manera que se fomente la propagación de las estrategias exitosas y su prevalencia entre la población, mientras que las estrategias menos exitosas disminuyen gradualmente o incluso se eliminan. La EGT tiene aplicaciones [26-28], en diversos campos, como la biología, la economía, la sociología e incluso la informática. Puede ayudar a explicar la aparición y persistencia de determinados comportamientos y estrategias de cooperación observados en la naturaleza y la sociedad.

En la EGT, las interacciones entre los individuos de la población se modelan considerando dos paradigmas: estructurado y no estructurado. Los enfoque estructurados [29] tienen en cuenta la estructura espacial o social de las interacciones entre los individuos de una población. En este contexto, “estructurado” se refiere a los patrones o restricciones específicos que rigen la forma en que los individuos interactúan entre sí, creando una red de relaciones. Los modelos no estructurados [21] consideran escenarios en los que los individuos interactúan entre sí de forma aleatoria, sin patrones ni restricciones específicas. Este esquema de interacción aleatoria suele modelarse utilizando un concepto conocido como supuesto de “campo medio”, en el que cada individuo interactúa con un gran número de otros individuos de la población con la misma probabilidad.

Los principios de la EGT han demostrado su eficacia a la hora de comprender y modelizar cómo evolucionan y se adaptan las estrategias para aumentar el éxito individual dentro de una población. Sin embargo, a pesar de su eficacia en diversas áreas y aplicaciones, el uso directo de los mecanísmos de la EGT para producir nuevos métodos metaheurísticos es muy escaso. Una excepción es el enfoque presentado en [30], donde algunos elementos de la EGT se han integrado en un algoritmo PSO para mejorar su convergencia.

El campo de la optimización es muy diverso y abarca una amplia gama de problemas con complejidades, tamaños y características variables. Aunque algunos problemas pueden tener estructuras y propiedades bien definidas que permiten a algoritmos específicos destacar, no existe un algoritmo único que pueda resolver todos los problemas de optimización de forma competitiva. El desarrollo de nuevos métodos metaheurísticos es, por tanto, crucial para mejorar el rendimiento algorítmico y abordar con eficacia la complejidad cada vez mayor de los problemas de optimización del mundo real. Este artículo presenta un nuevo enfoque de optimización que tiene en cuenta los principios de la EGT. El nuevo enfoque comienza inicializando todos los individuos mediante la técnica Metropolis-Hasting, posicionando las soluciones más cerca de las regiones óptimas o casi óptimas del problema. Tras el proceso de inicialización, cada individuo de la población recibe una estrategia original. A medida que avanza el algoritmo, se producen interacciones y competencia entre estos individuos, lo que permite una adaptación y ajuste continuos de sus estrategias. El objetivo primordial es promover la propagación de las estrategias exitosas en toda la población, al tiempo que se reducen o eliminan gradualmente las fracasadas. Al refinar y actualizar continuamente las estrategias en función de su rendimiento, el algoritmo utliza eficazmente las estrategias de búsqueda más prometedoras para explorar y explotar el espacio de soluciones. Este mecanismo de adaptación estratégica mejora la capacidad del algoritmo para identificar el óptimo global o soluciones cercanas al óptimo en problemas de optimización complejos. El método se comparó con ocho algoritmos metaheurísticos bien conocidos utilizando un conjunto diverso de 30

funciones de prueba comunmente utilizadas en la investigación de la optimización. Los resultados empíricos demostraron la superioridad del método propuesto en múltiples métricas importantes. En términos de calidad de la solución, el enfoque propuesto superó sistemáticamente a los algoritmos competidores, ofreciendo soluciones de mayor calidad y precisión. También demostró su capacidad para converger a soluciones óptimas o casi óptimas con una precisión impresionante, incluso cuando se trata de funciones complejas y multimodales.

Este artículo está organizado como sigue: En la sección 2 se describen los conceptos preliminares. Despues, en la sección 3 se presenta el enfoque de optimización de teoría de juegos evolutiva propuesto. Además, en la sección 4 se analizan y discuten los resultados experimentales. Por útimo, en la sección 5 se abordan las conclusiones y el trabajo futuro.

2. Conceptos preliminares

Esta sección proporciona una introducción completa a los conceptos fundamentales de la teoría de juegos y el método de inicialización Metrópolis-Hasting, que son muy relevantes para el método propuesto. La teoría de juegos es un marco analítico robusto para estudiar interacciones estratégicas entre decisores racionales. Permite modelar y analizar escenarios complejos en los que los resultados dependen de las decisiones tomadas por múltiples actores.

Además, el método de inicialización Metrópolis-Hasting es un enfoque novedoso para los algoritmos metaheurísticos. La fase de inicialización desempeña un papel crucial en la eficiencia y eficacia de los algoritmos metaheurísticos. Sin embargo, a menudo ha recibido menos atención que otros componentes como los operadores de movimiento y los métodos de selección. La integración de estos métodos en el marco propuesto tiene como objetivo crear nuevas estrategias de búsqueda para el proceso de optimización con el fin de aumentar el rendimiento y la velocidad de convergencia, dando lugar a soluciones mejoradas para problemas de optimización complejos.

2.1 Teoría evolutiva de juegos

En la Teoría Evolutiva de Juegos (TEJ) [31], los individuos de una población participan en competiciones recurrentes entre ellos. Durante estas competiciones, los individuos evalúan y contrastan sus propias estrategias. A través de este proceso competitivo, los individuos evalúan el éxito relativo de sus estrategias observando los resultados de su comparación con los demás. Los resultados de estas evaluaciones influyen significativamente en la adaptación y evolución de las estrategias dentro de la población [32].

Durante las competiciones repetidas, los individuos ponen en práctica sus estrategias e interactúan con los demás. Los resultados de estas interacciones generan una medida del éxito o la aptitud de cada estrategia. Los individuos evalúan el rendimiento de sus estrategias en relación con las de los demás y observan el éxito asociado a cada estrategia. Basándose en estas comparaciones y observaciones, los individuos pueden adaptar y modificar sus estrategias para mejorar su éxito relativo. Las estrategias que producen mayores beneficios o superan a otras en la competición tienen más posibilidades de mantenerse y propagarse en la población. Por el contrario, las estrategias menos exitosas pueden ser descartadas o sustituidas con el tiempo. El proceso de comparar estrategias y adaptarlas en función de los resultados de las competiciones es fundamental en la TEJ. Impulsa la dinámica de la evolución de las estrategias dentro de la población. A medida que las estrategias exitosas se hacen más frecuentes, la población en su conjunto evoluciona, favoreciendo el predominio de estrategias que conducen a una mayor aptitud en el entorno dado.

En TEJ, los procesos más comunes y cruciales se consiguen mediante la comparación por pares y la actualización probabilística de estrategias. Considerando una población de n elementos, el proceso comienza con un individuo i que selecciona aleatoriamente otro elemento j de la población (i,j ∈ 1, ...,n). A continuación, se comparan sus respectivas estrategias y, si se comprueba que la estrategia de i es peor que la del elemento j (es decir, que la estrategia de j produce una aptitud mayor), el individuo i tiene dos acciones posibles [25]. En primer lugar, puede imitar la estrategia de j, adoptando el mismo enfoque que el elemento más exitoso. Alternativamente, i puede alterar su estrategia para parecerse más a la de j, con el objetivo de aumentar su rendimiento. La decisión de cambiar la estrategia de i se toma de forma probabilística, siendo la probabilidad de actualización directamente proporcional a las diferencias de estrategias entre i y j. Este enfoque probabilístico permite la exploración en la búsqueda de mejores estrategias y, a lo largo de las iteraciones, las estrategias exitosas tienden a extenderse y dominar en la población, mientras que las menos exitosas disminuyen.

TEJ considera dos paradigmas para modelar las interacciones entre individuos dentro de una población: enfoques estructurados y no estructurados. En los enfoques estructurados [29], la comparación entre las estrategias de los individuos se limita a sus elementos vecinos. El concepto central de los métodos estructurados es centrarse en la estructura espacial o social de las interacciones, donde patrones o restricciones específicas dictan cómo interactúan los individuos, lo que lleva a la creación de una red de relaciones. Por el contrario, los modelos no estructurados [21] exploran escenarios en los que las interacciones o comparaciones entre estrategias se producen de forma aleatoria, sin ningún vecindario o patrón específico. En los enfoques no estructurados, cada individuo tiene la misma probabilidad de interactuar con cualquier otro individuo de la población. Los modelos no estructurados proporcionan una representación más abstracta, en la que el énfasis se pone en estudiar la dinámica a nivel de población y la aparición de comportamientos colectivos sin la influencia de interacciones específicas.

2.2 Inicialización de Metropolis Hasting

El proceso de crear una población inicial de soluciones al principio del proceso de optimización se denomina inicialización en metaheurística. Esta población inicial sirve como punto de partida para que el algoritmo explore y mejore iterativamente. El objetivo de un método de inicialización es generar un conjunto diverso de soluciones candidatas que cubran una amplia gama del espacio de soluciones, proporcionando una base para que el algoritmo comience su búsqueda de soluciones óptimas o casi óptimas. En la literatura se han documentado ampliamente varios métodos de inicialización [33]. Todos los métodos de inicialización comparten un elemento común fundamental. Su objetivo principal es conseguir una distribución uniforme de las soluciones iniciales por todo el espacio de búsqueda, con la intención de cubrirlo exhaustivamente. El objetivo de estos métodos es dispersar espacialmente las soluciones candidatas, sin tener en cuenta su calidad o rendimiento con respecto a la función objetivo.

El enfoque Metropolis-Hastings [34] genera un conjunto inicial de soluciones candidatas mediante el muestreo de la función objetivo f(x). utilizando la técnica Metropolis-Hastings [35]. Con este método, las soluciones iniciales tienden a agruparse en torno a los valores destacados de la función objetivo a optimizar, aproximándose mucho a ellos. Los pasos para utilizar el algoritmo Metropolis-Hastings como método de inicialización pueden resumirse como sigue:

1.	Comience con un valor inicial $\mathbf{x}^k$ o una muestra de la función objetivo $f(\mathbf{x})$.
2.	Proponer un nuevo valor $\mathbf{y}$ o una muestra de $f(\mathbf{y})$. $\mathbf{y}$ se genera como una perturbación del estado actual $\mathbf{x}^k$ con un valor aleatorio producido por la distribución normal $N(0,1)$.
3.	La probabilidad de aceptación α de la muestra propuesta $\mathbf{y}$ depende de la relación que mantengan los valores de la muestra actual $f(\mathbf{x}^k)$ y la muestra candidata $f(\mathbf{y})$. Por lo tanto, la probabilidad α se calcula de la siguiente manera: $$\alpha = min\left\{\frac{f(\mathbf{y})}{f(\mathbf{x}^k)}, 1\right\} \quad (1)$$ Esta probabilidad α determina si la muestra propuesta será aceptada o rechazada.
4.	Bajo esta regla probabilística, se genera un número aleatorio r uniformemente distribuido $U(0,1)$. Si $r < \alpha$, la muestra candidata $\mathbf{y}$ se acepta como el nuevo estado $\mathbf{x}^{k+1}$, en caso contrario, $\mathbf{y}$ se descarta, de modo que $\mathbf{x}^k$ queda como el nuevo estado $\mathbf{x}^{k+1}$.
5.	Repita los pasos 2-4 durante *n* veces para obtener el conjunto inicial de soluciones.

3. Enfoque propuesto

En el contexto de la optimización, el teorema del "no-free-lunch" [20] es un concepto fundamental que pone de manifiesto las limitaciones inherentes a cualquier algoritmo de optimización. Este teorema tiene profundas connotaciones porque implica que para cada algoritmo de optimización que funcione bien en un conjunto específico de problemas, habrá otros problemas para los que funcione mal. Algunos algoritmos pueden destacar en determinados espacios problemáticos o clases de problemas, mientras que otros pueden ser más adecuados para distintos tipos de retos de optimización. Dadas las limitaciones de cualquier algoritmo individual, la creación de nuevas metodologías de optimización resulta esencial. El desarrollo de enfoques novedosos puede dar lugar a herramientas más especializadas que se adapten a las características específicas de los problemas. Mediante el diseño de nuevas técnicas de optimización, los investigadores pueden profundizar en el conocimiento de las estructuras de los problemas, identificar heurísticas eficaces y explotar conocimientos específicos de cada ámbito.

En este estudio se presenta un algoritmo metaheurístico novedoso e innovador, cuyo objetivo es abordar problemas de optimización con eficacia. Nuestro algoritmo pretende identificar la solución global para un problema no lineal formulando un problema de optimización con la siguiente descripción:

$$\begin{array}{lll} \text{Maximizar} & f(\mathbf{p}) & \mathbf{p} = (p_1, \ldots, p_d) \in \mathbb{R}^d \\ \text{Sujeto a} & \mathbf{p} \in \mathbf{X} & \end{array} \quad (2)$$

donde $f: \mathbb{R}^d \to \mathbb{R}$ es una función no lineal d-dimensional y $\mathbf{X}$ representa el espacio de búsqueda $(\mathbf{x} \in \mathbb{R}^d | l_i \leq p_i \leq u_i, i = 1, \ldots, d)$ definido por los limites inferior (l_i) y superior (u_i). El método propuesto combina dos elementos clave para lograr su capacidad de optimización. En primer lugar, el algoritmo adopta la técnica Metrópolis-Hasting [34] como método de inicialización. En segundo lugar, el algoritmo incorpora un procedimiento basado en los principios de TEJ para evolucionar las posiciones de todas las soluciones de la población.

3.1 Inicialización con Metropolis-Hasting

El paso inicial de nuestro método consiste en obtener un conjunto de n posiciones iniciales $\{\mathbf{p}_1^0, \ldots, \mathbf{p}_n^0\}$ utilizando el método Metropolis-Hastings. Mediante esta técnica, posicionamos estratégicamente estas soluciones iniciales alrededor de las regiones prominentes de la función objetivo. Con ello, pretendemos garantizar que nuestro algoritmo inicie su proceso de optimización a partir de zonas prometedoras del espacio de soluciones, aumentando la probabilidad de encontrar soluciones de alta calidad. El uso de Metropolis-Hastings nos permite crear una población inicial que explora la vecindad de las características importantes de la función objetivo, proporcionando una base sólida para las iteraciones de optimización posteriores.

3.2 Modificación de soluciones mediante la TEJ

El enfoque propuesto en nuestro algoritmo implica dos ciclos distintos de actualización durante el proceso de optimización. El primer ciclo se centra en la actualización de las soluciones candidatas, mientras que el segundo ciclo se dedica a la actualización de las estrategias empleadas por los individuos.

En el primer ciclo, actualizamos iterativamente las soluciones candidatas en el espacio de búsqueda. Esto se consigue mediante un proceso sistemático de exploración de las regiones vecinas en cada iteración. Al actualizar continuamente las posiciones, el algoritmo se esfuerza por navegar por el espacio de soluciones y converger hacia regiones más prometedoras que puedan contener la solución global. El segundo ciclo gira en torno a la actualización de las estrategias de los individuos de la población. Las estrategias representan el comportamiento o enfoque adoptado por cada individuo durante el proceso de optimización. A través de este ciclo, el algoritmo permite a los individuos adaptar y evolucionar sus estrategias basándose en su rendimiento anterior y en los resultados de sus interacciones con otros individuos. Esta estrategia permite al algoritmo afinar el comportamiento de la población y responder dinámicamente a los cambios en el panorama de optimización.

La TEJ considera la dinámica de las interacciones entre los individuos de una población. Por tanto, los individuos participan en un ciclo recurrente de competiciones, en el que evalúan y comparan continuamente sus estrategias. En el contexto de nuestro algoritmo, las "estrategias" representan las distintas opciones o elecciones de comportamiento de que disponen los individuos de la población. En nuestro algoritmo, a cada individuo *i* de la población se le asigna inicialmente una estrategia o comportamiento aleatorio $S(i)$. La estrategia implica un punto de partida $\mathbf{p}_i^0$ y un vector de desviación estandar $\boldsymbol{\sigma}_i = \{\sigma_{1,i}, \ldots, \sigma_{d,i}\}$, donde cada valor σ_m $(m \in 1, \ldots, d)$ se genera como un valor aleatorio entre 0 y 1.

A) Modificación de las soluciones candidataS

En cada iteración k, cada individuo utiliza su estrategia $S(i)$ para seleccionar una nueva posición candidata $\mathbf{p}_i^{k+1}$ realizando una pequeña perturbación aleatoria en la solución actual $\mathbf{p}_i^k$. Esta perturbación se genera en base a una distribución de probabilidad normal $N(0, \boldsymbol{\sigma}_i)$ considerando el siguiente modelo:

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + R \cdot N(0, \boldsymbol{\sigma}_i) = S(i) \tag{3}$$

donde *R* representa la perturbación máxima admisible.

B) Modificación de estrategias

En TEJ, el proceso más importante se consigue mediante un mecanismo de comparación por pares con actualización de estrategias. Según este mecanismo, en cada iteración k del algoritmo, se aplica el siguiente proceso a cada individuo de la población.

1. **Selección.** Para el individuo $\mathbf{p}_i^k$, selecciona aleatoriamente otro elemento $\mathbf{p}_j^k$, de la población, con la condición de que $i \neq j$.
2. **Comparación de estrategias.** Una vez que el individuo $\mathbf{p}_i^k$ ha elegido el element $\mathbf{p}_j^k$, compara sus respectivas estrategias $S(i)$ y $S(j)$ en función de sus valores de aptitud $f(\mathbf{p}_i^k)\ y\ f(\mathbf{p}_j^k)$, respectivamente.
3. **Evaluación de la estrategia.** Si la estrategia $S(i)$ del individuo i resulta ser peor que la estrategia $S(j)$ del elemento j (es decir, la estrategia de j produce un valor de fitness más alto), entonces el individuo $\mathbf{p}_i^k$ considera actualizar su estrategia.
4. **Actualización probabilística de la estrategia.** La decisión de si el individuo i cambiará su estrategia se toma de forma probabilística. La probabilidad Pr de actualizar la estrategia es directamente proporcional a las diferencias de estrategias entre el individuo i y el elemento j. En estas condiciones, el valor Pr se calcula de la siguiente manera:

$$Pr = \left| \frac{f(\mathbf{p}_j^k) - f(\mathbf{p}_i^k)}{MF} \right| \quad (4)$$

donde *MF* simboliza el peor valor de la función objetivo encontrado hasta el momento. Según este modelo, cuanto más disímiles sean las estrategias, mayor será la probabilidad de que el individuo *i* modifique su estrategia.

5. **Estrategia de actualización**. En nuestro enfoque, la probabilidad **Pr** desempeña un papel crucial a la hora de determinar si la estrategia de un individuo se adaptará o no. Cuando **Pr** favorece la adaptación de la estrategia, el algoritmo inicia un proceso de modificación en dos pasos para cada individuo **i** de la población, que implica la actualización tanto del vector de desviación típica como de su posición.

   En la primera modificación, el individuo *i* adopta el mismo vector de desviación típica que otro individuo *j* de la población ($\sigma_i = \sigma_j$). Este proceso se conoce como proceso de imitación en el contexto de la TEJ. Al adoptar la estrategia exitosa del individuo *j*, el individuo *i* pretende imitar la estrategia que ha mostrado un rendimiento superior. Este proceso de imitación permite que las estrategias exitosas se extiendan y ganen prevalencia en la población, aumentando las posibilidades de lograr mejores soluciones.

   En la segunda modificación, el individuo *i* sufre un ajuste en su posición p_i^k para parecerse más a la posición p_j^k del individuo j. Este ajuste es una modificación basada en la diferencia aleatoria entre las posiciones de ambos individuos. Este ajuste se modela de la siguiente manera:

$$\mathrm{p}_i^{k+1} = r \cdot D - \mathrm{p}_i^{k+} \tag{5}$$

   donde *r* es un número aleatorio entre 0,5 y 1, $D = \mathrm{distancia}\left(\mathrm{p}_i^k, \mathrm{p}_j^k\right)$. Al realizar esta adaptación, el algoritmo permite al individuo *i* explorar la región alrededor de la posición del individuo *j* que ha tenido éxito y que ha dado resultados positivos en el proceso de optimización. Esta exploración permite la posibilidad de encontrar soluciones nuevas y potencialmente mejores en las proximidades de las estrategias exitosas.

   La combinación de estas dos modificaciones garantiza que los individuos de la población adapten sus estrategias en función del éxito y los resultados de los demás. Este mecanismo adaptativo permite al algoritmo responder dinámicamente a los cambios en el panorama de optimización, fomentando la propagación de estrategias exitosas y la exploración de regiones prometedoras en el espacio de soluciones.

   Al aplicar iterativamente estas modificaciones a cada individuo de la población, el algoritmo evoluciona y refina las estrategias de la población a lo largo de múltiples generaciones. Este proceso contribuye a la convergencia general del algoritmo hacia soluciones óptimas en el contexto del problema de optimización dado.

4. Resultados de la Experimentación

En esta sección, se han realizado una serie de experimentos para de este modo evaluar el rendimiento del algoritmo propuesto U-EGTA. Los resultados obtenidos fueron comparados con aquellos obtenidos por los siguientes algoritmos: La Colonia de Abejas Artificiales (ABC) [5], el Algoritmo del Murciélago (BA) [6], la Estrategia de Evolución de Adaptación de la Matriz de Covarianza (CMAES) [11], la Evolución Diferencial (DE) [7], el Algoritmo de Búsqueda de Armonía (HS) [12], el Algoritmo de Optimización por Enjambre de Partículas (PSO) [8], el Algoritmo Cuckoo Search (CS) [9], y el Algoritmo de Búsquedas de Cuervos (CSA) [10], los cuales han sido implementados en una gran variedad de estudios de la optimización.

Para simplificar la comparación de los resultados, todos los métodos fueron configurados de acuerdo con lo establecido en sus documentos guías. Las especificaciones de las configuraciones se muestran en la Tabla 1:

Algoritmo	Parámetros	Referencia
ABC	Limite = 50	[5]
BA	Volumen (A = 2), Frecuencia de Pulsos (r = 0.9), Frecuencia Mínima (Qmin = 0) y Frecuencia Máxima (Qmax = 1)	[6]
CMAES	Lambda = 50, número de padres = 25, sigma = 60, csigma = 0.32586, dsigma = 1.32586	[11]
DE	CR = 0.5, y F = 0.2	[7]
HS	$HCMR$ = 0.7, y $PArate$ = 0.3	[12]
PSO	c_1 = 2, y c_2 = 2; el factor de peso decrece linealmente de 0.9 hasta 0.2	[8]
CS	Pa = 0.25	[9]
CSA	Bandada = 50, probabilidad de conciencia = 0.1, longitud del vuelo = 2	[10]

Tabla 1. Parámetros de los algoritmos

En todos los experimentos realizados, se utilizó una población de 50 agentes para la búsqueda en cada uno de los algoritmos. Además, el criterio de paro se estableció en 50,000 accesos a función para lograr un análisis optimo. Del mismo modo, para evaluar el rendimiento de cada uno de los algoritmos se usaron tres indicadores: la mejor solución promedio obtenida (AB), la mejor solución mediana (MD) y la desviación estándar (SD) de las mejores soluciones. En cada uno de los problemas de optimización, los algoritmos se ejecutaron 30 veces, resultando en 30 diferentes soluciones. El valor de la variable AB es calculada de estos resultados mientras que la mediana es usada para determinar la media, y la desviación estándar es usada para medir el SD de las mejores soluciones.

La sección de experimentación está dividida en tres subsecciones: en la primera subsección 4.1 se analiza el rendimiento del algoritmo U-EGTA en comparación de otros algoritmos metaheurísticos; en la segunda subsección 4.2 se implementan diversos experimentos para identificar las diferencias significativas en el comportamiento de los algoritmos usando la prueba de Wilcoxon rank sum; y en la tercera subsección 4.3 se presenta la discusión de las curvas de convergencia más significativas de las funciones implementadas.

4.1. Comparación de resultados en términos de calidad

En esta subsección, se analiza el rendimiento del algoritmo propuesto U-EGTA comparándolo contra ocho de los más conocidos algoritmos metaheurísticos, siendo estos el ABC, BA, CMAES, CS, CSA, DE, HS, y el PSO. Para demostrar la efectividad del algoritmo propuesto, un set de 30 funciones ($f_1(x) - f_{30}(x)$ justo como se describe en el Apéndice A) fueron implementadas para su análisis. De acuerdo con los resultados de la Tabla 2, U-EGTA sobresale en 23 funciones a comparación del resto de algoritmos siendo esto en las funciones $f_1(x)$- $f_3(x)$, $f_9(x)-f_{10}(x)$, $f_{12}(x)-f_{17}(x)$, y en $f_{19}(x) - f_{30}(x)$.

Function		ABC	BA	CMAES	DE	HS	PSO	CS	CSA	UEGTA
$f_1(x)$	AB	2.17E+00	1.54E+01	5.62E-10	1.24E-07	1.22E+01	9.48E-05	6.86E+00	1.72E+01	**0.00E+00**
	MD	2.16E+00	1.53E+01	5.51E-10	3.71E-08	1.20E+01	2.54E-06	6.69E+00	1.74E+01	**0.00E+00**
	SD	4.64E-01	1.05E+00	1.67E-10	3.57E-07	5.46E-01	4.86E-04	1.11E+00	5.41E-01	**0.00E+00**
$f_2(x)$	AB	5.67E+02	4.87E+00	6.67E-01	6.87E-01	1.99E+04	1.07E+02	8.88E+00	1.49E+05	**2.47E-01**
	MD	5.02E+02	6.71E-01	6.67E-01	6.67E-01	1.87E+04	4.17E+01	8.81E+00	1.57E+05	**2.49E-01**
	SD	2.82E+02	1.07E+01	2.98E-08	1.13E-01	5.11E+03	1.36E+02	1.89E+00	3.61E+04	**7.79E-03**
$f_3(x)$	AB	1.04E+00	1.54E+02	7.40E-18	3.12E-03	4.32E+01	1.00E-02	1.15E+00	1.67E+02	**0.00E+00**
	MD	1.04E+00	1.34E+02	0.00E+00	4.34E-14	4.42E+01	3.70E-03	1.16E+00	1.70E+02	**0.00E+00**
	SD	1.83E-02	6.09E+01	4.05E-17	5.25E-03	6.55E+00	1.30E-02	4.11E-02	2.54E+01	**0.00E+00**
$f_4(x)$	AB	9.71E+00	1.70E+01	**2.34E-19**	4.82E-02	1.24E+01	7.57E-02	1.25E+01	5.68E+01	1.71E-06
	MD	9.78E+00	1.78E+01	**2.21E-19**	2.15E-15	1.24E+01	7.62E-12	1.28E+01	5.73E+01	8.68E-07
	SD	2.93E+00	7.41E+00	**1.46E-19**	1.16E-01	1.56E+00	1.72E-01	2.72E+00	7.76E+00	2.96E-06
$f_5(x)$	AB	-9.59E+03	-3.16E+07	-9.25E+01	**2.00E+00**	3.27E+01	3.41E+01	1.81E+09	2.15E+07	**2.00E+00**
	MD	-1.53E+03	-1.05E+06	-4.16E+01	**2.00E+00**	3.29E+01	9.00E+00	2.00E+00	1.73E+05	**2.00E+00**
	SD	3.24E+04	1.38E+08	1.51E+02	**0.00E+00**	1.15E+01	1.13E+02	3.80E+09	6.88E+07	**0.00E+00**

	AB	-4.32E+03	-2.24E+07	-8.51E+01	**2.00E+00**	4.81E+01	1.24E+01	2.33E+09	3.29E+07	**2.00E+00**
$f_6(x)$	MD	-1.19E+03	-2.16E+06	-2.33E+01	**2.00E+00**	4.37E+01	9.00E+00	2.00E+00	3.53E+05	**2.00E+00**
	SD	7.49E+03	7.59E+07	1.37E+02	**1.55E-15**	2.01E+01	1.83E+01	4.30E+09	1.40E+08	**0.00E+00**
	AB	1.49E-01	1.83E-15	1.33E-11	1.04E-12	1.25E-03	**0.00E+00**	**0.00E+00**	9.33E-02	**0.00E+00**
$f_7(x)$	MD	1.48E-01	1.44E-15	8.49E-12	2.05E-29	1.23E-03	**0.00E+00**	**0.00E+00**	8.84E-02	**0.00E+00**
	SD	3.53E-02	1.66E-15	1.33E-11	4.74E-12	4.53E-04	**0.00E+00**	**0.00E+00**	2.09E-02	**0.00E+00**
	AB	7.29E-05	1.23E+06	4.62E-205	**0.00E+00**	4.89E-40	**0.00E+00**	1.59E-04	2.32E+01	**0.00E+00**
$f_8(x)$	MD	4.76E-07	1.18E+00	1.70E-211	**0.00E+00**	4.59E-44	**0.00E+00**	2.65E-07	5.00E+00	**0.00E+00**
	SD	3.05E-04	4.09E+06	0.00E+00	**0.00E+00**	1.17E-39	**0.00E+00**	4.33E-04	4.93E+01	**0.00E+00**
	AB	1.38E+07	1.18E+07	7.22E+01	8.79E+01	6.57E+06	8.28E+01	1.36E+05	4.32E+07	**7.11E+01**
$f_9(x)$	MD	1.27E+07	1.41E+07	7.24E+01	7.37E+01	6.75E+06	8.29E+01	1.42E+05	4.52E+07	**7.11E+01**
	SD	3.69E+06	7.65E+06	1.09E+00	7.59E+01	1.18E+06	3.28E+00	2.89E+04	8.04E+06	**2.65E-03**
	AB	1.90E+06	9.38E+06	1.05E+02	1.06E+02	8.55E+06	1.02E+02	3.05E+04	7.58E+07	**4.64E+01**
$f_{10}(x)$	MD	1.83E+06	6.14E+06	1.05E+02	1.05E+02	8.69E+06	1.02E+02	3.11E+04	7.09E+07	**4.64E+01**
	SD	5.46E+05	1.10E+07	1.16E+00	3.51E+00	2.29E+06	4.71E+00	8.62E+03	1.99E+07	**6.01E-02**
	AB	**3.00E+01**	5.63E+01	**3.00E+01**	**3.00E+01**	3.45E+01	**3.00E+01**	**3.00E+01**	5.92E+01	**3.00E+01**
$f_{11}(x)$	MD	**3.00E+01**	5.60E+01	**3.00E+01**	**3.00E+01**	3.45E+01	**3.00E+01**	**3.00E+01**	5.90E+01	**3.00E+01**
	SD	**0.00E+00**	4.33E+00	**0.00E+00**	**0.00E+00**	1.36E+00	**0.00E+00**	**1.83E-01**	1.58E+00	**0.00E+00**
	AB	5.42E+02	8.48E-03	1.51E-03	7.52E-04	4.63E+02	2.62E+02	1.29E+00	2.07E+03	**4.17E-189**
$f_{12}(x)$	MD	4.94E+02	6.78E-03	1.29E-03	7.06E-04	4.52E+02	2.05E+02	1.28E+00	2.20E+03	**0.00E+00**
	SD	2.21E+02	4.60E-03	9.48E-04	4.13E-04	9.30E+01	2.90E+02	4.25E-01	3.78E+02	**0.00E+00**
	AB	1.19E+01	2.22E+01	9.05E+00	9.76E+00	1.46E+01	1.07E+01	1.06E+01	2.42E+01	**8.84E+00**
$f_{13}(x)$	MD	1.20E+01	2.04E+01	9.11E+00	9.76E+00	1.45E+01	1.07E+01	1.06E+01	2.38E+01	**8.95E+00**
	SD	7.80E-01	7.65E+00	2.93E-01	5.31E-01	6.82E-01	5.88E-01	3.41E-01	3.00E+00	**4.78E-01**
	AB	2.31E+02	9.37E+01	1.06E+02	1.52E+02	9.34E+01	4.84E+01	1.09E+02	2.60E+02	**0.00E+00**
$f_{14}(x)$	MD	2.31E+02	8.76E+01	1.60E+02	1.47E+02	9.53E+01	4.78E+01	1.07E+02	2.61E+02	**0.00E+00**
	SD	1.37E+01	4.21E+01	7.77E+01	2.37E+01	9.73E+00	1.89E+01	1.50E+01	1.22E+01	**0.00E+00**
	AB	1.54E+03	5.46E+01	1.89E+01	3.04E+01	1.58E+04	2.96E+04	1.92E+02	2.32E+05	**1.91E+00**
$f_{15}(x)$	MD	1.45E+03	2.94E+01	1.08E+01	2.56E+01	1.61E+04	2.53E+03	1.94E+02	2.30E+05	**1.00E-04**
	SD	5.41E+02	3.99E+01	2.51E+01	1.71E+01	3.38E+03	3.75E+04	3.92E+01	5.10E+04	**7.28E+00**
	AB	5.59E+01	4.44E+01	4.21E-07	1.20E+01	4.04E+01	6.81E+00	1.10E+01	5.12E+01	**0.00E+00**
$f_{16}(x)$	MD	5.58E+01	4.51E+01	3.88E-07	1.19E+01	4.03E+01	7.12E+00	1.09E+01	5.14E+01	**0.00E+00**
	SD	4.41E+00	6.67E+00	1.58E-07	5.37E+00	2.26E+00	2.31E+00	1.20E+00	2.40E+00	**0.00E+00**
	AB	1.99E+25	2.14E+32	2.05E-07	2.09E-05	1.56E+02	2.29E+02	1.00E+10	6.22E+29	**0.00E+00**
$f_{17}(x)$	MD	2.76E+23	5.97E+26	2.02E-07	5.91E-06	1.63E+02	2.14E+02	1.00E+10	1.72E+29	**0.00E+00**
	SD	8.15E+25	8.30E+32	6.77E-08	6.57E-05	2.94E+01	9.79E+01	0.00E+00	1.03E+30	**0.00E+00**
	AB	9.00E+00	1.94E+04	**0.00E+00**	1.33E-01	4.79E+03	3.33E-02	3.38E+01	1.87E+04	**0.00E+00**
$f_{18}(x)$	MD	9.00E+00	1.95E+04	**0.00E+00**	0.00E+00	4.71E+03	0.00E+00	3.40E+01	1.85E+04	**0.00E+00**
	SD	2.63E+00	6.37E+03	**0.00E+00**	3.46E-01	6.97E+02	1.83E-01	9.25E+00	2.28E+03	**0.00E+00**
	AB	-7.38E+02	-1.00E+03	-1.12E+03	-1.15E+03	-1.10E+03	-1.07E+03	-1.01E+03	-7.30E+02	**-1.17E+03**
$f_{19}(x)$	MD	-7.39E+02	-1.01E+03	-1.12E+03	-1.16E+03	-1.10E+03	-1.06E+03	-1.01E+03	-7.25E+02	**-1.17E+03**
	SD	3.56E+01	4.74E+01	2.33E+01	2.19E+01	1.40E+01	2.57E+01	1.24E+01	2.87E+01	**2.73E-02**
	AB	-2.62E+14	-2.54E+14	-8.90E+13	-2.33E+01	-2.78E+01	-2.84E+01	-2.86E+01	-6.81E+13	**-3.00E+01**
$f_{20}(x)$	MD	-2.60E+14	-2.52E+14	-7.63E+13	-2.31E+01	-2.79E+01	-2.81E+01	-2.85E+01	-6.96E+13	**-3.00E+01**
	SD	5.21E+13	6.19E+13	4.43E+13	2.14E+00	3.42E-01	1.23E+00	3.34E-01	1.91E+13	**4.55E-06**
	AB	1.52E+03	4.55E+02	8.78E+02	1.07E+00	1.67E+02	2.06E+02	1.48E+02	4.84E+02	**1.05E-106**
$f_{21}(x)$	MD	1.32E+03	2.86E+02	8.50E+02	8.68E-01	1.72E+02	1.92E+02	1.49E+02	4.52E+02	**1.10E-219**
	SD	4.97E+02	4.92E+02	2.16E+02	7.32E-01	2.48E+01	1.26E+02	2.33E+01	1.32E+02	**5.77E-106**
	AB	2.48E+01	6.13E+04	4.86E-16	1.10E-13	2.39E+04	3.58E+03	8.02E+01	1.15E+05	**0.00E+00**
$f_{22}(x)$	MD	2.31E+01	5.97E+04	4.03E-16	5.58E-14	2.49E+04	7.29E-10	7.21E+01	1.19E+05	**0.00E+00**
	SD	8.61E+00	4.42E+04	2.69E-16	1.82E-13	3.95E+03	5.30E+03	2.91E+01	1.51E+04	**0.00E+00**
	AB	8.56E+02	2.74E+06	9.77E-14	8.34E-12	6.73E+05	2.30E+05	2.86E+03	4.05E+06	**0.00E+00**
$f_{23}(x)$	MD	8.06E+02	2.57E+06	8.32E-14	1.25E-12	6.93E+05	1.35E+05	2.69E+03	4.26E+06	**0.00E+00**
	SD	2.69E+02	1.70E+06	7.60E-14	3.01E-11	1.16E+05	2.79E+05	7.83E+02	6.48E+05	**0.00E+00**
	AB	1.24E-02	1.26E-05	1.46E-20	7.29E-17	1.28E+01	7.55E-13	4.22E-02	4.87E+01	**0.00E+00**
$f_{24}(x)$	MD	1.17E-02	1.28E-05	1.52E-20	2.99E-17	1.30E+01	1.94E-13	4.38E-02	4.89E+01	**0.00E+00**
	SD	3.24E-03	1.84E-06	6.73E-21	1.12E-16	1.77E+00	2.03E-12	9.79E-03	5.57E+00	**0.00E+00**
	AB	5.73E-01	1.73E+01	1.41E-18	4.03E-15	5.88E+02	6.33E+01	2.02E+00	2.47E+03	**0.00E+00**
$f_{25}(x)$	MD	5.12E-01	1.77E-03	1.16E-18	1.52E-15	5.87E+02	3.94E-11	1.94E+00	2.63E+03	**0.00E+00**
	SD	2.27E-01	5.86E+01	8.43E-19	8.45E-15	6.35E+01	1.10E+02	4.67E-01	4.05E+02	**0.00E+00**
	AB	6.46E-04	1.02E-08	6.08E-10	6.69E-12	3.03E-05	1.07E-23	3.99E-11	8.21E-03	**5.58E-36**
$f_{26}(x)$	MD	4.18E-04	1.01E-08	4.14E-10	5.52E-18	3.03E-05	9.47E-29	3.94E-11	7.48E-03	**3.34E-64**
	SD	6.92E-04	3.84E-09	5.48E-10	3.57E-11	1.41E-05	5.82E-23	2.36E-11	4.28E-03	**3.05E-35**
	AB	7.03E+02	3.02E+04	3.60E-09	4.45E-08	4.46E+03	1.34E+03	1.53E+03	1.97E+04	**0.00E+00**
$f_{27}(x)$	MD	1.25E+02	2.83E+04	3.66E-09	3.97E-08	4.46E+03	4.28E-07	1.35E+03	2.02E+04	**0.00E+00**
	SD	2.50E+03	2.10E+04	1.20E-09	2.67E-08	7.40E+02	3.47E+03	7.46E+02	3.07E+03	**0.00E+00**

$f_{28}(x)$	AB	2.11E+02	5.39E+02	3.26E+01	3.63E+01	3.44E+02	6.60E+01	1.61E+02	7.33E+02	**2.90E+01**
	MD	2.14E+02	5.47E+02	2.90E+01	3.69E+01	3.46E+02	6.35E+01	1.60E+02	7.39E+02	**2.90E+01**
	SD	2.12E+01	1.28E+02	6.51E+00	7.08E+00	2.77E+01	1.81E+01	1.31E+01	6.32E+01	**5.64E-05**
$f_{29}(x)$	AB	2.72E+05	3.27E+07	3.30E+01	3.46E+01	6.61E+06	9.52E+01	3.12E+02	6.78E+07	**3.20E+01**
	MD	2.11E+05	1.44E+07	3.30E+01	3.32E+01	6.63E+06	8.72E+01	3.06E+02	7.05E+07	**3.20E+01**
	SD	2.46E+05	4.23E+07	5.64E-10	4.28E+00	1.98E+06	3.66E+01	4.74E+01	2.24E+07	**4.42E-06**
$f_{30}(x)$	AB	3.47E+02	9.24E+02	3.03E+01	3.24E+01	3.44E+02	7.46E+01	3.54E+02	9.18E+02	**2.90E+01**
	MD	3.34E+02	9.31E+02	2.90E+01	2.90E+01	3.49E+02	5.48E+01	3.42E+02	9.14E+02	**2.90E+01**
	SD	7.40E+01	3.06E+02	3.98E+00	6.44E+00	3.14E+01	6.80E+01	5.84E+01	1.13E+02	**0.00E+00**

Tabla 2. Comparación de resultados en términos de calidad

4.2 Comparación en términos de significancia

En esta segunda subsección, se realiza un estudio del tipo no paramétrico para evaluar la relevancia estadística comparado con los otros algoritmos. Para validar los resultados obtenidos se emplea un análisis de Wilcoxon rank-sum test [36], la cual es una prueba que determina si existe una diferencia importante entre ambos métodos. Este análisis específicamente compara el rendimiento del algoritmo U-EGTA contra el resto siendo el ABC, BA, CMAES, CS, CSA, DE, HS y PSO. Los resultados del análisis de Wilcoxon indican que existe una diferencia significativa, por lo cual se rechaza la hipótesis nula.

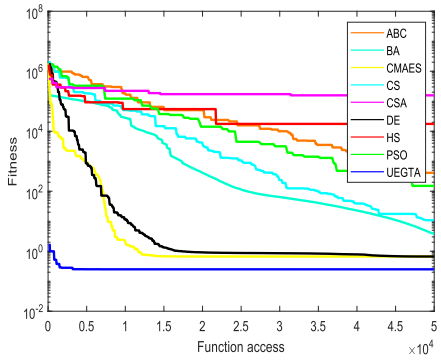
Function	**UEGTA vs ABC**	**UEGTA vs BA**	**UEGTA vs CMAES**	**UEGTA vs DE**	**UEGTA vs HS**	**UEGTA vs PSO**	**UEGTA vs CS**	**UEGTA vs CSA**
$f_1(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_2(x)$	3.02E-11▲	2.61E-10▲	5.57E-10▲	4.62E-10▲	3.02E-11▲	2.61E-10▲	3.02E-11▲	3.02E-11▲
$f_3(x)$	1.21E-12▲	1.21E-12▲	1.61E-01▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_4(x)$	3.02E-11▲	3.02E-11▲	3.02E-11▼	7.73E-02▲	3.02E-11▲	1.07E-07▲	3.02E-11▲	3.02E-11▲
$f_5(x)$	0.00E+00▲	1.35E-03▲	0.00E+00▲	0.00E+00►	1.21E-12▲	2.76E-07▲	2.92E-04▲	1.21E-12▲
$f_6(x)$	0.00E+00▲	1.25E-05▲	0.00E+00▲	0.00E+00►	1.21E-12▲	3.05E-07▲	2.78E-03▲	1.21E-12▲
$f_7(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	3.13E-04▲	1.21E-12▲	3.34E-01▲	4.19E-02▲	1.21E-12▲
$f_8(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	0.00E+00►	1.21E-12▲	0.00E+00►	1.21E-12▲	1.21E-12▲
$f_9(x)$	3.02E-11▲	3.02E-11▲	1.00E+00▲	5.57E-10▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{10}(x)$	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{11}(x)$	0.00E+00►	1.19E-12▲	0.00E+00►	0.00E+00►	1.00E-12▲	0.00E+00►	2.14E-02►	1.11E-12▲
$f_{12}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{13}(x)$	3.02E-11▲	3.02E-11▲	2.42E-02▲	7.09E-08▲	3.02E-11▲	9.92E-11▲	7.39E-11▲	3.02E-11▲
$f_{14}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{15}(x)$	3.02E-11▲	9.92E-11▲	5.07E-10▲	4.62E-10▲	3.02E-11▲	6.07E-11▲	3.02E-11▲	3.02E-11▲
$f_{16}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{17}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.69E-14▲	1.21E-12▲
$f_{18}(x)$	1.12E-12▲	1.21E-12▲	0.00E+00►	2.14E-02▲	1.21E-12▲	3.34E-01▲	1.20E-12▲	1.21E-12▲
$f_{19}(x)$	3.02E-11▲	3.02E-11▲	5.24E-10▲	1.07E-07▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{20}(x)$	1.90E-10▲	0.00E+00▲	4.05E-12▲	8.15E-11▲	3.02E-11▲	9.51E-06▲	3.02E-11▲	5.83E-10▲
$f_{21}(x)$	2.37E-12▲	2.37E-12▲	2.37E-12▲	2.37E-12▲	2.37E-12▲	2.37E-12▲	2.37E-12▲	2.37E-12▲
$f_{22}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{23}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{24}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{25}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{26}(x)$	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{27}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
$f_{28}(x)$	3.02E-11▲	3.02E-11▲	1.95E-03▲	1.95E-03▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{29}(x)$	3.02E-11▲	3.02E-11▲	3.02E-11►	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲	3.02E-11▲
$f_{30}(x)$	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲	1.21E-12▲
▲	29	30	26	26	30	28	30	30
►	1	0	3	4	0	2	0	0
▼	0	0	1	0	0	0	0	0

Tabla 3. Comparación en términos de significancia

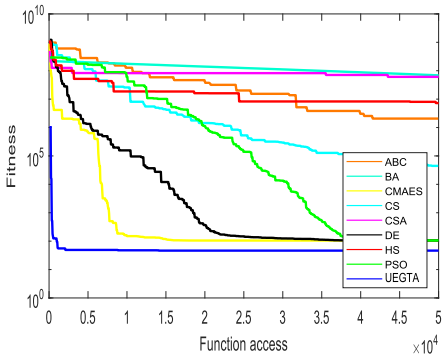
Esta validación estadística apoya y refuerza la confianza en el rendimiento del algoritmo propuesto en comparación con los otros algoritmos. En el análisis de Wilcoxon se indujeron 30 pruebas por función considerando un nivel de significancia de 0.05. Los resultados de los valores “p” son presentados en la Tabla 3, donde a su vez cada una de las pruebas realizadas de los algoritmos se llevaron a cabo en 30 dimensiones. Para mejorar la interpretación de los datos de la Tabla 3, se utilizó la siguiente simbología: ▲, ►, y ▼, donde cada uno de estos símbolos representa una condición diferente. El símbolo ▲ representa que la metodología propuesta supera a las otras metodologías en términos de calidad en sus soluciones. El símbolo ► indica que las dos metodologías comparadas producen soluciones similares. El símbolo ▼ representa que las soluciones generadas por la metodología propuesta son deficientes comparadas con las soluciones de las demás metodologías. Y con ello, como se puede observar en la Tabla 3, la mayoría de los resultados obtenidos superan la calidad demostrando así que la metodología propuesta mejora la calidad de las soluciones.

4.3. Análisis de las Gráficas de Convergencia

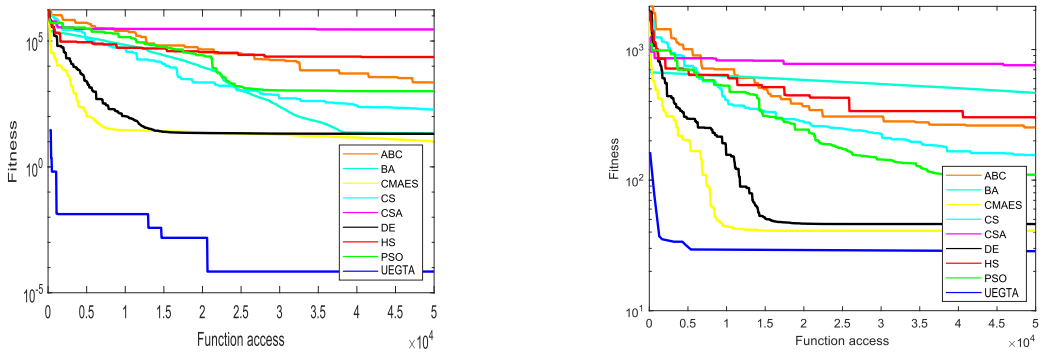
Evaluar la precisión de un algoritmo de optimización basándose solamente en las soluciones obtenidas es ineficiente para demostrar sus capacidades, por otro lado, la convergencia de un algoritmo juega un rol crucial para demostrar su eficacia y precisión. En esta tercera subsección se realiza un estudio de la convergencia para poder demostrar la velocidad con la que el algoritmo metaheurístico propuesto llega a la óptima solución. Para facilitar el entendimiento del estudio se realizaron graficas de convergencia para funciones representativas en 30 dimensiones en la Fig.1. Los gráficos fueron generados con los resultados obtenidos de la etapa de experimentación.



$f(x)_2$



$f(x)_8$



$f(x)_{14}$ $f(x)_{28}$

Fig.1. Graficas de la Curva de Convergencia

De acuerdo con las gráficas mostradas en la Fig. 1, el algoritmo propuesto U-EGTA demuestra una convergencia superior comparada con el resto de los algoritmos.

5. Conclusiones

Este trabajo introduce un nuevo algoritmo de optimización basado en los principios de la Teoría de Juegos Evolutivos, la cual da comienzo con la inicialización de todos los individuos usando la técnica de Metropolis-Hasting posicionando las soluciones cada vez más cerca de las regiones optimas. Cada uno de los individuos en la población tiene su propia estrategia, conforme el algoritmo progresa, interacciones y competiciones entre los individuos se llevan a cabo para poder llegar a una adaptación continua y a un ajuste de sus propias estrategias. El objetivo principal es que las estrategias más efectivas se difundan por toda la población para poco a poco ir disminuyendo y eliminar las estrategias menos efectivas, esto se logra refinando y actualizando constantemente las estrategias basándose en el rendimiento de estas mismas. Con ello, el algoritmo emplea, de manera efectiva, estrategias de búsqueda prometedoras para explorar y explotar el espacio de búsqueda, enriqueciendo su habilidad para encontrar el óptimo global en problemas de optimización complejos.

Para evaluar el rendimiento del algoritmo propuesto, se comparó contra ocho de los más conocidos algoritmos metaheurísticos usando 30 funciones benchmark. La metodología propuesta demuestra superioridad en términos de calidad en las soluciones, en términos significativos y en la convergencia cuando es comparado con otros. Así mismo, el algoritmo propuesto tiene el potencial para resolver problemas de la vida cotidiana, así como para resolver diferentes funciones; esto debido a que su versatilidad y efectividad lo convierte en una opción prometedora para pruebas prácticas esto basado en los excelentes resultados demostrados en la sección de experimentación. Puede explorarse y probarse más a fondo en diferentes dominios con diversos tipos de problemas para evaluar su eficacia y desempeño.

Referencias

[1] X.-S. Yang, *Engineering Optimization. An Introduction with Metaheuristic Applications*. United States of America: WILEY, 2010.

[2] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic Algorithms: A Comprehensive Review," *Comput. Intell. Multimed. Big Data Cloud with Eng. Appl.*, pp. 185–231, Jan. 2018.

[3] B. Chopard and M. Tomassini, *An introduction to metaheuristics for optimization*. 2018.

[4] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, no. August, p. 106040, 2019.

[5] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," 2005.

[6] X.-S. S. Yang, "A New Metaheuristic Bat-Inspired Algorithm BT - Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)," *Stud. Comput. Intell.*, vol. 284, pp. 65–74, 2010.

[7] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Australas. Plant Pathol.*, vol. 38, no. 3, pp. 284–287, 1995.

[8] J. Kennedy, R. Eberhart, and B. Gov, "Particle Swarm Optimization," *Encycl. Mach. Learn.*, pp. 760–6, 1995.

[9] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," *2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc.*, pp. 210–214, 2009.

[10] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.

[11] A. Auger and N. Hansen, "Performance evaluation of an advanced local search evolutionary algorithm," *2005 IEEE Congr. Evol. Comput. IEEE CEC 2005. Proc.*, vol. 2, pp. 1777–1784, 2005.

[12] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, Mar. 2014.

[13] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Syst.*, vol. 96, pp. 120–133, 2016.

[14] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, May 1983.

[15] E. Cuevas, A. Echavarría, and M. A. Ramírez-Ortegón, "An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation," *Appl. Intell.*, vol. 40, no. 2, pp. 256–272, Mar. 2014.

[16] A. R. Giri, T. Chen, V. P. Rajendran, and A. Khamis, "A Metaheuristic Approach to Emergency Vehicle Dispatch and Routing," *2022 IEEE Int. Conf. Smart Mobility, SM 2022*, pp. 27–31, 2022.

[17] A. Afzal *et al.*, "Optimizing the thermal performance of solar energy devices using meta-heuristic algorithms: A critical review," *Renew. Sustain. Energy Rev.*, vol. 173, p. 112903, Mar. 2023.

[18] E. Vaziri, F. Dehdar, and M. R. Abdoli, "Feasibility study of using meta-heuristic algorithms on optimizing of the integrated risk in banking system," *Int. J. Financ. Manag. Account.*, vol. 8, no. 28, pp. 143–158, Jan. 2023.

[19] S. Kaur, Y. Kumar, A. Koul, and S. Kumar Kamboj, "A Systematic Review on Metaheuristic Optimization Techniques for Feature Selections in Disease Diagnosis: Open Issues and Challenges," *Arch. Comput. Methods Eng. 2022 303*, vol. 30, no. 3, pp. 1863–1895, Nov. 2022.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol.*

Comput., vol. 1, no. 1, pp. 67–82, 1997.

[21] Wu, B., Bauer, B., Galla, T., & Traulsen, A. (2015). Fitness-based models and pairwise comparison models of evolutionary games are typically different—even in unstructured populations. New Journal of Physics, 17(2), 023043.

[22] Li, Q., Liu, S. Y., & Yang, X. S. (2020). Influence of initialization on the performance of metaheuristic optimizers. Applied Soft Computing, 91, 106193.

[23] E. Cuevas, H. Escobar, R. Sarkar, and H. F. Eid, "A new population initialization approach based on Metropolis–Hastings (MH) method," *Appl. Intell.*, vol. 53, no. 13, pp. 16575–16593, Dec. 2022.

[24] H. Gintis, *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton university press, 2000.

[25] L. R. Izquierdo, S. S. Izquierdo, and W. H. Sandholm, "Agent-Based Evolutionary Game Dynamics Agent-Based Evolutionary Game Dynamics," *Univ. Wisconsin Press.*, p. 207, 2020.

[26] O. Kapliński and J. Tamošaitiene, "Game theory applications in construction engineering and management," *Technol. Econ. Dev. Econ.*, vol. 16, no. 2, pp. 348–363, 2010.

[27] T. M. Choi, A. A. Taleizadeh, and X. Yue, "Game theory applications in production research in the sharing and circular economy era," *https://doi.org/10.1080/00207543.2019.1681137*, vol. 58, no. 1, pp. 118–127, Jan. 2019.

[28] M. Leng and M. Parlar, "Game Theoretic Applications in Supply Chain Management: A Review," *http://dx.doi.org/10.1080/03155986.2005.11732725*, vol. 43, no. 3, pp. 187–220, 2016.

[29] McAvoy, A., & Wakeley, J. (2022). Evaluating the structure-coefficient theorem of evolutionary game theory. Proceedings of the National Academy of Sciences, 119(28), e2119656119.

[30] C. Leboucher *et al.*, "An enhanced particle swarm optimization method integrated with evolutionary game theory," *IEEE Trans. Games*, vol. 10, no. 2, pp. 221–230, Jun. 2018.

[31] J. W. Weibull, *Evolutionary game theory*. MIT press, 1997.

[32] P. Hammerstein and R. Selten, "Game theory and evolutionary biology," *Handb. game theory with Econ. Appl.*, vol. 2, pp. 929–993, 1994.

[33] Agushaka, J. O., Ezugwu, A. E., Abualigah, L., Alharbi, S. K., & Khalifa, H. A. E. W. (2023). Efficient initialization methods for population-based metaheuristic algorithms: a comparative study. Archives of Computational Methods in Engineering, 30(3), 1727-1787.

[34] D. B. Hitchcock, "A History of the Metropolis–Hastings Algorithm," *http://dx.doi.org/10.1198/0003130032413*, vol. 57, no. 4, pp. 254–257, 2012.

[35] E. Cuevas, H. Escobar, R. Sarkar, and H. F. Eid, "A new population initialization approach based on Metropolis–Hastings (MH) method," *Appl. Intell.*, vol. 53, no. 13, pp. 16575–16593, Dec. 2022.

[36] F. Wilcoxon, "Probability Tables for Individual Comparisons by Ranking Methods," Biometrics, vol. 3, no. 3, p. 119, Sep. 1947.

Apendice A

	Nombre	Función	S	Mínimo
$f(x)_1$	Ackley	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\exp(1)$	$[-30,30]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_2$	Dixon	$(x_1-1)^2+\sum_{i=1}^{n}i(2x_i^2-x_{i-1})^2$	$[-10,10]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=2^{-\frac{2^i-2}{2^i}}$ for $i=1,\dots,n$
$f(x)_3$	Griewank	$\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$[-600,600]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_4$	Infinity	$\sum_{i=1}^{n}x_i^6\,sen(x_i+2)$	$[-1,1]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_5$	Mishra1	$(1+x_n)^{x_n};\quad x_n=n-\sum_{i=1}^{n-1}x_i$	$[0,1]^n$	$f(\mathbf{x}^*)=2;$ $\mathbf{x}^*=(1,\dots,1)$
$f(x)_6$	Mishra2	$(1+x_n)^{x_n};\quad x_n=n-\sum_{i=1}^{n-1}\frac{x_i+x_{i+1}}{2}$	$[0,1]^n$	$f(\mathbf{x}^*)=2;$ $\mathbf{x}^*=(1,\dots,1)$
$f(x)_7$	Penalty 1	$\frac{\pi}{n}\times\left\{10\sin^2(\pi\varphi_1)+\sum_{i=1}^{n-1}(\varphi_i-1)^2[1+10\sin^2(\pi\varphi_{i+1})]+(\varphi_n-1)^2\right\}+\sum_{i=1}^{n}u(x_i,a,k,m)$ $\varphi_i=1+\frac{1}{4}(x_i+1),\quad u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & if\ x_i>a\\0 & if-a\le x_i\\k(-x_i-a)^m & if\ x_i<a\end{cases}$ $a=10,\quad k=100,\quad m=4$	$[-50,50]^n$	$f(\boldsymbol{x}^*)=0;$ $\boldsymbol{x}^*=(-1,\dots,-1)$
$f(x)_8$	Penalty 2	$0.1\times\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\right\}+\sum_{i=1}^{n}u(x_i,a,k,m)$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & if\ x_i>a\\0 & if-a\le x_i\\k(-x_i-a)^m & if\ x_i<a\end{cases}$ $a=5,\quad k=100,\quad m=4$	$[-50,50]^n$	$f(\boldsymbol{x}^*)=0;$ $\boldsymbol{x}^*=(1,\dots,1)$
$f(x)_9$	Plateau	$30+\sum_{i=1}^{n}\lvert x_i\rvert$	$[-5.12,5.12]^n$	$f(\mathbf{x}^*)=30;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{10}$	Powell	$\sum_{i=1}^{\frac{n}{4}}[(x_{4i-3}+10x_{4i-2})^2+5(x_{i-1}+x_{4i})^2+(x_{4i-2}+2x_{4i-1})^4+10(x_{4i-3}+x_{4i})^4]$	$[-4,5]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{11}$	Quartic	$\sum_{i=1}^{n}i*x_i^4+rand(0,1)$	$[-1.28,1.28]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{12}$	Quintic	$\sum_{i=!}^{n}\lvert x_i^5-3x_i^4+4x_i^3+2x_i^2-10x_i-4\rvert$	$[-10,10]^n$	$f(\boldsymbol{x}^*)=0;$ $\boldsymbol{x}^*=(-1,\dots,-1)$
$f(x)_{13}$	Rastringin	$10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$	$[-5.12,5.12]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{14}$	Rosenbrock	$\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-5,10]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(1,\dots,1)$
$f(x)_{15}$	Schwefel21	$418.9829n-\sum_{i=1}^{n}x_i\sin(\sqrt{\lvert x_i\rvert})$	$[-100,100]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{16}$	Schwefel22	$\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-100,100]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$

$f(x)_{17}$	Step	$\sum_{i=1}^{n}(x_i+0.5)^2$	$[-100,100]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{18}$	Stybtang	$\frac{1}{2}\sum_{i=1}^{n}(x_i^4-16x_i^2+5x_i)$	$[-5,5]^n$	$f(\boldsymbol{x}^*)=-39.1659n;$ $\boldsymbol{x}^*=(-2.90,\dots,2.90)$
$f(x)_{19}$	Trid	$\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$	$[-n^2,n^2]^n$	$f(\boldsymbol{x}^*)=-n(n+4)(n-1)/6;$ $\boldsymbol{x}^*=[i(n+1-i)]$ $for\ i=1,\dots,n$
$f(x)_{20}$	Vincent	$-\sum_{i=1}^{n}\sin(10log x_i)$	$[0.25,10]^n$	$f(\mathbf{x}^*)=-n;$ $\mathbf{x}^*=(7.70,\dots,7.70)$
$f(x)_{21}$	Zakharov	$\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^4$	$[-5,10]^n$	$f(\boldsymbol{x}^*)=0;$ $\boldsymbol{x}^*=(0,\dots,0)$
$f(x)_{22}$	Rothyp	$\sum_{i=1}^{n}\sum_{j=1}^{i}x_j^2$	$[-65.536,65.536]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{23}$	Schwefel2	$\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_i\right)^2$	$[-100,100]^n$	$[-100,100]^n$
$f(x)_{24}$	Sphere	$\sum_{i=1}^{n}x_i^2$	$[-5,5]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=0,\dots,0$
$f(x)_{25}$	Sum2	$\sum_{i=1}^{d}ix_i^2$	$[-10,10]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{26}$	Sum of different powers	$\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	$[-1,1]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{27}$	Rastringin+ Schwefel2+ Sphere	$10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]+\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_i\right)^2+\sum_{i=1}^{n}x_i^2$	$[-100,100]^n$	$f(\mathbf{x}^*)=0;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{28}$	Griewank+ Rastringin+ Rosenbrock	$\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1+10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$ $+\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-100,100]^n$	$f(\mathbf{x}^*)=n-1;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{29}$	Ackley+ Penalty2+ Rosenbrock+ Schwefel2	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\exp(1)$ $+(0.1\{\sin(3\pi x_i)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]$ $+[(x_n-1)^2[1+\sin^2(2\pi x_n)]]\}+\sum_{i=1}^{n}u(x_i,5.100,4)+\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_i\right)^2$	$[-100,100]^n$	$f(\mathbf{x}^*)=(1.1n)-1;$ $\mathbf{x}^*=(0,\dots,0)$
$f(x)_{30}$	Ackley+ Griewank+ Rastringin+ Rosenbrock+ Schwefel22	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\exp(1)$ $+\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1+10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$ $+\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]+\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-100,100]^n$	$f(\mathbf{x}^*)=n-1;$ $\mathbf{x}^*=(0,\dots,0)$

Esta obra está bajo una licencia de Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

COMPUTATION E INFORMATICA

Recibido 02 May. 2024 *ReCIBE, Año 13 No. 1, May. 2024*

Aceptado 16 May. 2024

Segmentación rápida de imágenes con múltiples características

Fast Image Segmentation with Multiple Features

Sergio Gomez Vega[1]
checoskleyxd@gmail.com[*]

Alberto Luque Chang[1]

Hector Joaquin Escobar Cuevas[1]

Fernando Vega Parra[1]

[1]Universidad de Guadalajara

[*]Autor para correspondencia

Resumen:
La segmentación de características múltiples es superior a los enfoques unidimensionales en escalas de grises. El algoritmo del cambio medio (CM) se utiliza comúnmente para esta tarea. A pesar de sus interesantes resultados, el CM sigue siendo computacionalmente prohibitivo para escenarios de segmentación en los que el mapa de funciones está formado por características multidimensionales. El enfoque propuesto considera un mapa de características bidimensional que incluye el valor en escala de grises y la varianza local de cada píxel de la imagen. Para reducir el coste computacional, se modifica el algoritmo de CM clásico para que opere sobre un número menor de puntos. En tales condiciones, se diferencian dos conjuntos de elementos: datos implicados (el conjunto reducido de datos considerados en la operación CM) y datos no implicados (el resto de datos disponibles). A diferencia del CM clásico, que emplea funciones gaussianas, en nuestro enfoque el proceso de estimación del mapa de características se lleva a cabo utilizando un enfoque más preciso, como la función kernel de Epanechnikov. Una vez obtenidos los resultados del CM, se generalizan para incluir los datos no utilizados. Así, cada característica no utilizada se asigna al mismo clúster de datos utilizados más cercano. Por último, los grupos con menos características se fusionan con otros grupos vecinos. El método de segmentación propuesto se ha comparado con otros algoritmos del estado de la técnica usando de la base de datos de Berkeley. Los resultados experimentales confirman que el esquema propuesto produce imágenes segmentadas con un 50% más de calidad de percepción visual.

Palabras clave: Segmentación de imagen, Algoritmo de cambio medio, Estimador de densidad del kernel (EDK)

Summary:

Multifeature segmentation is superior to unidimensional approaches in grayscale images. The Mean Shift (MS) algorithm is commonly used for this task. Despite its interesting results, MS remains computationally prohibitive for segmentation scenarios where the feature map consists of multidimensional features. The proposed approach considers a two-dimensional feature map that includes the grayscale value and the local variance of each pixel in the image. To reduce computational costs, the classic MS algorithm is modified to operate on a smaller number of points. Under these conditions, two sets of elements are differentiated: involved data (the reduced set of data considered in the MS operation) and uninvolved data (the remaining available data). Unlike classic MS, which employs Gaussian functions, our approach uses a more precise method for the feature map estimation process, specifically the Epanechnikov kernel function. Once the MS results are obtained, they are generalized to include the unused data. Each unused feature is assigned to the same cluster as the nearest involved data. Finally, clusters with fewer features are merged with neighboring clusters. The proposed segmentation method has been compared with other state-of-the-art algorithms using the Berkeley database. Experimental results confirm that the proposed scheme produces segmented images with 50% higher visual perception quality.

Keywords: Image segmentation, Mean Shift algorithm, Kernel Density Estimator (KDE)

1 Introducción

Una de las principales etapas del procesamiento de imágenes es la segmentación de las mismas, que separa una imagen en distintas sub-áreas que no se solapan en función de un determinado criterio de similitud. Los resultados finales de distintos algoritmos de reconocimiento de objetos, recuperación basada en el contenido y clasificación semántica están totalmente condicionados por la eficacia del método de segmentación de imágenes empleado. Sin embargo, segmentar una sección o un elemento de una imagen complicada representa un proceso difícil. En las tres últimas décadas, se han introducido varios métodos potentes de segmentación de imágenes. Entre ellos se incluyen esquemas como los basados en histogramas [1,2], los basados en gráficos [3-5], los basados en la detección de contornos [6], los basados en umbrales [7,8], los difusos [9,10], los basados en campos aleatorios de Markkov [11], los basados en texturas [12], los basados en el análisis de componentes principales [13]y los basados en agrupaciones [14,15]. De estos enfoques, las técnicas de segmentación basadas en umbrales e histogramas se consideran las más populares en diversos campos de aplicación.

Los esquemas de segmentación basados en umbrales pueden clasificarse en métodos de segmentación de dos niveles y de varios niveles en función del número de objetos o regiones que contenga la imagen. En general, las imágenes con más de dos objetos o zonas requieren umbrales multinivel para su correcta segmentación. En función de la información que procesan, los métodos de segmentación de imágenes basados en umbrales también podrían dividirse en seis clases. también podrían clasificarse en seis clases [10]. Esta clasificación incluye métodos basados en clustering, métodos basados en histogramas, métodos basados en histogramas, métodos basados en histogramas, métodos basados en atributos de objetos, métodos basados en entropía, métodos locales y métodos espaciales.

Por otro lado, los esquemas basados en histogramas utilizan la información contenida en un histograma unidimensional, como sus curvaturas, valles y picos, para determinar el número de elementos presentes en la imagen. Sin embargo, dado que los histogramas unidimensionales no tienen en cuenta la asociación espacial entre píxeles, estos enfoques mantienen un rendimiento insatisfactorio. Para mejorar los resultados producidos por los histogramas unidimensionales, se han propuesto técnicas de histograma bidimensional para la segmentación de imágenes [16]. En estos enfoques, los valores de gris originales considerados en los histogramas unidimensionales se combinan con información sobre la posición de los píxeles. combinados con información sobre la posición de los píxeles [16,17] para producir histogramas bidimensionales (posición de grises). Según estos métodos, el conjunto de elementos diagonales del histograma bidimensional implica información esencial para clasificar objetos de fondo, mientras que los elementos no diagonales contienen información de imagen sobre píxeles de borde y datos de ruido. Normalmente, con las técnicas basadas en histogramas bidimensionales, los detalles finos como puntos, bordes y líneas no se pueden segmentar correctamente, ya que los histogramas bidimensionales suavizan los datos de la imagen [18] calculando la información media en escala de grises alrededor de los píxeles objetivo. Desde su introducción, varios algoritmos nuevos han tenido en cuenta el concepto de histograma bidimensional para producir resultados de segmentación interesantes [19,20]. Un ejemplo típico es el enfoque propuesto por Xueguang y Shu-hong [21]en 2012. En este método, se introduce un nuevo histograma bidimensional para segmentar objetos en imágenes genéricas. El histograma integra valores de píxeles en escala de grises y su respectiva información de gradiente. Aunque el algoritmo presenta resultados interesantes, nuevos experimentos [22]han demostrado que, en varios escenarios, este enfoque ofrece resultados inferiores a los esquemas de posición de grises. inferiores a los esquemas de posición de grises.

Técnicas metaheurísticas [20,23] También se han utilizado en combinación con histogramas bidimensionales como esquemas de segmentación [19,20,23-25]. Los métodos metaheurísticos son técnicas computacionales inspiradas en fenómenos naturales o sociales, que se utilizan habitualmente para resolver formulaciones de optimización [26] caracterizadas por su alta complejidad. Las técnicas de segmentación que integran métodos metaheurísticos e histogramas bidimensionales formulan la información proporcionada por el histograma como un problema de optimización.

Por tanto, mediante el uso de una función objetivo, se utiliza una técnica metaheurística particular para determinar el número de elementos presentes en la imagen, que minimizan/maximizan su valor. Se han propuesto varios enfoques metaheurísticos para la segmentación bajo el esquema de histograma bidimensional. Algunos ejemplos incluyen enfoques como la optimización del enjambre de partículas (PSO).[27-29], Evolución Diferencial (DE)[23], Optimización de colonias de hormigas (OCA)[30], Algoritmos genéticos (AG)[31], Colonia de abejas artificiales (CAA)[32], Recocido simulado (SA) [33], Algoritmo de enjambre de peces artificiales (AFS)[34], Optimización de enjambre de golondrinas (SSO) [35]Optimización similar al electromagnético (EMO) [36]y algoritmo de búsqueda gravitacional [37]. La mayoría de estas técnicas de segmentación tienen un problema crítico: el número de clases debe ser previamente fijado antes de su ejecución [36,37]. Este hecho limita seriamente su aplicación cuando se desconoce la naturaleza y características de la imagen. Los métodos de segmentación basados en técnicas metaheurísticas presentan buenos resultados en términos de optimización de características que relacionan el problema de segmentación con una función de costos, pero estos resultados no siempre se reflejan en la calidad visual [37]. Otra característica adversa de estos enfoques es su alto coste computacional. Todos los enfoques de segmentación basados en histogramas bidimensionales consideran el histograma como un mapa de características para relacionar dos características de píxeles diferentes. De hecho, los histogramas representan el método más sencillo para producir mapas de densidad. Para su generación, el espacio de datos se divide en clases disjuntas o contenedores, mientras que la densidad se estima calculando cuántos puntos de datos caen en cada contenedor. A pesar de su simplicidad, los histogramas presentan graves inconvenientes como baja precisión, discontinuidad de densidad y alto costo computacional [38]. Otro método útil para producir mapas de características es el método del estimador de densidad del kernel (KDE). KDE mantiene muchas características interesantes como regularidad asintótica, fundamentos matemáticos bien conocidos, propiedades de continuidad y diferenciabilidad [39,40]. A diferencia de los histogramas, los enfoques de KDE producen mapas de características con mayor precisión y propiedades suaves. Bajo el esquema KDE, la densidad en cada punto se calcula mediante el uso de funciones simétricas (funciones kernel) que calculan la acumulación ponderada de elementos vecinos. Varias funciones del kernel [38,41]Se han propuesto para su adopción en KDE métodos como las funciones gaussiana, triangular, uniforme, triweight, biweight y Epanechnikov. Aunque la función gaussiana es la más utilizada debido a su popularidad, el núcleo de Epanechnikov [42] presenta la mejor precisión de estimación cuando la cantidad de datos para calcular la densidad del mapa de características es muy limitada [42].

El esquema KDE también se ha empleado con fines de agrupación y clasificación. El ejemplo más representativo de su uso es el desplazamiento medio (CM) [43]algoritmo. El funcionamiento de CM considera dos pasos [44]. En la primera etapa, se realiza la estimación del mapa de características. Esta operación se logra mediante el uso del enfoque KDE. Luego, CM busca las posiciones que corresponden a los máximos locales del mapa de características de densidad mediante el uso de un método de gradiente. Estas posiciones, llamadas atractores, representan los prototipos de cúmulos de cada clase. Los resultados precisos y sólidos de CM han motivado su aplicación en varios contextos, como el filtrado de imágenes [45]y segmentación de imágenes [46]. En el proceso de segmentación de imágenes, CM detecta secciones unidimensionales homogéneas combinando el grupo de píxeles que corresponde a puntos suficientemente cercanos. puntos del tractor. En la literatura se han introducido varios algoritmos de segmentación basados en CM. Algunos ejemplos incluyen el enfoque propuesto por Tao et al [47]que incorpora un algoritmo de árbol con CM para la segmentación eficiente de barcos en imágenes infrarrojas. En [48], Parque y col. introdujo una técnica de segmentación basada en la fusión de un modelo de mezcla gaussiana con CM.

A pesar de sus interesantes resultados, CM mantiene un grave inconveniente: su coste computacional [42,49,50]. Durante la operación de CM, el mapa de características se calcula en cada punto, considerando la información de todos los datos disponibles. Asimismo, el proceso de asignación de conglomerados se lleva a cabo mediante la iteración de un método de gradiente para cada punto del mapa de características. Bajo esta limitación, el uso de CM es prohibitivo en escenarios de segmentación donde el mapa de características consta de características multidimensionales como las representadas por histogramas bidimensionales. Esta es la razón principal porque la mayoría de los algoritmos de segmentación de CM combinan datos

unidimensionales (valores en escala de grises) con otras técnicas computacionales que extraen información de clasificación sin utilizar características de píxeles adicionales (es decir, más dimensiones) [42]. El uso de más dimensiones en el mapa de características aumenta significativamente el costo computacional del método CM. Por lo tanto, el desafío es encontrar un esquema computacionalmente eficiente para manejar mapas de características multidimensionales. Una técnica común para reducir el costo computacional es centrar la operación del algoritmo en una pequeña cantidad representativa de elementos de todo el conjunto de datos. Los resultados parciales obtenidos luego se generalizan para incluir información que no está involucrada en el conjunto de datos de enfoque inicial. Luego, los resultados parciales obtenidos deben generalizarse para incluir información no involucrada. Esta metodología se ha aplicado con éxito en enfoques como Random Hough Transform (RHT) [51] y Consenso de muestra aleatoria (RANSAC) [52], para nombrar unos pocos.

En este artículo, se presenta un nuevo algoritmo de segmentación competitivo para imágenes en escala de grises. El enfoque propuesto considera un mapa de características bidimensional que incluye el valor de escala de grises y la varianza local para cada píxel de la imagen. Para reducir el costo computacional, el algoritmo de cambio medio (CM) se opera utilizando un conjunto representativo de elementos que contienen una cantidad muy limitada de puntos de todos los datos disponibles. En tales condiciones, se diferencian dos conjuntos de elementos: datos involucrados (el conjunto de datos reducido considerado en la operación CM) y datos no involucrados (el resto de datos disponibles). A diferencia de los enfoques tradicionales de CM que emplean funciones gaussianas, en nuestro enfoque, el proceso para estimar los valores de densidad en el mapa de características utiliza la eficiente función del núcleo de Epanechnikov. Una vez obtenidos los resultados de CM, se generalizan para incluir los datos no involucrados. Por lo tanto, cada elemento no utilizado se asigna al mismo grupo de datos utilizados más cercanos. Finalmente, los grupos con la menor cantidad de elementos se fusionan con otros grupos vecinos. El método de segmentación propuesto se ha comparado con otros algoritmos de última generación considerando el número total de imágenes del conjunto de datos de Berkeley. Los resultados experimentales confirman que el esquema propuesto presenta un mejor desempeño en términos de calidad, consistencia y precisión.

El resto del artículo está organizado de la siguiente manera: En la Sección 2, se da una breve explicación del algoritmo de cambio medio; en la Sección 3, se ilustran las principales características del problema ICE; en la Sección 4, se presentan los elementos más importantes sobre la CMA; En la Sección 5, se presenta el enfoque propuesto; La sección 6 exhibe los resultados experimentales; finalmente, en la Sección 7, se extraen las conclusiones.

2 Algoritmo de cambio medio

cambio medio [43]es un algoritmo que se ha utilizado comúnmente para la segmentación de múltiples características. Durante la operación CM, a cada característica $\boldsymbol{x}$ presente en el espacio de características, se le asigna un grupo $\boldsymbol{C_i}$ cuya posición prototipo $\boldsymbol{x_*}$ Corresponde a los máximos locales del mapa de densidad. Por lo tanto, a partir del punto $\boldsymbol{x}$, se calcula una nueva u bicación en la dirección del mayor aumento del mapa de densidad hasta un máximo local $\boldsymbol{x_*}$ ha sido alcanzado. La operación del desplazamiento medio considera dos etapas. Primero, se realiza el mapa de densidad del espacio de características. Esta operación se logra mediante el uso de un enfoque KDE. Luego, se lleva a cabo un paso de optimización basado en el ascenso del gradiente para encontrar los máximos locales (atractores de densidad) sobre el mapa de densidad [53].

2.1 Mapa de densidad

Un método útil para estimar el mapa de características de densidad $f(x)$ forman un conjunto de datos es el KDE no paramétrico. Bajo el esquema KDE, la densidad en cada punto se calcula mediante el uso de funciones del núcleo que calculan la acumulación ponderada de elementos vecinos. Una función del núcleo $\boldsymbol{k}$ se define como un modelo de kernel simétrico y no negativo $\boldsymbol{k}$, tal que:

$$K(x) \geq 0; K(-x) = K(x); \int K(x)dx = 1 \qquad (1)$$

Suponiendo un conjunto de $\boldsymbol{n}$ características $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots \boldsymbol{x_n}\}$, la función de densidad estimada $\boldsymbol{f(x)}$ se puede calcular en términos de $\boldsymbol{K}$ de la siguiente manera:

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \qquad (2)$$

Dónde $\boldsymbol{h}$ representa la influencia del área de $\boldsymbol{K}$.

Aunque se han propuesto varias funciones del núcleo para los métodos de KDE, la mayoría de los esquemas de KDE consideran la función gaussiana. $\boldsymbol{K_G(x)}$ como modelo de núcleo. El núcleo gaussiano o normal se define en Ec.(3) mientras que su representación gráfica es como se muestra en la figura 1.

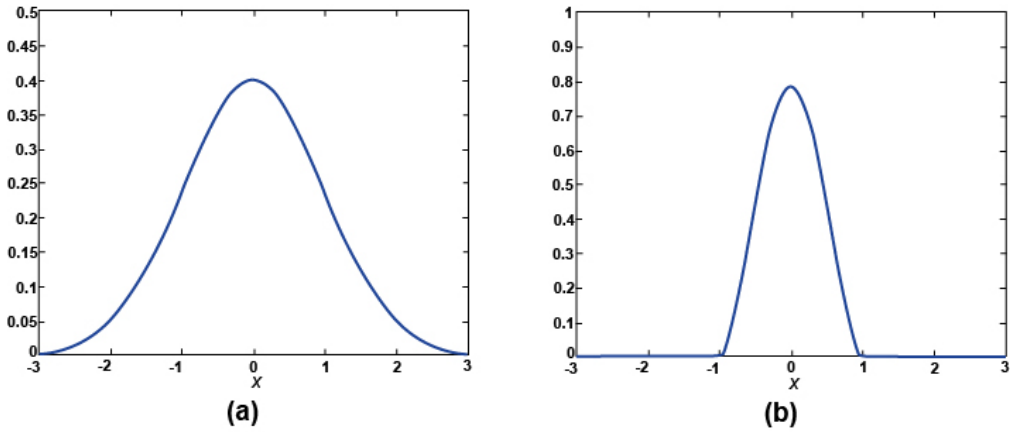


Figura 1. Diferentes modelos de kernel(s); kernel gaussiano $K_G(x)$ y (b) núcleo de Epanechnikov $\boldsymbol{K_E(x)}$

$$K_G(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2h}} \qquad (3)$$

Otra interesante alternativa del KDE es la función Epanechnikox $\boldsymbol{K_E(x)}$ (como se aprecia en la Figura 1(b)) [49], la cual está definida por el siguiente modelo:

$$K_E(x) = \begin{cases} \frac{3}{4}(1 - x^2), if\ |x| \leq 1 \\ 0, de\ otra\ manera \end{cases} \qquad (4)$$

Una forma común de evaluar la precisión de la estimación de un enfoque KDE es el análisis del error cuadrático medio (ECM) producido entre el mapa de características de densidad real $\boldsymbol{f}(\boldsymbol{x})$ y su representación estimada $\hat{\boldsymbol{f}}(\boldsymbol{X})$ de modo que

$$MSE\left(\hat{f}(x)\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{f}(x_i) - f(x_i)\right)^2 \qquad (5)$$

En tales circunstancias, el núcleo de Epanechnikov $\boldsymbol{K_E}(\boldsymbol{x})$ presenta la mejor precisión de estimación en términos de CME criterio desde su expresión $\boldsymbol{K_E}(\boldsymbol{x})$, según varios estudios [38,41], representa la solución funcional que minimiza la CME formulación de Ec. (5). Este comportamiento se mantiene incluso cuando la cantidad de datos para calcular la densidad del mapa de características es muy limitada [41].

Para mostrar las capacidades de modelado del núcleo de $\boldsymbol{K_E}(\boldsymbol{x})$. Se considera el siguiente ejemplo. Suponga la función de densidad de probabilidad (PDF) producida por una mezcla gaussiana bidimensional $\boldsymbol{M}(\boldsymbol{X_1}, \boldsymbol{X_2})$. La mezcla $\boldsymbol{M}(\boldsymbol{X_1}, \boldsymbol{X_2})$ combina tres funciones gaussianas $(\boldsymbol{j} = \boldsymbol{1}, \boldsymbol{2}, \boldsymbol{3})$ según el modelo $\boldsymbol{N_j}(\mu_j, \boldsymbol{\Sigma_j})$, dónde μ corresponde a su punto central mientras $\sum$ simboliza su covarianza. La Figura 2 (a) presenta una mezcla gaussiana definida como $(\boldsymbol{X_1}, \boldsymbol{X_2}) = \left(\frac{2}{3}\right)\boldsymbol{N_1} + \left(\frac{1}{6}\right)\boldsymbol{N_2} + \left(\frac{1}{6}\right)\boldsymbol{N_3}$. De este PDF, se muestra un conjunto reducido de 120 puntos de datos (verFigura 2(b)). Teniendo en cuenta este espacio de características, los núcleos $\boldsymbol{K_G}(\boldsymbol{x})$ y $\boldsymbol{K_E}(\boldsymbol{x})$ se han utilizado para estimar el mapa de densidad $\hat{\boldsymbol{f}}(\boldsymbol{x_1}, \boldsymbol{x_2})$ obtención de Figura 2(c) y 2(d), respectivamente. De la Figura 2(c) y 2(d), está claro que el núcleo de Epanechnikov $\boldsymbol{K_E}(\boldsymbol{x})$ alcanza un mejor resultado de estimación que el kernel normal $\boldsymbol{K_G}(\boldsymbol{x})$. Es evidente que el núcleo gaussiano produce varios máximos locales falsos en$\hat{\boldsymbol{f}}(\boldsymbol{x_1}, \boldsymbol{x_2})$ como consecuencia de su incapacidad para modelar adecuadamente el mapa de densidad cuando el conjunto de datos disponibles se reduce drásticamente.

2.1 Atractores de densidad

La idea principal detrás de la estimación de un mapa de densidad. $\hat{\boldsymbol{f}}(\boldsymbol{x_1}, \boldsymbol{x_2})$para un conjunto de datos dado es encontrar los puntos característicos $\boldsymbol{x_i^*}$ que representan los prototipos de clúster que resumen mejor la distribución de los puntos de datos modelados de $\hat{\boldsymbol{f}}(\boldsymbol{x})$. Estos puntos corresponden a los máximos locales presentes en el mapa de densidad estimado $\hat{\boldsymbol{f}}(\boldsymbol{x})$. Estos puntos se pueden encontrar aplicando un método de optimización basado en gradientes. Considerando un conjunto de datos $\boldsymbol{x}$ de $\boldsymbol{d}$ elementos $(\boldsymbol{x} = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_d}\})$, en este esquema, un punto aleatorio $\boldsymbol{x_j}$ primero se selecciona $(\boldsymbol{j} \in \boldsymbol{1}, \ldots, \boldsymbol{d})$. Luego, una nueva ubicación $\boldsymbol{x_i^{t+1}}$ se calcula en la dirección del gradiente de densidad $\boldsymbol{\nabla}\hat{\boldsymbol{f}}(\boldsymbol{x_j})$ hasta el máximo local $\boldsymbol{x_i^*}$ ha sido alcanzado. Esta regla de actualización se formula de la siguiente manera:

$$x_j^{t+1} = x^t + \delta \cdot \nabla\hat{f}\left(x_i^t\right) \qquad (6)$$

Dónde $\boldsymbol{t}$ denota el número de iteración actual, mientras que δ representa el tamaño del paso. Bajo esta formulación, el gradiente de densidad en cualquier punto $\boldsymbol{x}$ se calcula calculando la derivada de la PDF de la siguiente manera:

$$\nabla\hat{f}(x) = \frac{\partial}{\partial x}\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n}\frac{\partial}{\partial x}K\left(\frac{x - x_i}{h}\right) \qquad (7)$$

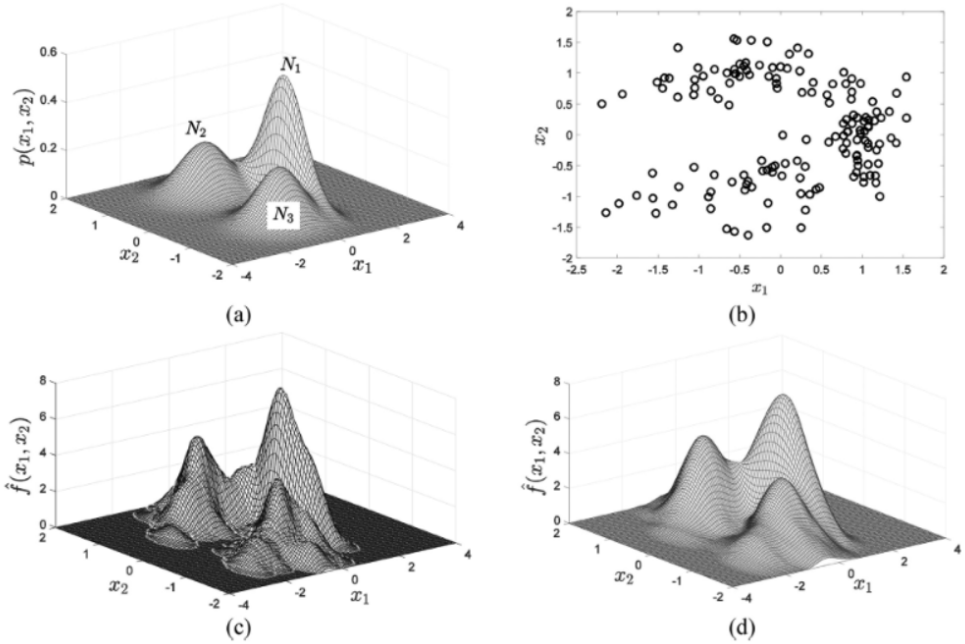


Figura 2. Capacidades de modelado del kernel gaussiano. $\boldsymbol{K_G(x)}$ y Epanechnikov $\boldsymbol{K_E(x)}$ cuando se reduce el conjunto de datos disponibles. (a) PDF bidimensional, (b) 120 características muestreadas del PDF, (c) estimación del mapa de densidad $\boldsymbol{\hat{f}(x_1, x_2)}$ considerando el núcleo kernel gaussiano y (d) $\boldsymbol{K_G(x)}$ y Epanechnikov $\boldsymbol{K_E(x)}$.

Dónde $\boldsymbol{n}$ representa el número de dimensiones de $\boldsymbol{x} = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_n})$. Una vez aplicado el método del gradiente a todos de los $\boldsymbol{d}$ elementos del conjunto de datos $\boldsymbol{x}$, varios puntos de datos serán atraídos por el mismo máximo local $\boldsymbol{x_i^*}$. Estos puntos se clasificarán luego como una parte del grupo $\boldsymbol{C_i}$. Bajo estas condiciones, como resultado final, obtenemos una lista de $\boldsymbol{C}$ atractores de densidad $\{\boldsymbol{x_1^*}, \ldots, \boldsymbol{x_c^*}\}$ dónde cada atractor $\boldsymbol{x_i^*}$ debe mantener un conjunto de elementos $X_i = \{\boldsymbol{x_a}, \ldots, \boldsymbol{x_g}\}$ que corresponde a todos los datos atraídos por $\boldsymbol{x_i^*}(\boldsymbol{x_a}, \boldsymbol{x_c} \in \boldsymbol{C_i})$. La figura 3 presenta una ilustración del proceso de gradiente para encontrar los máximos locales. En la imagen se muestra la información de contorno del mapa de densidad $\boldsymbol{\hat{f}(x_1, x_2)}$. En tales condiciones, comenzando en puntos $\boldsymbol{x_p}$,$\boldsymbol{x_q}$ y $\boldsymbol{x_r}$ se calculan nuevas ubicaciones en la dirección del mayor aumento del mapa de densidad$\boldsymbol{\hat{f}(x_1, x_2)}$. hasta los máximos locales, $\boldsymbol{x_1^*}$, $\boldsymbol{x_2^*}$ **y** $\boldsymbol{x_3^*}$ respectivamente, ha sido alcanzado.

3 Método propuesto

En este enfoque, llamado CAM-SEG, la segmentación es producida mediante el análisis del espacio de características bidimensionales, los cuales incluyen el valor de escala de grises y la varianza local para cada pixel en la imagen. El proceso agrupa los pixeles de acuerdo con su intensidad dentro del mapa de objetos bidimensional. Esto se logra mediante el algoritmo CM modificado. Para reducir el costo computacional, el algoritmo CM es adaptado para contener elementos representativos, que tienen un número limitado de pixeles de las características limitando así, la información disponible, por lo tanto, dos grupos de elementos son identificados: características usadas (la reducción del conjunto de datos) y características no usadas (el resto de la información). En este enfoque, el proceso para estimar la densidad de los valores en el mapa de características es adoptado de la función de Epanechnikov para su kernel. Una vez se obtiene el CM resultante, se generaliza para incluir las características no integradas. A continuación, a cada elemento no utilizado se le asigna el grupo más cercano y para aumentar la eficiencia grupos con poca cantidad de datos se fusionan con otros grupos.

3.1 Definición de características

Si se asume que $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ corresponden a niveles de escala de grises de 0 a L-1 para cada pixel en el espacio coordenado $(\boldsymbol{x},\boldsymbol{y})$ en cualquier imagen de dimensión **M*N.** $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{y})$ representa la varianza del valor del pixel en las coordenadas $(\boldsymbol{x},\boldsymbol{y})$, generando una ventana cuadrada de dimensión **n*n**. Por lo tanto, el objeto $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})$ de un pixel localizado en $(\boldsymbol{x},\boldsymbol{y})$ se simboliza bidimensionalmente como el siguiente vector:

$$\boldsymbol{F}(x,y) = [\boldsymbol{I}(x,y)\boldsymbol{V}(x,y)] \qquad (8)$$

En $\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})$ ambos elementos $\boldsymbol{I}(\boldsymbol{x},\boldsymbol{y})$ y $\boldsymbol{V}(\boldsymbol{x},\boldsymbol{y})$ son normalizados de 0 a 1.

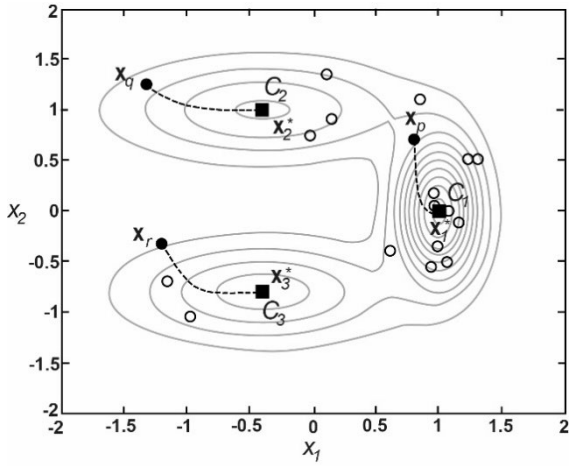


Figura 3 Ilustración del proceso de gradiente para encontrar los máximos locales.

3.2 Reducción del conjunto de datos

Para reducir el conjunto de datos, CM usa elementos representativos que contienen información limitada de toda la disponible $F(x,y)$. Bajo las condiciones anteriores, todo el conjunto de datos es dividido en dos grupos: elementos usados U y elementos no usados $\widehat{U}$. El grupo U son los elementos considerados para el procesamiento, mientras que $\widehat{U}$ representa el resto de los elementos, con las definiciones anteriores, se puede concluir que $\widehat{U}$ junto con U forman el conjunto F.

El número de elementos presentes en U es un aspecto determinante en la segmentación propuesta. Para ver el impacto de este parámetro, se a conducido un experimento de sensibilidad, en el experimento, se sabe que el tamaño de U es un porcentaje de toda la información disponible, por lo que se varía desde 1% hasta 100% mientras los otros parámetros permanecen constantes.

Los elementos U son seleccionados de manera aleatoria de la información disponible F. Por lo tanto, para minimizar los efectos estocásticos, se ejecutan 30 corridas independientes para cada tamaño de U. Los resultados sugieren que después de aumentar el porcentaje arriba del 5% el costo computacional crece exponencialmente mientras que mejora marginalmente el resultado final, se puede concluir que el mejor porcentaje para el tamaño de U es de 5%.

Al reflexionar sobre lo anterior se puede concluir que de toda la información disponible **M*N** del conjunto de datos $\boldsymbol{F}$ solo el 5% se usara para producir el conjunto $\boldsymbol{U}$. Los elementos de $\boldsymbol{U}$ serán seleccionados aleatoriamente de $\boldsymbol{F}$. Por lo tanto, el conjunto de datos $\boldsymbol{U}$ incluye vectores bidimensionales tal que $\boldsymbol{0} = \boldsymbol{int}(\boldsymbol{0.05} \cdot \boldsymbol{M} \cdot \boldsymbol{N})$, donde $\boldsymbol{int}(\cdot)$ produce un valor entero. Por otro lado, el grupo de elementos en $\widehat{\boldsymbol{U}} = \{\widehat{\boldsymbol{u}}_{\boldsymbol{i}}, \widehat{\boldsymbol{u}}_{\boldsymbol{j}}, \cdots, \widehat{\boldsymbol{u}}_{\boldsymbol{k}}\}$es integrado por elementos donde $\widehat{\boldsymbol{U}} = \{\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \in \boldsymbol{F} | \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k} \notin \boldsymbol{U}\}$.

3.3 operación CM

Ya reducido el conjunto a $\boldsymbol{U}$, el CM se puede ejecutar. En la operación **CM**, se deben configurar dos parámetros importantes: El núcleo $\boldsymbol{h}$ y el tamaño de paso $\boldsymbol{\delta}$. Para que el CM procese la información de las características bidimensionales de $\boldsymbol{F}$, ambos elementos vectores $\boldsymbol{h}$ y $\boldsymbol{\delta}$ deben ser bidimensionales con valores para cada dimensión. Estos factores son inicializados automáticamente atreves del análisis del conjunto de datos $\boldsymbol{U}$. El proceso usado para esta configuración fue reportado en la regla de Scott [54]. Bajo esta regla el valor de ambos factores $\boldsymbol{h}$ y $\boldsymbol{\delta}$ están estrechamente relacionados con la desviación estándar $\boldsymbol{\sigma}$ de los datos $\boldsymbol{U}$. Cada dato $\boldsymbol{u_d} = [\boldsymbol{u_d^1}, \boldsymbol{u_d^2}]$ donde representa $\boldsymbol{u_d^1}$ la información en escala de grises de $\boldsymbol{u_i}$ mientras que $\boldsymbol{u_d^2}$ corresponde a su varianza. Por lo tanto, la desviación estándar para cada dimensión $\boldsymbol{\sigma_1}$ y $\boldsymbol{\sigma_2}$ son computadas con las siguientes formulas:

$$\sigma_1 = \sqrt{\frac{1}{o}\sum_{d=1}^{o}(u_d^1 - \bar{u}^1)^2} \tag{9}$$

$$\sigma_2 = \sqrt{\frac{1}{o}\sum_{d=1}^{o}(u_d^2 - \bar{u}^2)^2} \tag{10}$$

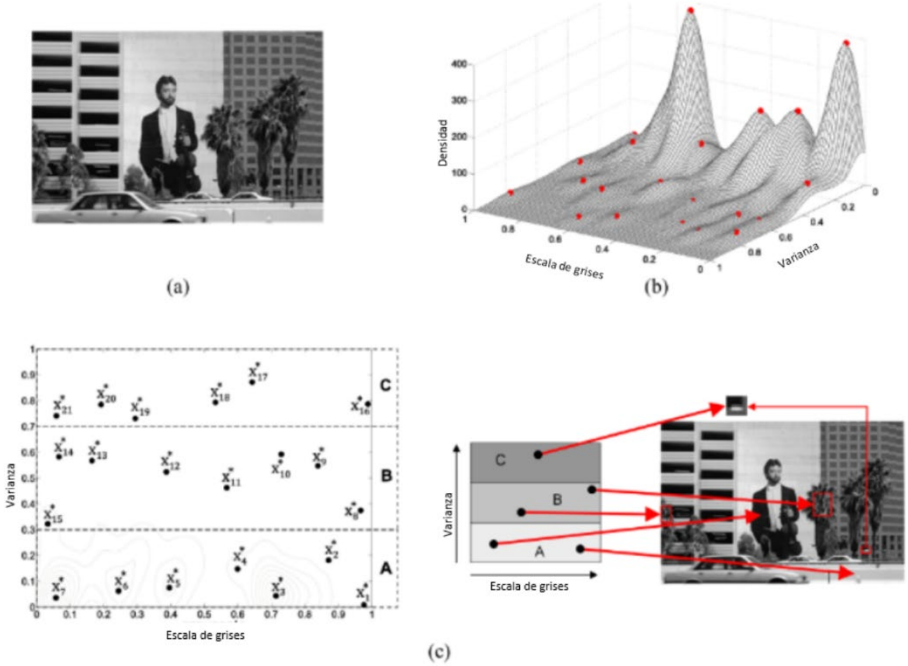
Donde $\bar{\boldsymbol{u}}^{\boldsymbol{1}}$ y $\bar{\boldsymbol{u}}^{\boldsymbol{2}}$representan el valor medio tal que $\bar{\boldsymbol{u}}^{\boldsymbol{s}} = (\boldsymbol{1}/\boldsymbol{o})\sum_{\boldsymbol{d}=\boldsymbol{1}}^{\boldsymbol{o}} \boldsymbol{u_d^s}$. Atreves de una exhaustiva experimentación, la regla de Scott [54] establece que los factores los factores están relacionados de la siguiente manera:

$$h_s = 3.5 \cdot \sigma_s \cdot o^{-\frac{1}{3}}, \qquad \delta_s = \sqrt{0.5} \cdot \sigma_s \tag{11}$$

Donde $\boldsymbol{s} \in [\boldsymbol{1}, \boldsymbol{2}]$.

3.4 Análisis de los resultados

La operación CM se genera con dos pasos. Primero, la densidad del mapa es producida, después se detecta el máximo local con la información del mapa de densidad. Por lo tanto, cuando se le



aplica este procesoal conjunto de datos $\boldsymbol{U}$, la función CM produce un mapa de densidad, un conjunto $\boldsymbol{c}$ de puntos $\{\boldsymbol{x_1^*}, \ldots, \boldsymbol{x_c^*}\}$ atractores y una lista de elementos $\boldsymbol{x_j} = (\boldsymbol{u_a}, \ldots, \boldsymbol{u_g})$ para cada atractor $\boldsymbol{x_j^*}$donde enlista los puntos atractores como $\boldsymbol{u_a}, \boldsymbol{u_g} \in \boldsymbol{C_i}$.

Figura 4 Análisis del efecto de los puntos de atracción. (a) Imagen original, (b) mapa de densidad producido con el conjunto de datos reducido, (c) división del mapa de densidad y sus características de correspondencia en la imagen.

En la figura 4 se puede observar una imagen y como esta se segmentará. Esta imagen tiene un tamaño de 214 X 320. De esta imagen, se computa un conjunto de características $\boldsymbol{F}$ integradas por valores de escala de grises y varianza de cada pixel. Entonces, un conjunto de 3,424 elementos $(\boldsymbol{o} = \boldsymbol{int}(\boldsymbol{0.05 \cdot 214 \cdot 320}))$ son aleatoriamente seleccionados de $\boldsymbol{F}$ para producir el conjunto reducido $\boldsymbol{U}$. para usar los puntos $(u_1, \ldots, u_o)$ de U y posteriormente ejecutar el algoritmo CM. Como resultado, CM produce la información mostrada en la ilustración 2 (b). La ilustración 2 también representa el mapa de densidad, así como sus puntos de atracción dibujados en rojo. Acorde con la información, el mapa de densidad puede dividirse en 3 diferentes regiones: área A, B, C ilustración 2 (c).

Los puntos de atracción localizados en el área A son caracterizados por tener una varianza pequeña, bajo estas condiciones, se pueden representar en regiones homogéneas de escala de grises en una imagen. Por otro lado, el área B representa objetos texturizados o detalles de la imagen tal como bordes. Finalmente, el área C corresponde en general a ruido no deseado. El uso del espacio bidimensional permite que el método propuesto obtenga un mayor desempeño en los resultados al detectar mejor los detalles e incorporarlos al resultado final.

3.5 Generalización de los resultados del CM

Una vez obtenidos los grupos los resultados del CM se pueden reducir a un conjunto de datos $\boldsymbol{U}$, los resultados parciales deben de generalizar la información no incluida $\widehat{\boldsymbol{U}}$. En este enfoque, para cada uno de los elementos $\widehat{\boldsymbol{u}}_{\boldsymbol{k}} \in \widehat{\boldsymbol{U}}$, es asignado al mismo grupo $\boldsymbol{C}_{\boldsymbol{i}}$ o en su defecto al grupo mas cercano $\boldsymbol{u}_{\boldsymbol{a}} \in \boldsymbol{X}$, asumiendo que $\boldsymbol{u}_{\boldsymbol{a}}$ pertenece a:

$$u_a = \arg\min d\left(\hat{u}_k, u_j\right), \qquad \hat{u}_k \in \widehat{U}, \quad u_a \in X_i \wedge u_a \in U \tag{12}$$

Donde $\boldsymbol{d}\left(\widehat{\boldsymbol{u}}_{\boldsymbol{k}}, \boldsymbol{u}_{\boldsymbol{j}}\right)$ representa la distancia euclidiana entre $\widehat{\boldsymbol{u}}_{\boldsymbol{k}}$ y $\boldsymbol{u}_{\boldsymbol{j}}$.

Una vez asignado los elementos no usados $\widehat{\boldsymbol{U}}$ a su respectivo grupo $\boldsymbol{C}_{\boldsymbol{i}}$ ($\boldsymbol{i} \in \boldsymbol{1}, \ldots, \boldsymbol{c}$), todas las listas de los grupos incrementaran sus números de elementos al incluir los elementos no usados $\widehat{\boldsymbol{U}}$. Por lo tanto, la nueva lista de grupos $\boldsymbol{X}_{\boldsymbol{i}}^{\boldsymbol{new}}$ son actualizadas como $\boldsymbol{X}_{\boldsymbol{i}}^{\boldsymbol{new}} = \boldsymbol{X}_{\boldsymbol{i}} \cup \left\{\widehat{\boldsymbol{u}}_{\boldsymbol{k}} \in \widehat{\boldsymbol{U}} \ \wedge \ \widehat{\boldsymbol{u}}_{\boldsymbol{k}} \in \boldsymbol{C}_{\boldsymbol{i}}\right\}$.

3.6 Generalización de los resultados del CM

con la operación CM muchos grupos son producidos. Sin embargo, no todos son visualmente predominantes. Por esta razón, es necesario combinar aquellos que son irrelevantes en terminaos del número de elementos contenidos. La idea principal es integrar grupos con poca población con otros grupos que presentan características similares para preservar la información visual. Por lo tanto, la idea principal es mantener los grupos con suficiente concentración mientras que los demás son fusionados.

El problema de determinar el número de elementos que caracterizan un clúster predominante es similar a detectar el punto de quiebre en ingeniería de sistemas. La ubicación del punto de quiebre representa el "punto de decisión correcto" en el cual el valor relativo de una variable ya no es significativo en términos de su contribución final. La detección mediante el punto de quiebre se aplica porque requiere un costo computacional muy reducido. En esta investigación, se experimentó con varios enfoques para discriminar los grupos producidos según su importancia. Algunos de ellos incluyen el análisis de componentes principales, la teoría del diseño de experimentos, etc. Sin embargo, estos métodos aumentan el costo computacional sin una mejora significativa en comparación con el punto rodilla. El enfoque principal de CAM-SEG es reducir el costo computacional, por eso el método de punto rodilla para separar grupos relevantes de grupos irrelevantes, es la mejor opción. Bajo este esquema, un conjunto de c pares es producidos. Cada par $\boldsymbol{V}_{\boldsymbol{l}} = (\boldsymbol{v}_{\boldsymbol{1}}^{\boldsymbol{x}}, \boldsymbol{v}_{\boldsymbol{2}}^{\boldsymbol{x}})$ es asociado a la siguiente información:

$$l = g(|X_k^{New}|); \quad v_l^x = \frac{g(|X_k^{New}|)}{c}; \quad v_l^x = \frac{|X_k^{New}|}{\sum_{i=1}^{c}|X_k^{New}|}, \ k \in 1, \ldots, c; \tag{13}$$

Donde $\left|\boldsymbol{X_k^{New}}\right|$ representa el número de elementos contenidos en el grupo $\boldsymbol{c_k}$. $\boldsymbol{g}\left(\left|\boldsymbol{X_k^{New}}\right|\right)$ es una función que renquea $\left|\boldsymbol{X_k^{New}}\right|$ dependiendo de su numero de elementos. Por lo tanto, el número uno $\boldsymbol{g}\left(\left|\boldsymbol{X_k^{New}}\right|\right) = \mathbf{1}$ es producido cuando el grupo $\boldsymbol{c_b}$ tienen el máximo número de elementos. Por otro lado, el índice más grande $\boldsymbol{g}\left(\left|\boldsymbol{X_k^{New}}\right|\right) = \boldsymbol{c}$ es asignado al grupo más pequeño $\boldsymbol{c_s}$.

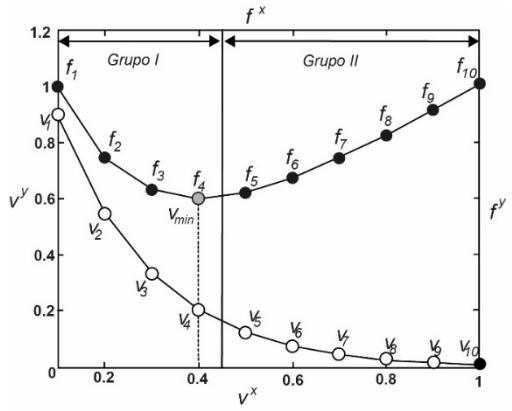


Figura 5. Detección del punto de inflexión para dividir los grupos en relevantes e irrelevantes.

La Figura 5 representa la información contenida por el conjunto de pares. Considerando la distribución $\boldsymbol{v_l}$ $(\boldsymbol{l} \in \mathbf{1}, \ldots, \boldsymbol{c})$, se genera una función de costo $\boldsymbol{f_l} = (\boldsymbol{f_l^x}, \boldsymbol{f_l^y})$ de $\boldsymbol{v_l}$, que relaciona el incremento $\boldsymbol{v_l^y}$respecto a $\boldsymbol{v_l^x}$. Esta función es calculada como:

$$f_l^x = v_l^x; \quad f_l^y = v_l^y + v_l^x; \tag{14}$$

Una característica importante de la función $\boldsymbol{f_l^y}$es que posee solo un mínimo global $\boldsymbol{v_{min}}$. Este valor corresponde al punto rodilla, el cual se define como:

$$v_{min} = \arg min\ f_l^y; \quad 0 \leq l \leq c; \tag{15}$$

En este método el punto $\boldsymbol{v_{min}}$ divide todos los grupos obtenidos por CM en los dos diferentes grupos (relevante e irrelevantes) plasmados en la Figura 5. Los grupos relevantes representan los pixeles predominantes en la segmentación de la imagen. Los grupos irrelevantes serán combinados con sus vecinos debido a su población reducida. Por lo tanto, el conjunto de grupos $\boldsymbol{c_a}, \ldots, \boldsymbol{c_d}$ es relevante, por consecuente $\boldsymbol{c_a}, \ldots, \boldsymbol{c_d}$ ∈Grupo I. Por otro lado, el conjunto de grupos $\boldsymbol{c_q}, \ldots, \boldsymbol{c_t}$ no es relevante, por consiguiente $\boldsymbol{c_q}, \ldots, \boldsymbol{c_t}$ ∈Grupo II. Grupos irrelevantes se combinan mientras que los relevantes prevalecen. En la combinación, los pixeles de cada grupo $\boldsymbol{c_w}$; $\boldsymbol{w} \in \boldsymbol{Grupo\ II}$ son integrados con los grupos $\boldsymbol{c_z}$; $\boldsymbol{z} \in \boldsymbol{Grupo\ I}$, donde $\boldsymbol{c_z}$ corresponde al *Grupo I* y la distancia entre los grupos $\left|\left|\boldsymbol{X_w^*} - \boldsymbol{X_z^*}\right|\right|$es la mínima posible, por lo tanto, estos dos grupos se fusionan. La operación final consiste en cada grupo $\boldsymbol{X_q^*}$ incluye 2 dimensiones una para la escale de grises $\boldsymbol{X_{q1}^*}$ y otra para la varianza $\boldsymbol{X_{q2}^*}$, todos los elementos de c_q son asignados a la misma escala de grises dada por $\boldsymbol{X_{q1}^*}$.

Los grupos irrelevantes contienen tan pocos elementos que son irreconocibles en la imagen a simple vista. En general, la mayoría de los grupos irrelevantes están ubicados cerca de un grupo representativo como consecuencia de diferentes factores de la imagen, como la iluminación, el ruido, etc. Para construir un grupo representativo a partir de varios grupos irrelevantes, es necesario fusionar grupos irrelevantes con características muy diferentes, lo que produce resultados inexactos en la imagen segmentada final.

3.7 Procedimiento computacional

El método propuesto fue diseñado en base a un procesamiento secuencial. El procedimiento se puede resumir en forma de seudo código en Algoritmo 1. Este esquema considera como entrada

una imagen con tamaño M x N. como primer paso, es computar la varianza local para cada pixel $\boldsymbol{V(x,y)}$ generada usando una ventana de 3 x 3. Se continúa, construyendo el espacio bidimensional de características agrupando las dos características (línea 3) escala de grises y varianza $\boldsymbol{F(x,y) = [I(x,y), V(x,y)]}$. Procediendo con el algoritmo, se selecciona el grupo reducido $\boldsymbol{U}$ (línea 5) a partir de la población $\boldsymbol{F}$, al hacer esto, se diferencian los dos conjuntos $\boldsymbol{U}$ y $\boldsymbol{\hat{U}}$, al diferenciar los dos conjuntos se puede procesar la desviación estándar $\boldsymbol{\sigma_1}$ y $\boldsymbol{\sigma_2}$ de los grupos relevantes $\boldsymbol{U}$ (línea 6). Considerando ambas desviaciones estándar, se calculan los parámetros $\boldsymbol{h_1, h_2, \delta_1}$ y $\boldsymbol{\delta_2}$ (línea 7) para ejecutar el algoritmo (línea 8) y obtener el conjunto $\boldsymbol{c}$ de puntos atractores $\{\boldsymbol{x_1^*, \dots, x_c^*}\}$ y sus respectivas listas $\boldsymbol{X_i = (u_a, \dots, u_g)}$ para cada punto de atracción $\boldsymbol{x_i^*}$ donde se enlista los puntos de datos atraídos por el ($\boldsymbol{u_a, u_g \in C_i}$). Ya generala la agrupación del algoritmo, los resultados parciales deben ser generalizados para incluir la información irrelevante $\boldsymbol{\hat{U}}$ (líneas 9-13). En esta operación, a cada elemento no utilizado $\boldsymbol{u_k \in \hat{U}}$ se le asigna el mismo clúster $\boldsymbol{C_i}$ del dato utilizado más cercano $\boldsymbol{u_a}$. Luego, se produce una función de costo $\boldsymbol{f}$, que se relaciona con el incremento del número de elementos con respecto a su importancia (línea 14). Con la información de $\boldsymbol{f}$, se detecta el punto de quiebre $\boldsymbol{v_{min}}$ (línea 15). Una vez calculado $\boldsymbol{v_{min}}$, se clasifica el número total de clústeres (línea 16) en clústeres relevantes (Grupo I) y clústeres irrelevantes (Grupo II). Posteriormente, en el proceso de combinación (líneas 17-20), los elementos de píxeles de cada clúster $\boldsymbol{C_j}$ ($\boldsymbol{j \in Grupo\ II}$) se integran con los elementos del clúster $\boldsymbol{C_a}$ ($\boldsymbol{a \in Grupo\ I}$), donde $\boldsymbol{C_a}$ corresponde al clúster del Grupo I de manera que la distancia entre sus prototipos de clúster $\boldsymbol{\left|\left|X_j^* - X_a^*\right|\right|}$ sea lo más mínima posible. Por lo tanto, los elementos de $\boldsymbol{C_a}$ también incluirán los elementos de $\boldsymbol{C_j}$ ($\boldsymbol{C_a = C_a \cap C_j}$). Finalmente, la salida del algoritmo es el clúster acumulado final del Grupo I.

Algoritmo 1

Seudo-código para el CAM-SEG propuesto

1 $input: I(x,y) de\ tamaño\ MxN$
2 $V(x,y) \leftarrow CalcularVarianza(I(x,y))$
3 $F(x,y) \leftarrow Construirlascaracterísticasdelespacio(I(x,y), V(x,y);$
4 $o \leftarrow int(0.05MxN)$
5 $[U, \hat{U}] \leftarrow Selecciódecaracterísticasparareducciondedataset(o)$
6 $[\sigma_1, \sigma_2] \leftarrow Calculardesviaciónestandard(U)$
7 $[h_1, h_2, \delta_1, \delta_2] \leftarrow ObtenerParametrosCM(\sigma_1, \sigma_2)$
8 $[(x_1^*, \dots, x_c^*), \{X_1, \dots, X_c\}] \leftarrow correrCM(U, h_1, h_2, \delta_1, \delta_2)$
9 $para\ cada\ \hat{u}_k \in \hat{U}$
10 $u_a = argmind(\hat{u}_k, u_j), u_a \in X_i \wedge u_a \in U$
11 $si\ U_a \in X_i entonces\ \hat{u}_k \in C_i$
12 $X_i^{New} \leftarrow X_i \cup \{\hat{u}_k\}$
13 $fin\ del\ para$
14 $f \leftarrow obtenerfunciónobjetivo(X_1^{New}, \dots, X_2^{New})$
15 $v_{min} \leftarrow obtenerpuntoderodilla(f)$
16 $[Grupo\ I, Grupo\ II] \leftarrow Divide\ Grupos(v_{min}, X_1^{New}, \dots, X_c^{New})$
17 $Para\ cada\ C_i \in Group\ II$
18 $C_a = argmin\ d(x_i^*, x_w^*), c \in Grupo\ I$
19 $C_a \leftarrow C_a \cup C_j$
20 $fin\ del\ para$
21 $Salida: \{C_q, C_r, \dots, C_t\} \in Grupo\ I$

4 Resultados experimentales

Con el propósito de ilustrar los resultados obtenidos con este algoritmo se utilizó un set de 300 imágenes para comparar con los resultados de otros algoritmos obtenido de Berkeley Segmentation Dataset and Benchmark (BSDS300), este data set fue seleccionado con el propósito de tener mayor compatibilidad con otros métodos. Además de contar con las imágenes, el dataset tiene un dataset con las imágenes llamadas ground truth, las cuales son imágenes segmentadas por humanos expertos en estos temas.

En los experimentos el algoritmo CAM-SEG fue comparado con los algoritmos Fast and Robust Fuzzy C-Means Clustering (FRFCM), 2D histogram and exponential Kbest gravitational search (EKGSA) y Electromagnetics-like Optimization segmentation (EMO),. En la comparación de estos algoritmos, se utilizó la mejor configuración para cada algoritmo según se especifica en su referencia, todos los experimentos fueron procesados en MATLAB en una PC con un Rysen 9 12-core a 3.7 GHz y una memoria RAM de 32 GB.

Las métricas utilizadas para evaluar el procesamiento de cada algoritmo en este análisis fueron: Structural similarity index measurement (SSIM), Feature Similarity index (FSIM), Root mean squared error (RCME), Peak Signal to Noise Ratio (PSNR), Normalized Cross-Correlation (NK), Average Difference (AD), Structural Content (SC), Maximum Difference (MD), Normalized Absolute Error (NAE), Visual Perception (VP) and Computational Cost (CC). Las primeras 9 métricas son para evaluar la calidad de la segmentación en la imagen resultante I procesada de cada algoritmo, contrastada con la imagen R la cual será su respectiva imagen segmentada por humanos o también llamada ground truth antes del procesamiento.

The Structural similarity index measurement (SSIM) evalúa la similitud entre la imagen segmentada a su referencia (ground truth). Asumiendo que I es una imagen de las mismas dimensiones que su referencia, SSIM es calculado con la siguiente formula:

$$SSIM = \frac{(2u_I u_R + Q_1)(2\sigma_{IR} + Q_2)}{(u_I^2 + u_R^2 + Q_1)(\sigma_I^2 + \sigma_R^2 + Q_2)} \qquad (16)$$

Donde $\boldsymbol{Q_1}$ y $\boldsymbol{Q_2}$ simbolizan dos constantes positivas (típicamente 0.01). $\boldsymbol{u_I}$ y $\boldsymbol{u_R}$ corresponden al valor promedio de la imagen segmentada y la de referencia respectivamente. $\boldsymbol{\sigma_I}$ y $\boldsymbol{\sigma_R}$ son las variancias de la imagen segmentada y su respectivo ground truth respectivamente y $\boldsymbol{\sigma_{IR}}$ es la covarianza de entre la imagen procesada y el ground truth el valor SSIM varia entre 0 y 1 mas cerca de 1 representa una mejor similitud.

Feature Similarity index (FSIM) es la comparación basada en índices que contrasta las estructuras locales mediante la extracción de la fase de congruencia (PC) y el valor de los gradientes (GM). El mapa de similitud S_{pc} se calcula con la siguiente formula:

$$S_{pc}(\mathrm{I},\mathrm{R}) = \frac{2 * I * R + T_1}{I^2 * R^2 + T_1} \qquad (17)$$

Donde $\boldsymbol{T_1}$ representa un valor constante positivo (típicamente 0.01). El gradiente de magnitud es se calcula utilizando las mascaras convoluciones GI y GR obtenidas de I y R respectivamente. Así, el mapa de gradiente es calculado de la siguiente manera:

$$S_{GM}(\mathrm{GI},\mathrm{GR}) = \frac{2 * GI * GR + T_1}{GI^2 * GR^2 + T_1} \qquad (18)$$

Al calcular $\boldsymbol{S_{pc}}$ y $\boldsymbol{S_{GM}}$ es posible calcular el FSIM sustituyendo valores en la siguiente formula:

$$FSIM = \frac{\sum_{j=1}^{2} S_{pc} * S_{GM} * I_j}{GI^2 * GR^2 + T_1} \tag{19}$$

Un valor más alto de FSIM representa una mejor calidad en el procesamiento de la imagen.

El valor RCME también evalúa la similitud substrayendo los valores del ground truth en la procesada para después obtener la raíz del promedio del error total, en esta ocasión un valor menor en el error representa una mejor similitud y se calcula de la siguiente manera:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(I(i,j) - R(i,j))^2}{MN}} \tag{20}$$

Donde I es la imagen procesada, R es la orignal, M y N son las dimensiones de las imágenes.

Peak Signal to Noise Ratio (PSNR) relaciona el valor máximo posible de un píxel de una imagen MAXI con su respectiva similitud en términos de la magnitud del PSNR. Un valor de PSNR más bajo implica un mejor rendimiento. Bajo estas condiciones, el PSNR se calcula de la siguiente manera:

$$PSNR = 20log_{10}\left(\frac{MAXI}{RMSE}\right) \tag{21}$$

Normalized Cross-Correlation (NK) evalúa la coincidencia entre la imagen segmentada y la imagen de referencia en términos de su correlación. Un valor alto de NK representa un mejor procesamiento y se calcula de la siguiente manera:

$$NK = \frac{\sum_{i=1}^{M}\sum_{i=1}^{N} I(i,j) \cdot R(i,j)}{\sum_{i=1}^{M}\sum_{i=1}^{N} R(i,j)} \tag{22}$$

The Average Difference (AD) evalúa la diferencia entre la imagen segmentada y la de referencia simplemente obteniendo el promedio de la diferencia entra la procesada menos su respectivo ground truth, el valor absoluto de lo anterior fija un valor que diferencia la imagen de referencia respecto a la procesada, por lo tanto, un valor más cerca del 0 representa una menor diferencia y mejor procesamiento, se obtiene de la siguiente manera:

$$AD = \left|\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j) - R(i,j)}{MN}\right| \tag{23}$$

The Structural Content (SC) evalúa similitud mediante la autocorrelación indexada. Un menor valor implica un mejor procesamiento. Se calcula con la siguiente formula:

$$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} I(i,j)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N} R(i,j)^2} \tag{24}$$

Maximun Difference (MD) mide la máxima disparidad entre una imagen procesada y la imagen de referencia. Un valor menor de MD representa una mejor segmentación, calculándose de la siguiente manera:

$$MD = MAX|I(i,j) - R(i,j)|;\ \forall i,j \in M \times N \tag{25}$$

The Normalized Absolute Error (NAE) evalúa la similitud entre las imágenes considerando el valor absoluto de los errores normalizados producido entre las imágenes, un valor menor implica un mejor resultado, calculándose de la siguiente manera:

$$NAE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}|I(i,j) - R(i,j)|}{\sum_{i=1}^{M}\sum_{j=1}^{N}R(i,j)} \tag{26}$$

Visual Perception (VP) provee una manera cuantitativa de una imagen en términos de percepción humana. Aunque aun es tema de estudio, algunos estudios [63-64] indican que la percepción humana esta relacionado con las diferencias de los detalles percibidos por un espectador. VP considera las diferencias entre las principales características de los resultados de una imagen. Para evaluar VP de una imagen I, el modelo adoptado involucra la intensidad de los bordes de los pixeles, el numero de bordes existentes y la entropía completa de la imagen. Esta información se procesa de la siguiente manera:

$$VP = \log\big(log\big(E(I)\big)\big) \cdot \frac{NE(I)}{M \times N} \cdot H(I) \tag{27}$$

Donde $E(I)$ representa la suma de todas las intensidades de los bordes obtenidos por el filtro de sobel. $NE(I)$ corresponde a el numero de bordes obtenido del procesamiento. $H(I)$ simboliza la entropía de la imagen. Un valor más alto de VP reprecenta más información de la imagen y por tanto un mejor procesamiento. $H(I)$ es calculado de la siguiente manera:

$$H(I) = \sum_{i=0}^{L-1} P(q) \cdot \log\left(P(q)\right) \tag{28}$$

Donde $P(q)$ se refiere a la función de densidad de probabilidad en el nivel de intensidad $P\ (q)\ \in 0, \dots, l-1)$ de la imagen. Es evidente hasta cierto punto las imágenes son visualmente aceptables por el ojo humano, sin embargo, después de cierto umbral la imagen se vuelve muy compleja para su visualización, algo que no toma en cuenta esta fórmula.

Los métodos de segmentación considerados en las comparaciones son, en general, procesos complejos con varias operaciones aleatorias y subrutinas estocásticas. Por lo tanto, es impráctico realizar un análisis de complejidad desde un punto de vista determinista. Por esta razón, se evalúa el Costo Computacional (CC) considerando el esfuerzo computacional invertido para cada algoritmo. El CC se evalúa, asumiendo el esfuerzo computacional promediado en segundos de 30 ejecuciones independientes. Un valor de CC bajo significa un mejor rendimiento.

Tabla 1
Comparación de características del método propuesto CAM-SEG con otros métodos considerando la dataset de Berkleley.

Métricas	FRFCM	EKGSA	EMO	**CAM-SEG**
SSIM	0.6621	0.6087	0.5901	0.7981
FSIM	0.7821	0.7897	0.7697	0.8025
RMSE	50.5218	52.0279	54.0974	48.05110
PSNR	22.3987	23.1187	24.9874	20.4410
NK	0.9875	0.8821	0.7889	1.02174
AD	42.8714	43.1101	44.0124	42.1621
SC	1.3781	1.2871	1.2721	1.0874
MD	110.9871	111.1498	112.0074	110.2187
NAE	0.1674	0.1724	0.1821	0.1081
VP	0.2887	0.3887	0.4150	0.6825
CC	48.97	84.87	108.21	27.14

4.1 Comparación

Esta subsección muestra la comparación de rendimiento de los diferentes esquemas de segmentación cuando se utilizan en las 300 imágenes del conjunto de datos Berkeley. El objetivo es determinar, en términos generales, la eficiencia y efectividad de cada método. La Tabla 1 muestra los valores promedio de cada índice considerado sobre el conjunto de 300 imágenes de BSDS300. Para el funcionamiento de los métodos EKGSA y EMO, es necesario definir de antemano el número de clases en las cuales se segmentarán las imágenes. Por lo tanto, en las comparaciones, el número de clases se fija en cuatro niveles. Este número se ha decidido para mantener la compatibilidad con los resultados obtenidos para tales esquemas en sus respectivas referencias [36,37].

Inspeccionando la tabla 1, se observa que el método propuesto presenta mejores valores promedios entre los competidores. Esto da a entender que el algoritmo CAM-SEG conlleva la mejor semejanza entre los resultados segmentados y sus correspondientes referencias (ground truth). El algoritmo FRFCM presenta la segunda mejor segmentación mientras que el EKGSA y EMO producen los peores resultados. Por otro lado, la propuesta CAM-SEG ofrece el mejor valor en cuanto a percepción visual (VP), seguida de cerca por EKGSA y EMO, mientras que FRFCM presenta la peor evaluación de percepción visual. De acuerdo con estos valores, queda claro que la propuesta CAM-SEG mantiene más del 50% de la VP considerando su competidor más cercano (EMO). También es importante destacar que la propuesta CAM-SEG mantiene el menor esfuerzo computacional en comparación con otras técnicas de segmentación. Para el costo computacional, se puede decir que el algoritmo propuesto es cerca de dos veces más rápido que el FRFCM, aproximadamente tres veces más rápido que el EKGSA y cerca de cuatro veces más rápido que el EMO.

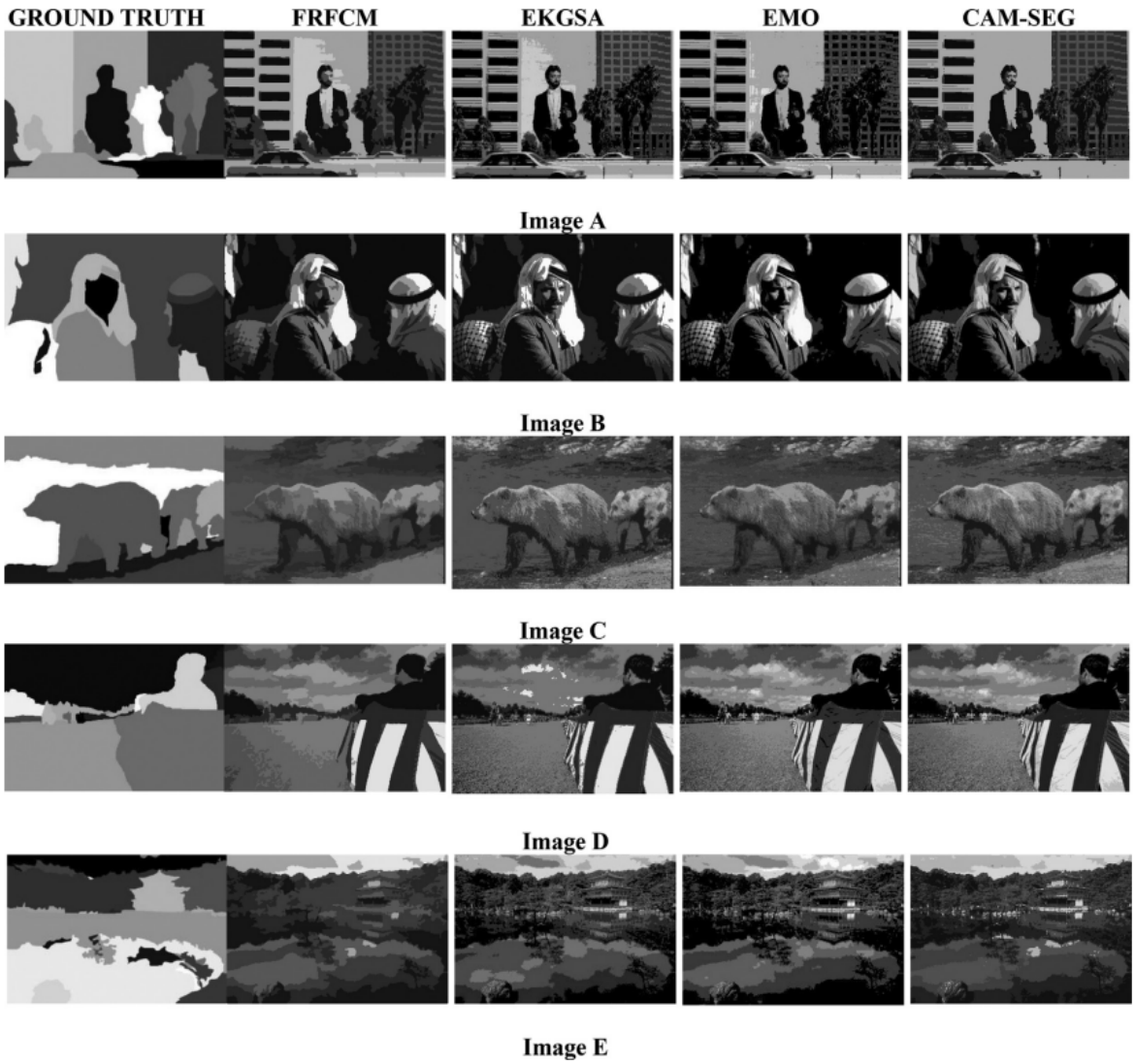


Figura 6. Resultado de la segmentación visual de un conjunto representativo de imágenes del conjunto de datos de Berkeley.

Con objeto de visualización de los resultados se utilizaron 6 imágenes para ilustrar los resultados obtenidos. Estas imágenes se obtuvieron de un conjunto de datos llamado BSDS300 generado por la universidad de Berkeley. Estas imágenes fueron seleccionadas por su complejidad al momento de generar la segmentación. Se puede observar que el método propuesto presenta una mejor percepción visual en comparación con los otros competidores. En todas las imágenes, detalles como textura o pequeños elementos son claramente mostrados por CAM-SEG, mientras que los otros algoritmos eliminan detalles interesantes. Este rendimiento notable es consecuencia de su mecanismo robusto para detectar detalles a partir del mapa de características generado por el algoritmo CM.

La tabla 2 presenta el resumen de los resultados de las imágenes seleccionadas. La tabla reporta las métricas anteriormente descritas.

Tabla 2
Métricas de características para el conjunto representativo de las imágenes presentadas en la Figura 6.

Imagen	Métricas	FRFCM	EKGSA	EMO	**CAM-SEG**
Imagen A	SSIM	0.7093	0.7030	0.6103	0.7630
	FSIM	0.7539	0.6950	0.7788	0.8364
	RMSE	24.9397	19.2821	32.2896	23.0181
	PSNR	20.1929	22.4277	17.9495	20.8893
	NK	0.8520	0.9327	0.8103	0.9746
	AD	31.1213	37.254	36.9640	19.7554
	SC	1.3461	1.1112	1.4582	1.0282
	MD	80	70	89	65
	NAE	0.2024	0.1391	0.2585	0.1207
	VP	0.2170	0.4954	0.4087	0.6414
	CC	42.14	80.12	100.12	26.21
Imagen B	SSIM	0.4531	0.4381	0.5901	0.8883
	FSIM	0.7703	0.8553	0.7697	0.8551
	RMSE	22.2047	22.7306	54.0974	12.8758
	PSNR	22.9871	20.9985	24.9874	25.9352
	NK	0.9505	0.8073	0.7889	0.9669
	AD	32.1436	39.5537	44.0124	25.7359
	SC	1.0733	1.4834	1.2721	1.0493
	MD	150	56	112.0074	111
	NAE	0.1654	0.3115	0.1821	0.1129
	VP	0.3010	0.4971	0.4150	0.7121
	CC	39.14	81.14	108.21	25.14
Imagen C	SSIM	0.6411	0.7652	0.3557	0.8030
	FSIM	0.6934	0.8473	0.8047	0.8667
	RMSE	13.2809	18.3313	24.5442	10.6946
	PSNR	25.6662	22.8669	20.3317	27.5474
	NK	0.9600	0.8521	0.8491	0.9921
	AD	41.9045	45.1565	35.9246	38.9610
	SC	1.0669	1.3569	1.3000	1.0045
	MD	41	45	59	74
	NAE	0.0941	0.1594	0.1339	0.0757
	VP	0.4107	0.6921	0.5472	0.8721
	CC	37.14	87.97	97.24	26.87
Imagen D	SSIM	0.6597	0.7702	0.7521	0.7776
	FSIM	0.7406	0.7733	0.8417	0.7985
	RMSE	17.7907	17.7575	19.0758	17.3683
	PSNR	23.1269	23.1413	22.5211	23.3356
	NK	0.9654	0.8931	0.8498	0.9734
	AD	41.8971	44.6256	46.2638	31.7830
	SC	1.0517	1.2406	1.3602	1.0354

	MD	154	36	44	135
	NAE	0.0972	0.1403	0.1710	0.0978
	VP	0.3802	0.5901	0.6014	0.8511
	CC	40.14	90.01	89.21	28.57
Imagen E	SSIM	0.6286	0.6875	0.7190	0.8015
	FSIM	0.6754	0.8233	0.7364	0.8242
	RMSE	17.8809	22.9722	20.0168	14.0111
	PSNR	23.0829	20.9067	22.1028	25.2013
	NK	0.9595	0.8251	0.8796	0.9832
	AD	41.1631	49.7261	46.9961	37.3525
	SC	1.0521	1.4242	1.2748	1.0144
	MD	121	54	46	50
	NAE	0.1424	0.2320	0.1526	0.1201
	VP	0.5109	0.8001	0.6587	0.9127
	CC	49.01	110.61	114.07	30.17

En la tabla 2, se puede observar la mejora basada en las métricas de segmentación mediante el método propuesto. En general EKGSA y EMO tiene un mal desempeño, Esto dado que se ejecutaron variar veces para producir resultados consistentes. En el caso del FRFCM, mantiene un buen procesamiento desde la perspectiva numérica. Sin embargo, la percepción visual respecto a las mejores métricas que obtiene no es congruentes, observándose valores en las métricas altos mientras que la imagen presenta una mala calidad de segmentación. A diferencia de FRFCM, la propuesta CAM-SEG obtiene los mejores valores de VP. Este comportamiento notable es consecuencia de su capacidad para identificar grupos de píxeles que representan detalles en las imágenes. Por otro lado, los métodos EKGSA y EMO alcanzan mejores valores de VP que FRFCM, pero peores que CAM-SEG. Estos resultados son producto de sus principios de funcionamiento.

Una inspección detallada de la Tabla 2 muestra que el método propuesto CAM-SEG presenta valores deficientes en algunos índices como MD o PSNR en algunas imágenes. El algoritmo propuesto presenta una notable capacidad para identificar regiones homogéneas que representan detalles finos en las imágenes. En tales circunstancias, dado que las imágenes de referencia discriminan principalmente información semántica (como un oso, una cara, una persona, etc.), en algunas regiones (secciones con detalles) podrían aparecer grandes diferencias entre la imagen segmentada de CAM-SEG y su respectiva imagen de referencia. Aunque estas diferencias producen resultados numéricos deficientes en algunos índices de rendimiento, aumentan la percepción visual (VP) de la imagen segmentada.

En la tabla 2, al observar el costo computacional de los algoritmos, está absolutamente claro que el método propuesto mantiene el valor mas bajo y por tanto un menor costo computacional. Aunque FRFCM compite con el algoritmo propuesto mejor que EKGSA y EMO, sus resultados son peores que los del método CAM-SEG, acorde con los resultados se puede concluir que el método propuesto presente el mejor balance entre calidad de segmentación y velocidad.

5. Conclusiones

En este artículo, se presenta un nuevo algoritmo de segmentación competitivo para imágenes en escala de grises. El enfoque propuesto basado en el algoritmo de desplazamiento medio considera un mapa de características bidimensional que incluye el valor de escala de grises y la varianza local para cada píxel de la imagen.

Para reducir el costo computacional, el algoritmo de cambio medio (MS) se modifica para operar con un número muy limitado de puntos de todos los datos disponibles. En tales condiciones, se diferencian dos conjuntos de elementos: elementos usados (el conjunto de datos reducido considerado en la operación MS) y elementos no utilizados (el resto de datos disponibles).

A diferencia del algoritmo MS clásico que emplea funciones gaussianas, en nuestro enfoque, el proceso para estimar los valores de densidad en el mapa de características utiliza la función del núcleo de Epanechnikov. Este núcleo presenta la mejor precisión de estimación cuando la cantidad de datos para calcular la densidad del mapa de características es muy limitada [39].

Una vez obtenidos los resultados del MS, se generalizan para incluir los datos no implicados. Por lo tanto, cada elemento no utilizado se asigna al mismo grupo de datos utilizados más cercanos. Finalmente, los clusters con la menor cantidad de elementos se fusionan con otros clusters vecinos.

En los experimentos, hemos aplicado el algoritmo CAM-SEG a las 300 imágenes del conjunto de datos de Berkeley, y sus resultados se comparan con los producidos por el algoritmo de agrupación de medias C difusa rápido y robusto (FRFCM) [53], el histograma 2D y el algoritmo de búsqueda gravitacional exponencial Kbest (EKGSA)[24]y la segmentación de optimización similar a la electromagnética (EMO) [54]. El primer método, FRFCM, es un enfoque de agrupación de última generación para fines de segmentación, mientras que EKGSA y EMO representan dos algoritmos de segmentación representativos basados en esquemas metaheurísticos. En nuestro análisis se han considerado once índices de desempeño: Medición del índice de similitud estructural (SSIM), índice de similitud de características (FSIM), raíz del error cuadrático medio (RMSE), Relación pico señal-ruido (PSNR), correlación cruzada normalizada (NK), diferencia promedio (ANUNCIO), Contenido Estructural (CAROLINA DEL SUR), diferencia máxima (Maryland), error absoluto normalizado (NAE), Percepción visual (vicepresidente) y costo computacional (CC). Los resultados experimentales confirman que el esquema propuesto produce imágenes segmentadas con una calidad de percepción visual un 50% mejor aproximadamente dos veces (≈1,8 − 2) más rápido que sus competidores.

Referencias

[1] K.S. Tan, N.A.M Isa, Color image segmentation using histogram thresholding– fuzzy c-means hybrid approach, Pattern Recognit. 44 (2011) 1–15.

[2] H.-D. Cheng, X. Jiang, J. Wang, 2002. Color image segmentation based on homogram thresholding and region merging, Pattern Recognit 35 (2002) 373–393.

[3] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2002) 888–905.

[4] Y. Tian, J. Li, S. Yu, T. Huang, Learning complementary saliency priors for foreground object segmentation in complex scenes, Int. J. Comput. Vis. 111 (2015) 153–170.

[5] P.F. Felzenszwalb, D.P Huttenlocher, Efficient graph-based image segmentation, Int. J. Comput. Vis. 59 (2004) 167–181.

[6] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 898–916.

[7] X. Zhang, C. Xu, M. Li, X. Sun, Sparse and low-rank coupling image segmentation model via nonconvex regularization, Int. J. Pattern Recognit. Artif. Intell. 29 (2015) 1–22.

[8] A. Dirami, K. Hammouche, M. Diaf, P.. Siarry, Fast multilevel thresholding for image segmentation through a multiphase level set method, Signal Process. 93 (2013) 139–153.

[9] H. Zhang, J.E. Fritts, S.A Goldman, Image segmentation evaluation: a survey of unsupervised methods, Comput. Vis. Image Underst. 110 (2008) 260–280.

[10] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, J. Electron. Imaging 13 (2004) 146–168.

[11] M. Mignotte, A label field fusion Bayesian model and its penalized maximum rand estimator for image segmentation, IEEE Trans. Image Process. 19 (2010) 1610–1624.

[12] M. Krinidis, I. Pitas, Color texture segmentation based on the modal energy of deformable surfaces, IEEE Trans. Image Process. 18 (2009) 1613–1622.

[13] Y. Han, X.-C. Feng, G. Baciu, Variational and PCA based natural image segmentation, Pattern Recognit. 46 (2013) 1971–1984.

[14] Z. Yu, O.C. Au, R. Zou, W. Yu, J. Tian, An adaptive unsupervised approach toward pixel clustering and color image segmentation, Pattern Recognit. 43 (2010) 1889–1906.

[15] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, A.K. Nandi, Significantly fast and robust fuzzy C-Means clustering algorithm based on morphological reconstruction and membership filtering, IEEE Trans. Fuzzy Syst. 26 (5) (2018) 3027–3041.

[16] A.S. Abutaleb, Automatic thresholding of gray-level pictures using two-dimensional entropy. Comput. Vis. Graph. Image Process.. 47, (1089), 22–32.

[17] A. Brink, Thresholding of digital images using two-dimensional entropies, Pattern Recognit. 25 (1992) 803–808.

[18] A. Buades, B. Coll, J.-M. Morel, A non-local algorithm for image denoising, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005.

[19] A.B Ishak, Choosing parameters for Rényi and Tsallis entropies within a two-dimensional multilevel image segmentation framework, Physica A 466 (2017) 521–536.

[20] X. Zhao, M. Turk, W. Li, K.-c Lien, G. Wang, A multilevel image thresholding segmentation algorithm based on two-dimensional K–L divergence and modified particle swarm optimization, Appl. Soft Comput. 48 (2016) 151–159.

[21] W. Xue-guang, C. Shu-hong, An improved image segmentation algorithm based on two-dimensional Otsu method, Inform. Sci. Lett. 1 (2012) 77–83.

[22] C. Sha, J. Hou, H. Cui, A robust 2D Otsu's thresholding method in image segmentation, J. Vis. Commun. Image Represent. 41 (2016) 339–351.

[23] S. Sarkar, S. Das, Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach, IEEE Trans. Image Process. 22 (2013) 4788–4797.

[24] A. Nakib, S. Roman, H. Oulhadj, P. Siarry, Fast brain MRI segmentation based on two-dimensional survival exponential entropy and particle swarm optimization, in: Proceedings of the International Conference on Engineering in Medicine and Biology Society, 2007.

[25] A. Nakib, H. Oulhadj, P. Siarry, Image thresholding based on pareto multiobjective optimization, Eng. Appl. Artif. Intell. 23 (2010) 313–320.

[26] X.-S. Yang, Nature-inspired Optimization AlgorithCM, Elsevier, 2014.

[27] Y.-g. Tang, D. Liu, X.-p. Guan, Fast image segmentation based on particle swarm optimization and two-dimension otsu method, Control Decis. 22 (2007) 202–205.

[28] X. Lei, A. Fu, Two-dimensional maximum entropy image segmentation method based on quantum-behaved particle swarm optimization algorithm, in: Proceedings of the International Conference on Natural Computation, 2008.

[29] C. Qi, Maximum entropy for image segmentation based on an adaptive particle swarm optimization, Appl. Math. 8 (2014) 3129–3135.

[30] X. Shen, Y. Zhang, F. Li, An improved two-dimensional entropic thresholding method based on ant colony genetic algorithm, in: Proceedings of the WRI Global Congress on Intelligent SysteCM, 2009.

[31] H. Cheng, Y. Chen, X. Jiang, Thresholding using two-dimensional histogram and fuzzy entropy principle, IEEE Trans. Image Process. 9 (2000) 732–735.

[32] S. Kumar, T.K. Sharma, M. Pant, A. Ray, Adaptive artificial bee colony for segmentation of CT lung images, in: Proceedings of the International Conference on Recent Advances and Future Trends in Information Technology, 2012.

[33] S. Fengjie, W. He, F. Jieqing, 2D Otsu segmentation algorithm based on simulated annealing genetic algorithm for iced-cable images, in: Proceedings of the International Forum on Information Technology and Applications, 2009.

[34] L. Xiao-Feng, L. Hui-Ying, Y. Ming, W. Tai-Ping, Infrared image segmentation based on AAFSA and 2D-renyi entropy threshold selection, in: Proceedings of the Joint International Conference on Artificial Intelligence and Computer Engineering and International Conference on Network and Communication Security, 2016.

[35] R. Panda, S. Agrawal, L. Samantaray, A. Abraham, An evolutionary gray gradient algorithm for multilevel thresholding of brain MR images using soft computing techniques, Appl. Soft Comput. 50 (2017) 94–108.

[36] D. Oliva, E. Cuevas, G. Pajares, D. Zaldivar, V. Osuna, A multilevel thresholding algorithm using electromagnetism optimization, Neurocomputing 139 (2014) 357–381.

[37] H. Mittal, M. Saraswat, An optimum multi-level image thresholding segmentation using non-local means 2D histogram and exponential Kbest gravitational search algorithm, Eng. Appl. Artif. Intel. 71 (2018) 226–235.

[38] M.P. Wand, M.C. Jones, Kernel Smoothing, Springer, 1995.

[39] J.E. Chacón, Data-driven choice of the smoothing parametrization for kernel density estimators, Can. J. Stat. 37 (2009) 249–265.

[40] K.S Duong, Kernel density estimation and Kernel discriminant analysis for multivariate data in R, J. Stat. Softw. 21 (2007) 1–16.

[41] A. Gramacki, Nonparametric Kernel Density Estimation and Its Computational Aspects, Springer, 2018.

[42] V.A. Epanechnikov, Non-parametric estimation of a multivariate probability density, Theory Probab. Appl. 14 (1969) 153–158.

[43] Y.Z. Cheng, Mean Shift, mode seeking, and clustering, IEEE Transact. Pattern Anal. Mach. Intel. 17 (8) (1995) 790–799.

[44] Y. Guo, A. S¸engür, Y. Akbulut, A. Shipley, An effective color image segmentation approach using neutrosophic adaptive mean shift clustering, Measurement 119 (2018) 28–40.

[45] G. Domingues, H. Bischof, R. Beichel, Fast 3D mean shift filter for CT images, in: Proceedings of the Scandinavian Conference on Image Analysis, Sweden, 2003, pp. 438–445.

[46] D. Comaniciu, P. Meer, Meanshift: a robust approach toward feature space analysis, IEEE Trans. Pattern Anal. Mach. Intel. 24 (5) (2002) 603–619.

[47] W.B. Tao, J. Liu, Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation, Opt. Eng. 46 (2007) 12.

[48] J.H. Park, G.S. Lee, S.Y. Park, Color image segmentation using adaptive mean shift and statistical model-based methods, Comput. Math. Appl. 57 (2009) 970–980.

[49] Q. Li, J.S Racine, Nonparametric Econometrics: Theory and Practice, Princeton University Press, 2007 ISBN 978-0-691-12161-1.

[50] Y. Luo, K. Zhang, Y. Chai, Y. Xiong, Muiti-parameter-setting based on data original distribution for DENCLUE optimization, IEEE Access 6 (2018) 16704–16711.

[51] L. Xu, E. Oja, P. Kultanen, A new curve detection method: Randomized Hough transform (RHT), Pattern Recognit. Lett. 11 (5) (1990) 331–338.

[52] M.A. Fisher, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.

[53] I. Horová, J. Kolácek ˇ , J. Zelinka, Kernel Smoothing in Matlab, World Scientific, 2012.

[54] D.W. Scott, Scott's rule, Wires Comput. Stat. 2 (4) (2010) 497–502.

[55] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[56] S.K. Choy, T.C. Ng, C. Yu, Unsupervised fuzzy model-based image segmentation, Signal Process. 171 (2020) 107483.

[57] N. Dhanachandra, Y.J. Chanu, An image segmentation approach based on fuzzy c-means and dynamic particle swarm optimization algorithm, Multimed. Tools Appl. (2020), https://doi.org/10.1007/s11042-020-08699-8.

[58] C. Wu, Y. Chen, Adaptive entropy weighted picture fuzzy clustering algorithm with spatial information for image segmentation, Appl. Soft Comput. 86 (2020) 105888.

[59] A.A. Hernandez del Rio, E. Cuevas, D. Zaldivar, Multi-level image thresholding segmentation using 2d histogram non-local means and metaheuristics algorithCM, in: D. Oliva, S. Hinojosa (Eds.), Applications of Hybrid Metaheuristic AlgorithCM for Image Processing. Studies in Computational Intelligence, 890, Springer, 2020.

[60] B. Vinoth Kumar, S. Sabareeswaran, G. Madumitha, A decennary survey on artificial intelligence methods for image segmentation, in: R. Venkata Rao, J. Taler (Eds.), Advanced Engineering Optimization Through Intelligent Techniques. Advances in Intelligent SysteCM and Computing, 949, Springer, 2020.

[61] M. Chouksey, R.K. Jha, R. Sharma, A fast technique for image segmentation based on two Meta-heuristic algorithCM, Multimed. Tools Appl. (2020), https://doi.org/10.1007/s11042-019-08138-3.

[62] A. Draa, A. Bouaziz, An artificial bee colony algorithm for image contrast enhancement, Swarm Evol. Comput. 16 (2014) 69–84.

[63] A. Draa, A. Bouaziz, An artificial bee colony algorithm for image contrast enhancement, Swarm Evol. Comput. 16 (2014) 69–84.

[64] L.D.S. Coelho, J.G. Sauer, M. Rudek, Differential evolution optimization combined with chaotic sequences for image contrast enhancement, Chaos Sol. Fract. 42 (2009) 522–529.

[65] C. Munteaunu, A. Rosa, Gray-scale image enhancement as an automatic process driven by evolution, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 34 (2) (2004) 1992–1998.

[66] M. Braik, A. Sheta, A. Ayesh, Particle swarm optimisation enhancement approach for improving image quality, Int. J. Innov. Comput. Appl. 1 (2) (2007) 138–145.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.

COMPUTATION E INFORMATICA

Recibido 11 Mar. 2024 *ReCIBE, Año 13 No. 1, May. 2024*

Aceptado 18 Mar. 2024

Estudio bibliográfico sobre las metodologías de big data y su aplicación en la movilidad inteligente

Bibliographical study on big data methodologies and their application in smart mobility

Viteri Chica Miguel Andrés*
mviteri2994@utm.edu.ec
Chancay-García Leonardo*
leonardo.chancay@utm.edu.ec

*Universidad Técnica de Manabí, Facultad de Ciencias Informáticas, Portoviejo, Ecuador.

Resumen

El avance de las tecnologías de la información ha revolucionado numerosos aspectos de la vida cotidiana, incluyendo la manera en que gestionamos y planificamos la movilidad en las ciudades. La implementación de big data en la movilidad inteligente promete transformar la eficiencia, seguridad y sostenibilidad del transporte urbano, ofreciendo soluciones innovadoras para los desafíos modernos.

El presente artículo científico tiene el propósito de analizar el impacto del uso de big data en la movilidad inteligente y cómo puede mejorar la planificación y gestión del transporte urbano. Se examinan las principales áreas de investigación, evaluando las contribuciones y limitaciones de la literatura existente.

Se realizó una revisión sistemática de la literatura en las principales bases de datos científicas como IEEE Xplore, Scopus y Web of Science, seleccionando artículos publicados entre los años 2017 y 2022 en inglés, español y portugués. Este estudio bibliográfico busca proporcionar una visión general de las investigaciones existentes sobre el uso de big data para la movilidad inteligente y destacar las tendencias emergentes en esta área.

Las aplicaciones del big data en la movilidad inteligente incluyen la predicción de congestiones, el análisis de patrones de tráfico y la optimización de rutas de transporte. Estas aplicaciones tienen un impacto significativo en el mundo real, mejorando la eficiencia del transporte, reduciendo la congestión y promoviendo un entorno urbano más sostenible. En definitiva, el uso de big data en la movilidad inteligente se presenta como un área de investigación en constante evolución, con un gran potencial para transformar la gestión del transporte en las ciudades modernas.

Palabras clave: Big data, movilidad inteligente, sistemas de transporte inteligente, dispositivos móviles.

Abstract

The advancement of information technologies has revolutionized numerous aspects of daily life, including the way we manage and plan mobility in cities. The implementation of big data in smart mobility promises to transform the efficiency, safety and sustainability of urban transportation, offering innovative solutions to modern challenges.

The purpose of this scientific article is to analyze the impact of the use of big data on smart mobility and how it can improve the planning and management of urban transportation. The main areas of research are examined, evaluating the contributions and limitations of the existing literature.

A systematic review of the literature was carried out in the main scientific databases such as IEEE Xplore, Scopus and Web of Science, selecting articles published between 2017 and 2022 in English, Spanish and Portuguese. This literature study seeks to provide an overview of existing research on the use of big data for smart mobility and highlight emerging trends in this area.

Applications of big data in smart mobility include congestion prediction, traffic pattern analysis, and transportation route optimization. These applications have a significant real-world impact, improving transportation efficiency, reducing congestion and promoting a more sustainable urban environment. In short, the use of big data in smart mobility is presented as an area of research in constant evolution, with great potential to transform transportation management in modern cities.

Keywords: Big data, smart mobility, smart transportation systems, mobile devices.

1. Introducción

El ser humano es una especie que ha desarrollado una tendencia a ser nómada, es decir, a moverse de un lugar a otro en busca de alimento, refugio y mejores condiciones de vida. Dicha característica se ha mantenido a lo largo de la historia, desde los primeros humanos hasta la actualidad, aunque con diferentes motivos y medios de transporte acortando de alguna manera las distancias. Por otro lado, las tecnologías informáticas se han venido utilizando desde hace varios años con la finalidad de automatizar la vida humana.

La movilidad inteligente es un concepto que se basa en el uso de las tecnologías digitales y de la información para mejorar la eficiencia, seguridad, accesibilidad y sostenibilidad de los sistemas de transporte y movilidad. Según (Carlo Ratti, 2020), "la movilidad inteligente implica la integración de las tecnologías digitales con la infraestructura física de transporte, para mejorar la eficiencia y la accesibilidad del sistema de transporte urbano y ofrecer a los usuarios una experiencia de transporte más cómoda y sostenible".

Por su parte el big data se refiere a conjuntos de datos muy grandes y complejos que no se pueden manejar con herramientas de procesamiento de datos tradicionales. Así mismo, el trabajo presentado por (Babak Hodjat, 2019), define al big data como: "un conjunto de tecnologías y herramientas que permiten recolectar, almacenar, procesar y analizar grandes cantidades de datos, para obtener información valiosa que ayude a tomar decisiones más informadas y eficientes."

El uso de tecnologías de big data para la movilidad inteligente tiene el potencial de revolucionar la forma en que las ciudades y los gobiernos manejan el transporte y la movilidad de sus ciudadanos, lo que va a reducir notablemente el consumo de combustibles fósiles y la contaminación ambiental en las zonas urbanas que generalmente son altamente pobladas, todo esto va a permitir mejorar notablemente la salud de sus ciudadanos. Los avances en big data y la movilidad inteligente han sido significativos en los últimos años, y con la creciente urbanización y la necesidad de soluciones de transporte más eficientes y sostenibles en las ciudades, se han realizado diversos estudios que han sido documentados por varios autores. Por ejemplo, (Seguí-Amortegui et al. 2019) en su trabajo orientado a ciudades de España señala que la urbanización es uno de los mayores desafíos a los que se enfrenta la movilidad en la actualidad, y que la gestión del transporte es fundamental para lograr un desarrollo sostenible en las ciudades.

Por otro lado, (Cheng et al. 2020) señalan que el análisis de big data puede ayudar a mejorar la planificación del transporte y la toma de decisiones en la gestión de la movilidad en las ciudades chinas. Es así que el uso de aplicaciones móviles y otros dispositivos conectados para recopilar datos de movilidad de los usuarios ha sido documentado por varios autores. Por ejemplo, Leng et al. (2019) en el análisis a las aplicaciones móviles chinas les pueden ayudar a proporcionar datos valiosos sobre la movilidad de los usuarios, incluyendo la ubicación, el tiempo de viaje y la preferencia de ruta, con el fin de mejorar notablemente la toma de decisiones a la hora de automatizar el tráfico.

El presente artículo sobre el big data orientado a la movilidad inteligente muestra un estudio bibliográfico el cual es importante debido a la gran cantidad de información que se utiliza para automatizar el tráfico, así como la movilidad inteligente se está convirtiendo rápidamente en una prioridad para las ciudades y las comunidades, se está considerando como una forma de mejorar la eficiencia del transporte, reducir la congestión y aumentar la sostenibilidad del medio ambiente. Por lo tanto, es importante conocer las técnicas y herramientas utilizadas en este ámbito, así como los datos pueden ser utilizados para mejorar la movilidad inteligente y cómo se pueden integrar en los sistemas existentes de transporte. En definitiva, se trata de un tema de gran relevancia e interés en la actualidad, que merece ser investigado y analizado en profundidad para mejorar la movilidad en las ciudades modernas.

2. Marco Teórico Big Data

Big data se refiere a la recopilación, almacenamiento y análisis de grandes volúmenes de datos que son demasiado complejos para ser manejados por las herramientas tradicionales de procesamiento de datos. En el contexto de la movilidad inteligente, big data permite analizar patrones de tráfico, prever congestiones y optimizar rutas en tiempo real.
Aplicación: Utilización de técnicas de minería de datos y análisis predictivo para mejorar la eficiencia del transporte urbano y reducir los tiempos de viaje.

Movilidad Inteligente:

La movilidad inteligente implica el uso de tecnologías avanzadas como IoT, inteligencia artificial y big data para crear sistemas de transporte más eficientes, seguros y sostenibles. Se centra en la integración de diversos modos de transporte y la optimización del uso de infraestructuras existentes.
Aplicación: Implementación de sistemas de gestión de tráfico adaptativos, vehículos autónomos y plataformas de movilidad como servicio (MaaS).

Internet de las Cosas (IoT)

IoT se refiere a la interconexión de dispositivos a través de internet, que pueden recopilar y compartir datos en tiempo real. En la movilidad inteligente, IoT permite la monitorización continua del tráfico, las condiciones de las carreteras y el comportamiento de los usuarios.
Aplicación: Uso de sensores inteligentes en semáforos, vehículos y paradas de autobús para gestionar el tráfico de manera más eficiente y mejorar la seguridad vial.

Inteligencia Artificial (IA) y Aprendizaje Automático (ML):

La IA y el ML son tecnologías que permiten a los sistemas aprender y mejorar a partir de datos. En el ámbito de la movilidad inteligente, estas tecnologías se utilizan para analizar datos de tráfico, prever demandas de transporte y optimizar rutas.
Aplicación: Implementación de algoritmos de aprendizaje profundo para la predicción de congestiones y la optimización dinámica de rutas de transporte público.

Aplicaciones Más Novedosas y de Mayor Impacto Sistemas de Gestión de Tráfico

Inteligente:

Descripción: Sistemas que utilizan IA y big data para monitorizar y gestionar el flujo de tráfico en tiempo real.
Impacto: Reducción de congestiones, mejora de la seguridad y optimización del uso de infraestructuras viales.

Vehículos Autónomos:

Descripción: Vehículos equipados con sensores y algoritmos de IA que pueden operar sin intervención humana.
Impacto: Incremento en la seguridad vial, reducción de accidentes y mejora en la eficiencia del transporte.

Plataformas de Movilidad como Servicio (MaaS):

Descripción: Servicios que integran diversos modos de transporte en una única plataforma accesible a través de aplicaciones móviles.
Impacto: Facilita el acceso a diferentes opciones de transporte, mejora la experiencia del usuario y promueve el uso de modos de transporte más sostenibles.

3. Materiales y Métodos

Para el desarrollo de este estudio se utilizó el procedimiento de RSL (Revisión Sistemática de la Literatura), recolectando información existente sobre big data y movilidad inteligente con el fin de conocer los avances y limitaciones en el tema.

La metodología utilizada en este artículo se fundamenta en el método PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) apoyado en su implementación mediante la herramienta en línea Parsifal. El proceso de revisión se dividió en tres fases principales: planificación, ejecución e informe, como se describe en la Figura 1.

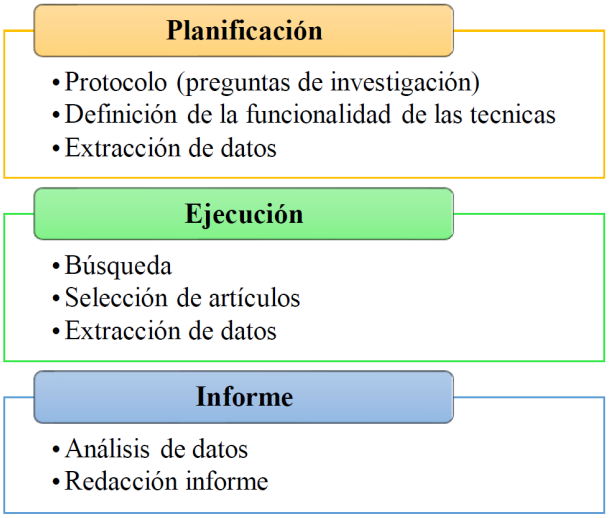


Figura 1. Esquema del proceso de revisión.

Fuente: Elaboración propia.

Planificación

En esta fase se elaboró el protocolo a seguir durante la revisión, identificando palabras clave como big data, movilidad inteligente, transporte y dispositivos móviles. Se definieron las siguientes preguntas de investigación (PI):

- **PI1**. ¿Qué dispositivos o herramientas existen para administrar el big data orientado a movilidad inteligente?
- **PI2**. ¿Qué tecnología de comunicación utilizan los dispositivos o herramientas para procesar el big data orientado a la movilidad inteligente?
- **PI3**. ¿Qué tipo de funcionalidad tienen estos dispositivos o herramientas?
- **PI4**. ¿Son estos dispositivos o herramientas portátiles o fijos?

La cadena de búsqueda utilizada en todas las colecciones de información y almacenes de datos consultados fue:

("Big data"; OR "movilidad inteligente"; OR "transporte"; OR "dispositivos móviles".) AND ("ciudades inteligentes") AND ("Internet de las cosas" OR "IoT").

Se establecieron las palabras clave a través del método Población, Intervención, Comparación, Salidas y Contexto (PICOC) que se muestran en la Tabla 1 a continuación.

Tabla 1. Palabras clave de PICOC.

Término	Palabras clave
Población	Big data, movilidad inteligente, transporte, dispositivos móviles
Intervención	Ciudades inteligentes
Comparación	Internet de las cosas

Fuente: Elaboración propia.

A continuación, se seleccionaron los estudios en base a los criterios de inclusión y exclusión presentados en la Figura 2.

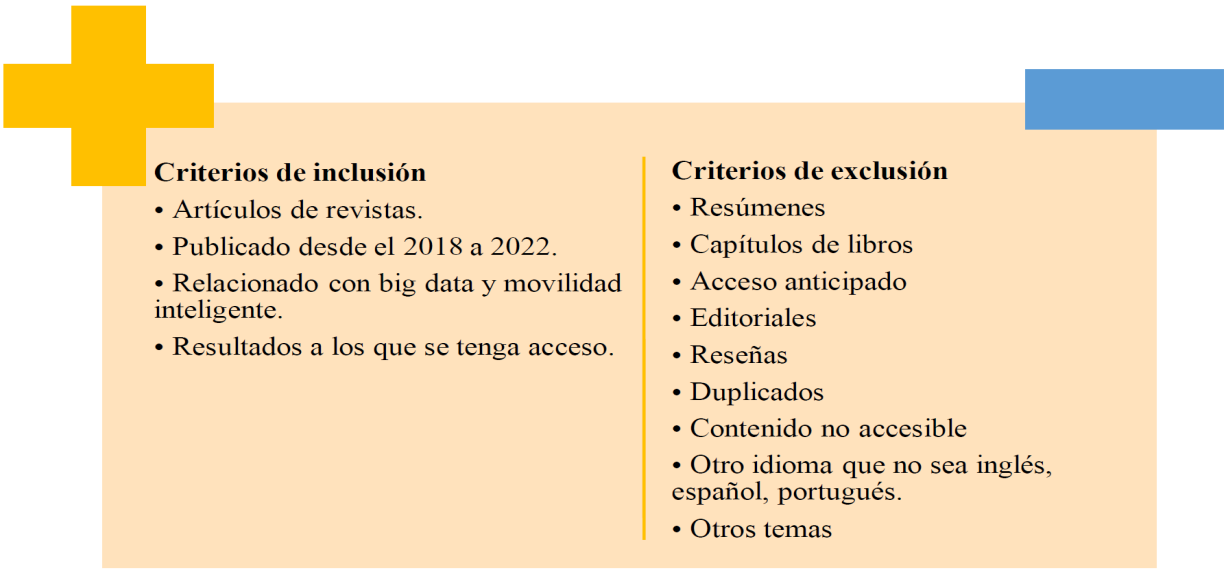


Figura 2. Criterios de selección y exclusión.
Fuente: Elaboración propia.

Ejecución
Se eligieron tres bases de datos científicas (IEEE Xplore, Scopus y Web of Science) para realizar la búsqueda en el proceso de revisión, ajustando algunos parámetros de búsqueda como el rango de años o el tipo de publicación.

Informe
A partir de los datos obtenidos en la última selección se llevaron a cabo diversos análisis, incluyendo la obtención de respuestas a las preguntas de investigación y una evaluación breve de las herramientas relacionadas con el tema. Finalmente, se procedió a la redacción del presente informe.

Propuesta de Tecnologías

Inteligencia Artificial y Aprendizaje Automático:
- **Descripción:** Integración de algoritmos avanzados de IA y aprendizaje automático para análisis predictivo y optimización en tiempo real de rutas de transporte.
- **Aplicación:** Implementar redes neuronales profundas para predecir patrones de tráfico y ajustar dinámicamente las rutas de transporte público y privado.
- **Impacto:** Mejora en la eficiencia del transporte, reducción de tiempos de espera y disminución de emisiones de carbono.

Internet de las Cosas (IoT) y Sensores Inteligentes:
- **Descripción:** Implementación de una red de sensores y dispositivos IoT para la recolección de datos en tiempo real sobre tráfico, condiciones meteorológicas y estado de infraestructuras.
- **Aplicación:** Desplegar sensores en semáforos, autobuses y estaciones para gestionar el tráfico de manera adaptativa.
- **Impacto:** Aumento de la seguridad vial, reducción de congestiones y mejora de la experiencia del usuario.

Vehículos Autónomos y Conectados:
- **Descripción:** Integración de vehículos autónomos y sistemas de comunicación V2V (vehículo a vehículo) y V2I (vehículo a infraestructura).
- **Aplicación:** Coordinación de movimientos de vehículos autónomos para mejorar la eficiencia y seguridad del tráfico.
- **Impacto:** Reducción de accidentes, mejora en la eficiencia del tráfico y facilización de servicios de transporte autónomo compartido.

Análisis de Datos en la Nube y Computación en el Borde:
- **Descripción:** Uso de plataformas de análisis de datos en la nube y computación en el borde para procesamiento en tiempo real.
- **Aplicación:** Arquitecturas híbridas que combinan nube y borde para procesar grandes volúmenes de datos y proporcionar recomendaciones instantáneas.
- **Impacto:** Aceleración del procesamiento de datos, reducción de latencia y mejora de la capacidad de respuesta del sistema de transporte.

4. Resultados y Discusión

4.1. Selección de artículos y extracción de información

Se encontraron un total de 37 artículos en las bases de datos que se muestran en la tabla 2. Se realizó una selección de artículos basada en criterios de inclusión y exclusión, seguida de una evaluación de la calidad. Después de aplicar estos criterios, se seleccionaron 23 artículos que cumplían con dichos criterios. A continuación, se aplicó el PI a estos artículos, quedando sólo 12 artículos seleccionados para el estudio que si cumplían con todas las PI. Los datos extraídos se resumen en la Tabla 3, en esta tabla, la columna que corresponde a cada PI también está descrita acorde a lo que se quiere.

Tabla 2: Documentos recuperados y seleccionados.

Base de datos	Cantidad	1ra selección	2nd selección
IEEE Xplore	8	3	3
Scopus	11	4	4
Web of Science	18	5	5

Fuente: Elaboración propia.

Tabla 3: Datos extraídos de los estudios seleccionados.

ID	Referencia PI-1	Nombre	País		Año	Tecnología PI-2	Funcionalidad PI-3	Movilidad PI-4
1	Raul M. Abril et al.	Apache Hadoop / Apache Spark	Bélgica		2020	Bluetooth / GPS	Es un motor de procesamiento de datos en clústeres que permite realizar análisis en tiempo real de grandes conjuntos de datos.	Portátil / Fijo
2	Stefan Poslad et al.	Apache Cassandra / Apache Flink	Reino Unido		2018	Bluetooth / GPS	Se utiliza en una base de datos distribuida para gestionar grandes conjuntos de datos en varios servidores.	Portátil / Fijo
3	Mattias Hjerpe et al.	SAS	Suecia		2018	Wifi	Realiza análisis de datos que permite el procesamiento y la visualización de grandes conjuntos de datos.	Portátil / Fijo
4	Josep M. Salanova Grau et al.	QlikView	España		2020	GPS	Se centra en el análisis exploración, Y visualización de datos para tomar decisiones empresariales.	Portátil / Fijo
5	Mocholí et al.	RFID	España		2019	WAVE / RFID	Identificación de autobús mediante una tarjeta RFID (Identificación por radio frecuencia).	Portátil
6	Li Zhuet al.	Tableau	China		2019	Wifi	Establece la visualización de datos que permite crear informes y paneles interactivos para su respectivo análisis.	Portátil / Fijo
7	Guo et al.	Sistema BSP	China		2021	GNSS	Activación virtual de la señal de tráfico, mediante el uso de la tecnología de valla electrónica GNSS.	Portátil
8	Godfrid et al.	MobilityDB	Argentina		2022	GPS	Analizar la movilidad del transporte público en tiempo real.	Portátil
9	Zanchett et al.	Share Bus	Brasil		2019	WiFi	Traza rutas de forma precisa, maximizando la comodidad y minimizando el tiempo de viaje.	Portátil
10	Nassar and Vieira	Smart Bus	Brasil		2018	NFC/ RFID	Comparte datos con otras unidades de transporte.	Portátil
11	Souza et al.	Dispositivos ubicuos	Brasil		2020	Sensores IoT	Captura y transmite un gran volumen de datos en tiempo real, para monitorizar el transporte y generar alguna alerta.	Portátil
12	Belenko, Krundyshev, and Kalinin	VANET	Rusia		2018	Wifi / GPS	Establecer una comunicación inalámbrica entre vehículos.	Portátil

Fuente: Elaboración propia.

4.2. Análisis de datos

En esta sección se lleva a cabo un análisis individual de los campos de datos relacionados con cada PI.

PI-1 Estudios encontrados

La figura 3 muestra la distribución de los trabajos seleccionados según el año y el tipo de publicación (revista o congreso). Se observa un aumento en los primeros años en comparación con los años más recientes incluidos en el estudio. La cantidad de trabajos es similar para ambos tipos de publicación: 8 trabajos en revistas y 4 en congresos.

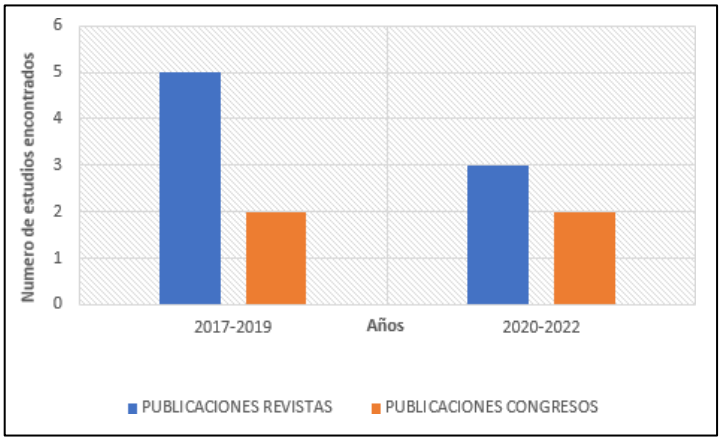


Figura 3: Clasificación de estudio por año.

Fuente: Elaboración propia.

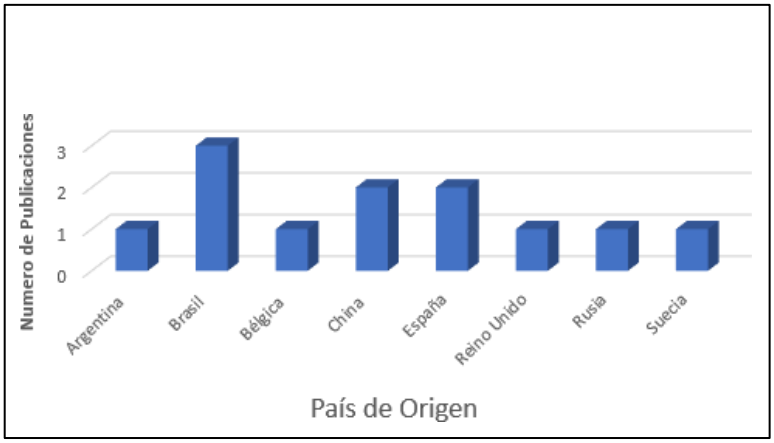


Figura 4: Distribución de los estudios encontrados por su país de origen.

Fuente: Elaboración propia.

La figura 4 representa la distribución de los trabajos de acuerdo a su país de origen, donde podemos observar que la mayoría de estudios provienen de países europeos, aunque el país con mayor número de estudios encontrados es Brasil, seguidos por China y España.

PI-2 Tecnologías de comunicación

Se identificaron diversas opciones tecnológicas que abarcan desde aquellas diseñadas para distancias cortas como el Bluetooth y aquellas de largo alcance como el GPS. La Figura 5 presenta la distribución de estas tecnologías en los estudios seleccionados.

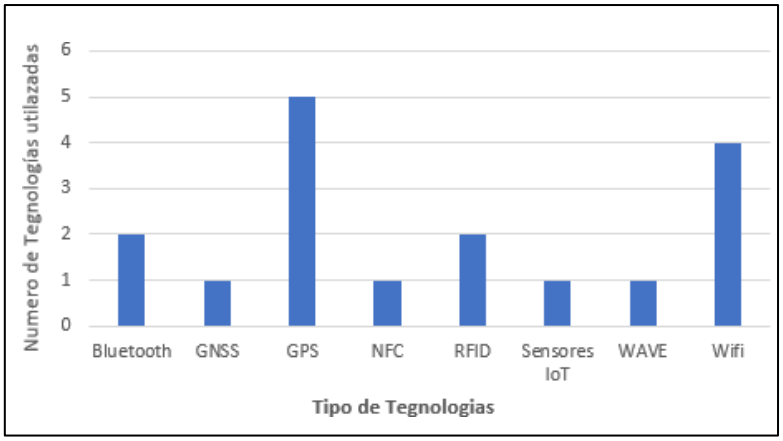


Figura 5: Tecnologías de comunicación utilizadas en los estudios encontrados.

Fuente: Elaboración propia.

En la figura 5 se muestran las tecnologías de comunicación más utilizadas por sistemas y dispositivos dentro del campo del big data orientado a la movilidad inteligente. Cabe recalcar que muchos de los sistemas o dispositivos pueden utilizar tecnologías conjuntamente, ya sean tecnologías de corto o de largo alcance. Las tecnologías de comunicación utilizadas en los estudios encontrados fueron las siguientes:

- Bluetooth: es una tecnología de comunicación inalámbrica que permite la transferencia de datos y la conexión entre dispositivos electrónicos a corta distancia.
- GPS (Sistema de Posicionamiento Global): es un sistema de navegación por satélite que proporciona información precisa de ubicación y tiempo en cualquier parte del mundo.
- GNSS (Sistema Global de Navegación por Satélite): es un término general que engloba varios sistemas de navegación por satélite, incluyendo el GPS, GLONASS y otros, que proporcionan información precisa de posición y tiempo.
- NFC (Comunicación de Campo Cercano): es una tecnología de comunicación inalámbrica de corto alcance que permite la transferencia de datos entre dispositivos compatibles, como teléfonos móviles y tarjetas de pago, simplemente acercándolos entre sí.
- RFID (Identificación por Radiofrecuencia): es una tecnología de identificación automática que utiliza etiquetas o tags con chips integrados para transmitir datos mediante radiofrecuencia, permitiendo la identificación y seguimiento de objetos.
- Sensores IoT (Internet de las Cosas): son dispositivos que recopilan datos del entorno físico, como temperatura, humedad, movimiento, entre otros, y los transmiten a través de internet para su posterior procesamiento y análisis. Estos sensores forman parte de la infraestructura de la Internet de las Cosas.
- WAVE (Wireless Access in Vehicular Environments): es una arquitectura de comunicación inalámbrica para vehículos que se basa en el estándar IEEE 802.11p.
- WiFi: es una tecnología de comunicación inalámbrica que permite la conexión a internet y la transferencia de datos a través de ondas de radio.

PI-3 Funcionalidad

Las herramientas y sistemas de big data en la movilidad inteligente permiten recopilar, almacenar y analizar grandes volúmenes de datos generados por dispositivos y sensores en el entorno de la movilidad. Estos datos pueden ser utilizados para obtener información valiosa, generar conocimientos y respaldar la toma de decisiones en el ámbito del transporte y la movilidad. La funcionalidad se clasifico en tres partes:

- **Comunicación:** En la investigación de Orozco O. (2017), en el transporte urbano, los sistemas inteligentes hacen uso de tecnologías de comunicación para establecer conexiones entre vehículos, sensores y sistemas de control. Esto permite la transmisión de datos en tiempo real relacionados con el tráfico, la velocidad y la ubicación de los vehículos, lo que a su vez facilita la optimización de la circulación y la reducción de los tiempos de espera. Un ejemplo de estos sistemas son las redes vehiculares ad hoc (VANETs, por sus siglas en inglés), que utilizan la tecnología de comunicación vehículo a vehículo (V2V) para intercambiar información entre los automóviles y otros elementos del entorno vial. Gracias a estos sistemas, se pueden tomar decisiones basadas en datos actualizados en tiempo real, lo que contribuye a mejorar la eficiencia y la seguridad del transporte urbano.

Figura 6: Comunicación en los **s**ervicios inteligentes
Fuente: Autor

- **Monitorización:** Existen diversos sistemas para el monitoreo urbano de la accesibilidad en ciudades inteligentes. Un ejemplo es el proyecto mencionado por Mora y Gilart-Iglesias en 2017. Este proyecto utiliza varios dispositivos físicos como RFID (del inglés Radio Frequency Identification), Bluetooth, GPS y sistemas en la nube, que se integran con dispositivos móviles. Se implementan varios sistemas autónomos que registran las rutas más utilizadas por los transeúntes. Además, se utilizan aplicaciones instaladas en dispositivos móviles para recopilar y analizar esta información. Entre las aplicaciones más conocidas de este tipo se encuentran Uber, Cabify, Easy Taxi y BlaBlaCar.

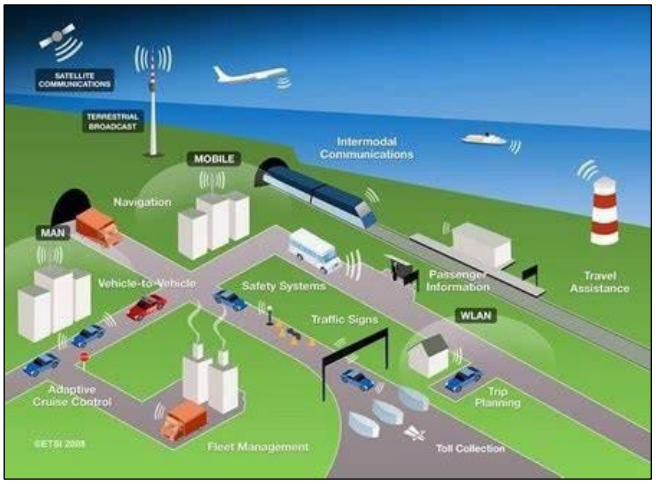


Figura 7: Sistema Inteligente de gestión de tráfico ITS**. Fuente:** Revista Mexican

- **Seguridad:** Como mencionan Tian, Zhang, y Hui 2018, los sistemas inteligentes en el transporte urbano desempeñan un papel fundamental en el mejoramiento de la seguridad tanto dentro como fuera de los vehículos. En este sentido, se implementan diversas tecnologías y soluciones para minimizar los riesgos y prevenir accidentes. Además, los vehículos pueden contar con sensores y sistemas de frenado automático que permiten evitar colisiones al detectar objetos u otros vehículos en su trayectoria. Estos sistemas emplean tecnologías como el radar, cámaras y LIDAR para monitorear el entorno y reaccionar de manera rápida y precisa.

Figura 8: Sensores ubicados en parqueaderos de Dahua Security**.**
Fuente: Autor

En la figura 9 se muestra la tabulación de los artículos filtrados al final para cada categoría, donde podemos ver la funcionalidad con mayor uso es la comunicación, seguido de la monitorización y por último la seguridad.

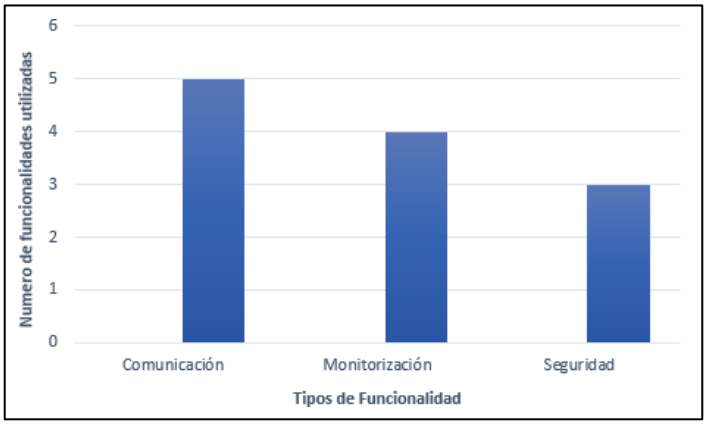


Figura 9: Función de los sistemas inteligentes.

Fuente: Elaboración propia.

PI-4 Movilidad

A continuación, se muestra la clasificación de las características de movilidad en dos categorías distintas:

Portátil: estos dispositivos son compactos, livianos y generalmente están diseñados para ser usados de manera autónoma, sin necesidad de estar conectados a una fuente de alimentación fija. Los dispositivos portátiles suelen tener funcionalidades específicas y están equipados con tecnologías como pantallas táctiles, conectividad inalámbrica, sensores y baterías recargables Estos dispositivos ofrecen comodidad y flexibilidad al permitir a los usuarios llevar consigo la tecnología y acceder a diversas funciones y servicios en movimiento. Santander M (2020).

Fijo: estos dispositivos están diseñados para ser instalados y utilizados en una ubicación específica sin la intención de ser fácilmente transportados o movidos. Estos dispositivos están conectados a una fuente de alimentación fija y pueden requerir una instalación permanente o semipermanente. Estos dispositivos están diseñados para funcionar de manera estable en un entorno fijo y pueden tener mayor capacidad de procesamiento, almacenamiento y conectividad en comparación con los dispositivos portátiles. Santander M (2020).

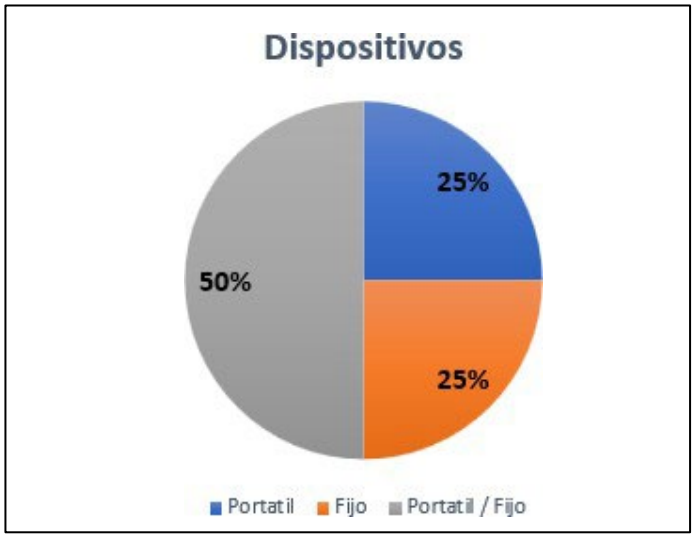


Figura 10: Movilidad de los dispositivos.

Fuente: Elaboración propia.

La figura 10 representa la distribución de sistemas inteligentes por categoría, donde se destaca que el modo preferido de uso para los usuarios es a través de dispositivos móviles. Esto permite a los ciudadanos aprovechar los beneficios de los sistemas de transporte inteligentes utilizando tanto dispositivos físicos como software, lo que les brinda movilidad y un servicio eficiente.

Como podemos observar, a través de la revisión de diversos estudios en este RSL, se presentan los resultados y la discusión sobre las herramientas de big data en la identificación de patrones de tráfico, la optimización de rutas, la mejora de la seguridad y la identificación de oportunidades de mejora en la movilidad inteligente. Además, se discuten los beneficios y limitaciones de la aplicación de big data en la movilidad inteligente, los desafíos de la implementación de tecnologías de big data y las oportunidades futuras en este campo. La presente revisión proporciona una comprensión de cómo las tecnologías de big data están ayudando a mejorar la eficiencia y la seguridad en la movilidad inteligente y puede servir como una guía para futuras investigaciones y desarrollos en este campo.

5. Conclusiones

El análisis realizado en este artículo demuestra que la integración de big data en la movilidad inteligente tiene un impacto profundo en la eficiencia, seguridad y sostenibilidad del transporte urbano. Las tecnologías de inteligencia artificial y aprendizaje automático permiten predicciones precisas de patrones de tráfico y la optimización de rutas, reduciendo los tiempos de viaje y mejorando la experiencia del usuario. La implementación de IoT y sensores inteligentes proporciona datos en tiempo real esenciales para una gestión dinámica del tráfico y una toma de decisiones más informada. Además, los vehículos autónomos y conectados, a través de su capacidad de comunicarse entre sí y con la infraestructura urbana, aumentan la seguridad vial y facilitan la movilidad autónoma, reduciendo potencialmente los accidentes y mejorando la eficiencia del tráfico.

Mirando hacia el futuro, se anticipa que las tecnologías de inteligencia artificial y aprendizaje automático seguirán evolucionando, permitiendo una mayor precisión en las predicciones de tráfico y una optimización en tiempo real de las rutas de transporte. La expansión de la implementación de sensores inteligentes en la infraestructura vial, vehículos y dispositivos personales mejorará aún más la precisión y el alcance de los datos recogidos. En el ámbito de los vehículos autónomos y conectados, se prevé un aumento en la adopción de estos vehículos y el desarrollo de sistemas de comunicación vehículo a vehículo (V2V) y vehículo a infraestructura (V2I) más avanzados, lo que permitirá una coordinación más eficiente entre vehículos e infraestructuras.

La combinación de análisis de datos en la nube y computación en el borde permitirá el procesamiento rápido y eficiente de grandes volúmenes de datos, reduciendo la latencia y mejorando la capacidad de respuesta del sistema de transporte. Se espera un mayor uso de arquitecturas híbridas que integren la computación en la nube y en el borde, optimizando el procesamiento de datos y mejorando la toma de decisiones en tiempo real.

En resumen, las tecnologías abordadas en este artículo tienen el potencial de transformar radicalmente la movilidad urbana. La inteligencia artificial, IoT, los vehículos autónomos y la computación en la nube ofrecen soluciones innovadoras para los desafíos del transporte moderno, mejorando la eficiencia, seguridad y sostenibilidad. A medida que estas tecnologías continúan evolucionando, se espera que desempeñen un papel cada vez más crucial en la configuración del futuro de la movilidad inteligente, proporcionando beneficios tangibles para las ciudades y sus habitantes.

Agradecimientos

A mi madre por estar siempre apoyándome y alentándome para seguir adelante, y también a los docentes de la Facultad de Ciencias Informáticas de la Universidad Técnica de Manabí en especial al Dr. Leonardo Chancay por hacer posible mi aprendizaje durante el desarrollo de la carrera y esta tesis.

Contribución de los autores

Autor	Contribución
Viteri Chica Miguel Andrés	Redacción: borrador original del artículo, conceptualización, metodologías, análisis y conclusiones
Chancay García Leonardo	Supervisión: Revisión y edición del artículo

Referencias

1. Belenko, V., V. Krundyshev, and M. Kalinin. (2018). “Synthetic Datasets Generation for Intrusion Detection in VANET.” ACM International Conference Proceeding Series. doi: 10.1145/3264437.3264479.

2. Bovalis, K., Peristeras, V., Abecasis, M., Abril-Jimenez, R.-M., Rodriguez, M. A., Gattegno, C., Karalopoulos, A., Sagias, I., Szekacs, S., & Wigard, S. (2014). Promoting Interoperability in Europe’s E- Government. Computer, 47(10), 25–33. https://doi.org/10.1109/mc.2014.295.

3. Burrieza Galán, J., Rodríguez Vázquez, R., Cantú Ros, O. G., Ayfantopoulou, G., Salanova Grau, J. M., Konstantinidou, M., Frederix, R., & Pápics, P. (2021). Future scenarios for mobility innovations and their impacts in cities and transport models. En Advances in Mobility-as-a-Service Systems (pp. 1129– 1138). Springer International Publishing.

4. Cheng, S., Liu, B., Ting, T. O., Qin, Q., Shi, Y., & Huang, K. (2016). Survey on data science with

population-based algorithms. Big Data Analytics, 1(1). https://doi.org/10.1186/s41044-016-0003-3.

5. Godfrid, J., P. Radnic, A. Vaisman, and E. Zimányi. (2022). Analyzing Public Transport in the City of Buenos Aires with MobilityDB. Vol. 14. Springer Berlin Heidelberg.

6. Guo, J., X. Wu, H. Wu, Y. Liu, and B. Zhang. (2021). “Bus Signal Priority System Based on GNSS Electronic Fence Technology.” ACM International Conference Proceeding Series 17–22. doi: 10.1145/3459012.3459015.

7. Hodjat, B. (2019). AI’s Biggest Challenge: Making The Black Box Transparent.

8. Mocholí, F., A. Martínez, A. Mocholí, and J. Arroyo. (2019). “Vehicle Identification by Means of Radio- Frequency-Identification Cards and Magnetic Loops.” IEEE Transactions on Intelligent Transportation Systems 21(12):5051–59. doi: 10.1109/TITS.2019.2948221.

9. Massoud, R., Berta, R., Poslad, S., De Gloria, A., & Bellotti, F. (2021). IoT sensing for reality-enhanced serious games, a fuel-efficient drive use case. Sensors (Basel, Switzerland), 21(10), 3559. https://doi.org/10.3390/s21103559.

10. Mora, H., & Gilart-Iglesias, V. (2017). A Comprehensive System for Monitoring Urban. Sensors — Open Access Journal, 7,10.

11. Nassar, V., and M. Vieira. (2018). “O Compartilhamento de Informações No Transporte Público Com as Tecnologias RFID e NFC: Uma Proposta de Aplicação.” Scielo 9(2):327–40. doi: 10.1590/2175- 3369.009.002.AO12.

12. Orozco, O., and G. Llano. (2017). “Aplicaciones Para Redes VANET Enfocada En La Sostenibilidad Ambiental, Una Revisión Sistemática.” Dialnet. 24(2):111. doi: 10.18359/rcin.396.

13. Ratti, C. (2020). From Big Data to Smart Data: The Case of Urban Planning. IEEE Intelligent Systems, 35(4), 92-96. DOI: 10.1109/MIS.2020.2999554.

14. Ren, Y., Leng, Y., Cheng, Y., & Wang, J. (2019b). Secure data storage based on blockchain and coding in edge computing. Mathematical Biosciences and Engineering, 16(4), 1874–1892. https://doi.org/10.3934/mbe.2019091.

15. Santander, M. R., & Colazo, Z. S. L. (2020). Las aplicaciones móviles: Su uso educativo para compartir la información en la Bioinformática. Serie Científica de la Universidad de las Ciencias Informáticas, 13(9), 55-67.

16. Seguí-Amortegui, Clemente-Almendros, Medina, & Grueso. (2019). Sustainability and competitiveness in the tourism industry and tourist destinations: A bibliometric study. Sustainability, 11(22), 6351. https://doi.org/10.3390/su11226351.

17. Sköld Gustafsson, V., Hjerpe, M., & Strandberg, G. (2023). Construction of a national natural hazard interaction framework: The case of Sweden. IScience, 26(4), 106501. https://doi.org/10.1016/j.isci.2023.106501.

18. Souza, L., J. De Almeida, T. Melo, M. Soares, F. Rocha, and R. Nascimento. (2020). “An Investigation into Technology Solutions for Urban Public Monitoring through Ubiquitous Devices in the Context of Smart Cities.” ACM International Conference Proceeding Series Part F1667. doi: 10.1145/3401895.3402082.

19. Tian, L., Y. Zhang, and J. Hui. (2018). “Service Optimization for Bus Corridors with Group Bus Strategy.” ACM International Conference Proceeding Series 80–85. doi: 10.1145/3285957.3285972.

20. Zanchett, D., J. Junior, A. Monteiro, D. Haddad, and L. Assis. (2019). “Collaborative Information System to Find Efficient Routes Using Public Transport.” ACM International Conference Proceeding Series 473–76. doi: 10.1145/3323503.3361717.

21. Zhu, L., F. R. Yu, Y. Wang, B. Ning and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 383-398, Jan. 2019, doi: 10.1109/TITS.2018.2815678.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-Compartirlgual 4.0 Internacional.

ELECTRONICA

Recibido 23 Dic. 2023 *ReCIBE, Año 13 No. 1, May. 2024*

Aceptado 11 Jun. 2024

Identificación y Clasificación de Imágenes Farmacéuticas mediante Técnicas de Visión e Inteligencia Artificial: Un Enfoque en Máquinas de Vectores de Soporte (SVM)

Support Vector Machines (SVM) based identification and classification of pharmaceutical images

Jesus Alfonso Medrano Hermosillo[1*]
jesus.mh@chihuahua.tecnm.mx
Dr. Rodriguez Mata[1]
Dr. Djilali[1]
Dr. Edgar Macias[2]
Dr. Oscar Suárez
Raul Vazquez

[1]Tecnológico Nacional de México
[2] INTEL

RESUMEN

Este estudio se enfoca en la identificación y clasificación de imágenes con fármacos para facilitar su selección y administración adecuada a pacientes postrados. La implementación del algoritmo se lleva a cabo mediante la combinación de metodologías convencionales de visión e inteligencia artificial. Por el lado de visión artificial, se emplea preprocesamiento de imágenes para el proceso de segmentación, donde la técnica utilizada es la llamada umbralización. Una vez completado dicho proceso, se procede a la clasificación de los fármacos mediante aprendizaje de máquina, conocido en inglés como *machine learning*. En este estudio específico, se eligen las máquinas de vectores de soporte (SVM por sus siglas en inglés), que han demostrado una notable eficacia en la clasificación de datos linealmente separables. Luego de la clasificación de las imágenes, se genera una representación visual (mediante una imagen) que exhibe los medicamentos ordenados y situados en un plano cartesiano, listos para ser dispensados a cualquier paciente.

Palabras clave: Aprendizaje de Máquina, Inteligencia Artificial, Visión artificial.

ABSTRACT

The present study focuses on the identification and classification of images that contain drugs, with the purpose of facilitating their selection and/or appropriate administration. The implementation of the algorithm is carried out by combining conventional techniques and machine learning methods. Preprocessing is used to carry out image segmentation, using the thresholding technique. Once the segmentation is completed, the pills are classified using machine learning techniques. In the specific context of this work, the use of support vector machines (SVM) is chosen, which demonstrate notable effectiveness in the classification of linearly separable data.

Keywords: Machine Learning, Artificial Intelligence, Artificial vision.

1. Introducción

Según la OMS (Organización Mundial de la Salud), al menos 500,000 personas sufren lesiones medulares cada año, lo cual se traduce a cambios en el estilo y calidad de vida de los afectados (Organization, W. H., & Society, I. S. C., 2013). Las personas que padecen diferentes cuadros de paraplejia, esclerosis múltiple, esclerosis lateral amiotrofia, entre otras, permanecen postrados por la degeneración de la capacidad motriz de su cuerpo, lo cual, implica la utilización permanente de equipos de asistencia médica e incrementa la dependencia hacia terceros para llevar a cabo sus actividades cotidianas. Se han propuesto avances tecnológicos para ayudar a estas personas con distintos tipos de parálisis, como paraplejia, esclerosis múltiple y esclerosis lateral amiotrófica. Estos abarcan dispositivos de autoayuda (Orejuela-Zapata 2019), rehabilitación robótica (Mekki 2018), exoesqueletos para extremidades inferiores (Guntara 2021, Kapula 2021), desde sillas de ruedas (García & Páez, 2021), exoesqueletos (Medrano et. al, 2014), interfaces hombre-máquina (Macías-Macías et. al, 2021), entre otros trabajos (McGibbon et. al, 2021; Pascuas-Rengifo et. al, 2015) y exoesqueletos para extremidades superiores (Sanngoen 2020). Estas tecnologías tienen el objetivo de mejorar la movilidad, minimizar la fatiga muscular y mejorar la calidad de vida en general. No obstante, esta rama de la ciencia sigue siendo ciencia abierta es necesario seguir investigando para satisfacer las necesidades específicas de las personas con lesiones de la neurona motora superior.

Sin embargo, uno de los casos más severos de deficiencia motriz es la cuadriplejia, la cual impide el movimiento de los miembros de la persona (Mogahed & Ibrahim, 2023). Esta situación impide que la persona pueda realizar la toma de cualquier objeto, como puede ser el caso de sus medicamentos (English, 2023). Es por tal motivo, que el propósito del presente trabajo es el desarrollo de un algoritmo para la detección, clasificación y localización de medicamentos en un ambiente controlado. Para como trabajo a futuro, cualquier sistema mecatrónico robótico pueda obtener el medicamento adecuado y suministrarlo al paciente con discapacidad de movimiento. El diagrama del algoritmo completo se puede ver en la Figura 1.

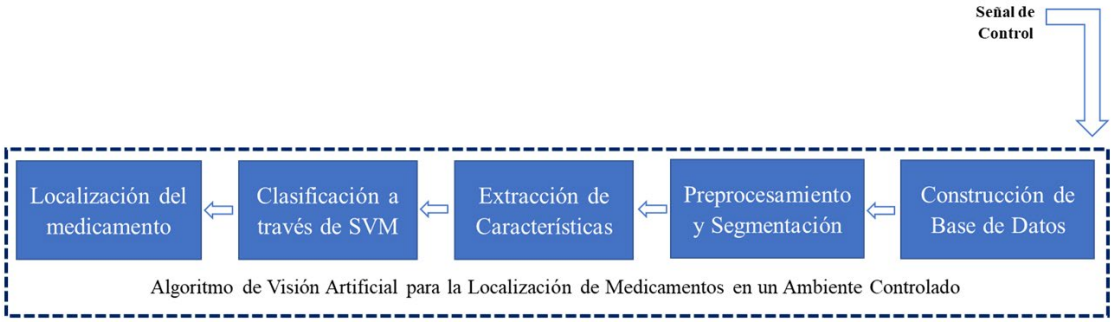


Figura 1. Diagrama a bloques del algoritmo implementado.

El presente trabajo está organizado de la siguiente manera: en la Sección 2 se ilustran los trabajos relacionados, en la Sección 3 se muestran los fundamentos matemáticos del algoritmo propuesto. Por otro lado, los resultados del trabajo se observan en la Sección 4. Finalmente, enla Sección 5 se comentan las conclusiones y el trabajo a futuro.

2. Antecedentes

Con base al diagrama a bloques observado anteriormente, son necesarias diversas técnicaspara el desarrollo del algoritmo propuesto. Los trabajos relacionados con técnicas de visión y clasificación se muestran a continuación. (Andhare & Rawat, 2016) desarrollaron un sistema de visión capaz de identificar las coordenadas de un objeto dentro del espacio de trabajo de un brazo robótico de seis grados de libertad, a partir de las coordenadas de píxeles que obtiene elsensor de visión de dicho objeto. En (Huiling et. al, 2019) se desarrolla un sistema de reconocimiento de medicamentos para máquinas expendedoras. El sistema emplea SVM (*Support Vector Machines*) y componentes conectados para identificar la región de texto del medicamento.

Luego, realiza la segmentación del texto mediante el método *fragment link*. Finalmente, utiliza el *software* de reconocimiento óptico de caracteres (OCR, por sus siglas en inglés) para reconocer con precisión el nombre del medicamento.

En (Hnoohom et. al, 2018) los autores presentan el diseño de una aplicación de teléfono inteligente capaz de identificar de manera automática distintos fármacos utilizando la arquitectura de aprendizaje profundo Inception-V3, obteniendo un 92.75% de precisión. Las pruebas de uso real demostraron que el modelo de redneuronal propuesto podía identificar correctamente el medicamento indicado en los datos de entrenamiento. En (Shinde et. al, 2020) se presenta un trabajo de investigación enfocado en la detección de medicamentos falsificados a través de sus firmas hiperespectrales. El sistema implementado utiliza las firmas hiperespectrales de los medicamentos originales y falsificados para entrenar un modelo de aprendizaje automático basado en una red neuronal para reconocerla autenticidad del medicamento. El modelo propuesto logra una media superior al 90% de precisión en la clasificación de los medicamentos. En (Ou, & Lin, 2018), los autores proponen eluso de redes neuronales convolucionales para la detección de píldoras, donde el trabajo obtuvoun rendimiento de más del 80% en la clasificación de los medicamentos. Cabe resaltar, que se utilizó una base de datos de 1,680 imágenes para el entrenamiento de la red neuronal. Es posibleobservar que existe una vasta literatura sobre la detección de objetos, sin embargo, algunos deellos suelen ser complejos para su aplicación en la vida real, es por lo que el algoritmo propuesto, para el presente trabajo, es el uso de máquinas de vector se soporte, por su simplezay eficiencia (Gidudu et. al, 2007).

3. Fundamentos matemáticos

3.1 Análisis por medio de visión artificial

Precedentemente de iniciar el procesamiento de imágenes y la implementación de clasificadores para el entrenamiento del SVM, es esencial contar con muestras representativas. Con este propósito, se capturaron imágenes de diversas medicinas en un entorno controlado. Estas imágenes exhiben pastillas, cápsulas y tabletas dispuestas en posiciones y orientaciones aleatorias. La Figura 2 ilustra un ejemplo visual de estas imágenes obtenidas.

Figura 2. Imagen de la base de datos propuesta.

El factor de calibración (FC) juega un papel fundamental en la localización de objetos en una imagen, ya que facilita la determinación de la relación entre píxeles y distancia. En el entorno controlado propuesto, el campo visual abarca un espacio de 42.19 cm x 56.25 cm, mientras que las superficies de las imágenes capturadas son 1200 píxeles x 1600 píxeles. Así, el FC se establece de la siguiente manera:

$$FC = \frac{Dimensiones\ de\ la\ imagen\ en\ centimetros}{Dimensiones\ de\ la\ imagen\ en\ pixeles} \quad (1)$$

En relación con imágenes de la base de datos, el resultado del factor de calibración se establece en 0.0352 $\frac{cm}{píxel}$. Además, al considerar la distancia entre el centro óptico de la cámara y la superficie del entorno controlado, se determina el campo visual total de la cámara en las direcciones vertical ($\theta_{vertical}$) y horizontal ($\theta_{horizontal}$):

$$\theta_{vertical} = 2tan^{-1}\left(\frac{21.04\ cm}{45\ cm}\right) = 50.23° \quad (2)$$

$$\theta_{horizontal} = 2tan^{-1}\left(\frac{28.125\ cm}{45\ cm}\right) = 64.01° \quad (3)$$

Con lo anterior, es posible detectar la posición de los diferentes objetos. Para más información respecto al procedimiento anterior, consultar (González & Woods, 2006).

Preprocesamiento y segmentación

Diversos métodos se emplean en la segmentación de imágenes (González, 2002), siendo la media de intensidad de píxeles una herramienta eficaz comúnmente utilizada con este propósito. En otras palabras, al calcular el valor medio de la intensidad de los píxeles, se aplica la umbralización local para lograr la segmentación de la imagen respectiva (Chacón, 2007). En el marco del algoritmo propuesto, esta técnica se implementa para obtener las imágenes binarias deseadas, y la ecuación que rige dicha segmentación se presenta en (4).

$$g(x,y) = \begin{cases} 1 & si\ f(x,y) > T \\ 0 & si\ f(x,y) \leq T \end{cases} \quad (4)$$

Donde $g(x,y)$ es valor del píxel en la nueva imagen segmentada, $f(x,y)$ es el valor del píxel de la imagen a segmentar y T representa el umbral global. Considerando las imágenes en un estado binario, se continua con la exclusión de todos los componentes conectados que posean $2000 \leq T$ ya que esta cantidad representa el mínimo de píxeles conectados que contienen los medicamentos en las imágenes. Es crucial destacar que este proceso se aplica exclusivamente al canal R del espacio de color RGB (Red, Green, Blue; por sus siglas en inglés) de las imágenes originales. Esto se realiza con el propósito de facilitar el procesamiento de las imágenes en las siguientes fases del algoritmo. La Figura 3 exhibe el resultado de una imagen segmentada correctamente, donde se visualizan fácilmente los objetos a clasificar.

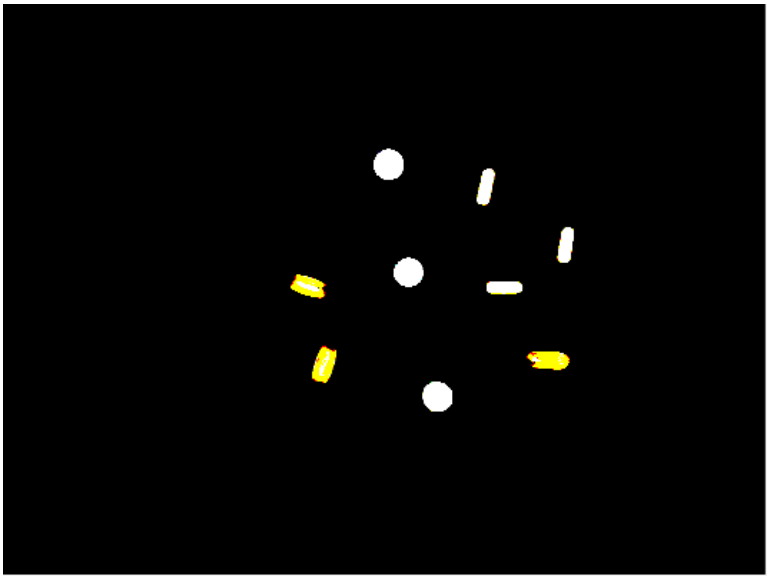


Figura 3. Segmentación de los objetos de interés

Extracción de características

En la fase de extracción de características, se procede a identificar las partes más significativas de un objeto en la imagen, como señalan en (Shah et al., 2016). En el algoritmo propuesto, se destacan el área, redondez y solidez como características clave para la clasificación de los fármacos. El cálculo de estas propiedades se lleva a cabo mediante las ecuaciones (5), (6) y (7), respectivamente. Una vez obtenidas estas características, se forma un vector compuesto por tres elementos, proporcionando así la base para el desarrollo de un clasificador adecuado.

$$Área = Número\ de\ pixeles\ de\ la\ región \tag{5}$$

$$Redondez = \frac{4 * Área * \pi}{Perímetro^2} \tag{6}$$

$$Solidez = \frac{Área}{Área\ Convexa} \tag{7}$$

Con el propósito de evaluar el potencial de las características extraídas para la clasificación de medicamentos en pastillas, cápsulas o tabletas, se procede al cálculo de la matriz de correlación de estas características. La Tabla 1 presenta la matriz de correlación correspondiente, destacando que el área y la solidez exhiben el menor grado de correlación entre ellas (0.3970), mientras que la redondez y la solidez muestran un mayor grado de correlación (0.7917). Con base en este análisis, se avanza hacia la implementación de un clasificador binario basado en SVM utilizando estas propiedades descriptivas.

Tabla 1. Correlación de los medicamentos a clasificar

Características	Área	Redondez	Solidez
Área	1	0.7613	0.3970
Redondez	0.7613	1	0.7917
Solidez	0.3970	0.7917	1

3.2 Clasificador utilizando máquinas de vector de soporte

Una técnica ampliamente reconocida para la clasificación de objetos es el algoritmo SVM. Estas máquinas emplean datos, también denominados vectores de soporte, para establecer un umbral distintivo entre diferentes clases (Suthaharan et al., 2016).

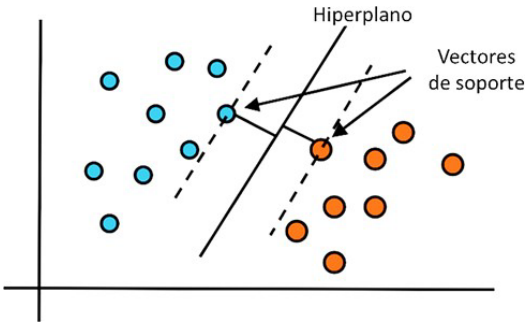


Figura 4. Descripción del hiperplano creado por el algoritmo SVM

En la Figura 4, se observan dos clases claramente divididas por un hiperplano, el cual es definido por el algoritmo de máquinas de vectores de soporte. En el escenario específico de una aplicación como clasificador binario lineal, el hiperplano se expresa mediante:

$$g(x) = w^T X + b \;\leftrightarrow\; g(x) = 0 \tag{8}$$

En la ecuación anterior, X representa los patrones de entrada, w los pesos del sistema y b es el bias. La determinación de los pesos se logra al minimizar la función de costo definida en la ecuación (9) con respecto a los pesos del sistema.

$$J(w, b, \propto) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left[d_i (w^T x_i + b) - 1 \right] \tag{9}$$

Donde λ_i son los multiplicadores de Lagrange. La clasificación de los distintos fármacos en el presente trabajo se realiza utilizando SVM, el cual es un clasificador supervisadobinario descrito por una función discriminante lineal (LDA, por sus siglas en inglés) descrita anteriormente. Considerando que las clases son linealmente separables, es necesario definir lospesos ω = (ω1, ω2 , … , ω_l) a través de la ecuación dada por (Pisner et al., 2020):

$$\omega = \sum_{i=1}^{q} \lambda_i y_i x_i \tag{10}$$

Aquí, y_i son las etiquetas, x_i el vector de características y q la cantidad de vectores de soporte. Para tomar la decisión de a que clase (W_1 o W_2) pertenece el vector $X = \{x_1, x_2, \dots, x_n\}$, el clasificador toma las decisiones de la siguiente manera:

$$\omega^T X > 0 \rightarrow \quad X \in W_1 \tag{11}$$

$$\omega^T X \leq 0 \rightarrow \quad X \in W_2 \tag{12}$$

No obstante, en la aplicación de fármacos propuesta, el clasificador debe tener la capacidad de identificar los tres tipos de medicamentos. Por consiguiente, se requiere diseñar dos SVM de clasificación binaria: uno para reconocer pastillas ($g(x)_{pastillas}$) y otro para las cápsulas ($g(x)_{capsulas}$). De esta manera, se propone un flujo de trabajo que se describe en la Figura 5.

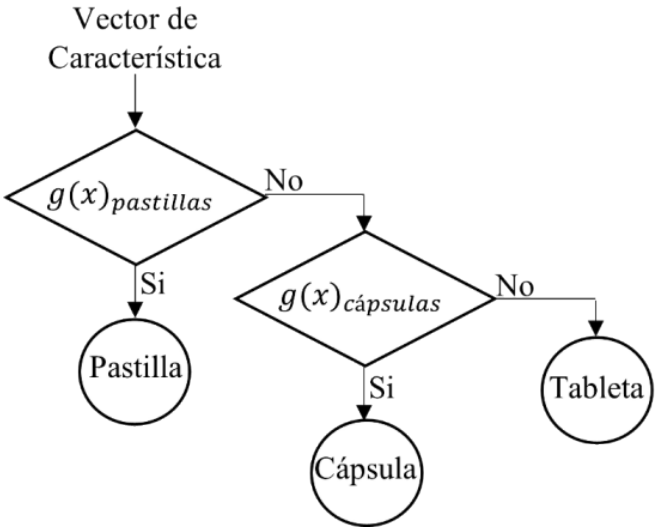


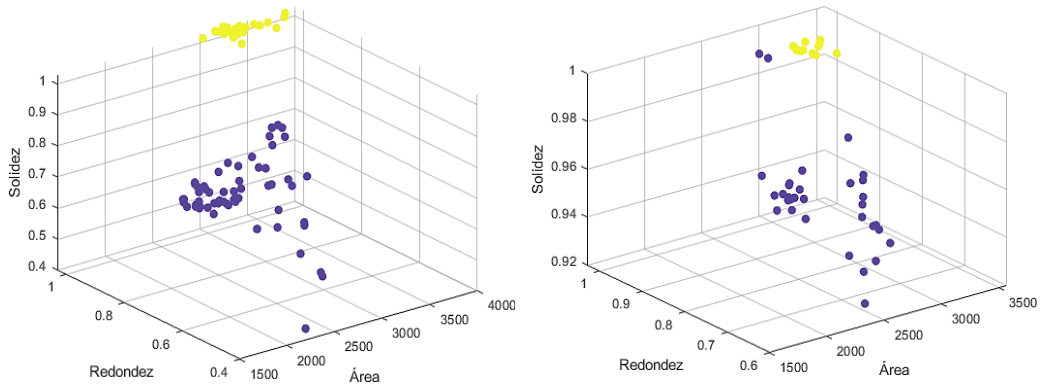
Figura 5. Descripción del algoritmo de clasificación implementado

El algoritmo recibe como entrada el vector de características de un medicamento que se va a clasificar. Primero, pasa por $g(x)_{pastillas}$ para evaluar si es una pastilla o cualquier otro tipo de medicamento. Luego, se somete a $g(x)_{cápsulas}$ para determinar si es una cápsula o si se clasifica como una tableta. El análisis del conjunto de entrenamiento y prueba para ambos clasificadores arroja las siguientes funciones discriminantes lineales, que definen los respectivos hiperplanos:

$$g(x)_{pastillas} = 0.0095x_1 + 1.2332x_2 + 0.0878x_3 - 30.8028 \tag{13}$$

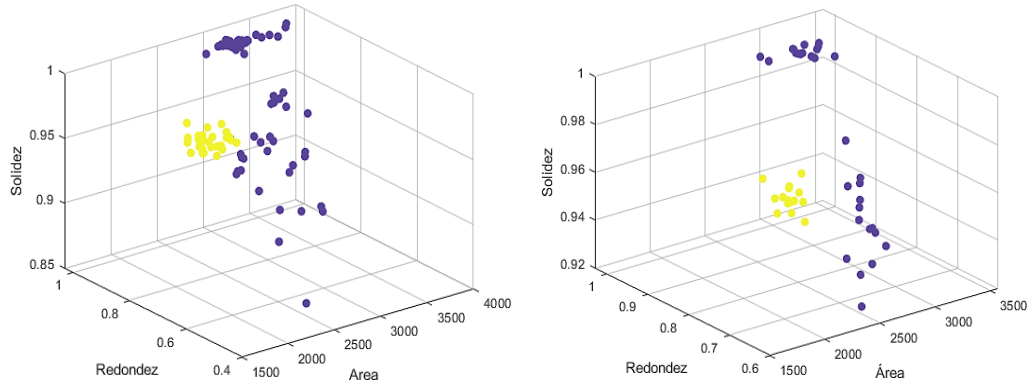
$$g(x)_{capsulas} = -0.0158x_1 + 0.4259x_2 + 0.1659x_3 + 36.4810 \tag{14}$$

En la Figura 6, se presentan los resultados del entrenamiento y la prueba del clasificador de pastillas. En esta representación, los puntos amarillos indican los elementos clasificados como pastillas, mientras que los puntos azules representan los elementos clasificados como otro tipo de medicamento. De manera análoga, la Figura 7 muestra los resultados correspondientes al clasificador de cápsulas, donde los puntos amarillos indican los elementos clasificados como cápsulas y los puntos azules representan los elementos clasificados como otro tipo de medicamento. Estas visualizaciones reflejan la efectividad de los clasificadores en la diferenciación de pastillas y cápsulas de otros tipos de medicamentos.



a) Resultados del entrenamiento b) Resultado de la prueba

Figura 6. Resultados del clasificador de pastillas



a) Resultados del entrenamiento b) Resultado de la prueba

Figura 7. Resultados del clasificador de cápsulas

A partir de los resultados obtenidos en las pruebas de los clasificadores de pastillas y cápsulas, se generan las correspondientes matrices de confusión, presentadas en la Tabla 2 y Tabla 3, respectivamente. Estas matrices ofrecen una visión detallada de la eficacia de los clasificadores al evaluar la precisión de las predicciones para cada clase.

Tabla 2. Matriz de confusión del clasificador de pastillas

		Valor Predicho	
		No Pastilla	**Pastilla**
ValorReal	**No Pastilla**	TN=15	FP=0
	Pastilla	FN=2	TP=13

Tabla 3. Matriz de confusión del clasificador de cápsulas.

		Valor Predicho	
		No Cápsula	**Cápsula**
ValorReal	**No Cápsula**	TN=15	FP=0
	Cápsula	FN=0	TP=15

Considerando las matrices de confusión, se procede al cálculo de las métricas de desempeño del clasificador. La precisión evalúa la calidad del modelo en las tareas de clasificación, mientras que la exhaustividad cuantifica la cantidad que el modelo es capaz de identificar (Campbell et al., 2022). La precisión se calcula mediante la ecuación (15), mientras que la exhaustividad se determina a través de la ecuación (16).

$$precisión = \frac{TP}{TP + FP} \quad (15)$$

$$exhaustividad = \frac{TP}{TP + FN} \quad (16)$$

Al examinar los resultados previos, se destaca que el clasificador de pastillas logra identificar el 86% de las pastillas, mientras que el clasificador de cápsulas alcanza un reconocimiento del 100%. Estos resultados se muestran coherentes y competitivos en relación con los trabajos previamente descritos en el capítulo de antecedentes.

3.3 Localizador de medicamentos

En el presente trabajo, la determinación de la posición de los fármacos se lleva a cabo mediante el cálculo del centroide de los objetos en la imagen. Donde el centroide representa el centro de masa de los objetos, detallando sus coordenadas en píxeles en la propia imagen (González, 2002). Dicho centroide se obtiene mediante las siguientes ecuaciones:

$$c_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (17)$$

$$c_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (18)$$

Donde A es el área de la figura, y x_i, y_i corresponden a las coordenadas de sus vértices. Una vez encontrado el centroide, se utiliza el factor de calibración calculado en (1) para definir las coordenadas en centímetros de la posición de los medicamentos.

4. Resultados

A continuación, a manera de comprobar el funcionamiento del algoritmo completo se realizaron ciertas tomas de diferentes medicamentos en una superficie. Donde el algoritmo además de poder detectar y posicionar correctamente los diferentes fármacos, fue capaz de clasificarlos de manera correcta. En la Figura 9 se ilustran diversas pruebas para la identificación de las diferentes tablillas, donde fueron elegidas de la siguiente manera:

- Si el fármaco tiene forma circular y su área tiene una cierta dimensión, el medicamento pertenece a la clase de “Pastilla 1”.
- Si el fármaco tiene forma de ovalo con una cierta dimensión, el medicamento puede pertenecer a la clase de la “Tableta 2” o la “Pastilla 3”. Donde su área, redondez y solidez van a determinar a cuál clase pertenece.

Cabe resaltar que lo mencionado anteriormente es una descripción muy general, ya que las características que se observaron en la Sección 3 fueron sometidas al algoritmo SVM para su clasificación de acuerdo con los hiperplanos nombrados en (13) y (14). Donde en los resultados observados en la Figura 9, se aprecia que el algoritmo hizo una correcta clasificación e identificación de los medicamentos en la imagen. Es así, que el algoritmo puede ser utilizado para futuras aplicaciones, cabe resaltas que la programación es sumamente sencilla, ya que sólo es necesario implementar los hiperplanos mencionados anteriormente, no es necesario volver a entrenar el algoritmo nuevamente, a menos de que se deseen aplicar más fármacos u otra aplicación.

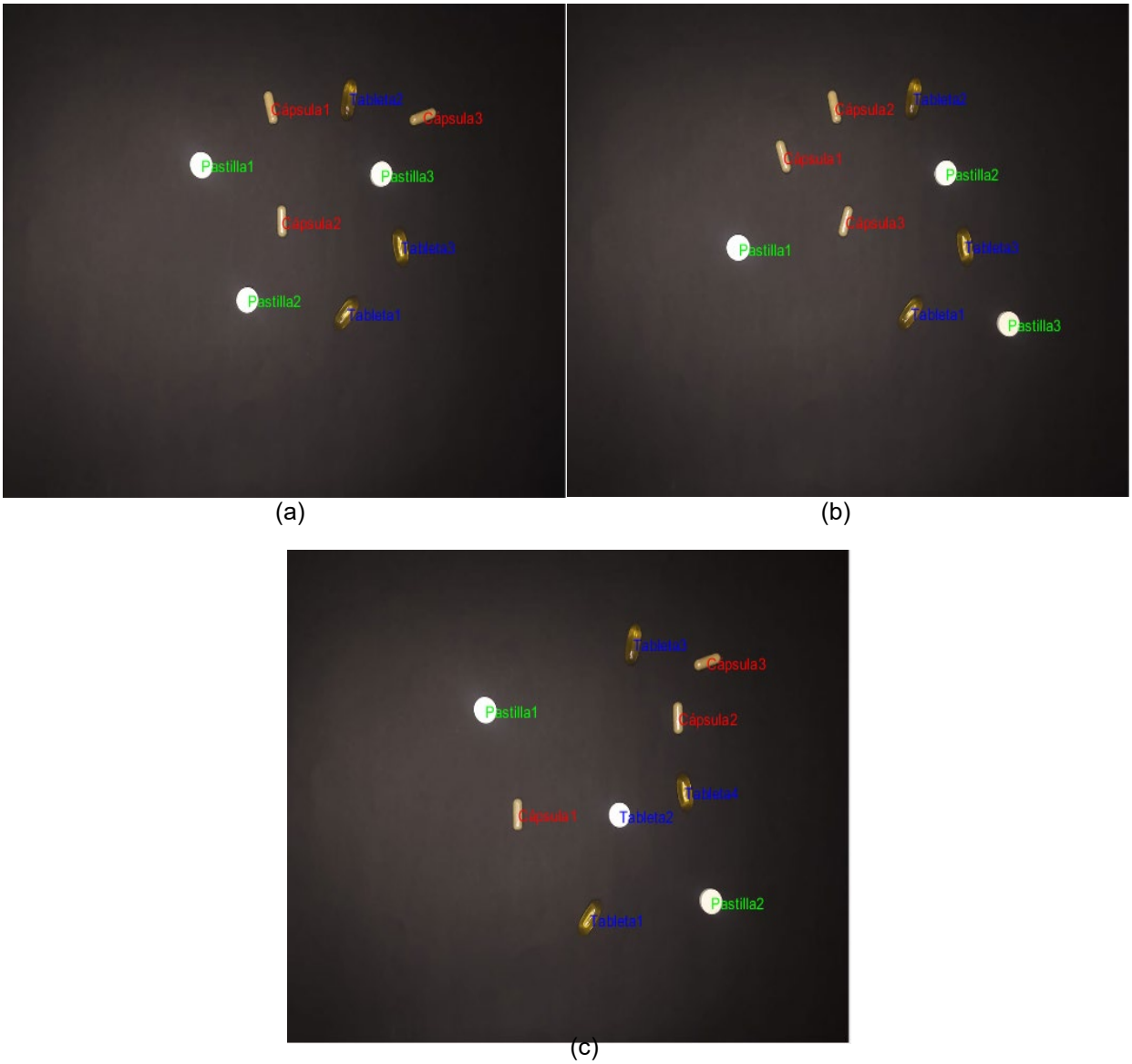


Figura 8. Pruebas de funcionamiento del algoritmo de visión para la detección de fármaco

Conclusiones y trabajo a futuro

Este estudio resalta la eficacia del algoritmo SVM en la detección de fármacos en un entorno controlado, proporcionando una clasificación y localización precisa de los diferentes medicamentos. La utilidad de SVM no solo radica en su eficacia, sino también en su facilidad de programación, lo que permite su implementación en cualquier sistema embebido y facilita su acceso a regiones con limitaciones tecnológicas.

En cuanto a futuras investigaciones, se contempla el diseño de un sistema mecatrónico dedicado a la selección y suministro automatizado de los medicamentos identificados en el presente trabajo. Esta perspectiva no solo constituye una mejora práctica en la administración de medicamentos, sino que también promete elevar la calidad de vida de aquellos pacientes con restricciones en el movimiento corporal y/o de miembros superiores. Este trabajo sienta las bases para avanzar hacia soluciones más integrales en el ámbito de la atención médica, orientadas a mejorar la accesibilidad y el bienestar de los pacientes con discapacidades físicas.

5. REFERENCIAS BIBLIOGRÁFICAS

Andhare, P., & Rawat, S. (2016). Pick and place industrial robot controller with computer vision. 2016 International Conference on Computing Communication Control and Automation (ICCUBEA), 1-4.

Campbell, C., & Ying, Y. (2022). Learning with support vector machines. Springer Nature.

García, W., & Páez, E. (2021). Diseño de un sistema de conducción autónoma adaptado a una silla de ruedas eléctrica.

Gidudu, A., Hulley, G., & Tshilidzi, M. (2007). Classification of images using support vector machines.

Gonzalez, R. C., & Woods, R. E. (2002). Digital image processing (2nd ed.). Prentice Hall.

Xia, H., Wang, C., Yan, L., Dong, X., & Wang, Y. (2019). Machine Learning Based Medicine Distribution System. 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 912-915.

Macías-Macías, J. M., Quintana, J. A. R., Aguirre, J. S. A. M., Murguía, M. I. C., & Sáenz, A. D. C. (2021). Procesamiento embebido de p300 basado en red neuronal convolucional para interfaz cerebro-computadora ubicua. RECIBE.

Chacón, M. I. (2007). Procesamiento digital de imágenes. Editorial Trillas.

McGibbon, C. A., Sexton, A., Gryfe, P., Dutta, T., Jayaraman, A., Deems-Dluhy, S., Novak, A. C., Fabara, E., Adans-Dester, C., & Bonato, P. (2021). Effect of using of a lower-extremity exoskeleton on disability of people with multiple sclerosis. Disability and Rehabilitation: Assistive Technology, 18(5), 475–482.

Medrano, J. A., Zendejas-Hernandez, I., & Sandoval-Rodríguez, R. (2014). Diseño de un Exoesqueleto de Miembros Inferiores para la Movilidad de Personas con Paraplejía.

Mogahed, H. S., & Ibrahim, M. M. (2023). Development of a motion controller for the electric wheelchair of quadriplegic patients using head movements recognition. IEEE Embedded Systems Letters, 1.

Hnoohom, N., Yuenyong, S., & Chotivatunyu, P. (2018). MEDiDEN: Automatic Medicine Identification Using a Deep Convolutional Neural Network. 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), 1-5.

World Health Organization, & International Spinal Cord Society. (2013). International perspectives on Spinal cord Injury. World Health Organization.

Ou, Y., Tsai, A., Wang, J., & Lin, J. (2018). Automatic Drug Pills Detection based on Convolution Neural Network. 2018 International Conference on Orange Technologies (ICOT), 1-4.

Pascuas-Rengifo, Y., Vargas-Jara, E. O., & Sáenz-Núñez, M. (2015). Tecnologías de la información y las comunicaciones para personas con necesidades educativas especiales. Entramado, 11(2), 240–248.

Pisner, D. A., & Schnyer, D. M. (2020). Support vector machine. In Machine learning (pp. 101-121). Academic Press.

González, R. C., & Woods, R. E. (2006). Digital Image Processing (3rd ed.). Prentice-Hall, Inc., USA.

Shinde, S. R., Bhavsar, K., Kimbahune, S., Khandelwal, S., Ghose, A., & Pal, A. (2020). Detection of Counterfeit Medicines Using Hyperspectral Sensing. 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).

Shah, F. P., & Patel, V. (2016, March). A review on feature selection and feature extraction for text classification. In 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 2264-2268). IEEE.

Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification: Thinking with examples for effective learning (pp. 207-235).

Jacob, S. (2017). A review of technology advances for assisting paralyzed people [leading edge]. IEEE Technology and Society Magazine, 36(2), 36-37.

Orejuela-Zapata, J. F., Rodríguez, S., & Ramírez, G. L. (2019). Self-help devices for quadriplegic population: A systematic literature review. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 27(4), 692-701.

Mekki, M., Delgado, A. D., Fry, A., Putrino, D., & Huang, V. (2018). Robotic rehabilitation and spinal cord injury: a narrative review. Neurotherapeutics, 15(3), 604-617.

Guntara, A., & Rahyussalim, A. J. (2021, March). The uses of lower limb exoskeleton, functional electrical stimulation, and future improvements for leg paralysis management–A systematic review. In AIP Conference Proceedings (Vol. 2344, No. 1). AIP Publishing.

Kapula, P. R., Pasha, I. A., Kumar, B. A., & Sowmya, V. (2021, November). Wearable Lower Limb Exoskeletons for paraplegia: A Review. In 2021 International Conference on Technological Advancements and Innovations (ICTAI) (pp. 441-445). IEEE.

Krakowska, N., Szmelcer, B. P., Zaborna, D., Fortuna, A., Wszelaki, P., Florczak, A., & Kędziora-Kornatowska, K. (2019). Rehabilitation of patients with paraplegia-a review of the diversity of forms. Journal of Education, Health and Sport, 9(5), 534-546.

Sirawattanakul, S., & Sanngoen, W. (2020). Review of upper limb exoskeleton for rehabilitation and assistive application. International Journal of Mechanical Engineering and Robotics Research, 9(5), 752-758.

Esta obra está bajo una licencia de Creative Commons
Atribución-NoComercial-CompartirIgual 4.0 Internacional.