

Ciencias de la Educación y Humanidades

UNA REVISIÓN Y CLASIFICACIÓN DE LOS ALGORITMOS PARA EL PROBLEMA SBPR (SORTING PERMUTATIONS BY PREFIX REVERSALS)

A REVIEW AND CLASSIFICATION OF THE ALGORITHMS FOR SBPR PROBLEM (SORTING PERMUTATIONS BY PREFIX REVERSALS)

Wilmer José Palacios López¹
Fernando José Hernández Gómez²

RESUMEN

El ordenamiento de permutaciones por reversión de prefijos (SPBR) es un clásico problema de optimización combinatoria cuyo objetivo es ordenar una permutación de n elementos invirtiendo los bloques más a la izquierda (prefijos) de dicha permutación. En términos técnicos, se busca transformar una permutación dada π en una permutación identidad llamada ι . Podría compararse, en magnitud, a una pila de tortitas que se debe reordenar con una espátula en cualquier punto, invirtiendo su orden para que, tras varias iteraciones, la pila quede ordenada. Sin embargo, desde 2012 se ha demostrado que este problema pertenece a la clase NP-Hard, lo que lo hace inviable en términos prácticos para la construcción de una solución óptima en tiempo polinómico o inferior. En los últimos años, la comunidad académica ha tratado de utilizar algoritmos heurísticos y metaheurísticos de aproximación que proporcionen mejores resultados y se acerquen al óptimo global. Durante el desarrollo de esta investigación se ha realizado y comparado el estado del arte del problema en estudio, se diseñarán e implementarán algoritmos heurísticos y metaheurísticos basados en criterios propuestos por la literatura revisada. Asimismo, se propondrá una familia de metaheurísticos inspiradas en la vida, cuyo rendimiento se espera que sea eficiente en la búsqueda de soluciones en grandes espacios de búsqueda. Para llevar a cabo la implementación de los algoritmos, se utilizará el lenguaje de programación Python, en combinación con la herramienta Google Colab, lo que permitirá un análisis rápido y eficiente de los algoritmos. Además, se revisaron los conceptos básicos relacionados con la teoría moderna de permutaciones y sus métodos elementales de ordenación. Se evaluaron distintos algoritmos heurísticos y metaheurísticos utilizando un conjunto de resultados obtenidos en la literatura científica de los últimos 10 años. Se compararon las ventajas y desventajas de cada método y se determinó cuál es el más adecuado en función del contexto del problema en cuestión.

PALABRAS CLAVE: ORDENAMIENTO DE PERMUTACIONES, ALGORITMOS HEURÍSTICOS, ALGORITMOS METAHEURÍSTICOS, OPTIMIZACIÓN COMBINATORIA.

¹ Máster en Matemática Aplicada, Estudiante de doctorado en Matemática Aplicada, Universidad Nacional Autónoma de Nicaragua, UNAN - Managua, FAREM Matagalpa. <https://orcid.org/0000-0002-9783-6271>, wilmer.palacios@unan.edu.ni

² Dr. en Matemática Aplicada, Docente de Keiser University y Universidad Nacional Autónoma de Nicaragua, UNAN - Managua. <https://orcid.org/0000-0002-5132-2769>, fernando.hernandez@keiseruniversity.edu

Ciencias de la Educación y Humanidades

ABSTRACT

The ordering of permutations by prefix reversal (SPBR) is a classical combinatorial optimization problem whose main objective is to order a permutation of n elements by reversing the leftmost blocks (prefixes) of that permutation. Technically, a given permutation π must be transformed into an identity permutation called ι . Similarly, it could be compared in magnitude to a stack of pancakes that must be rearranged with a spatula by inserting at any point and reversing their order so that after several iterations the stack is ordered. However, since 2012 it has been shown that this problem belongs to the NP Hard class, which makes it infeasible in practical terms to construct an optimal solution in polynomial time or lower. In fact, in recent years, the academic community has been trying to use heuristic and metaheuristic approximation algorithms that provide better results approaching the global optimum. In this research, heuristic and metaheuristic algorithms were designed and implemented based on the criteria proposed by the reviewed literature. In addition, a family of life-inspired metaheuristics was proposed, which have proven to be very efficient in finding solutions in large search spaces. The algorithms used during the implementation of this research work were written with Python, using the Google Colab tool. In addition, the basic concepts related to modern permutation theory and its elementary sorting methods were reviewed. The evaluation of the algorithms was performed using the set of results provided by the scientific literature in the last 10 years. The advantages and disadvantages of the use of the different proposed methods were shown and, ultimately, it is shown which of them will prove to be the most appropriate depending on the context in which the problem is placed.

KEYWORDS: SORTING PERMUTATIONS, HEURISTICS ALGORITHMS, METAHEURISTICS ALGORITHMS, COMBINATORIAL OPTIMIZATION.

INTRODUCCIÓN

Etimológicamente, las palabras meta y heurístico vienen de los vocablos griegos: meta que significa más allá o nivel superior y heurístico definido como el arte de descubrir o plantear nuevas soluciones. Usualmente se cita al Dr. Fred Glover como el creador del término en 1986 con su trabajo *Future paths for integer programming and links to artificial intelligence*.

En principio, las metaheurísticas más que algoritmos son un conjunto de técnicas o metodologías que permiten aproximar soluciones de problemas de optimización, que por naturaleza son considerados complejos. Un factor clave que permite diferenciar estas técnicas de las heurísticas es su carácter estocástico.

Por definición, si existe un algoritmo A que resuelve un problema A se tiene un conjunto de entradas I que corresponden respectivamente a un conjunto de salidas O , es decir, para cada entrada ik existe una salida ok . Sin embargo, el carácter estocástico de un algoritmo metaheurístico permite afirmar que para una entrada ik no necesariamente siempre existirá una misma salida ok .

De acuerdo con Osman Laporte (1996) El término metaheurística se define como “un proceso iterativo que guía una heurística subordinada, combinando de forma inteligente diferentes conceptos para explorar y explotar el espacio de búsqueda”(p. 21).

Ciencias de la Educación y Humanidades

Sin embargo, Du y Swamy (2016) lo define como:

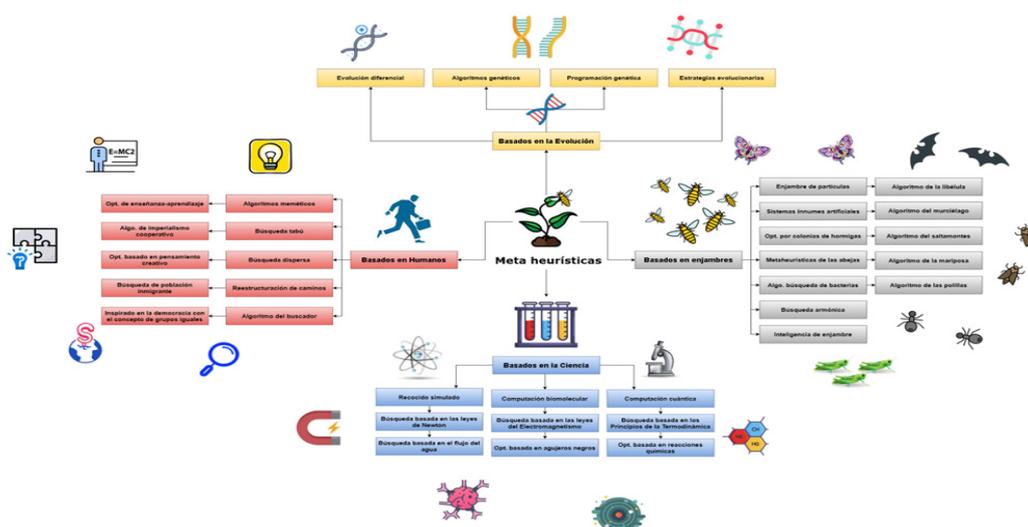
Un procedimiento o heurística de nivel superior diseñado para encontrar, generar o seleccionar un procedimiento o heurística de nivel inferior (algoritmo de búsqueda parcial) que pueda proporcionar una solución suficientemente buena a un problema de optimización. Al buscar en un gran conjunto de soluciones factibles, la metaheurística puede encontrar a menudo buenas soluciones con menos esfuerzo computacional que los métodos basados en el cálculo, o la heurísticas simples. (p.10)

En su defecto, casi cualquier planteamiento sistemático conjugado con una componente de aleatoriedad podría considerarse un algoritmo metaheurístico. Es decir, pueden existir multitud de estrategias diferentes (unas más formales que otras) que permitan aproximar una solución a la solución óptima para cualquier problema. Por esa razón, la literatura especializada ofrece un catálogo amplio de procedimientos que a menudo plantean nuevos puntos de vista sobre el problema a tratar.

En este sentido es importante destacar que existe una gran cantidad de meta heurísticas diseñadas para tratar conjuntos o familias de problemas en concreto. Dentro de este enorme espectro de ideas hay un subconjunto interesante de meta heurísticas que siguen comportamientos inspirados en la naturaleza. Todo esto, por supuesto, bajo la idea de que la propia naturaleza tiene una tendencia intrínseca a mejorar continuamente y adaptarse.

Uno de los principales exponentes de este movimiento es Yang y He (2019) cuyas compilaciones y publicaciones han inspirado a la comunidad a continuar desarrollando estos conceptos que han resultado ser claves para la resolución de problemas reales de enorme complejidad. En la figura 1 se muestra una clasificación generalizada muy compacta de las metaheurísticas basadas en la naturaleza, en el comportamiento humano y distintos hechos científicos.

Para efectos prácticos, en este estudio sólo se abordarán aquellos que gozan de una alta popularidad dentro de la comunidad, tales como: algoritmo genético, algoritmo memético y algoritmo de colonia de hormigas



Ciencias de la Educación y Humanidades

Algoritmos metaheurísticos

Desde que Bulteau et al. (2012) publicó su demostración sobre SBPR la dinámica para plantear posibles soluciones al problema cambió radicalmente. Muchos autores optaron por desarrollar soluciones heurísticas, ya que algorítmicamente es la opción que tiene más sentido y en la que se debería invertir un mayor esfuerzo. De esta forma, inició la era de las heurísticas y metaheurísticas para SBPR.

Antes de analizar algunos de los estudios más importantes, hay que aclarar que los trabajos de Zhongxi y Tao (2006), Sharmin et al. (2008) fueron excluidos, en parte por las razones explicadas en el apartado anterior y adicionalmente porque muchos de los trabajos posteriores mejoraron en gran medida sus propuestas. Por tal razón se consideró redundante brindarles un espacio adicional, por el mínimo valor que estos aportan.

En contraste, autores como Soncco Álvarez y Ayala Rincón (2013) podría considerarse como el más prolífico de todos. En esta ocasión, su aporte está relacionado con un algoritmo genético modificado capaz de eliminar puntos de interrupción, especialmente para permutaciones con signo. Para cualquier caso del problema en enfoque podría funcionar ya que uno de los objetivos es reducir dichos puntos de interrupción a cero lo más rápido posible y así obtener la permutación identidad deseada.

Por otro lado, también existen métodos alternativos en comparación con los algoritmos genéticos como aquellos basados en SI (hormigas, abejas, aves, murciélagos, libélulas, entre otros) cuyos procedimientos se han estandarizado con el paso del tiempo. Ante esto, Al Daoud (2014) propuso una solución mejorada con base en la optimización por colonia de hormigas que en su defecto necesita encontrar la secuencia de ordenamiento óptima haciendo uso nuevamente del concepto de puntos de interrupción. Este método propone el uso de la función de evaporación de feromonas, propuesta inicialmente por Dorigo et al. (1996).

Sin embargo, en cuanto a las implementaciones genéticas, Soncco Álvarez y Ayala Rincón (2014) presentaron una novedosa variación referente a algoritmos meméticos, esta vez haciendo un verdadero énfasis al tema de estudio de esta tesis. De hecho, esta será la única propuesta metaheurística directa que estudiará a fondo sobre el problema durante el desarrollo y análisis de algoritmos mediante programación. Es importante aclarar que este enfoque obtuvo mejores resultados en comparación con un algoritmo genético estándar e inclusive a su versión propia del autor utilizando computación paralela.

Como consecuencia, dicho estudio, desarrollado por da Silveira et al. (2017) fue excluido nuevamente por ser redundante, pero sobre todo por las dificultades técnicas que éste presentaba debido a que las especificaciones del algoritmo requieren de una red distribuida de potentes computadoras con elevados requerimientos de hardware. Sin embargo, este trabajo únicamente se limita únicamente al uso del plan gratuito de Google Colab, en el cual no está permitido tener un cómputo demasiado elevado ni realizar configuraciones avanzadas.

Algoritmos heurísticos

Suele afirmarse que los algoritmos metaheurísticos son más poderosos porque intentan cubrir y

Ciencias de la Educación y Humanidades

esquivar esos óptimos locales que muchas veces suelen ser problemáticos. Sin embargo, una técnica heurística correctamente planteada también suele ser muy poderosa.

En algunas ocasiones este tipo de propuestas suelen presentar lagunas o problemas con su tiempo de implementación. Tal fue el caso de Caprara et al. (1999) quienes desarrollaron su trabajo basados en una solución clásica dentro de la optimización combinatoria como es la generación de columnas utilizando branch-and-bound. A pesar de su enorme popularidad y efectividad, faltó muy poco para ser destronada por técnicas mucho más modernas.

Como respuesta a lo anterior, Helmert (2010) propuso el algoritmo de punto de referencia basado en conceptos provenientes de la AI (en español, inteligencia artificial) y el ML (en español, aprendizaje automático). Aunque, por desgracia, ha sido complicado encontrar el artículo original con sus novedosos resultados, debido a que la única versión que existe está incompleta. Fueron estas razones las que llevaron nuevamente a excluir ambos estudios tanto de su análisis como de su implementación.

Por su parte, Dias et al. (2014) diseñaron un marco de trabajo completo que consiste en aproximar soluciones para varios de los casos más populares de problemas de reordenamiento genómico entre ellos el SBPR. Ellos mismos llamaron a su algoritmo principal ventana corrediza (ver figura 2) debido a que se trata de un método de mejora continua, siguiendo una filosofía muy similar a un algoritmo greedy o voraz.

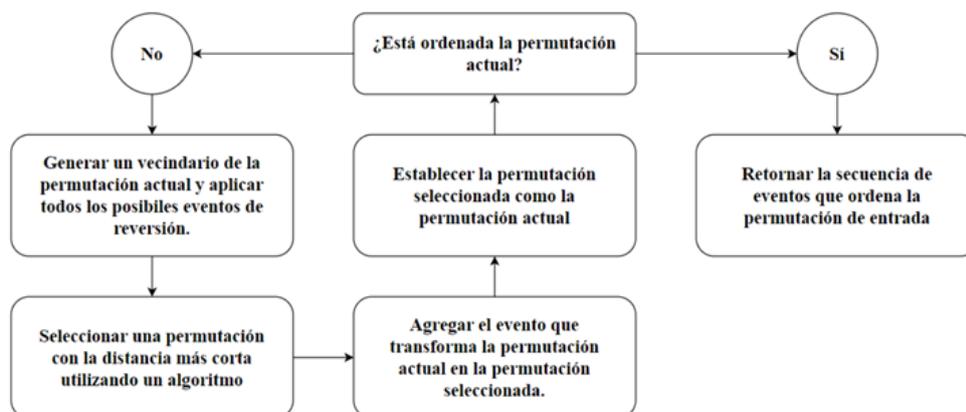


Figura 2: Diagrama de flujo de ventana corrediza. Fuente: Brito et al. (2019)

De forma similar, Brito et al. (2019) tomaron como base la propuesta y desarrollaron una nueva versión de la ventana corrediza junto a dos propuestas adicionales: Mirada hacia el futuro y Ventana Corrediza Iterativa. Entre otras cosas mostraron que, el primero de estos puede ejecutarse en tiempo lineal a diferencia de su primera versión. Además los otros también fueron diseñados para muy buenas soluciones con instancias del problema lo suficientemente grandes (permutaciones con 500 elementos).

Hasta la fecha de elaboración del presente trabajo no se tiene conocimiento de nuevas contribuciones y estudios relacionados con el tema de esta investigación. Por tal razón, se considera 2019 como el último año de actividad académica solo sustituido por el presente.

Ciencias de la Educación y Humanidades

De esta forma, en la tabla 1 se muestra a modo de resumen el conjunto de todos los algoritmos a implementar en el capítulo 4 y cuya importancia ha tenido un impacto significativo en el estado del arte del problema.

Autor y año	Tipo de algoritmo	Cod. Algoritmo
Soncco -Álvarez y Ayala -Rincón (2013)	MHA	soncco2013
Al Daoud (2014)	MHA	al2014
Soncco -Álvarez y Ayala -Rincón (2014)	MHA	soncco2014
Dias <i>et al.</i> (2014)	HA	dias2014
Brito <i>et al.</i> (2019)	HA	brito2019

Tabla 1: Algoritmos a implementar. Fuente: Elaboración propia

MATERIALES Y MÉTODOS

Para el desarrollo de esta investigación se ha empleado una metodología de análisis documental en base a la literatura técnica relacionada con el problema. Por otro lado, se ha realizado un análisis de los diferentes resultados obtenidos por los autores de la tabla 1 basados en los indicadores de rendimiento propuestos por Du y Swany (2016).

Criterio	Descripción	Ejemplo
Eficiencia	Evalúa la calidad de los resultados de un algoritmo sin tomar en cuenta su velocidad de ejecución.	La MBF es la media de la mejor aptitud en el último conjunto de todas las ejecuciones. De hecho, estos valores pueden utilizarse como una medida más absoluta de la eficacia.
Eficacia	Hace referencia a encontrar rápidamente la solución óptima global.	AES se define como el número medio de evaluaciones que tardan las ejecuciones exitosas. Si un algoritmo no tiene ejecuciones exitosas, su AES es indefinido.
Fiabilidad	Indica hasta qué punto el algoritmo puede proporcionar resultados aceptables	La tasa de éxito (SR) se define como el porcentaje de ejecuciones terminadas con éxito. Una ejecución exitosa se define como la diferencia entre el mejor valor de aptitud en la última generación f^* y un valor predefinido f_0 bajo un umbral predefinido ϵ .

Tabla 2: Indicadores de rendimiento para algoritmos. Fuente: Elaboración propia

Ciencias de la Educación y Humanidades

Para alcanzar los objetivos de este trabajo, como se muestra en la figura 3, se han tomado en cuenta algoritmos tanto heurísticos como metaheurísticos. De hecho el único criterio de selección ha sido que su solución se aplique al problema de estudio, ya que la cantidad de artículos en los últimos años ha disminuido. Inclusive, **al2014** estudió uno de los tantos casos del problema general, sin embargo se escogió por su facilidad de adaptación. Por otro lado, **soncco2014** es la única publicación que estudia de forma directa el problema.

Posterior al análisis de cada artículo se procedió a su reproducción y codificación en Python. Actualmente, el proceso de investigación se encuentra en esta etapa, tratando de desarrollar un marco de trabajo simple que permita luego comparar los resultados utilizando distintos parámetros, ya sean los valores estándar o los planteados por los autores en cada estudio de forma individual.

Finalmente, el análisis del rendimiento de los algoritmos consiste en comparar los resultados variando la cantidad de instancias y entradas, de tal manera que se pueda obtener información valiosa para analizar estadísticamente. En este punto de la metodología se va a demostrar cual de todos los algoritmos es el más óptimo en función de los datos analizados y la relación que existe con su complejidad computacional.

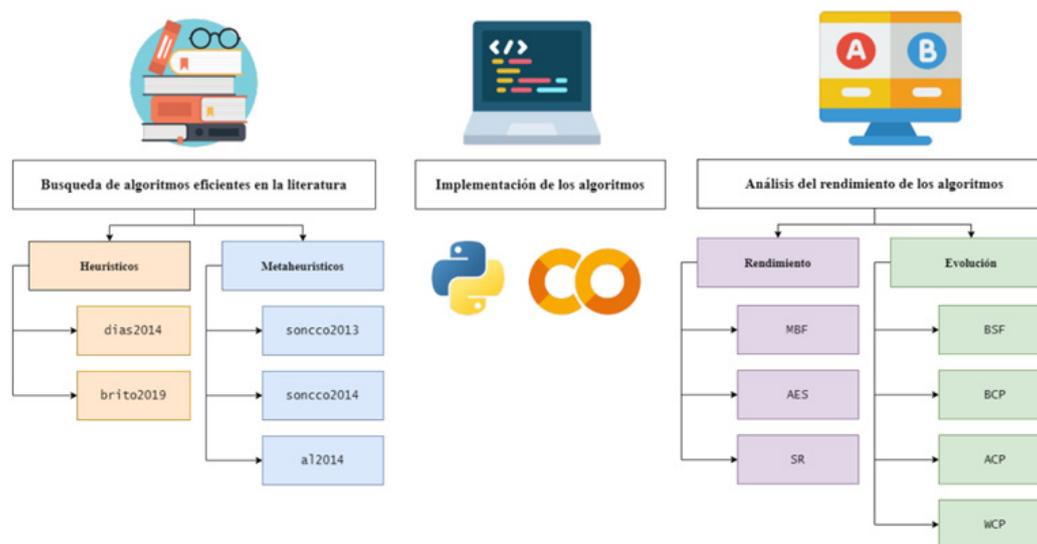


Figura 3: Resumen de la metodología aplicada. Fuente: Elaboración propia

RESULTADOS Y DISCUSIÓN

Es importante destacar el análisis individual de los resultados obtenidos por cada uno de los autores, que en términos generales ha sido diverso y poco estandarizado.

En el caso de la figura 4, Dias et al. (2014) obtuvo, en promedio, una LB (en español, cota inferior) eficiente al implementar el algoritmo de la ventana corrediza. Es importante notar que en su caso el tamaño de las permutaciones aleatorias usadas como muestra fue de 300 unidades.

Ciencias de la Educación y Humanidades

Por otro lado, en la figura 5 se observa un resultado similar con el algoritmo de búsqueda anticipada, aunque los parámetros son distintos. En este caso Brito et al. (2019) decidieron comparar basados en el longitud del radio de cada permutación, y no la distancia como en el problema original. Además, el tamaño de las permutaciones aleatorias utilizadas fue de 500 unidades.

En cuanto a los resultados del algoritmo genético, la figura 6 muestra como la versión mejorada propuesta por Soncco-Álvarez y Ayala-Rincón (2013) llamada iGA (en español, algoritmo genético mejorado) se ejecutó más rápido que los demás a pesar que el tamaño de las permutaciones apenas llegó a 160 unidades.

Luego, en la figura 7, Al Daoud (2014) mostró sus resultados en la versión mejorada del algoritmo de colonia de hormigas enfocándose en la comparación del radio y la distancia de cada permutación. En este estudio se utilizó un tamaño de permutaciones de 200 unidades.

Finalmente, en la figura 8 se muestran los resultados del algoritmo memético de Soncco-Álvarez y Ayala-Rincón (2014) fueron comparados con la eficiencia del algoritmo de paralelización, obteniendo casi la misma efectividad. Sin embargo, en este caso se utilizó un tamaño de permutaciones aleatorias de 150 unidades.

A pesar de los resultados anteriores, un parámetro valioso para analizar la eficiencia de cada algoritmo está relacionado con su complejidad computacional. En la figura 9 se muestran las funciones aproximadas que modelan la complejidad de estos algoritmos en función del tiempo que tardan en ejecutarse. El más destacado de todos es el de días2014 debido a que se ejecuta en tiempo lineal, aunque dicha linealidad depende de la complejidad de una base de datos diseñada especialmente para la ejecución de este algoritmo.

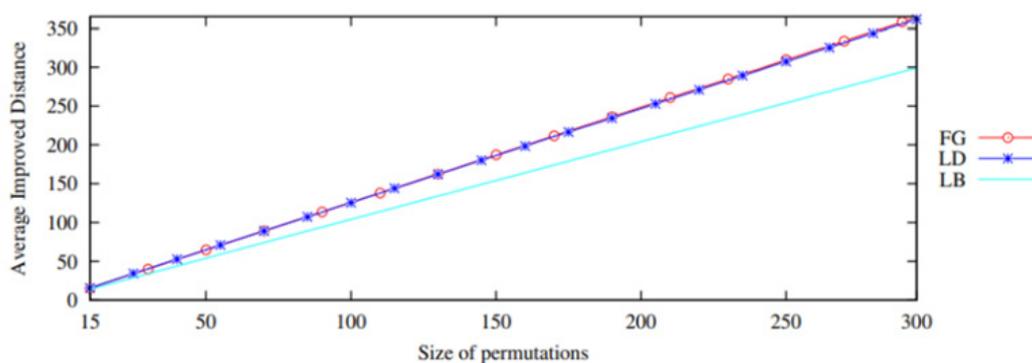


Figura 4: Resultados del algoritmo de ventana corrediza. Fuente: Dias et al. (2014)

Ciencias de la Educación y Humanidades

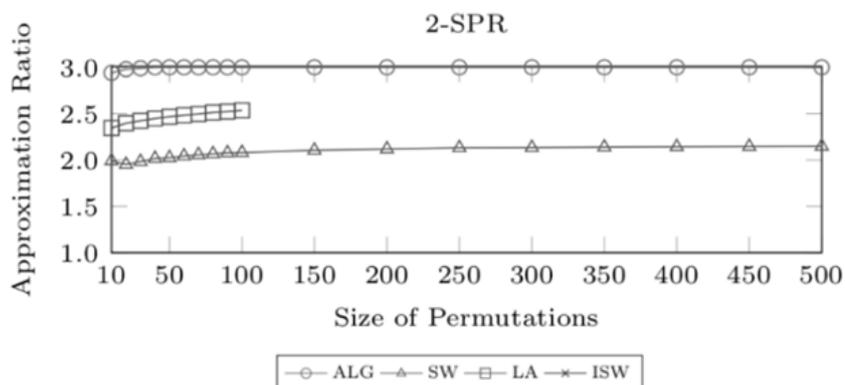


Figura 5: Resultados del algoritmo de búsqueda anticipada. Fuente: Brito et al. (2019)

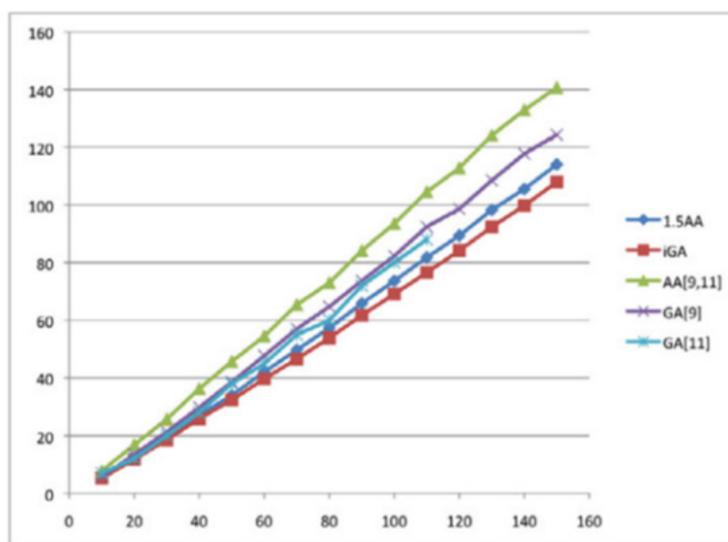


Figura 6: Resultados del algoritmo genético. Fuente: Soncco-Álvarez y Ayala-Rincón (2013)

No	$b(\pi)$	Exact $d(\pi)$	IBRS		New Ant Algorithm	
			$d(\pi)$	Ratio	$d(\pi)$	Ratio
1	113	81	140	1.73	91	1.12
2	122	84	133	1.58	102	1.21
3	105	74	112	1.52	105	1.43
4	90	66	101	1.53	77	1.17
5	107	71	108	1.52	79	1.11
6	98	72	116	1.61	85	1.18
7	118	78	123	1.58	93	1.19
8	114	81	130	1.60	88	1.09
9	94	65	102	1.57	85	1.31
10	102	72	108	1.50	79	1.10
Avg.	106.3	74.35	117.30	1.57	88.40	1.19

Ciencias de la Educación y Humanidades

Figura 7: Resultados del algoritmo de colonia de hormigas. Fuente: Al Daoud (2014)

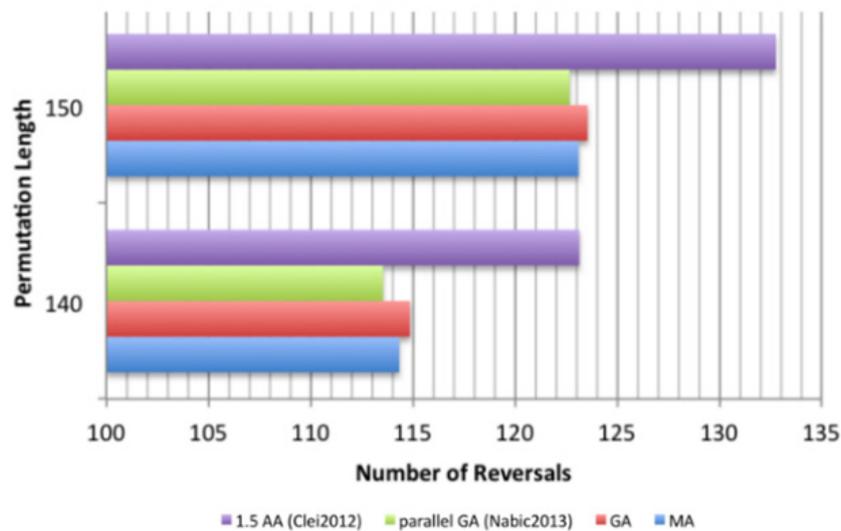


Figura 8: Resultados del algoritmo memético. Fuente: : Soncco-Álvarez y Ayala-Rincón (2014)

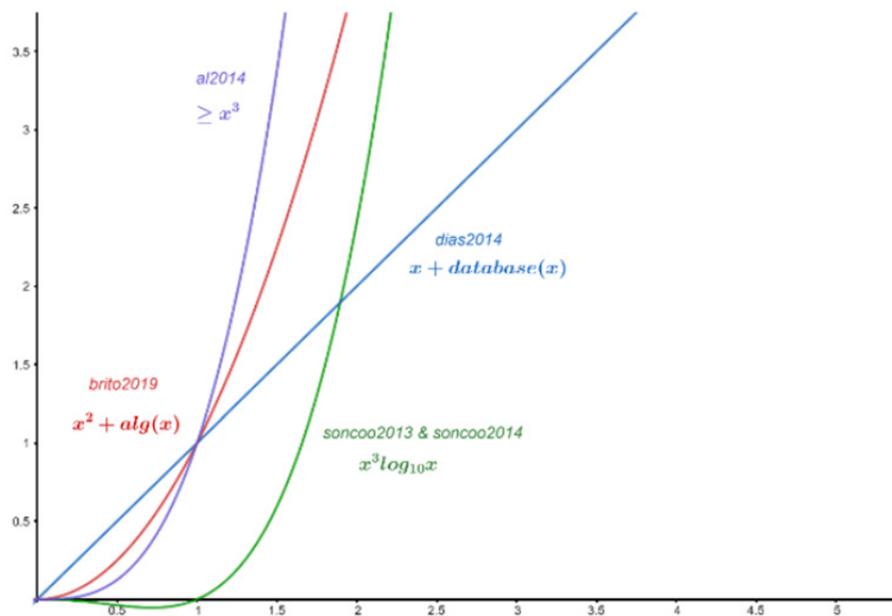


Figura 9: Comparativa de complejidad computacional de los algoritmos analizados. Fuente: Elaboración propia

Ciencias de la Educación y Humanidades

CONCLUSIONES

En este artículo se ha realizado una descripción teórica sobre las disciplinas matemáticas que han servido de base para plantear soluciones exactas, heurísticas y metaheurísticas al problema en cuestión. La teoría combinatoria clásica, álgebra lineal, la complejidad computacional y técnicas de análisis estadístico son tan solo algunas de las tantas herramientas utilizadas en este estudio.

Por otro lado, se tomaron los algoritmos heurísticos y metaheurísticos más importantes para realizar un análisis de su eficiencia. El algoritmo más destacado ha sido días2014 la heurística general planteada por Dias et al. (2014) y los algoritmos genéticos Soncco-Álvarez y Ayala-Rincón (2012) y Soncco-Álvarez y Ayala-Rincón (2013) que de igual forma han servido de base para continuar desarrollando nuevas propuestas metaheurísticas.

En este sentido, ha sido muy importante la utilización de un lenguaje de programación como Python para implementar tanto los algoritmos seleccionados como los resultados que estos producen de acuerdo a la cantidad de instancias evaluadas, el tamaño de las entradas y el tiempo de ejecución. Todo esto ha sido posible gracias a la herramienta de Google Colab la cual permite realizar todo esto desde el navegador agilizando en gran medida el proceso de programación y análisis debido a que posee un entorno completo de desarrollo completo de forma gratuita.

REFERENCIAS

- Al Daoud, E. (2014). An Improved Ant Colony Algorithm for Genome Rearrangements. *International Journal of Bioengineering and Life Sciences*, 8(5), 768–771.
- Brito, K. L., Oliveira, A. R., Dias, U., & Dias, Z. (2019). Heuristics for the reversal and transposition distance problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(1), 2–13.
- Bulteau, L., Fertin, G., & Rusu, I. (2012). Pancake flipping is hard. *Proceedings of the 37th International Conference on Mathematical Foundations of Computer Science*, 247–258.
- Caprara, A., Lancia, G., & Ng, S.-K. (1999). A column-generation based branch-and-bound algorithm for sorting by reversals (Vol. 47).
- da Silveira, L. A., Soncco-Álvarez, J. L., & Ayala-Rincón, M. (2017). Parallel genetic algorithms with sharing of individuals for sorting unsigned genomes by reversals. *2017 IEEE Congress on Evolutionary Computation (CEC)*, 741–748.
- Dias, U., Galvão, G. R., Lintzmayer, C. N., & Dias, Z. (2014). A general heuristic for genome rearrangement problems. *Journal of Bioinformatics and Computational Biology*, 12(03), 1450012.
- Du, K.-L., & Swamy, M. (2016). *Search and Optimization by Metaheuristics: Techniques and Algorithms Inspired by Nature*. Birkhäuser.
- Helmert, M. (2010). Landmark heuristics for the pancake problem. *Third Annual Symposium on Combinatorial Search*.

Ciencias de la Educación y Humanidades

- Okwu, M. O., & Tartibu, L. K. (2020). *Metaheuristic optimization: Nature-inspired algorithms swarm and computational intelligence, theory and applications* (Vol. 927). Springer Nature.
- Osman, I. H., & Kelly, J. P. (1997). Meta-heuristics theory and applications. *Journal of the Operational Research Society*, 48(6), 657–657.
- Sharmin, M., Yeasmin, R., & Hasan, M. (2008). Sorting by Prefix Reversals and Prefix Transpositions with Forward March. *ArXiv Preprint ArXiv:0812.3933*.
- Soncco-Álvarez, J. L., & Ayala-Rincón, M. (2013). Sorting permutations by reversals through a hybrid genetic algorithm based on breakpoint elimination and exact solutions for signed permutations. *Electronic Notes in Theoretical Computer Science*, 292, 119–133.
13. Soncco-Álvarez, J. L., & Ayala-Rincón, M. (2014). Memetic algorithm for sorting unsigned permutations by reversals. *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2770–2777.
- Yang, X.-S., & He, X.-S. (2019). *Mathematical foundations of nature-inspired algorithms*. Springer.
- Zhongxi, M., & Tao, Z. (2006). An improved genetic algorithm for problem of genome rearrangement. *Wuhan University Journal of Natural Sciences*, 11(3), 498–502.