

**DESEMPEÑO DE ALGORITMOS HEURÍSTICOS EN LA
SOLUCIÓN DE PROBLEMAS CUADRÁTICOS NO CONVEXOS CON
RESTRICCIONES DE CAJA**

**PERFORMANCE OF HEURISTIC ALGORITHMS IN THE SOLUTION OF THE
NON-CONVEX QUADRATIC PROBLEM WITH BOX CONSTRAINTS**

PhD. Ridelio Miranda Pérez*, **Ing. Juan Manuel Castellanos Hernández****

* Departamento de Matemática, Facultad de Ciencias Económicas y Empresariales,
Universidad de Cienfuegos, Carretera de Rodas Km 4, Cienfuegos, Cuba.
rmiranda@ucf.edu.cu

** Departamento de Matemática, Facultad de Ciencias Económicas y Empresariales,
Universidad de Cienfuegos, Carretera de Rodas Km 4, Cienfuegos, Cuba.
jmcastellanos@yandex.com

Resumen: En el trabajo se proponen varios algoritmos heurísticos para resolver el problema cuadrático no convexo con restricciones de caja y se compara su rendimiento tomando en consideración la calidad de las soluciones y el tiempo de ejecución empleado. Adicionalmente, comparamos los resultados obtenidos con los algoritmos heurísticos con los algoritmos híbridos heurísticos, que consisten, una vez hallada la mejor solución del algoritmo heurístico, realizar una exploración exhaustiva de su vecindad, para determinar un mínimo local contenido en ella. Los resultados muestran un mejor desempeño de las metaheurísticas Enjambre de Partículas (PSO) y Algoritmos Genéticos (GA) en problemas de hasta 200 variables. Por último se muestra un mejor desempeño de las metaheurísticas híbridas en comparación con las no híbridas, aunque no poseen diferencias significativas entre ellas.

Palabras clave: programación cuadrática, algoritmos heurísticos, programación paramétrica, relación doblemente no negativa, métodos de continuación, optimización global.

Abstract: We present several heuristic algorithms to solve the non-convex quadratic problem with box constraints. The performance of these algorithms is compared taking into account the quality of the solutions and the execution time used to obtain it. Additionally, we compare the results obtained with the heuristic algorithms with the hybrid heuristic algorithms, which consist, once the best solution of the heuristic algorithm is found, to carry out an exhaustive exploration of its neighborhood, to determine a local minimum contained in it. The results show a better performance of the Particle Swarm metaheuristics (PSO) and Genetic Algorithms (GA) in problems up to 200 variables. Finally, a better performance of hybrid metaheuristics is shown compared to non-hybrid ones, although they do not have significant differences between them.

Keywords: quadratic programming, heuristic algorithms, parametric programming, doubly non-negative relaxation, continuation methods, global optimization.

1. INTRODUCCIÓN

Resulta incuestionable el hecho de que el uso de la metaheurísticas, muy de moda en nuestros días, en comparación con los métodos determinísticos, permiten la obtención de “buenas soluciones”, con un ahorro de tiempo considerable, sobre todo cuando se trata de resolver problemas de grandes dimensiones. Pero, también es conocido que el empleo de estas estrategias no garantiza la obtención de la solución óptima (por regla general no la obtienen). Y si en la práctica existen muchos casos en que tales “buenas soluciones” satisfacen las expectativas del cliente, también existen muchos otros en que este no es el caso. En este trabajo se estudia el comportamiento de algunos algoritmos heurísticos aplicados a la solución del problema cuadrático no convexo con restricciones de caja, cuya formulación estándar puede escribirse como sigue.

$$[\text{BQP}] \min\{f(x) = \frac{1}{2}x^T Qx + q^T x \mid x \in B\} \quad (1)$$

$$B = \{x \in \mathbb{R}^n \mid 0 \leq x \leq e\}$$

donde $Q \in \mathbb{R}^{n \times n}$ denota una matriz simétrica indefinida o definida negativa, $q \in \mathbb{R}^n$, y $e = (1, 1, \dots, 1)^T \in \mathbb{R}^n$.

Este problema resulta uno de los casos especiales más estudiados dentro de la programación cuadrática y uno de los problemas clásicos fundamentales de la programación no lineal, y de la optimización global (Horst, Pardalos, & Van Thoai, 2000).

El hecho de que la matriz Q es no semidefinida positiva, implica que el problema [BQP] es no convexo, que por su alta complejidad computacional pertenece a la clase de problemas NP-Duro (Vavasis, 1990).

Existen múltiples aplicaciones prácticas de (1), entre las que se cuentan: problemas de diseño del chip de Integración a Escala Muy Grande (VLSI) (Floudas & Visweswaran, 1995), en telecomunicaciones, el problema de máxima probabilidad de detección sobre el acceso múltiple por división de código (CDMA) (Peng Hui Tan, Rasmussen, & Teng Joon Lim, 2000; Peng Hui Tan, Rasmussen, & Lim, 2001). En la programación no lineal con la programación cuadrática secuencial (Izmailov & Solodov, 2016). En rastreo en línea de cientos de personas en escenarios con gran multitud (Dehghan & Shah, 2018), en el Control de Modelos Predictivos para la toma de decisiones bajo limitaciones físicas de plantas controladas (Otta, 2013). En problemas de superficies de respuesta (Asachi & Marandi, 2015; Iqbal, Nadeem, Sherazi, & Khan, 2015); y por su relación directa con el

problema cuadrático con variables 0-1, en la solución de problemas combinatorios, como por ejemplo, el particionamiento de grafos (Hager, James, & Hungerford, 2015).

Todo ello ha motivado la aparición y desarrollo de numerosas técnicas de optimización especialmente diseñadas para su solución. Entre los principales enfoques explorados se encuentran la utilización de procedimientos de la programación no lineal, tales como, métodos de conjuntos activos (Gill & Wong, 2015), Métodos de barrera (Krumke, 1994), y algoritmos de ramificación y acotación (Burer & Vandembussche, 2008; Chen & Burer, 2012, Miranda, Alonso, Cañedo, & Allende, 2018). La función *quadprog* del Toolbox de optimización en Matlab implementa un grupo de estas técnicas.

Todos estos algoritmos aseguran alcanzar el mínimo global del problema, o demuestran que no existe solución factible para este. Sin embargo, todos ellos resultan significativamente lentos, en muchos casos el tiempo de ejecución crece exponencialmente con las dimensiones del problema. Una alternativa muy utilizada para superar esta dificultad de los métodos exactos, es el empleo de algoritmos heurísticos, los cuales permiten la obtención de “buenas soluciones” con un considerable ahorro de tiempo en comparación con los métodos determinísticos, especialmente cuando se resuelven problemas de grandes dimensiones.

En este trabajo se investiga el comportamiento de algunas de las heurísticas más utilizadas, aplicadas ahora a la solución del problema de programación cuadrática no convexa con restricciones de caja (1). Al artículo está organizado de la siguiente forma.

2. METODOLOGIA

Dentro de la gran variedad de algoritmos heurísticos reportados en la literatura: algoritmos golosos (GRASP por sus siglas en inglés), búsqueda tabú (TS), recocido simulado (SA), Colonias de Hormigas (ACO), Enjambre de partículas (PSO), algoritmos genéticos (GA), etc. (Dorigo, Maniezzo, & Colomi, 1991; Feo & Resende, 1995; Glover, 1986), hemos seleccionados tres de ellas, recocido simulado, algoritmos genéticos y enjambre de partículas.

Esta selección está basada, en primer lugar, por su amplia utilización, en segundo lugar, buscando diversidad de enfoques, y tercero, por la disponibilidad de un código eficiente para su ejecución (Utilizaremos las implementaciones disponibles en el Toolbox de optimización de Matlab).

2.1. Recocido Simulado

El algoritmo Recocido Simulado (Simulated annealing) fue propuesto por Kirkpatrick en 1983 (Kirkpatrick, Gelatt, & Vecchi, 1983). El mismo consiste en una búsqueda aleatoria local que contiene una estrategia implícita para escapar de los mínimos locales encontrados. La idea original del algoritmo parte del proceso físico de enfriamiento de los metales. Para escapar de la solución encontrada, el algoritmo incluye una función de costo que acepta toda solución que mejore la calidad esta, pero también permite aceptar peores soluciones con cierta función de probabilidad. La probabilidad de aceptar nuevas soluciones está dada por la expresión $C + \exp(\Delta \max(T))$, donde C es una constante, $\Delta = \text{Nuevo Valor Objetivo} - \text{anterior Valor Objetivo}$, $T_0 =$ temperatura inicial, $T =$ temperatura actual, y $T = T_0 * 0.95^k$.

Cuando Δ y T son positivos, la probabilidad de aceptación está entre 0 y $\frac{1}{2}$. Otras temperaturas conducen a menores probabilidades de aceptación. A su vez, valores grandes de Δ también disminuyen la probabilidad de aceptación.

El algoritmo funciona de acuerdo a cierto esquema de enfriamiento controlado, haciendo descender el parámetro temperatura desde altos valores hasta valores muy bajos. Cuando la temperatura es suficientemente baja, el método se detiene entregando la mejor solución calculada hasta ese momento. Recocido Simulado es una de las heurísticas más efectivas, no solo en la solución de problemas combinatorios, sino también para problemas de optimización con variable continua (Connolly, 1990).

2.2. Algoritmos Genéticos

Algoritmos Genéticos (Genetic Algorithm) es una metaheurística de tipo poblacional desarrollada por John Holland en los 1960's. (Holland, 1975). GA se basa en el principio de evolución de las especies que actúa sobre una población de individuos sometidos a la selección natural en presencia de operadores como mutación y cruzamiento. Para la generación y reproducción de los individuos se define una función conveniente. En cada iteración se genera una nueva solución, a partir de la generación actual. Seguidamente se recombinan las mejores soluciones entre sí para formar una nueva solución. Finalmente, las nuevas soluciones, así generadas, sustituyen a las peores soluciones de la actual generación, para formar una nueva. Este procedimiento se repite iterativamente hasta que se satisface cierto criterio

de parada predeterminado, en cuyo caso el algoritmo entrega la mejor solución calculada. La efectividad de este algoritmo ha sido probada para numerosos problemas de optimización con variables continuas (Baghel, Agrawal, & Silakari, 2012).

2.3. Enjambre de Partículas

El algoritmo Enjambre de Partículas (Particle Swarm Optimization) fue formulado por Kennedy y Eberhart (Kennedy & Eberhart, 1995). El algoritmo (PSO) inicia creando las partículas iniciales (puntos de R^n), y asignándoles velocidades iniciales. Luego se evalúa la función objetivo en la localización de cada partícula y determina el mejor valor (valor mínimo de la Función Objetivo) y la mejor localización. Luego escoge nuevas velocidades, basado en la velocidad actual, la mejor localización individual de cada partícula y en la mejor localización de sus vecinos. A continuación actualiza iterativamente la localización de las partículas, la velocidad y los vecinos. La generación de las partículas y de sus velocidades se realiza empleando una distribución probabilística específica. Este procedimiento se repite hasta que se cumple el criterio de parada establecido. Los reportes de aplicación de esta metaheurística sobre problemas de optimización con variable continua evidencian un buen comportamiento (Mezura-Montes & Coello, 2011; Pedersen, 2010).

2.4. Formulación de una estrategia híbrida

Con el fin de obtener mejores soluciones a las reportadas por los algoritmos heurísticos seleccionados se implementó, adicionalmente, una estrategia híbrida para cada uno de ellos. Esta estrategia consiste en lo siguiente. Partiendo de la mejor solución detectada por el algoritmo heurístico, se realiza una exploración exhaustiva de su vecindad, para determinar un mínimo local contenido en ella, y con ello mejorar la calidad de la solución reportada inicialmente. Para la ejecución de la búsqueda local se utiliza la función quadprog contenida en el Toolbox de Optimización de Matlab.

2.5. Reportes de Aplicaciones

Desafortunadamente, en la revisión bibliográfica realizada, no hemos encontrado reportes de la aplicación de estas metaheurísticas sobre el problema (1) objeto de la investigación, aunque si aplicados a la solución de otros tipos de problemas cuadráticos (Ghandeshtani, Mollai, Seyedkashi, & Neshati, 2010; Kvasov & Mukhametzhanov, 2018; Laguna, 2000; Munapo & Kumar, 2015; Peltonen, 2015; Schlüter, Egea, & Banga, 2009). Este es un

elemento adicional que justifica la importancia de este trabajo.

3. RESULTADOS

En esta sección se describen los resultados de la experimentación computacional realizada para medir el rendimiento de las tres heurísticas seleccionadas en la solución del problema [BQP]. Para ello proponemos, inicialmente, comparar los resultados alcanzados tras la aplicación de las heurísticas, en su versión pura original, entre ellas, y luego comparar estos resultados con las soluciones óptimas de los problemas. Empleando la misma metodología se comparan luego los resultados de los procedimientos combinados con la búsqueda local a posteriori (algoritmos híbridos).

Para conducir la experimentación hemos seleccionado una librería de problemas pruebas de programación cuadrática con restricciones de caja ampliamente utilizada por los investigadores, disponible en: <http://dollar.biz.uiowa.edu/~sburer>.

La misma incluye las 54 instancias propuestas en (Vandenbussche & Nemhauser, 2005a) compuesta por problemas de hasta 60 variables (a los que referiremos aquí como *problemas básicos*), y 36 ejemplos que contienen problemas entre 70 y 100 variables (referidos como *problemas extendidos*), más los 9 problemas adicionales de 125 variables propuestos por Burer (Chen & Burer, 2012), referidos también como *extendidos*. Para un total de 99 problemas a resolver.

Para comparar el comportamiento de los algoritmos seleccionados se utilizarán dos parámetros fundamentales: calidad de las soluciones y tiempo de ejecución empleado. Como medida para la calidad de las soluciones, además del valor mínimo f^* calculado por los métodos heurísticos, usaremos también el gap-relativo

($gap = \left| \frac{f^* - f^{glob}}{f^{glob}} \right|$) entre la mejor solución f^* obtenida en cada caso y el mínimo global f^{glob} reportado en la librería.

Como se indicó antes, para la ejecución de los algoritmos se utilizaron las implementaciones de estos disponibles en el toolbox de la plataforma de Matlab.

3.1. Resultados obtenidos por las heurísticas

Como resultado de la aplicación de las metaheurísticas sobre la colección de problemas pruebas seleccionados se pudo comprobar que las metaheurísticas GA y PSO tienen un mejor desempeño que SA, pues independientemente de la

dimensión de los problemas (Básicos o Extendidos), los resultados alcanzados por estas resultan mejores. A su vez se puede verificar que las soluciones obtenidas por GA y SW se acercan bastante a las soluciones óptimas reportadas por la librería (ver figura-1)

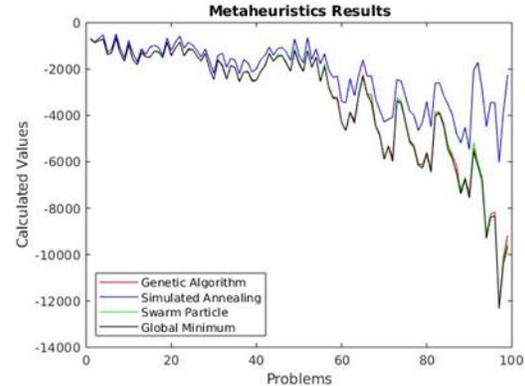


Fig. 1: Resultado de las Metaheurísticas aplicados en los problemas [BQP]

Con respecto a la calidad de las soluciones obtenidas, atendiendo al gap-relativo correspondiente, se observa que las metaheurísticas GA y PSO alcanzan muy buenas soluciones, con valores bien pequeños. En el caso de GA se tiene un gap-relativo promedio de 0.0158, en todos los casos menor que 0.09014, mientras que el gap-relativo para SW resulta en promedio 0.0117 e inferior, en todos los casos, a 0.065236. Por su parte, SA tiene un gap-relativo promedio de 0.26805 y su mayor valor es de 0.79825. (ver figura-2).

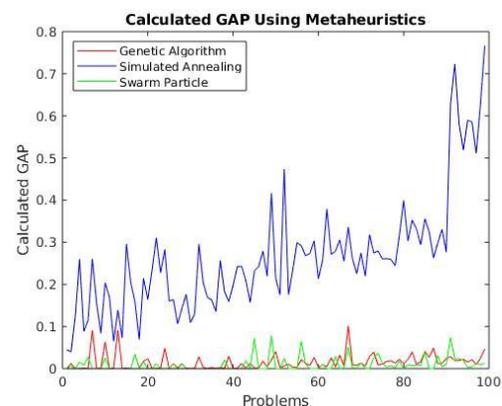


Fig. 2: Gap-relativo correspondiente a las 3 metaheurísticas para los problemas [BQP]

Comparando los gap-relativo de ambos algoritmos se encontró, aunque PSO es mejor que GA en 64 problemas, y GA es mejor que PSO en 35 de ellos, la diferencia entre ellos es bien pequeña, todo lo cual se puede apreciar en la gráfica log-log que se muestra en la figura-3.

Para comprobar si las diferencias entre GA y PSO son o no significativas, se aplica entonces una prueba de suma de rangos de Wilcoxon para muestras pareadas (Berry, Mielke & Johnston, 2012). Esta prueba (estadística no paramétrica) permite identificar si las muestras comparadas proceden de poblaciones con la misma distribución de probabilidad (hipótesis nula H_0). La hipótesis alternativa (H_1) por su parte, establece que hay diferencias significativas respecto a la tendencia central de las poblaciones.

Como resultado de la comparación de las muestras pareadas correspondientes a los resultados GA y PSO, se obtiene, que se existen evidencias suficientes para rechazar H_0 (valor- $p = 0,94$, con un nivel de significación $\alpha=0.05$). Este resultado permite concluir que no existe una diferencia significativa entre los valores del gap-relativo para ambas metaheurísticas. Sin embargo, al aplicar la misma prueba para comparar los valores mínimos calculados por esas metaheurísticas con los mínimos globales reportados en la librería, obtenemos que existe una diferencia significativa importante entre estos (en ambos casos con un valor- $p=5.45 \times 10^{-34}$, con $\alpha=0.05$). Este resultado evidencia que, aunque los resultados son muy buenos, tienen aun una diferencia significativa del valor mínimo global de los problemas resueltos.

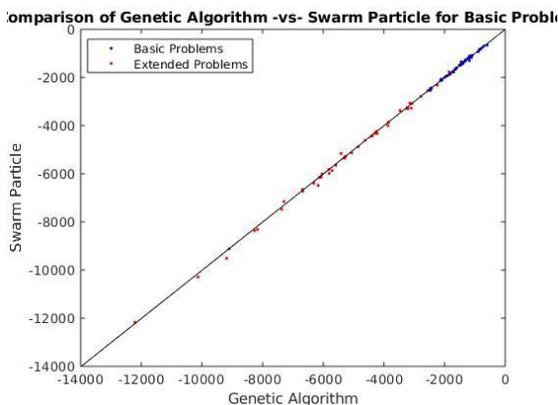


Fig. 3 Comparación de las metaheurísticas GA y PSO en cuanto a resultados alcanzados

3.2. Resultados obtenidos por las estrategias híbridas

Siguiendo la misma metodología del epígrafe anterior, comparamos ahora los resultados obtenidos cuando se aplican las estrategias híbridas diseñadas en el epígrafe 2.4.

En la figura-4 se observa un mejor desempeño en los algoritmos híbridos GA y PSO, no así con el algoritmo SA.

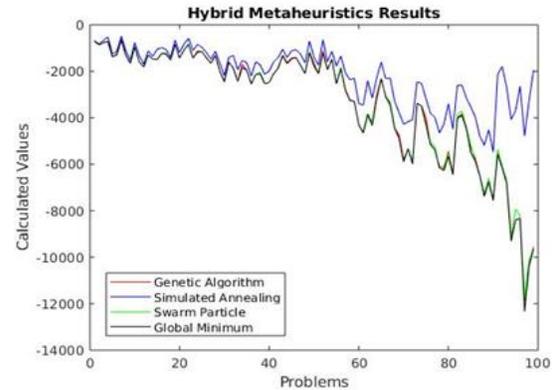


Fig. 4: Resultado de las Metaheurísticas híbridas aplicados en los problemas [BQP]

En la Tabla-1 se muestran tres indicadores con respecto a las tres metaheurísticas seleccionadas en el estudio. El primer indicador muestra el número de problemas [BQP] donde las metaheurísticas puras alcanzaron el mínimo global óptimo (NMG), el segundo, el número de problemas con mejoras al aplicar las metaheurísticas híbridas (NHM), y por último, el número de problemas en los que se mantuvo el mismo resultado al aplicar metaheurísticas híbridas respecto a las puras (NMHI).

Tabla 1. Mejora al emplear algoritmo híbrido

	GA	SA	PSO
NMG	0	0	16p*
NHM	99p	0	74p
NMHI	0	99p	25p

* p : problemas [BQP].

En la Figura 5 se observa un gap-relativo menor en las metaheurísticas híbridas GA y PSO con gap-relativo promedio de 0.01189 y 0.01042 respectivamente y valores máximos de 0.08787 y 0.05681 respectivamente. La metaheurística híbrida SA posee gap-relativo promedio de 0.26805 y su máximo valor alcanzado es de 0.79825. Se observa que todos los valores de gap-relativo promedio y máximo obtenidos son menores en la variante híbrida que en la pura.

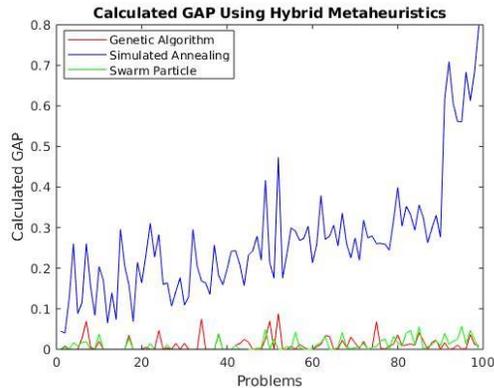


Fig. 5: Gap-relativo correspondiente a las 3 metaheurísticas híbridas para los problemas [BQP]

Comparando los gap-relativo de ambos algoritmos híbridos se encontró, aunque PSO es mejor que GA en 57 problemas, y GA es mejor que PSO en 42 de ellos, la diferencia entre ellos es bien pequeña, todo lo cual se puede apreciar en la gráfica log-log que se muestra en la figura-6.

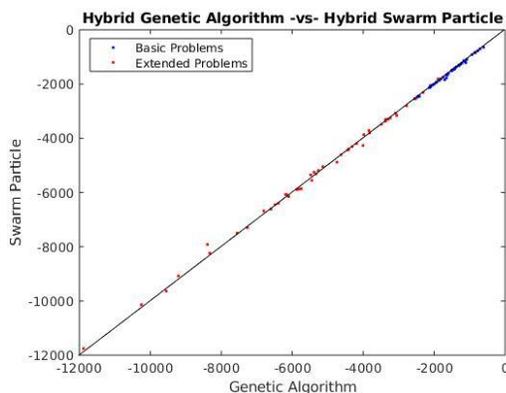


Fig. 6 Comparación de las metaheurísticas híbridas GA y PSO en cuanto a resultados alcanzados

Como resultado de la comparación de las muestras pareadas correspondientes a los resultados GA y PSO, se obtiene, que existen evidencias suficientes para rechazar H_0 (valor- $p = 0,97$, con un nivel de significación $\alpha=0.05$). Este resultado permite concluir que no existe una diferencia significativa entre los valores del gap-relativo para ambas metaheurísticas híbridas. Sin embargo, al aplicar la misma prueba para comparar los valores mínimos calculados por esas metaheurísticas híbridas con los mínimos globales reportados en la librería, obtenemos que existe una diferencia significativa importante entre estos (en ambos casos con un valor- $p=5.45 \times 10^{-34}$, con $\alpha=0.05$). Este resultado evidencia que, aunque los resultados son muy

buenos, son aun diferentes significativamente del valor mínimo global de los problemas resueltos.

Por último, la comparación de los resultados obtenidos en las metaheurísticas puras y las metaheurísticas híbridas, se obtiene, que existen evidencias suficientes para rechazar H_0 (GA: valor- $p=0,87$, PSO: valor- $p=0,92$, SA: valor- $p=1$, con un nivel de significación $\alpha=0.05$). Este resultado permite concluir que no existe una diferencia significativa entre los valores hallados por las metaheurísticas híbridas en relación a las puras en los tres algoritmos seleccionados en el estudio.

4. CONCLUSIONES

Los resultados de este estudio permiten inferir que los algoritmos Genéticos y Enjambre de Partículas al ser empleados para resolver problemas de programación cuadrática no convexa con restricciones de caja, alcanzan mejores resultados que la heurística Recocido Simulado. También se puede concluir que los resultados alcanzados GA y PSO son muy similares, con una diferencia estadísticamente insignificante. Sin embargo, la calidad de las soluciones calculadas por estos dos algoritmos distan aun de los valores mínimos absolutos de los problemas resueltos.

El estudio del desempeño de las estrategias híbridas (combinación de heurísticas con búsqueda local) indica que se obtienen mejores valores en la solución de problemas [BQP], pero no poseen diferencia significativa respecto a no aplicarlos. Consecuencia de estos resultados, podemos concluir que las metaheurísticas estudiadas permiten obtener “buenas soluciones”, pero no alcanzan la calidad requerida para ser aceptadas por clientes exigentes.

REFERENCIAS

- Asachi, M., & Marandi, R. (2015). Response surface modeling of lead and copper removal from oil field brine by potential sorption method. *Particulate Science and Technology*, 33(3), 290-300. Recuperado de [ResponseSurfaceModelingofLeadandCopperRemovalfromOilFieldBrinebyPotentialSorptionMethod.pdf](https://www.researchgate.net/publication/266666666/figure/fig1/AS:631746151111111/5a58c46e4cde266d588263f4/AS:5819).
- Baghel, M., Agrawal, S., & Silakari, S. (2012). Survey of metaheuristic algorithms for combinatorial optimization. *International Journal of Computer Applications*, 58(19). Recuperado de https://www.researchgate.net/profile/Mohamed_Mourad_Lafifi/post/What_is_the_metaheuristic_that_best_works_to_solve_permutation-based_combinatorial_optimization_problems/attachment/5a58c46e4cde266d588263f4/AS:5819

- 82691704832@1515766894417/download/Survey+of+Metaheuristic+Algorithms+for+Combinatorial+Optimization.pdf
- Berry, K., Mielke, P., & Johnston, J. (2012). Thetwo-samplerank-sumtest:early development. *Electronic Journal for History of Probability and Statistics*, 8. Burer, S., & Vandembussche, D. (2008). A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations. *Math. Program.*, 2 Ser. A(113), 259-282.
- Chen, J. (2010). Convex relaxations in nonconvex and applied optimization (University of Iowa). Recuperado de <https://ir.uiowa.edu/etd/654>
- Chen, J., & Burer, S. (2012). Globally solving nonconvex quadratic programming problems via completely positive programming. *Mathematical Programming Computation*, 4(1), 33-52. Recuperado de 79-283-1-PB.pdf.
- Connolly, D. T. (1990). An improved annealing scheme for the QAP. *European Journal of Operational Research*, 46(1), 93-100.
- Dehghan, A., & Shah, M. (2018). Binary Quadratic Programming for Online Tracking of Hundreds of People in Extremely Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3), 568-581. <https://doi.org/10.1109/TPAMI.2017.2687462>
- Dorigo, M., Maniezzo, V., & Coloni, A. (1991). Positive feedback as a search strategy.
- Feo, T. A., & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2), 109-133.
- Floudas, C. A., & Visweswaran, V. (1995). Quadratic optimization. En *Handbook of global optimization* (pp. 217-269). Springer. 10.1.1.55.4573.pdf.
- Ghandeshtani, K. S., Mollai, N., Seyedkashi, S., & Neshati, M. M. (2010). New simulated annealing algorithm for quadratic assignment problem. 87-92. Recuperado de https://www.researchgate.net/profile/S_M_Hossain_Seyedkashi/publication/265943043_New_Simulated_Annealing_Algorithm_for_Quadratic_Assignment_Problem/links/5524ee1b0cf2b123c5175e65.pdf
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5), 533-549.
- Hager, W. W., & Hungerford, J. T. (2015). Continuous quadratic programming formulations of optimization problems on graphs. *European Journal of Operational Research*, 240(2), 328-337. Recuperado de graph.pdf.
- Holland, J. H. (1975). Adaptation in natural and artificial systems Ann Arbor. The University of Michigan Press, 1, 975.
- Horst, R., Pardalos, P. M., & Van Thoai, N. (2000). Introduction to global optimization. Springer Science & Business Media.
- Iqbal, M. K., Nadeem, A., Sherazi, F., & Khan, R. A. (2015). Optimization of process parameters for kitchen waste composting by response surface methodology. *International Journal of Environmental Science and Technology*, 12(5), 1759-1768.
- Izmailov, A. F., & Solodov, M. V. (2016). Some new facts about sequential quadratic programming methods employing second derivatives. *Optimization Methods and Software*, 31(6), 1111-1131. Recuperado de izmsol15SQP-modH.pdf.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization (PSO). 1942-1948. Recuperado de http://www.cs.cmu.edu/~arielpro/15381f16/c_slides/781f16-26.pdf
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Krumke, S. (1994). Eine modifizierte barriermethode fur konvexe quadratische optimierungsprobleme. Unpublished master's thesis, University of Wurzburg, Germany. (Diplomthesis)
- Kvasov, D. E., & Mukhametzhano, M. S. (2018). Metaheuristic vs. Deterministic global optimization algorithms: The univariate case. *Applied Mathematics and Computation*, 318, 245-259.
- Laguna, M. (2000). Global optimization and metaheuristics. Recuperado de <https://pdfs.semanticscholar.org/d88b/17e53bcca18398cb9fb9d73da680d0ec26ab.pdf>
- Mezura-Montes, E., & Coello, C. A. C. (2011). Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation*, 1(4), 173-194.
- Miranda, R., Allende, S., Bouza, G., & Perez, B. (2015). Algoritmo de corte y continuación con ramificación y acotación para la solución del problema cuadrático con restricciones de caja. *Revista Ciencias Matemáticas*, 29, 1, 19-29.
- Miranda, R. Alonso, S. A., Cañedo, B. P., & Allende, G. B. (2018). DESEMPEÑO COMPUTACIONAL DE ESTRATEGIAS HÍBRIDAS PARA LA SOLUCIÓN DE PROBLEMAS CUADRÁTICOS NO CONVEXOS CON RESTRICCIONES DE CAJA. *Investigación Operacional*, 39(1), 42-53.

- Munapo, E., & Kumar, S. (2015). A New Heuristic for the Convex Quadratic Programming Problem. *American Journal of Operations Research*, 5(05), 373.
- Otta, B. P. (2013). Numerical algorithms for quadratic programming for approximated predictive control. Recuperado de https://support.dce.felk.cvut.cz/mediawiki/images/6/63/Dp_2013_otta_pavel.pdf
- Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001, 1551-3203.
- Peltonen, T. (2015). Comparative study of population-based metaheuristic methods in global optimization. Recuperado de <https://jyx.jyu.fi/bitstream/handle/123456789/46236/URN:NBN:fi:jyu-201506082229.pdf?sequence=1>
- Peng Hui Tan, Rasmussen, L. K., & Teng Joon Lim. (2000). Box-constrained maximum-likelihood detection in CDMA. 2000 International Zurich Seminar on Broadband Communications. Accessing, Transmission, Networking. Proceedings (Cat. No.00TH8475), 55-62. <https://doi.org/10.1109/IZSBC.2000.829228>
- Peng Hui Tan, Rasmussen, L. K., & Lim, T. J. (2001). Constrained maximum-likelihood detection in CDMA. *IEEE Transactions on Communications*, 49(1), 142-153. <https://doi.org/10.1109/26.898258>
- Schlüter, M., Egea, J. A., & Banga, J. R. (2009). Extended ant colony optimization for non-convex mixed integer nonlinear programming. *Computers & Operations Research*, 36(7), 2217-2229.
- Vavasis, S. A. (1990). Quadratic programming is in NP. *Information Processing Letters*, 36(2), 73-77.