

Mecánica Computacional Vol XL, págs. 1107-1116 (artículo completo)  
**EVALUACIÓN DE DESEMPEÑO DEL OPTIMIZADOR HIPERHEURÍSTICO COOPERATIVO PARALELO (PCHO)**  
Storti (Eds.)  
Concordia, 6-9 Noviembre 2023

**PERFORMANCE EVALUATION OF THE PARALLEL COOPERATIVE  
HYPERHEURISTIC OPTIMISER (PCHO)**

**Diego O. Debortoli<sup>a,b</sup>, Paola P. Oteiza<sup>a,c</sup>, Juan I. Ardenghi<sup>a</sup> y Nélica B. Brignole<sup>a,b</sup>**

<sup>a</sup>Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC), Universidad Nacional del Sur (DCIC-UNS), Av. Alem 1253, Bahía Blanca, Argentina, ardenghi@criba.edu.ar

<sup>b</sup>Planta Piloto de Ingeniería Química (PLAPIQUI), Complejo CCT-UAT, CONICET-UNS, Camino La Carrindanga Km 7, 8000 Bahía Blanca, Argentina, dybrigno@criba.edu.ar

<sup>c</sup>Departamento de Ingeniería Química (DIQ), Universidad Nacional del Sur (UNS), B8000 Bahía Blanca, Argentina

**Palabras clave:** Metaheurísticas, Optimización, Funciones de prueba, Funciones multimodales

**Resumen.** El Optimizador Hiperheurístico Cooperativo Paralelo (PCHO) es un resolvidor de alto nivel que se desarrolló originalmente para la optimización en ingeniería de procesos. El algoritmo comprende las metaheurísticas clásicas denominadas Algoritmos Genéticos (GA), Recocido Simulado (SA) y Optimización por Enjambre de Partículas (PSO), cuyos parámetros se eligen de forma adaptativa durante las ejecuciones. Dado que PCHO ha mostrado un rendimiento satisfactorio con colecciones que contienen exclusivamente problemas de ingeniería química, es un reto ampliar su alcance realizando una mayor variedad de pruebas. Este trabajo se centra en evaluar el rendimiento algorítmico mediante pruebas que comprenden problemas multidimensionales duros y escalables, con funciones unimodales ó multimodales. Para analizar la eficacia algorítmica se adoptaron dos indicadores de rendimiento. El primero es una distancia que muestra la proximidad del resultado final a la mejor solución. El otro es la brecha de aptitud, que evalúa la calidad de la solución obtenida en términos del error relativo del valor de aptitud alcanzado. El estudio experimental también valida el teorema de "no hay almuerzo gratis" que estipula que ningún algoritmo es eficiente para tratar todos los problemas de optimización.

**Keywords:** Metaheuristics, Optimization, Test functions, Multimodal functions

**Abstract.** The Parallel Cooperative Hyperheuristic Optimiser (PCHO) is a high-level solver that was originally developed for process engineering optimisation. The algorithm comprises classical metaheuristics called Genetic Algorithms (GA), Simulated Annealing (SA) and Particle Swarm Optimisation (PSO), whose parameters are adaptively chosen during runs. Since PCHO has shown satisfactory performance with collections containing exclusively chemical engineering problems, it is a challenge to extend its scope by performing a wider variety of tests. Therefore, this work focuses on evaluating algorithmic performance using tests comprising hard and scalable multidimensional problems with unimodal or multimodal functions. Two performance indicators were adopted to analyse algorithmic efficiency. The first is a distance that shows how close the final result is to the best solution. The other is the fitness gap, which evaluates the quality of the solution obtained in terms of the relative error of the fitness value achieved. The experimental study also validates the "no free lunch" theorem, which stipulates that no algorithm is efficient in dealing with all optimisation problems.

## 1 INTRODUCCION

El desarrollo de algoritmos de optimización mediante metaheurísticas ha despertado gran interés en los últimos tiempos (Burke et al., 2013). Este enfoque consiste en el diseño e implementación de procedimientos iterativos que guían una heurística subordinada combinando de forma inteligente distintos conceptos para explorar y explotar adecuadamente el espacio de búsqueda (Luke, 2013). Las heurísticas son métodos que se basan en conocimientos específicos del problema y tienen la capacidad de generar soluciones eficientes en un tiempo razonable. Sin embargo, los enfoques heurísticos específicos no siempre ofrecen un rendimiento óptimo cuando se aplican sin modificaciones significativas. Esta limitación ha impulsado el desarrollo de "hiperheurísticas" de propósito general, término que se refiere a una combinación de metodologías de búsqueda automatizada de alto nivel que explora un espacio de búsqueda de heurísticas de bajo nivel (metaheurísticas) para resolver problemas computacionalmente difíciles (Drake et al., 2020). En los últimos años, se ha logrado un progreso significativo en el desarrollo de métodos de alto nivel que pueden aplicarse a una amplia gama de casos diferentes. Su utilidad para problemas de negocios actualmente constituye un tópico de investigación muy atractivo, pues su empleo asociado a la recolección de datos reales no ha sido explorado aún en ingeniería de procesos.

Además, en la literatura se puede encontrar un conjunto extenso de técnicas metaheurísticas, donde la mayoría de los desarrollos recientes tienden a alterar el proceso de búsqueda para lograr un balance entre las tareas de exploración y explotación. Las metaheurísticas son métodos de optimización aproximada que combinan procedimientos de mejora local con estrategias de nivel superior para crear un proceso de búsqueda iterativo capaz de superar óptimos locales y realizar una búsqueda robusta en amplios espacios. En los últimos años, se ha observado un creciente interés en la investigación acerca de integración de técnicas de aprendizaje automático (Machine Learning) en las metaheurísticas para abordar Problemas de Optimización Combinatoria (COP). Esta integración tiene como objetivo mejorar el rendimiento de las metaheurísticas en términos de calidad de solución, tasa de convergencia y robustez, mediante la extracción de conocimiento útil de los datos generados durante el proceso de búsqueda. Al incorporar este conocimiento, las metaheurísticas pueden tomar decisiones más acertadas, volviéndose más inteligentes y mejorando su rendimiento global. (Karimi-Mamaghan et al., 2022). Por otra parte, Cruz Duarte et al. (2023) presentaron MatHH. Se trata de una arquitectura basada en Matlab<sup>®</sup> con un enfoque orientado a objetos que permite armar en forma rápida prototipos de hiperheurísticas. Sin embargo, su enfoque no contempla que, ante un problema dado, la eficiencia de las metaheurísticas depende de la selección cuidadosa de sus parámetros.

Por lo tanto, para mejorar la inteligencia del optimizador y reducir sus tiempos de cómputo, otra alternativa viable es implementar un mecanismo de aprendizaje simple que incorpore características auto-adaptativas en cada metaheurística. Con este propósito dual fue concebido el Optimizador Hiperheurístico Cooperativo Paralelo (PCHO), el cual fue originalmente desarrollado para la optimización en ingeniería de procesos (Oteiza et al., 2021). Los resultados obtenidos demuestran la flexibilidad, precisión y eficiencia del PCHO, gracias al trabajo conjunto de sus agentes y a la incorporación de rasgos de auto-adaptabilidad en todas las metaheurísticas que lo componen.

## 2 OBJETIVOS

Sabemos que ningún algoritmo es eficiente para tratar todos los problemas de

optimización. Wolpert y Macready (1997) presentaron un marco formal en el que comparan algoritmos de optimización de propósito general e incluyeron una perspectiva geométrica. Derivaron una serie de teoremas "no hay almuerzo gratis" que demuestran el peligro de comparar algoritmos por su rendimiento en una pequeña muestra de problemas.

Las estructuras de cooperación han sido poco estudiadas hasta el presente y vale la pena continuar explorándolas en el futuro. Aunque el PCHO ha mostrado un rendimiento satisfactorio en diversos problemas de ingeniería química, el desafío radica en ampliar su alcance a través de una mayor variedad de pruebas. Se pretende explorar nuevos dominios y aplicaciones para evaluar el comportamiento general, la flexibilidad y la robustez del PCHO en diferentes contextos. Sobre esta base, en este trabajo identificamos mejoras para fortalecer su utilidad en varias áreas. El propósito de este estudio es perfeccionar nuestra metodología a fin de lograr un optimizador generalizado PCHO más robusto. Por lo tanto, hemos diseñado y ejecutado experimentos con problemas difíciles de optimización, para luego analizar los datos y extraer conclusiones.

### 3 FUNDAMENTOS

La optimización es una herramienta clave para la toma de decisiones en variadas ramas de ingeniería, economía y administración de empresas. Particularmente en ingeniería química, los algoritmos de optimización constituyen los instrumentos fundamentales para el diseño experimental, la estimación de parámetros, el desarrollo de modelos y el análisis estadístico; la síntesis, análisis, diseño y retroajuste de procesos; el control predictivo de modelos y la optimización en tiempo real; la planificación, programación e integración de las operaciones de proceso en la cadena de suministro para fabricación y distribución de productos.

La mayoría de estos problemas de optimización pueden encuadrarse en la categoría No Lineal Mezcla Entera (MINLP: Mixed Integer Non-Linear Problems). Un MINLP se representa formalmente con la Ec. 1, donde  $f(x, y)$  es la función objetivo;  $h(x, y) = 0$  son las ecuaciones que describen el proceso;  $g(x, y) \leq 0$  son especificaciones o restricciones del problema. Las variables  $x$  son continuas, mientras que  $y$  representa variables discretas que comúnmente definen decisiones lógicas, tales como la asignación de equipos ó la secuenciación de tareas (Biegler, 2010).

$$\begin{aligned} & \min_{x,y} f(x, y) \\ \text{sujeto a:} & \quad h(x, y) = 0 \\ & \quad g(x, y) \leq 0 \\ & \quad x \in \mathbb{R}^n; \quad y \in [0,1]^T \end{aligned} \tag{1}$$

Muchos de estos problemas pertenecen a la categoría NP-hard. Un problema es de esta clase cuando no se ha encontrado un algoritmo eficiente para resolverlo, y se cree, si bien aún no ha sido demostrado, que no existe un algoritmo de resolución general para este problema que funcione en tiempo polinómico (Goldreich, 2010). Por ende, estos problemas resultan intratables con algoritmos clásicos de optimización, pues para encontrar el óptimo en forma exacta mediante dichas técnicas se pueden llegar a requerir tiempos exponenciales. En este contexto, resultan muy útiles las metaheurísticas ya que permiten encontrar soluciones en tiempos de cómputo razonables.

### 3.1 Evaluación de desempeño de algoritmos metaheurísticos

Los algoritmos metaheurísticos han tenido éxito en la resolución de numerosos problemas en comparación con sus homólogos deterministas. A pesar de esta ventaja, la naturaleza estocástica de tales algoritmos da lugar, mediante un cierto número de ensayos, a un espectro de soluciones que puede conducir a la aparición de incertidumbres en la calidad de los resultados. Por lo tanto, es de suma importancia utilizar una herramienta correcta para medir el rendimiento del conjunto diverso de algoritmos metaheurísticos a fin de derivar un juicio apropiado sobre la superioridad de los algoritmos y también con el objeto de validar las afirmaciones planteadas por los investigadores en relación a sus objetivos específicos.

El rendimiento de un algoritmo metaheurístico aleatorio puede dividirse en medidas de eficiencia y eficacia. La eficiencia está relacionada con la velocidad del algoritmo para encontrar soluciones precisas, su convergencia y sus cálculos. Por otro lado, la eficacia se refiere a la capacidad del algoritmo para encontrar soluciones de calidad. Ambos alcances son cruciales para problemas continuos y discretos, ya sean de uno o varios objetivos. Cada tipo de problema tiene una formulación y unos métodos de medición diferentes dentro del ámbito del rendimiento asociado a la eficiencia y eficacia. Uno de los veredictos más decisivos para la medida de la eficacia es el análisis estadístico que depende de la distribución de los datos y la metodología adecuada para emitir juicios correctos (Halim et al., 2021).

En el campo de los algoritmos metaheurísticos se sabe que una sola ejecución no es suficiente para medir el rendimiento, sino que es necesario analizar la solución del algoritmo en una cantidad fija de  $n$  ejecuciones, típicamente se adopta  $n=30$ . A partir de este mínimo de ejecuciones se acostumbra establecer los parámetros de rendimiento mencionados en los diferentes tipos de problemas abordados.

Para los problemas de objetivo único, la principal preocupación en la medida de la eficiencia se refiere a la tasa de resolución del problema que incluye la tasa de convergencia, la diversidad de agentes de búsqueda, el costo computacional, la complejidad y medidas estadísticas, tales como la distribución acumulativa. Un ámbito similar se aplica a los problemas discretos de objetivo único. Algunas medidas adicionales, especialmente para los problemas combinatorios, incluyen medidas de tiempo de ejecución utilizando tasas de sub-solución óptima y mejoras del recorrido. Desde el punto de vista de la eficacia, las medidas generales para ambos dominios de objetivo único pueden dividirse principalmente en tasa de eficacia, medidas de perfil, escalabilidad, precisión de 100 dígitos y medidas estadísticas que incluyen las descriptivas y las inferenciales. La opción de inferencia estadística se divide a su vez en pruebas frecuentistas y bayesianas.

Entre las métricas significativas implementadas en varios trabajos de la literatura están los porcentajes de convergencia exitosa ya sea por la mejor solución o el promedio de soluciones. También se destaca que utilizar sólo la mejor solución como indicador de rendimiento no es suficiente. Se aconseja incluir también las medidas medias, así como la variabilidad de la solución, para describir la eficacia del algoritmo en relación con las soluciones convergentes globales. La comparación con otros algoritmos de referencia es también otra métrica de apoyo para comprender la calidad del algoritmo propuesto con respecto a otros algoritmos establecidos (Abualigah et al., 2021).

## 4 DISCUSIÓN

### 4.1 Sobre el Optimizador Hiperheurístico Cooperativo Paralelo (PCHO)

El PCHO (Oteiza et al., 2021) es un novedoso optimizador con estructura Master-Worker basado en hiperheurísticas, cuyos trabajadores son las siguientes metaheurísticas clásicas:

Algoritmos Genéticos (GA), Recocido Simulado (SA) y Optimización por Enjambre de Partículas (PSO). Para introducir diversidad, posee herramientas que imitan procesos de selección natural (algoritmos evolutivos) y mecánica estadística (SA).

En primer lugar, la entrada se adapta para ajustarse a la información del modelo que necesita cada agente. Después, todos ellos empiezan a trabajar. Durante todo el procedimiento, se aplica una técnica de aprendizaje cooperativo para mejorar la búsqueda. Cada vez que se encuentra una solución candidata, se informa a todos los agentes. El trabajador que ha conseguido encontrar una nueva solución candidata, es decir el ganador, mantiene su configuración, mientras que los demás agentes se resintonizan. A continuación, la clasificación, que es una lista ordenada basada en valores de aptitud, se actualiza con el nuevo candidato. El proceso continúa hasta que se completan 30 iteraciones externas. Es importante señalar que cada agente realiza 15 iteraciones internas, administrando por si solo sus propios criterios de terminación. Finalmente, el Maestro informa a todos sobre la mejor solución encontrada. Además, en el PCHO también se han contemplado entradas opcionales de soluciones sugeridas, ya que en muchos casos el usuario experimentado puede prever la localización aproximada del punto óptimo desde un punto de vista ingenieril.

Dado que el rendimiento de los algoritmos evolutivos depende en gran medida de los valores de sus parámetros y operadores, se lleva a cabo una selección en línea para elegirlos de forma adaptativa durante la búsqueda. Desde un punto de vista práctico, esta estrategia contribuye a favorecer la diversidad. Cuando un agente no consigue ser el ganador, se vuelven a ajustar algunos de sus parámetros basándose en su último rendimiento para el mismo problema. En el PCHO, los parámetros sintonizables son la temperatura para SA y el peso inercial para PSO. En cuanto al GA, tanto la tasa de cruzamiento como el mecanismo de cruce se adaptan automáticamente durante su evolución.

La forma de compartir la información establece el grado de cooperación entre los agentes. La política de actualización depende de la metaheurística. En el caso de PSO y GA, la población se actualiza convenientemente, mientras que en SA se cambia el punto inicial. Dependiendo de su entrada, cada meta-heurística adopta un comportamiento diferente con el fin de incluir el mejor descubrimiento hasta el momento. Hay que tener en cuenta que, en términos generales, la información compartida proporciona una idea de dónde se encuentran las zonas buenas del espacio. De ahí que se pueda generar un nuevo conjunto de muestras. Dado que el progenitor potencial (es decir, la solución candidata recientemente informada) es más prometedor, el conjunto enriquecido aportará diversidad a la población inicial del GA, favoreciendo así la convergencia.

En concreto, SA itera utilizando un parámetro sintonizable, calculando siempre los valores de aptitud para quedarse con la mejor solución. Dado un candidato mejor recién informado, la solución existente debe sustituirse para que el paseo aleatorio continúe desde un punto recién creado. De este modo, el reinicio podría ayudar a escapar de los óptimos locales. Si SA se atascara corriendo hacia un óptimo local, se podría acelerar la velocidad de cálculo y guiar la búsqueda para encontrar el óptimo global. Por su parte, en PSO se evalúa la aptitud de cada partícula y se actualiza la localización mejor que ha sido descubierta.

Una de las principales ventajas del PCHO es su flexibilidad, ya que se configura automáticamente para adaptarse a las características específicas del problema para reducir los tiempos de ejecución. Al ajustar el rendimiento de los agentes a las instancias particulares, se mejora significativamente la aplicabilidad del PCHO. Para evaluar el rendimiento del algoritmo, Oteiza et al. (2021) analizaron un conjunto de pruebas exclusivamente aplicadas a problemas de optimización no lineal que surgen normalmente en ingeniería de procesos.

En líneas generales, los problemas de optimización que se plantean en ingeniería química pueden estar en función de variables continuas y discretas. Sin embargo, Oteiza et al. (2021)

no evaluaron el desempeño del PCHO empleando problemas que involucren variables enteras. En principio, el PCHO podría aplicarse a casos de alta complejidad, tales como los problemas de optimización MINLP, donde las variables discretas tienden a generar un espacio de búsqueda gigante. Cabe aclarar que en este trabajo analizaremos la efectividad del PCHO para problemas difíciles de optimización continua exclusivamente.

Para ampliar la aplicabilidad del PCHO, se recomienda analizar su comportamiento con la batería de tests empleada por Abualigah et al. (2021), quienes comprobaron su rendimiento en veintinueve funciones de referencia y varios problemas de diseño de ingeniería del mundo real. Para mostrar su aplicabilidad, ellos efectuaron un análisis del rendimiento, de los comportamientos de convergencia y de la complejidad computacional mediante diferentes escenarios. Además, compararon su propuesta con otros once algoritmos bien conocidos.

Es importante mencionar que el PCHO no fue concebido con herramientas específicas para afrontar problemas multi-objetivo. En realidad, el usuario assume la existencia de un extremo, pero nunca sabe si se enfrenta a un problema multiobjetivo, a menos que el optimizador no logre éxito en su búsqueda. Por ende, la complejidad de los casos suele estar determinada por la cantidad de óptimos locales que el problema posee. En general, no se sabe cuántos óptimos tiene la función. Esto obliga a continuar la búsqueda a pesar de haber encontrado uno, dado que nunca se tiene la certeza de haber llegado a un mínimo global. En resumen, si no conociéramos una solución académica, siempre existe un grado de incertidumbre.

## 4.2 Diseño de experimentos

Teniendo en cuenta estas observaciones, los experimentos computacionales se realizaron a partir de problemas cuya solución es ya conocida. Se pretende observar cómo el algoritmo llega al óptimo global, es decir si el equipo de agentes del PCHO es capaz de equilibrar la búsqueda de objetivos con la reacción al entorno y permitir comportamientos complejos. Así, se espera lograr agentes inteligentes y un PCHO robusto, es decir que los agentes posean capacidades adicionales (Phillips-Wren, 2012) tales como:

- i. Reactividad y Adaptabilidad: que perciban el entorno y respondan a los cambios
- ii. Proactividad: que inicien acciones para lograr su objetivo
- iii. Cooperación: que tengan capacidad de comunicación con otros agentes
- iv. Persistencia: que puedan mantener su estado durante largos períodos de tiempo
- v. Movilidad: que puedan viajar por el sistema para adquirir conocimiento o hacer tareas.

Con respecto a las funciones de prueba numéricas de referencia, se han utilizado ampliamente como herramientas de evaluación (benchmarking) del rendimiento en la literatura metaheurística. También se considera que los algoritmos metaheurísticos que se desempeñan satisfactoriamente con estas funciones son capaces de resolver problemas de optimización difíciles de la vida real. Diferentes investigadores eligen diversos conjuntos de funciones con distintas configuraciones experimentales, aunque existen configuraciones experimentales comunes que pueden guiar a otros investigadores a seguir la tendencia y calibrar la robustez de cualquier algoritmo metaheurístico recién propuesto. En este sentido, Hussain et al. (2017) proporcionan una lista de funciones de referencia de uso común.

En contraste, los últimos avances en pruebas de software permiten derivar automáticamente pruebas que alcanzan casi cualquier punto deseado del código fuente. Fraser y Arcuri (2011) presentaron EVOSUITE, que se basa en optimizar colecciones completas de pruebas con el objeto de satisfacer un criterio de búsqueda específico, en lugar de generar casos de prueba diferentes dirigidos a objetivos de cobertura distintos.

Sin embargo, existe un problema fundamental con la idea general de centrarse en un objetivo de cobertura de pruebas distinto cada vez. El problema no es simple. Los objetivos de cobertura no son independientes entre sí, ni tampoco la generación de pruebas para un

objetivo concreto tiene garantías de éxito. Es por esta razón que hemos decidido identificar los rasgos de comportamiento del PCHO mediante la selección de funciones de prueba clásicas que presentan superficies de difícil exploración. Para las evaluaciones, hemos empleado además algunas nuevas funciones de prueba unimodales, multimodales y de ruido que fueron creadas por Layeb (2022) para evaluar el rendimiento de los algoritmos de optimización global. Todas estas funciones son problemas multidimensionales desafiantes cuya dureza fue demostrada mediante estudios experimentales.

Para analizar el comportamiento del algoritmo PCHO se seleccionaron funciones de prueba unimodales y multimodales disponibles en la literatura que son utilizadas para evaluar el rendimiento de los algoritmos de optimización global. Todas estas funciones seleccionadas representan problemas multidimensionales desafiantes cuya dureza fue demostrada mediante estudios experimentales.

### 4.3 La dificultad del estancamiento

El estancamiento se declara a partir de la obstinación del algoritmo en apuntar a una misma solución. En los problemas de ensayo, donde se conoce una solución teórica, el buen funcionamiento del algoritmo produce una insistencia sobre la solución conocida, situación que se observa en los casos de prueba testeados en Oteiza et al.(2021). Pero en los problemas donde no conocemos una solución preestablecida, no es posible determinar si el fenómeno del estancamiento se produce por insistencia a la solución definitiva o el algoritmo queda capturado en una región de búsqueda de la que no puede liberarse.

A partir de este inconveniente implementamos una alternativa a la versión original de PCHO que consiste en incorporar una lista de puntos a vetar en las iteraciones posteriores. Esto es, al finalizar cada iteración principal de PCHO, el punto obtenido con mejor fitness queda registrado como posible óptimo definitivo, pero al mismo tiempo se incorpora a una lista, aquí denominada lista tabú, donde queda asentado que ya fue seleccionado como óptimo. El veto se produce a través de una penalización en su valor de fitness que lo deja fuera de competencia frente a los otros puntos candidatos a óptimos. De esta forma el algoritmo evita repeticiones y es empujado a otras zonas de búsqueda. Establecer que un punto pertenece a esta lista tabú puede verse entorpecido por la precisión aritmética de la computadora utilizada en el cálculo. Entonces, acordamos que un punto  $x_k$  está en la lista tabú si su distancia a algún punto  $x_i$  existente en la lista es menor que una tolerancia establecida. Es decir,  $x_k \in \{ListaTaboo\} \leftrightarrow \|x_k - x_i\| \leq Tolerancia$  para algún  $i \in \{1, \dots, Long_{ListaTaboo}\}$ .

### 4.4 Análisis del desempeño general del PCHO

Para examinar el PCHO, utilizamos un conjunto compuesto por 27 funciones (Gavana, 2014), lográndose resultados satisfactorios en 21 de estos casos. Cabe señalar que la elección de estas funciones se basó en una selección previa de más de 100 funciones, utilizando la métrica denominada "Éxito Global". La Tabla 1 muestra los resultados obtenidos para una diversidad de problemas de optimización. Se realizaron exhaustivas pruebas del PCHO, fundamentadas en el resumen compilado por Gavana (2014). En este contexto, se llevó a cabo una evaluación de la capacidad intrínseca del optimizador para identificar soluciones óptimas o aproximadas en distintos contextos. Para cuantificar el desempeño del optimizador, se aplicaron dos medidas primordiales. La primera es la distancia euclidiana entre la solución final del PCHO y el valor verdadero. La otra medida calculada es el error relativo del fitness (brecha de aptitud).

Caso	Dimensión	Éxito global*	Distancia	Brecha de aptitud
DeVilliersGlasser02	5	0.00	3,50	5,78E <sup>-06</sup>
Damavandi	2	0.25	0,00	1,56E <sup>-11</sup>
CrossLegTable	2	0.83	89,38	-1,00
XinSheYang03	2	1.08	0,00	0,00
SineEnvelope	2	2.17	2,07	4,90E <sup>-01</sup>
Whitley	2	4.92	0,00	0,00
Zimmerman	2	4.92	3,42	2,29E <sup>+02</sup>
Griewank	2	6.08	0,00	2,26E <sup>-06</sup>
Trefethen	2	6.58	0,00	9,48E <sup>-06</sup>
Bukin06	2	6.83	177,39	1,17E <sup>-02</sup>
CrownedCross	2	8.42	3,14	9,70E <sup>+01</sup>
RosenbrockModified	2	8.42	0,34	3,49
Salomon	2	10.33	0,00	2,51E <sup>-05</sup>
PowerSum	4	10.50	1,56	1,53E <sup>-05</sup>
Stochastic	2	14.17	0,13	5,80E <sup>-06</sup>
Kowalik	4	14.83	>>1	2,57E <sup>+03</sup>
ZeroSum	2	15.08	447,51	0,00
Price02	2	18.00	0,00	0,90
EggHolder	2	18.92	0,14	-6,15E <sup>-04</sup>
Ripple01	2	19.25	0,00	-2,32E <sup>-02</sup>
Mishra03	2	20.42	>>1	-6,72E <sup>+67</sup>
XinSheYang01	2	20.92	0,00	6,69E <sup>-06</sup>
NewFunction01	2	21.17	>>1	1,02E <sup>+66</sup>
DropWave	2	21.25	0,00	0,00
Branin02	2	23.50	0,00	2,46E <sup>-05</sup>
Schaffer01	2	23.67	0,23	2,09E <sup>-13</sup>
Schaffer03	2	25.92	0,00	2,34E <sup>-01</sup>
Easom	2	26.08	0,00	0,00
HelicalValley	3	34.75	0,00	1,25E <sup>-07</sup>
Shubert01	2	39.42	386578,42	0,00
Trid	6	43.83	0,00	-2,00 E <sup>-05</sup>
Schwefel01	2	82.08	0,02	1,65E <sup>-17</sup>

\*Gavana (2014)

Tabla 1: Resultados numéricos

La distancia euclidiana se utilizó con el fin de discernir la proximidad de las soluciones obtenidas por el optimizador con respecto a las soluciones teóricas ya establecidas. Esta métrica brinda una apreciación cuantitativa de la precisión de la optimización en términos de localización. Para la evaluación del desempeño basada en el error relativo del fitness, se instauró un umbral en 1. Aquellos valores de error relativo superiores a 1 fueron

categorizados como "deficientes", lo cual denota que la optimización no logró mejorar el valor de aptitud en relación con el valor teórico. En contrapartida, los valores de error relativo menores que 1 fueron considerados "satisfactorios", sugiriendo que la optimización logró efectuar una mejora en el valor de aptitud con respecto al marco de referencia teórico. Este análisis otorgó la capacidad de clasificar los resultados obtenidos por el optimizador en diversas categorías, generando una panorámica comprensiva de su rendimiento en un amplio espectro de problemas de optimización. Los resultados considerados "satisfactorios" reflejan casos en los cuales el PCHO logró optimizar el valor de aptitud. Por el contrario, resultados catalogados como "deficientes" (indicados en la Tabla 1 mediante filas sombreadas) señalan situaciones en las que el PCHO no consiguió superar la referencia teórica establecida.

Se obtuvieron resultados particularmente precisos en ciertos escenarios. Por ejemplo, en el caso "DeVilliersGlasser02", se alcanzó un nivel de error relativo de fitness muy bajo (del orden de  $10^{-6}$ ). Este logro es significativo pues se trata de una función que es altamente desafiante desde el punto de vista de la optimización. En un tenor similar, se lograron avances notables en casos tales como "Damavandi", donde se obtuvo una brecha de aptitud del orden de  $10^{-11}$ , y "XinSheYang03", "DropWave", "Whitley" e "Easom", quienes alcanzaron una distancia euclideana igual a cero y un valor de fitness idéntico al teórico. Estos resultados resaltan la capacidad del optimizador para enfrentar eficazmente inclusive las funciones más complejas y difíciles. Sin embargo, es importante destacar que algunas funciones específicas, tales como "CrossLegTable", "Zimmerman", "RosenbrockModified", "Kowalik", "Mishra03" y "NewFunction01", exhibieron un desempeño más modesto, con resultados donde el PCHO no logró alcanzar los niveles deseados. Estas fallas podrían proporcionar puntos de partida valiosos para la identificación de áreas con potencial para mejoras futuras.

#### 4.5 Sobre la eficiencia del SA

En forma análoga al proceso físico en el que SA está inspirado, el algoritmo simula el proceso de enfriamiento reduciendo gradualmente la temperatura del sistema hasta que converge a un estado de congelación estable. A medida que la temperatura se hace relativamente más fría, la pelota no puede rebotar tan alto, y también puede asentarse para quedar atrapada en rangos relativamente más pequeños de valles. Este problema puede subsanarse cuando otro trabajador del PCHO propone un nuevo punto de exploración. Sin embargo, antes de que eso ocurra sería conveniente mejorar la eficiencia y velocidad del SA.

Con este propósito, se recomienda implementar un método de Recocido Simulado Adaptivo (ASA: Adaptive Simulated Annealing) empleando las ideas subyacentes en varias propuestas explicadas en la literatura. Moins (2002) implementó una variante del ASA, donde las restricciones se incluyen en la función de costo. Para salir de un mínimo local, se acepta un aumento de dicha función con una cierta probabilidad. Dos parámetros importantes que rigen el programa de enfriamiento son el tamaño del paso para la perturbación y la temperatura. Por su parte, Grohall y Jung (2003) demostraron cómo los métodos numéricos tradicionales fallan en los problemas multipaso de optimización. Para superar esta desventaja, desarrollaron un algoritmo híbrido que utiliza GA y SA.

## 5 CONCLUSIONES

En este trabajo, se presenta un análisis de desempeño del PCHO ante una variedad de experimentos que permiten validar su comportamiento. Con los estudios realizados, en primer lugar se vislumbra la necesidad de implementar varias modificaciones específicas en el algoritmo PCHO. En primer lugar, sería conveniente incorporar una lista tabú para evitar el estancamiento. Para lograr mayor eficiencia del GA, también es aconsejable agregar un

criterio de parada eficaz con el objeto de evitar iteraciones infructuosas. Por último, se sugiere implementar un módulo auxiliar que aplique el método ASA, el cual será convocado en caso de estancamiento grave del SA.

Además, se pueden proponer otras versiones mejoradas del PCHO con mayor especificidad, es decir con habilidades para resolver problemas de optimización con formulaciones binarias o software especializado en problemas multiobjetivo. Para mejorar su rendimiento, el PCHO también puede hibridizarse con otros componentes estocásticos, incluyendo métodos de búsqueda local o global. Finalmente, la investigación de la utilización del PCHO en otras disciplinas diversas sería una valiosa contribución. Por ejemplo, emplearlo en aplicaciones de industria e ingeniería, otras funciones de prueba de referencia y otros problemas del mundo real.

## 6 REFERENCIAS

- Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M., Gandomi, A., The arithmetic optimization algorithm. *Computer methods in applied mechanics and eng*, 376, 113609, 2021.
- Biegler, L. T., Nonlinear programming: concepts, algorithms, and applications to chemical processes. *Society for Industrial and Applied Mathematics*, 2010.
- Burke E., Gendreau M., Hyde M., Kendall G., Ochoa G., Özcan E., Qu R., Hyper-heuristics: A survey of the state of the art. *J. Operational Res. Soc.*, 64 (12), 1695-1724, 2013.
- Cruz-Duarte, J. M., Ortiz-Bayliss, J. C., Amaya, I., MatHH: A Matlab-based Hyper-Heuristic framework. *SoftwareX*, 18, 101047, 2022.
- Drake, J. H., Kheiri, A., Özcan, E., Burke, E. K., Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, 285(2), 405-428, 2020.
- Fraser, G., and Arcuri, A., Evolutionary generation of whole test suites. In *2011 11<sup>th</sup> International Conference on Quality Software*, 31-40, IEEE, 2011.
- Gavana, A., Global optimization benchmarks and AMPGO. Accessed Apr. 2016.
- Goldreich, O., P, NP and NP Completeness. Cambridge, 2010.
- Grohall, G., and Jung, J. Multiple objective step function maximization with genetic algorithms and simulated annealing. 2003.
- Halim, A. H., Ismail, I., Das, S., Performance assessment of the metaheuristic optimization algorithms: an exhaustive review. *Artificial Intelligence Review*, 54, 2323-2409, 2021.
- Hussain, K., Salleh, M. N. M., Cheng, S., Naseem, R., Common benchmark functions for metaheuristic evaluation: A review. *JOIV: International Journal on Informatics Visualization*, 1(4-2), 218-223, 2017.
- Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A. M., Talbi, E. G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: *European Journal of Operational Research*, 296(2), 393-422, 2022.
- Layeb, A., Tangent search algorithm for solving optimization problems. *Neural Computing and Applications*, 34(11), 8853-8884, 2022.
- Luke, S., Essentials of Metaheuristics. Lulu, *Metaheuristics in large-scale global continues optimization: A survey*, available at: <http://cs.gmu.edu/~sean/book/metaheuristics>, 2013.
- Moins, S., Implementation of a simulated annealing algorithm for Matlab, 2002.
- Oteiza, P. P., Ardenghi, J. I., Brignole, N. B., Parallel hyper-heuristics for process engineering optimization. *Computers & Chemical Engineering*, 153, 107440, 2021.
- Phillips-Wren, G., AI tools in decision making support systems: a review. *International Journal on Artificial Intelligence Tools*, 21(02), 1240005, 2012.
- Wolpert, D. H., and Macready, W. G., No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67-82, 1997.