Naturaleza y Tecnología
Enero Abril 2023
ISSN 2007-672X
Universidad de Guanajuato

METAVIZ: UNA HERRAMIENTA PARA VISUALIZAR EL FUNCIONAMIENTO DE ALGORITMOS METAHEURÍSTICOS

Oscar Daniel Lara Montaño [a*], Fernando Israel Gómez Castro [b], Claudia Gutiérrez Antonio [a], Sergio Iván Martínez Guido [a]

[a]Facultad de Ingeniería, Universidad Autónoma de Querétaro, Campus Amazcala, Carretera a Chichimequillas s/n km. 1, Amazcala, El Marqués, Querétaro, 76265, México.
od.laramontano@gmail.com

[b] Departamento de Ingeniería Química, Universidad de Guanajuato, Noria Alta s/n, Guanajuato, Gto., 36050, México.

Resumen

Los algoritmos metaheurísticos son estrategias fundamentales para resolver problemas con espacios de búsqueda complejos. La visualización del funcionamiento de este tipo métodos es importante en las etapas tempranas del desarrollo de nuevos algoritmos, y resulta muy útil en las aulas de clase para facilitar el aprendizaje de estos. En este trabajo se presenta la herramienta Metaviz que permite visualizar la trayectoria de las soluciones candidato en un espacio bidimensional, misma que puede ser empleada en los principales sistemas operativos.

Palabras clave: optimización; algoritmos metaheurísticos; visualización; funciones de prueba

METAVIZ: A TOOL FOR THE VISUALIZATION OF THE WORKING MODE OF METAHEURISTIC ALGORITHMS

Abstract

Metaheuristic algorithms are fundamental strategies to solve optimization problems with complex search space. Visualizing these algorithms is important in the early stages of developing new methods; also, it is very useful for scholars interested in this branch of optimization. This work presents the Metaviz tool that allows visualization of the path followed by the different candidate solutions in a bi-dimensional search space. This tool can be employed in the main operative systems.

Keywords: optimization; metaheuristic algorithms; visualization; benchmark function

1. Introducción

Para continuar con el quehacer diario, la humanidad cada vez afronta retos más complejos con la implementación y desarrollo de ciencia y tecnología. Los distintos retos exigen soluciones con los que no solo se satisfagan los objetivos establecidos, sino que también requieran la menor cantidad posible de recursos. Una herramienta imprescindible que ayuda a tomar la mejor solución para un problema dado es la optimización.

Desde el punto de vista matemático, un problema de optimización consiste en una o varias funciones objetivo y/o un conjunto de restricciones de igualdad y desigualdad que dependen de un vector de variables de decisión. Para éstas, se deben encontrar los valores de las variables de decisión que maximicen o minimicen a las funciones objetivo, satisfaciendo las restricciones establecidas.

Muchos de los problemas reales de optimización, al ser representados con modelos matemáticos, comprenden espacios de búsqueda complejos. Esta complejidad está dada por la presencia de ecuaciones no convexas, no diferenciables y/o multimodales; adicionalmente, es común que el conjunto de variables de decisión contenga tanto variables discretas y continuas.

Con el fin de encontrar la mejor solución en problemas de optimización, se han desarrollado diversos métodos que pueden encontrarse en la literatura. Particularmente, este trabajo se enfoca en los métodos conocidos como metaheurísticos (o estocásticos), que tienen las características de; implementar parámetros estocásticos en su estructura; dado que su funcionamiento no está basado en el gradiente, sino en manipulaciones realizadas de acuerdo con los valores obtenidos en la evaluación de soluciones candidato en la función objetivo; son métodos iterativos; y mantienen un balance en la capacidad de exploración (en la que se realiza un muestreo del espacio de búsqueda) y la explotación (que consiste aumentar la calidad de la solución en el sub-espacio que probablemente contiene a la mejor solución) (Yang, 2013).

Es importante mencionar que, aunque estos algoritmos son de aplicación general su desempeño depende del problema a resolver (Wolpert & Macready, 1997). Es común que, cuando se propone un nuevo algoritmo metaheurístico, este sea probado en funciones de prueba con el propósito de identificar en qué tipo de entornos tienen un funcionamiento adecuado. A partir del desempeño del algoritmo propuesto los investigadores ajustan el valor de los parámetros para que se tenga un desempeño adecuado en el mayor número de funciones de prueba.

Aunque existen diversas librerías para analizar la calidad de las soluciones desde el punto de vista puramente numérico, dentro del conocimiento de los autores no existe una herramienta disponible que permita el seguimiento visual del comportamiento de las soluciones candidato a lo largo del proceso iterativo. En este sentido, el presente trabajo se desarrolla una herramienta que permite visualizar el funcionamiento de algoritmos de optimización metaheurística en problemas de optimización de dos dimensiones. La herramienta puede tener un impacto importante al ser empleada como apoyo para desarrollar la estructura de un nuevo método que se desee proponer, analizar el impacto de modificaciones a algoritmos existentes, o bien, en el ámbito educativo, para que los estudiantes comprendan de mejor manera el comportamiento y funcionamiento de los algoritmos metaheurísticos.

En la literatura pueden encontrarse una gran cantidad de algoritmos de optimización metaheurística; sin embargo, en la herramienta, que lleva el nombre Metaviz, se contemplan cuatro. Estos son la Búsqueda Cuckoo (CS), Evolución Diferencial (DE), Optimizador por Enjambre de Partículas (PSO), y el Algoritmo Basado en la Enseñanza y el Aprendizaje (TLBO). Los cuatro se basan en poblaciones, y en todos ellos el usuario puede agregar algoritmos de optimización propios.

El algoritmo CS fue propuesto por Yang & Deb (2009), se inspira en la manera parasitaria en la que se reproducen los cucos. En su estructura se emplea una distribución de Lévy para explorar de manera eficiente el espacio de búsqueda. La DE es una estrategia inspirada en la teoría de evolución Darwiniana propuesta por Storn & Price (1997), esta emplea los operadores de selección, mutación y entrecruzamiento de las especies. El PSO es uno de los primeros métodos conocidos como inteligencia de enjambre. Fue propuesto por Kennedy & Eberhart (1995) y se inspira en el movimiento que siguen algunos animales, como las aves o los peces. El movimiento de las soluciones candidato en el espacio de búsqueda en el transcurso de las iteraciones está definido por un término de inercia, un término de comunicación social y un término cognitivo. Por último, el algoritmo TLBO presentado por Rao et al. (2012) tiene la virtud de no contener parámetros en su estructura; se inspira en el proceso de aprendizaje de los estudiantes en las aulas.

En la Sección 2 se muestra la estructura de la herramienta propuesta, así como la información que el usuario debe indicar para su funcionamiento. La Sección 3 contiene los resultados obtenidos se la aplicación de Metaviz en dos funciones de prueba. Por último, en la Sección 4 se dan las conclusiones del trabajo, así como algunas recomendaciones para el uso de la herramienta.

2. Metodología

Metaviz es una herramienta de acceso libre creada en Python, por lo que puede ser implementada en los diferentes sistemas operativos. La versión actual permite visualizar la trayectoria que siguen la soluciones candidato en el proceso iterativo en dos y tres dimensiones. Es decir, que se limita a problemas de optimización que contemplan dos y tres variables de decisión. Metaviz está disponible a través del siguiente enlace https://github.com/DanlaraIQ/Metaviz.

2.1 Estructura y uso de Metaviz

La herramienta se conforma de dos archivos; el primero se llama *optimizadores.py*, que contiene los algoritmos de optimización BC, ED, OEP y AB y es en este en el que el usuario puede agregar algoritmos propios siguiendo el formato establecido. El segundo archivo es nombrado *visaulizador.py*, el cual contiene el código que se encarga de realizar el seguimiento de la trayectoria que siguen las soluciones candidato y la creación de la animación.

Metaviz no cuenta con una interfaz gráfica, sin embargo, puede ser ejecutada desde el archivo *visaulizador.py* usando el terminal o bien empleando un entorno de desarrollo integrado.

Para cada problema de optimización, el usuario debe especificar el límite inferior de las variables de decisión, límite superior de las variables de decisión, número de soluciones candidato y número de iteraciones. Además, es necesario codificar la función objetivo en forma de función y seleccionar el algoritmo de optimización que se deseé implementar y definir la ruta para guardar el archivo que corresponde a la animación. Estos valores por indicar se muestran en la Figura 1. La plataforma también genera las imágenes de cada una de las iteraciones, permitiendo al usuario la posibilidad de hacer un análisis más detallado. Las imágenes producidas en cada una de las iteraciones contienen como nombre genérico de los ejes a x_1 y x_2, y los valores numéricos de acuerdo con los límites establecidos.

```python
        images.append(imageio.imread(filename))
        # Aquí se establece la ruta en la que se desean guardar las imágenes y la animación generadas.
    imageio.mimsave('/Users/oscarlara/Dropbox/proyectosGIT/MetaVisual/scripts/animacion.gif', images)

# Aquí se establecen los datos que requiere Metaviz para funcionar.
## Función objetivo
fobj = f2
# Límite superior (ub) e inferior (lb)
lb = np.array([-500, -500])
ub = np.array([500, 500])
dim = len(ub)
## Número de individuos
numInd = 8
## Número de iteraciones
numRun = 75
# Seleccionar el optimizador a emplear
optimizador = opt.cs
```

Figura 1. Variables requeridas por la herramienta Metaviz.

2.2 Casos de estudio

El archivo *visualizaror.py* contiene dos funciones de prueba en dos dimensiones con el que el usuario puede interactuar para familiarizarse con la herramienta. Estas funciones corresponden a las ecuaciones 1 y 2. Los límites inferiores y superiores para la ecuación 1 son [-100, -100] y [100, 100], respectivamente. Para la ecuación 2, los límites inferiores y superiores para las variables de decisión son [-500, 500], respectivamente. Para la ecuación 1 el óptimo global se encuentra en [0, 0] y el de la ecuación 2 en [420.9687, 420.9687].

$$f(x) = x_1^2 + x_2^2 \quad (1)$$

$$f(x) = 418.9829(2) - x_1 \, sin\left(\sqrt{(|x_1|)}\right) - x_2 \, sin\left(\sqrt{(|x_2|)}\right) \quad (2)$$

La función objetivo dada por la ecuación 1 es convexa, diferenciable y unimodal. Por otro lado, la ecuación 2 es no convexa, unimodal y diferenciable.

Ambas funciones de prueba se consideraron como caso de estudios. Se empleo una población de 8 y 30 individuos, así como 75 iteraciones. Con fines de comparación de emplearon los algoritmos de optimización CS, DE y TLBO.

3. Resultados

Esta sección contiene diferentes figuras obtenidas empleando Metaviz. En la parte superior se indica el método empleado, mientras que en el lateral izquierdo se tiene un número separado por un guion. El número previo al guion indica el tamaño de la población y el posterior la iteración en la que se obtiene la figura.

La Figura 2 muestra la posición de las soluciones candidato en las iteraciones 10, 30 y 60, con una población de 8 individuos cuando se emplea la ecuación 1. En cada gráfica se muestran líneas de contorno que muestra la forma de la función objetivo, los puntos indican la posición de las 8 soluciones candidato, y las flechas en color negro muestran hacia donde se mueve la solución candidato en la próxima iteración.

Naturaleza y Tecnología
Enero Abril 2023
ISSN 2007-672X
Universidad de Guanajuato

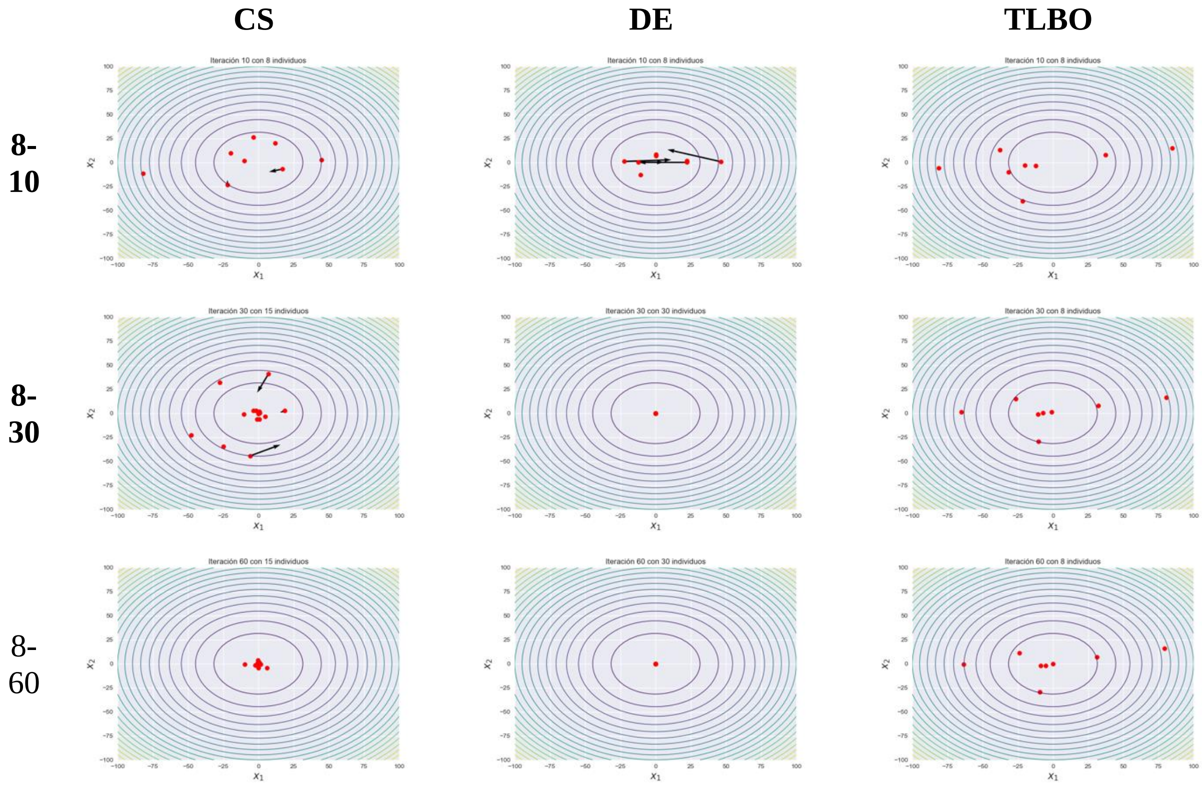


Figura 2. Trayectoria de las soluciones candidato en el proceso iterativo con la ecuación 1, usando una población de 8 individuos en la iteración 10, 30 y 60.

La herramienta permite visualizar que, para la función objetivo dada por la ecuación 1, la DE converge en menos iteraciones comparada con los otros dos algoritmos. La DE converge completamente en menos de 30 iteraciones, la CS en 60 iteraciones está en la etapa de explotación y el TLBO aún no ha pasado de la etapa de exploración hacia la explotación. También, a través de las flechas que indican la trayectoria seguida por las soluciones candidato se observa que los mayores movimientos se generan al inicio del proceso iterativo, sobre todo en la DE, después el movimiento es menor. Es importante mencionar que dentro del análisis que se puede realizar a través de la herramienta presentada, es adecuado concluir que el TLBO no funciona de manera adecuada para este tipo de regiones de búsqueda. La CS se comporta de manera satisfactoria, sin embargo, requiere un mayor número de iteraciones que la DE.

La Figura 3 muestra el desarrollo de las posiciones de las soluciones candidato en la gráfica de superficie de generada para la ecuación 2. Se visualiza que nuevamente la DE tiene el mejor desempeño en este entorno; converge rápidamente, aunque en un mínimo local. Al igual como sucedió para el problema de optimización dado por la ecuación 1, el TLBO no es capaz de converger en las iteraciones contempladas: este algoritmo de optimización presenta poca capacidad de exploración y explotación. Por otro lado, la CS ha convergido completamente en la iteración 30 en un mínimo local, después los operadores contemplados en el algoritmo ocasionan que el método converja fuera de la región factible.

Naturaleza y Tecnología
Enero Abril 2023
ISSN 2007-672X
Universidad de Guanajuato

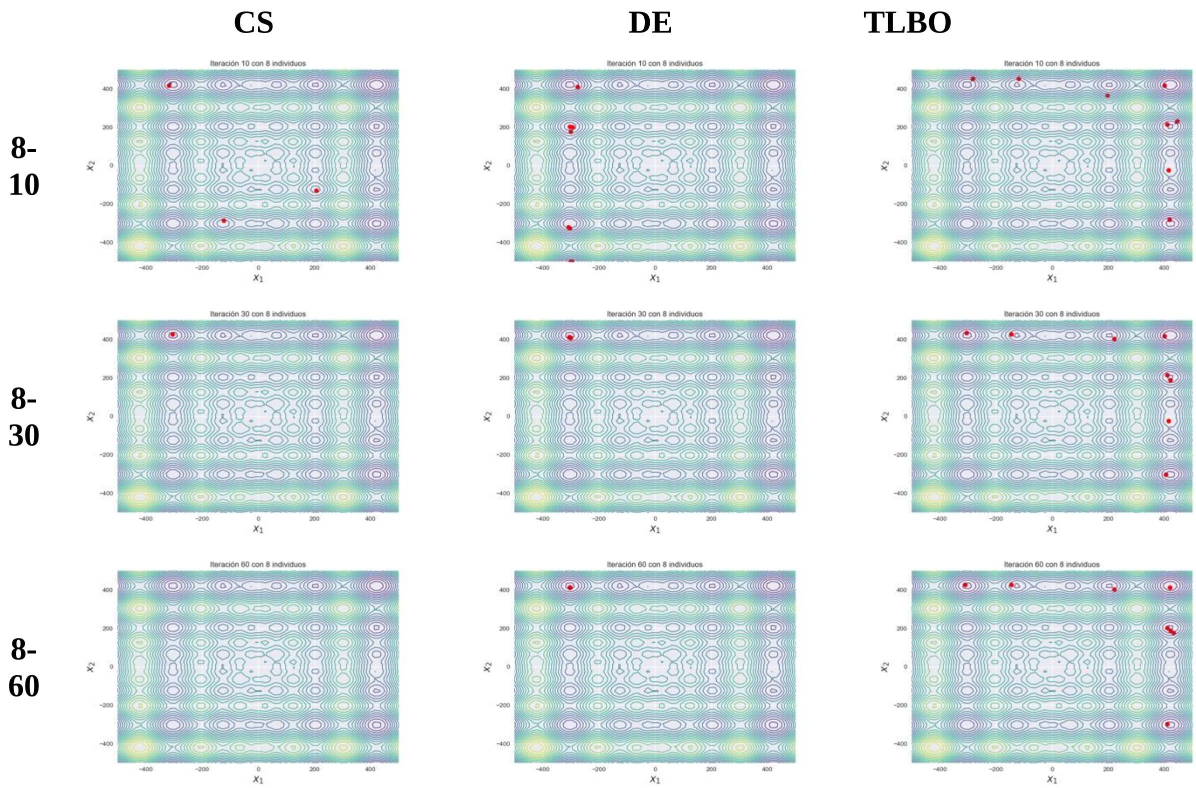


Figura 3. Trayectoria de las soluciones candidato en el proceso iterativo con la ecuación 2, usando una población de 8 individuos en la iteración 10, 30 y 60.

Uno de los aspectos que el usuario puede observar, relevantes sobre todo en el ámbito educativo, son los relacionados con el número de soluciones candidato con las que se conforma la población empleada en el proceso iterativo. En la Figura 4 se hace una aplicación similar a la mostrada en la Figura 2; no se usan 8 soluciones candidato, sino 30. El aumento del tamaño de la población afecta principalmente al TLBO. Lo observado en la Figura 4 se relaciona con los operadores empleados por el algoritmo; este método emplea una distribución gaussiana con la cual se comparan las distintas funciones candidato. La mayor diversidad de soluciones ocasiona un mayor movimiento de las soluciones candidato en el proceso iterativo. En la Figura 5, al igual que en la Figura 3, se emplea la función de prueba mostrada en la ecuación 2, pero con una población de 30 soluciones candidato. La CS encuentra soluciones en la región del óptimo global, sin embargo, nuevamente converge en una solución no factible. Al contar con una población más grande, la DE es capaz de realizar un mejor muestreo de la región factible para posteriormente converger en el óptimo global. El algoritmo TLBO logra encontrar la vecindad del óptimo global, teniendo como resultado lo que se conoce como convergencia prematura.

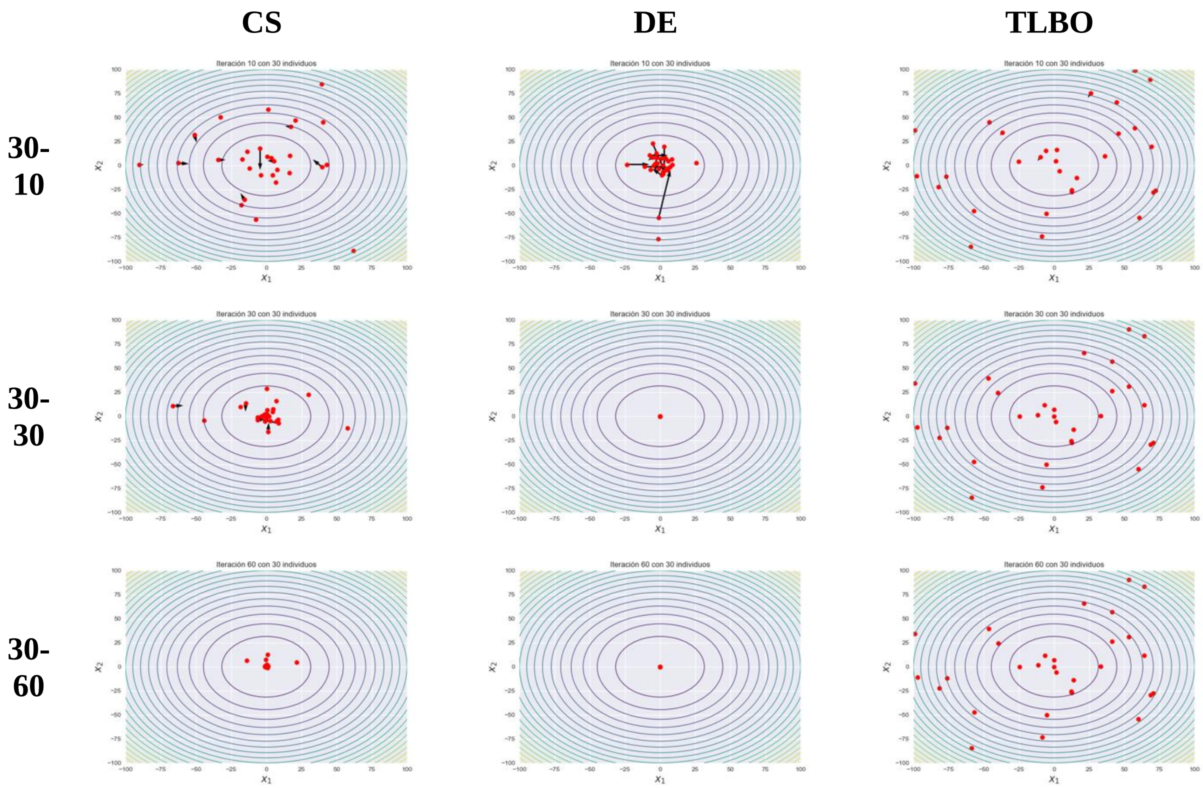
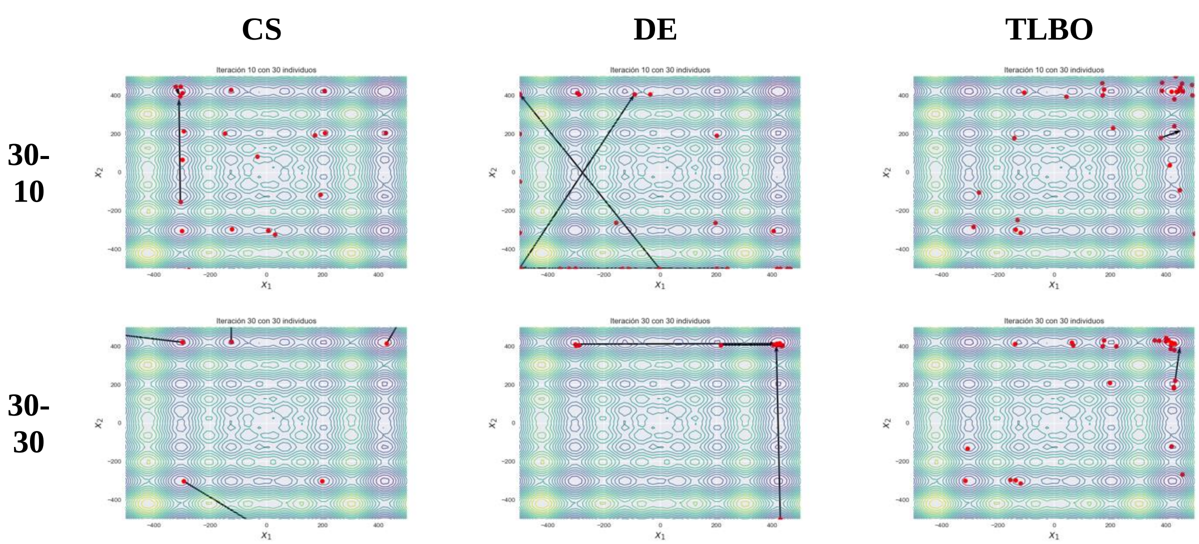


Figura 4. Trayectoria de las soluciones candidato en el proceso iterativo con la ecuación 1, usando una población de 30 individuos en la iteración 10, 30 y 60.



30-60

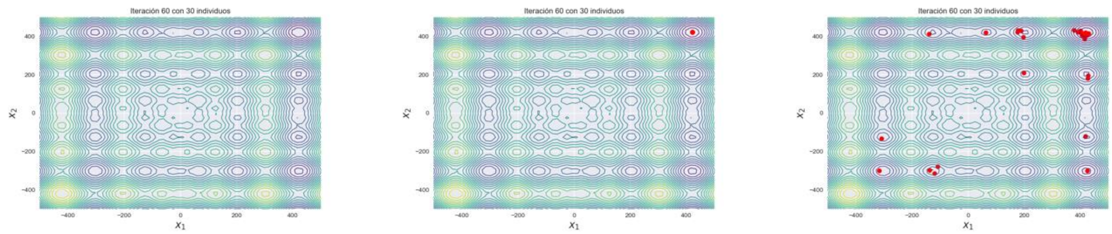


Figura 5. Trayectoria de las soluciones candidato en el proceso iterativo con la ecuación 2, usando una población de 30 individuos en la iteración 10, 30 y 60.

4. Conclusiones

En este trabajo se presenta la herramienta Metaviz que permite la visualización de la trayectoria que siguen las soluciones candidato de métodos de optimización metaheurística en el transcurso de las iteraciones en un espacio de dos dimensiones.

Metaviz facilita la comprensión del efecto que tienen los operadores propuestos cuando se están proponiendo nuevos métodos de optimización, así como el impacto del valor asignado a variables generales como el número de iteraciones o tamaño de la población. La herramienta es especialmente útil para analizar el funcionamiento de algoritmos de optimización que se encuentren en fases iniciales.

En las aulas de clase también puede ser una herramienta de gran utilidad debido a que puede ayudar al profesor a mostrar a los estudiantes como realizan el proceso de búsqueda diferentes algoritmos metaheurísticos, y cómo afecta su funcionamiento la manipulación del valor de los parámetros de los que dependen.

Referencias bibliográficas

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Rao, R. V, Savsani, V. J., & Vakharia, D. P. (2012). Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large-scale problems. Inf. Sci., 183(1), 1–15. https://doi.org/https://doi.org/10.1016/j.ins.2011.08.006

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

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67–82. https://doi.org/10.1109/4235.585893

Yang, X.-S. (2013). Metaheuristic Optimization: Nature-Inspired Algorithms and Applications. En X.-S. Yang (Ed.), Artificial Intelligence, Evolutionary Computing and Metaheuristics: In the Footsteps of Alan Turing (pp. 405–420). Springer Berlin Heidelberg.

https://doi.org/10.1007/978-3-642-29694-9_16

Yang, X.-S., & Deb, S. (2009). Cuckoo Search via Lévy Flights. 2009 World Congress on Nature & Biologically Inspired Computing: 9-11 December 2009, Coimbatore, India: proceedings.