Revista Peruana de Computación y Sistemas 2023 5(1):65-81
https://doi.org/10.15381/rpcs.v5i1.25806

ISSN Online: 2617-2003
Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos

Artículo de Contribución

Heurística de Intercambio 2Opt Best Improvement_r y Nivel de Eficacia de las soluciones del Problema del Agente Viajero Simétrico

Heuristic Exchange 2Opt Best Improvement_r and Level of Efficacy of the solutions of the Symmetric Travelling Salesman Problem

David J. Astoquillca-Yaranga[1,a], Esther Berger-Vidal[1,b]

[1] Universidad Nacional Mayor de San Marcos, Facultad de Ciencias Matemáticas, Unidad de Posgrado. Lima, Perú

[a] Autor de correspondencia: david.astoquillca@unmsm.edu.pe, ORCID: https://orcid.org/0009-0001-7027-7238

[b] E-mail: ebergerv@unmsm.edu.pe, ORCID: https://orcid.org/0000-0001-5282-6793

Resumen

En esta investigación se desarrolló un algoritmo híbrido denominado VMC_2OptBI_r, que a partir de la solución inicial construida bajo el pensamiento del Vecino más Cercano se buscó mejorar el criterio de búsqueda de soluciones aplicado por la Heurística de Intercambio 2Opt basada en una política de intercambio de nodos denominada Best Improvement (BI) e introduciendo un factor adicional "1+r" al criterio para realizar los intercambios 2Opt. El factor "1+r" permite modificar ligeramente la selección de nodos a intercambiar provocando así la exploración de nuevas soluciones. Para medir el nivel de eficacia del nuevo algoritmo se seleccionaron instancias del problema del agente viajero simétrico de TSPLIB, las cuales en primer lugar se compararon con sus versiones básicas: El Vecino más Cercano (VMC) y el VMC_2OptBI; luego se compararon con las soluciones de los algoritmos SCA_2Opt, SCA_2Opt_r, donde se comparó con las soluciones de cuatro metaheurísticas publicadas en artículos recientes (2019-2022). Los resultados mostraron que las soluciones obtenidas por el nuevo algoritmo VMC_2OptBI_r alcanzaron un nivel de eficacia de 100% con respecto al VMC, VMC_2OptBI, SCA_2Opt, un valor mayor al 81% con respecto al SCA_2Opt_r y un rango entre el 11% al 88% con respecto a las cuatro metaheurísticas revisadas para las instancias comparadas.

Palabras clave: Problema del Agente Viajero Simétrico, Intercambio 2Opt, Vecino más Cercano, Metaheurísticas.

Abstract

In this research, a hybrid algorithm called VMC_2OptBI_r was developed, which improved the solution search criteria applied by the 2Opt Exchange Heuristic based on a node exchange policy called Best Improvement (BI) and introducing an additional factor "1+r" to perform the 2Opt exchanges. The "1+r" factor allows slightly modifying the selection of nodes to be exchanged, thus exploring new solutions. To measure the level of efficacy of the new algorithm, instances of the Symmetric Traveling Salesman Problem from TSPLIB were selected and compared with their basic versions: Nearest Neighbor (VMC) and VMC_2OptBI. Then, they were compared with the solutions of the SCA_2Opt and SCA_2Opt_r algorithms, where they were compared with the solutions of four metaheuristics published in recent articles (2019-2022). The results showed that the solutions obtained by the new VMC_2OptBI_r algorithm achieved an efficacy level of 100% compared to VMC, VMC_2OptBI, SCA_2Opt, a value higher than 81% compared to SCA_2Opt_r, and a range between 11% to 88% compared to the four reviewed metaheuristics for the compared instances.

Keywords: Symmetric Travelling Salesman Problem, 2Opt Exchange, Nearest Neighbor, Metaheuristics.

Recibido: 30/04/2023 - Aceptado: 04/06/2023 - Publicado: 30/06/2023

Citar como:
Astoquillca-Yaranga, D. & Berger-Vidal, E. (2023). Heurística de Intercambio 2Opt Best Improvement_r y Nivel de Eficacia de las soluciones del Problema del Agente Viajero Simétrico. Revista Peruana de Computación y Sistemas, 5(1):65-81. https://doi.org/10.15381/rpcs.v5i1.25806

1. Introducción

Debido al desarrollo de las capacidades de procesamiento de los computadores es posible resolver problemas complejos por medio de algoritmos exactos, sin embargo, a pesar de dicho desarrollo computacional aún existen problemas para los cuales los procesadores actuales no logran encontrar soluciones en tiempos razonables, entre ellos está el problema del agente viajero el cual según las características de cada caso presenta variantes tal como el problema del agente viajero simétrico para el cual el costo de ir del nodo "a" al nodo "b" es igual al de ir del nodo "b" al nodo "a" para a ≠ b, Moroyoqui, Orozco y Picos [1]

El problema del agente viajero al cual en adelante denominaremos como TSP (Travel Salesman Problem) es considerado según López, Salas y Murillo [2] como uno de los problemas más conocidos en la planificación de rutas. Este problema es estudiado por distintas disciplinas entre ellas la Investigación de Operaciones que para resolverlos usa con frecuencia algoritmos heurísticos.

Al revisar las bases de datos bibliográficas Web of Science y Scopus, se evidencia que en los últimos diez años ha existido una alta preferencia por realizar investigaciones aplicando metaheurísticas tales como los algoritmos genéticos, y que heurísticas como el Vecino más Cercano y el Intercambio 2Opt han sido poco estudiadas, tal como se observa en la Tabla 1.

Tabla 1

Cantidad de Artículos en Web of Science y Scopus 2014-2023

Palabras clave buscada	Web of Science	%	Scopus	%
- Genetic Algorithms	81,713	49.8%	111,212	45.1%
- Heuristic Algorithms	38,460	23.4%	66,732	27.1%
- Metaheuristic Algorithms	10,048	6.1%	15,658	6.4%
- Greedy Algorithm	9,473	5.8%	17,744	7.2%
- Ants Colony Algorithm	6,989	4.3%	12,183	4.9%
- Simulated Annealing Algorithm	6,932	4.2%	10,201	4.1%
- Bee Colony Algorithm	5,682	3.5%	7,635	3.1%
- Particle Colony Algorithm	3,946	2.4%	3,681	1.5%
- Nearest Neighbor Heuristic	457	0.3%	829	0.3%
- 2-opt Heuristic o 2-opt	315	0.2%	554	0.2%

Nota. Con data de Web of Science y Scopus al 25.05.2023

La Tabla 1 muestra la baja cantidad de investigaciones sobre la heurística de Intercambio 2Opt, la cual mejora una ruta construida previamente por otro algoritmo. Esa heurística es considerada de mejora debido a que realiza una exploración local buscando mejorar la ruta en vecindarios próximos a la ruta inicial. En las investigaciones revisadas se observó pocos casos donde se busca mejorar el diseño del Intercambio 2Opt.

Esta problemática fue la motivación para estudiar el algoritmo heurístico Intercambio 2Opt y buscar una forma de mejorar su Nivel de Eficacia, definido como: el cociente entre la cantidad de instancias donde la solución de nuestro algoritmo logró superar a la solución obtenida por otro algoritmo con el cual se comparó, entre el total de instancias revisadas. Entendiendo como instancia a la especificación de los parámetros de un problema para lograr diferenciarlo de otros.

Por ello, se buscó un factor que permita alterar la dirección por la cual se realiza la búsqueda dentro del espacio de soluciones y facilite moverse ligeramente dentro del vecindario, denominado factor de perturbación, con lo cual se buscó explorar a mayor profundidad el espacio de soluciones factibles

2. Marco Teórico

2.1. Revisión de la literatura

Levin y Yovel [3] realizaron una modificación a la función que guía el intercambio 2Opt, algoritmo sobre el cual trataremos en el ítem 2.2.5., la cual es expresada por medio de la función $f(C(x)) = (C(x))^r$, donde $r \in [0;2]$ y C(x) representa la función costo usada en el criterio del 2Opt, de tal manera que cuando r=1 se estaría trabajando con el criterio original del intercambio 2Opt, donde para validar la mejora producida se comparó las soluciones obtenidas por esta variante con las originales de tal manera que las soluciones obtenidas por el (2,f)-Opt superaron significativamente a las soluciones del 2Opt.

Pérez y Jaramillo en 2014 [4] presentaron una modificación al pensamiento del algoritmo del Vecino más Cercano, el cual busca ir a la ciudad más próxima en cada iteración, introduciendo el término Sacrificio Cortoplacista Adaptativo, que consiste en que estando en una determinada ciudad sacrificar ir a la ciudad más cercana para elegir ir a la segunda más cercana, realizado esto, se continúa con el pensamiento del vecino más cercano para terminar de construir la ruta inicial que será mejorada por la heurística 2Opt, introduciendo el factor "r" de Pérez/ $r \in [0,1]$, y seguir iterando con un próximo nodo para aplicar el sacrificio cortoplacista y continuar actualizando la mejor ruta, es decir, una búsqueda tipo trayectoria. A ese algoritmo lo denominó SCA_2Opt_r.

Hansen, Mladenović, Todosijević y Hanafi [5] reúnen variantes de la heurística Variable NeighborHood Search (VNS), las cuales trabajan con diferentes vecindades para realizar la búsqueda del óptimo evitando caer en óptimos locales por medio de una fase denominada perturbación la cual realiza el movimiento a un determinado vecindario según un orden preestablecido inicialmente o de forma cíclica. También diferencia entre *la primera mejora* como la primera iteración que logra mejorar la ruta actual y *la mejor de las mejoras* como la ruta que logra mejorar a todas las mejoras en un vecindario o en un conjunto de iteraciones.

Da Costa, Rhuggenaath, Zhang y Akcay [6], indican que la mayoría de los enfoques para las heurísticas se

centran en las de construcción, las cuales se denominan así porque en cada iteración agregan un elemento a la solución hasta completarla.

2.2. Fundamentos Teóricos

2.2.1. Grafo: Es un conjunto de puntos, denominados vértices o nodos (V) y un conjunto de líneas que unen pares de nodos, y representan relaciones orientadas o no orientadas entre ellos. En el problema del agente viajero los nodos representan ciudades, una arista representa la existencia de una vía de comunicación en ambos sentidos entre un par de ciudades y un arco una vía de comunicación en el sentido del arco. Formalmente un grafo se define como una estructura matemática G = (V, E) donde $V \neq \emptyset$, (Zuñiga [7]).

Para diferenciar un grafo de otro surge el concepto de instancia de un problema (Zuñiga [7]) el cual consiste en un grafo G =(V, E) donde $V \neq \emptyset$ y una cota k e Z, con k ≤ orden de G. Es decir, las instancias corresponden a las características que diferencian un grafo de otro, tales como el orden del grafo, entre otras.

Un grafo G=(V,E) puede representarse gráficamente como también matricialmente. Una representación es a través de la matriz de costos, cuyos elementos C_{ij} indican el costo de ir desde un nodo "i" hasta un nodo "j".

2.2.2. El Problema del Agente Viajero: Es según Anaya-Fuentes [8] uno de los problemas más difíciles de resolver, teniendo amplias aplicaciones en la industria logística. Para presentar el TSP, utilizamos la definición de Da Costa et al. [6], que indica que el problema consiste en que dado un conjunto de nodos, que representan ciudades, y un conjunto de aristas o arcos, que representan caminos entre los nodos, se pide determinar el recorrido más corto para visitar una sola vez cada una de las ciudades y retornar a la ciudad inicial.

Guo, Hou y Ye [9], resaltan que el TSP, es un problema NP-Hard, debido a que presenta una complejidad O(n!). Esto significa que conforme aumenta el número de nodos aumenta la dificultad para resolverlo. Mientras que Thirugnanasambandam, Raghav, Saravanan, Prabu y Rajeswari [10] señalan que la primera formulación matemática para el TSP fue estudiada en 1930 por Karl Menger y posteriormente Hassler Whitney introdujo el término TSP.

Anaya-Fuentes [8] refiere que los primeros en intentar resolverlo fueron Dantzing, Fulkerson y Johnson en 1954 por medio de un algoritmo basado en ramificación y acotamiento percatándose que a medida que crecía el tamaño del problema crecía el tiempo para encontrar la solución, por ello surgieron métodos que encuentran una buena solución, no necesariamente óptima, en un tiempo aceptable.

El TSP, según el valor numérico que tomen las aristas, puede clasificarse como un TSP simétrico, cuando el costo de ir del nodo "i" al nodo "j" es igual al costo de ir del nodo "j" al nodo "i", para i ≠ j; y es considerado TSP asimétrico cuando para ir de i a j, i ≠ j ese costo es diferente.

2.2.3. Clasificación de los métodos de resolución del TSP: Utilizamos la clasificación indicada por Astoquillca [11], basada en Pérez De La Cruz [12] adicionando a las metaheurísticas indicadas por Glover [13]:

- *Los métodos exactos*. Estos métodos garantizan una solución óptima, sin embargo se presentan casos donde al crecer el número de nodos puede ocurrir que el tiempo para encontrar el óptimo se haga tan elevado que su aplicación sea inviable (Pérez de la Cruz [12]).
- *Los métodos heurísticos*. Cockbaine y Silva [14] refieren que cuando el tamaño de la instancia es muy grande y no puede resolverse por un algoritmo exacto se hace uso de los algoritmos heurísticos, que se acercan a la solución óptima en tiempo polinomial. Nilson (2003), citado por López [15] menciona que estos algoritmos resuelven el problema utilizando el sentido común, encontrando soluciones aproximadas al óptimo. Entre los métodos heurísticos más conocidos según Maguiña [16] está el de Inserción, el del Vecino más cercano entre otros.
- *Los métodos metaheurísticos*. Término utilizado por Glover [13], son variantes de los métodos heurísticos que agregan estrategias inteligentes para evitar quedar atrapados en óptimos locales como suele ocurrir con las heurísticas. Entre las más conocidas según Maguiña [16] están: el Recocido Simulado, Búsqueda Tabú, Colonia de Hormigas, entre otras.

Du, Liu, Zhang y Lu [17] refieren que los algoritmos metaheurísticos presentan la ventaja de no requerir una formulación matemática precisa y presentan aplicaciones en casi todos los campos de la ingeniería.

Guo, Hou y Ye [9], consideran a las heurísticas y metaheurísticas como un mismo grupo denominándolo heurísticas inteligentes al cual dividen en tres categorías:

- *De construcción*, las cuales a través de un proceso de análisis según criterios definidos construyen una ruta, como ejemplo el algoritmo del Vecino más Cercano.
- *De Mejoramiento*, los cuales buscan mejores soluciones en vecindades cercanas a la solución inicial proporcionada por otro algoritmo, razón por la cual también se conocen como Algoritmos de Búsqueda Local (Local Search), como por ejemplo el algoritmo Intercambio 2Opt.
- *Híbridas*, son el resultado de combinar criterios de búsqueda de dos o más heurísticas o

metaheurísticas, lo cual incluye un algoritmo de construcción de rutas y uno de mejoramiento, por ejemplo, el VMC_2Opt, que usa el VMC para construir una ruta y el Intercambio 2Opt para mejorarla.

2.2.4. Heurística del Vecino más Cercano (VMC): Considerada por Hoto, Bressany Rodrigues [18] como una heurística constructiva codiciosa, la cual como indican Pérez y Jaramillo [4] tiene el pensamiento de siempre tomar la mejor decisión en cada iteración al elegir el nodo más próximo posible al actual y así ir construyendo una ruta, de tal manera que el proceso termina cuando ya no se tiene más nodos por visitar.

Hoto, Bressan y Rodrigues [18] indican que sucede en muchos casos que para los últimos nodos a seleccionar, la heurística se ve obligada a elegir nodos distantes, debido a ello, la solución no resulta ser muy buena, razón por la cual Pérez y Jaramillo [4] la consideran una heurística miope al preocuparse solo en el corto plazo y buscar siempre el nodo más cercano sin pensar que eso puede ser perjudicial en la ruta finalmente construida al inhabilitar arcos que podrían representar menores costos en la selección final de nodos para la ruta.

Lima, Araújo y Schimit [19], indican que el algoritmo del Vecino más Cercano es sencillo de implementar y el tiempo de ejecución para encontrar una solución es bastante aceptable, pero por su naturaleza codiciosa puede dejar de lado rutas más cortas.

2.2.5. Heurística de Intercambio 2Opt: Considerada por Hougardy Zaiser y Zhong [20] como uno de los algoritmos más simples que es capaz de encontrar buenas soluciones al problema del agente viajero cuya solución que ya no puede ser mejorada es denominada 2-Óptimo.

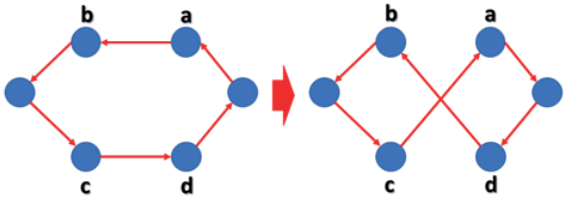
Como indican Schulz, Scarpin y de Souza [21] consiste en remover dos arcos de una solución dada e intercambiarlos así: el inicio del primer arco se conecta con el inicio del segundo y el nodo final del segundo arco se conecta con el nodo final del primero en búsqueda de reducir el costo de la solución actual. El algoritmo se detiene cuando ya no se pueden realizar más intercambios que mejoren la ruta. La versión generalizada consiste en realizar "k" intercambios.

Gutiérrez [22] indica que el algoritmo 2Opt intercambia arcos que no necesariamente mejoran la ruta, pudiendo empeorar, pero conforme se siga iterando la solución final sí mostrará la mejoría buscada.

Por último, el algoritmo de Intercambio 2Opt es una heurística de mejoramiento porque con base en una solución inicial, generada por otro algoritmo, busca mejorar la solución moviéndose en vecindarios cercanos a la última ruta obtenida. En la Figura 1 se muestra el funcionamiento de los Intercambios 2Opt para un ejemplo dado por Levin y Yovel [3].

Figura 1

Gráfico de Aplicación de Intercambio 2Opt



En el intercambio 2Opt se sustituyen los arcos (a,b) y (c,d) por (c,a) y (d,b) respectivamente solo si la nueva ruta representa una mejoría con respecto a la actual, por lo cual es necesario utilizar el criterio de intercambio de arcos en función del costo de la ruta.

Existen variantes del Intercambio 2Opt presentadas por diversos autores, entre ellas mostraremos la indicada por Pérez y Jaramillo [4] que trabaja con TSP's simétricos, la cual se utiliza en la presente investigación, donde el intercambio de los arcos se realiza como se muestra a continuación: (a, b) y (c, d) por (a, c) y (b, d). Para esa versión se usa el siguiente criterio: dada "C" una función que representa el costo de una ruta, el intercambio de nodos (a, c) y (b, d) será viable siempre y cuando:

$$[C(a,b) + C(b,d)] - [C(a,b) + C(c,d)] < Dmax \quad (1)$$

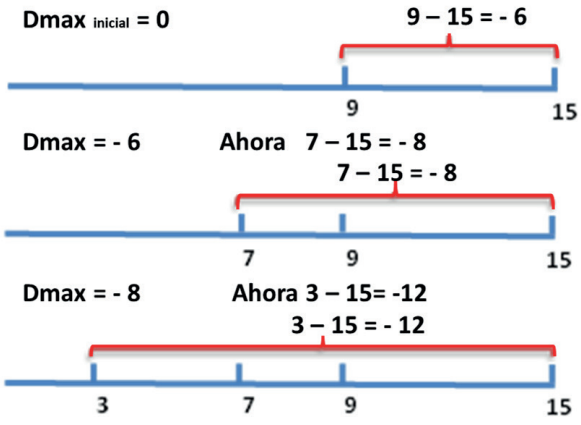
Donde Dmax comienza con un valor igual a cero al inicio de cada ejecución y se va actualizando con el valor más negativo alcanzado en cada iteración, con lo cual se busca que se realicen intercambios que mejoren el costo del anterior intercambio, hasta quedar con el mejor intercambio de nodos posibles, tal que ese criterio puede modificarse según cada investigador.

El algoritmo Intercambio 2Opt, necesita una primera solución, en este caso obtenido por el VMC, la cual busca mejorar por Intercambios 2Opt, de tal manera que se tiene un algoritmo denominado VMC_2Opt.

Para ilustrar cómo funciona el indicador Dmax (indicado en la Ecuación 1) y la selección de nodos según el valor más negativo posible alcanzado para realizar el intercambio 2Opt se presenta la Figura 2.

Figura 2

Variable Dmax en el Intercambio 2Opt



Según Da Costa, Rhuggenaath, Zhang y Akcay [6], pueden existir distintas políticas para el proceso de mejoramiento por el Intercambio 2Opt, cuya idea es reiniciar la búsqueda para evitan quedar atrapados en óptimos locales. Entre las políticas mencionamos a:

- *First Improvement (FI).* La cual selecciona el primer intercambio 2Opt de nodos que mejora la ruta y actualiza la ruta para la siguiente ejecución.
- *Best Improvement (BI).* La cual selecciona el intercambio 2Opt de nodos que reduce más el costo de la ruta actual, y al final se procede con la actualización de la ruta para una siguiente ejecución.

Al algoritmo desarrollado en esta investigación se le adicionó la política Best Improvement (BI), con la cual presenta la forma: VMC_2OptBI.

2.2.6. KPI's utilizados para medir la Calidad de las soluciones: Los KPI's son indicadores clave para evaluar el desempeño de las soluciones obtenidas por nuestro algoritmo en comparación con las de algoritmos procedentes de otras investigaciones. Por ello, se consideró KPI's tales como:

- *Error Relativo.* Mide la distancia entre la solución obtenida y la solución óptima reportada en TSPLIB, el cual es cuantificado de la siguiente manera:

$$Error\ Relativo = \frac{Solución\ Algoritmo - Solución\ Óptima\ TSPLIB}{Solución\ Óptima\ TSPLIB} \quad (2)$$

  Multiplicando la ecuación (2) por 100% se obtiene el Error Relativo Porcentual.

- *Desviación Estándar de los Errores Relativos:* Indicador que permite medir la dispersión de los Errores Relativos con respecto a la media, la cual es cuantificada de la siguiente manera:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N}(xi - \bar{X})^2}{N}} \quad (3)$$

- *Nivel de Eficacia.* Mide la cantidad de instancias en que un algoritmo logra alcanzar mejores soluciones que otro con el que se compara, se obtiene según:

$$Nivel\ de\ Eficacia = \frac{Cantidad\ de\ instrancias\ donde\ nuestro\ algoritmo\ encontró\ la\ mejor\ solución}{Cantidad\ de\ instrancias\ revisadas\ por\ los\ algoritmos\ comparados} \quad (4)$$

  Multiplicando la ecuación (4) por 100% se obtiene el Nivel de Eficacia Porcentual.

3. Propuesta

3.1. Metodología

El principal objetivo de esta investigación es la mejoría del Nivel de Eficacia de las soluciones obtenidas por el Intercambio 2Opt. Para ello, se tomó como base la propuesta formulada por Pérez y Jaramillo [4] para la introducción del factor de perturbación (1+r). Se utilizó el tipo de investigación denominada tecnológica formal, la cual indica Esteban [23], comprende campos como la Investigación de Operaciones, Programación de computadores, la Cibernética y el Análisis de sistemas.

Para la comparación de soluciones obtenidas por los algoritmos se utilizó instancias simétricas de la librería virtual TSPLIB. Así, la metodología aplicada consistió en los siguientes aspectos:

a. Determinación del lenguaje de programación a utilizar y del entorno de programación.
b. Selección de instancias del problema del agente viajero simétrico de la librería virtual TSPLIB a las cuales se les realizó distintas pruebas para la construcción del algoritmo.
c. Incorporación del factor "(1+r)" dentro del criterio para la realización del intercambio 2Opt.
d. Determinación del criterio de parada para el algoritmo propuesto.
e. Determinación del límite superior y límite inferior de valores de "r" recomendados.
f. Determinación de rango de valores de "r" para el algoritmo VMC_2OptBI_r
g. Descripción del algoritmo, por medio de un Diagrama de Flujo.

3.1.1. Determinación del lenguaje y entorno de programación para implementar el algoritmo propuesto: Se optó por el lenguaje de programación Java dentro del entorno libre de NetBeans IDE 16.

3.1.2. Selección de instancias simétricas TSPLIB: Para la evaluación de la calidad de las soluciones, se realizaron pruebas y análisis en instancias del problema del agente viajero simétrico de la librería virtual TSPLIB, que el año 2023 dispone de poco más de 110 instancias, las cuales presentan desde 14 hasta 85900 nodos, por ello, se tomó una muestra por conveniencia de 63 instancias (Apéndice B, Tabla 5) que cumplieron los siguientes criterios:

- La matriz de pesos cuyos elementos se calcularon con el Método Euclidiano_2D, o se especificaron en los archivos de TSPLIB.
- Un tamaño de instancia no mayor a 1002 nodos.

3.1.3. Adición del factor (1+r): Para agregar un factor que permita perturbar la dirección de búsqueda indicada por el Criterio de la Ecuación (1), se adicionó el factor (1+r), introducido por Pérez y Jaramillo [4], al minuendo de la resta en el Criterio de Intercambio 2Opt, es decir:

$$(1+r)x[C(a,b)+C(b,d)]-[C(a,b)+C(c,d)]<Dmax \quad (5)$$

Se introdujo el factor de perturbación "1+r" al minuendo de la ecuación (5) modificando el criterio para realizar el Intercambio 2Opt, el cual permitió explorar soluciones resultantes de los nuevos valores posibles de "r". A esta versión se le denominó 2Opt_r.

Dentro de nuestro Intercambio 2Opt_r se seleccionó la política de Best Improvement propuesta por Hansen et. al. [5], por lo cual para nuestra investigación se trabajó con un algoritmo denominado 2OptBI_r.

3.1.4. Determinación del Criterio de Parada: Al realizar distintas pruebas sobre el algoritmo VMC_2OptBI, se observó que la heurística tiende a empeorar en vez de mejorar las soluciones, incluso después de mejorar la solución existen casos donde vuelve a empeorar hasta lograr una convergencia a una solución mejor a las anteriores, tal como se muestra en la Figura 3, con ello, se logra evitar una convergencia prematura a un óptimo local.

Por el empeoramiento de la solución antes de estabilizarse, se consideró que el criterio de parada de "parar al no encontrar mejoría en iteraciones sucesivas" no resultaba viable, por esta razón se buscó determinar una cantidad de ejecuciones que se encuentre en función del orden del grafo de la instancia revisada.

En el algoritmo denominado VMC_2OptBI_r, se observa que, dependiendo del valor que tome "r", la solución puede oscilar o converger a un óptimo local, tal como se muestra en la Figura 4, por ello, se adicionó una memoria que almacena las soluciones a lo largo de distintas ejecuciones para posteriormente elegir la mejor.

Por medio de ensayo y error se proporcionó evidencia experimental, la cual indicó que el número de ejecuciones para las instancias seleccionadas tiene la forma indicada en la Ecuación 6, donde se observó que para k ≥ 6, se alcanzó la mejor solución disponible, la cual es la mejor solución que el algoritmo obtuvo entre todas las experimentaciones realizadas:

$$Número\ de\ Ejecuciones = (k)x(n) \quad (6)$$

Para el Número de Ejecuciones = 50 x n, se observó que solo para 16 de las 63 instancias mejoró la solución en un promedio de 1.2% con respecto a la óptima, sin embargo, esas mismas soluciones también llegaron a obtenerse con un valor de "6x(n)" ejecuciones. Por ello, para las instancias simétricas de TSPLIB resultó de gran importancia una selección adecuada de valores de "r".

3.1.5. Determinación del Límite Superior y Límite Inferior de valores recomendados de "r": El algoritmo desarrollado por Pérez y Jaramillo [4], recomienda un rango de valores para r = r1 / r1 ∈ [0, 1], es decir, un LInferior=0 y LSuperior=1. Con base a esto se tomó al rango [0;1] como un punto de partida para nuestro algoritmo y se

Figura 3

Convergencia de algoritmo VMC_2OptBI

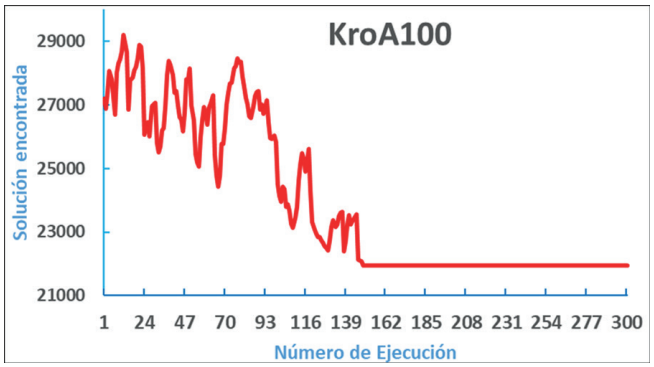
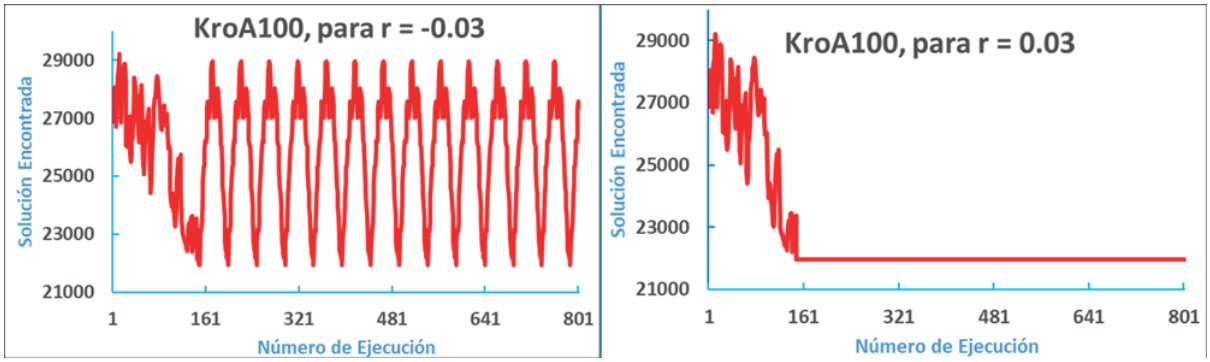


Figura 4

Convergencia del VMC_2OptBI_r según el valor de "r"



realizó una ampliación del rango para tomar también valores negativos para r = r2 / r2 ∈ [-1;0], debido a que al tomar dichos valores reducimos el valor del minuendo dentro del Criterio del Intercambio 2Opt (ver Ecuación 5) con lo cual se habilitan nuevos intercambios 2Opt y por tanto se explora nuevas soluciones factibles. Por esta razón, para nuestro valor de "r", se partió del rango que resulta de la unión de los dos rangos indicados (para r1 y r2), es decir r ∈ [-1;1].

Para determinar los mejores valores de r ∈ [-1;1], se ejecutó el VMC_2OptBI_r evidenciándose que las soluciones obtenidas presentaron un comportamiento similar a una función convexa, tal que al acercarse más a "0" se encontraron mejores soluciones y para valores mayores a "1" y menores a "-1" empeoraban(Apéndice A).

Posteriormente, se realizó un estudio analítico para las 63 instancias seleccionadas, donde el menor valor de "r" fue -0.27 y el mayor valor fue 0.11, con lo cual se asumió un rango de valores de r ∈ R / r ∈ [-0.3; 0,2], para el cual en 47 de las 63 instancias el mejor valor de "r" fue negativo (casi el 75% de los casos), lo cual muestra lo acertado de nuestro pensamiento de incluir el rango [-1; 0] para los valores admisibles para "r". Como decisión final se amplió el rango de r a [-0.3; 0.3] con el objetivo de que, para otras instancias en futuras investigaciones, se evite omitir potenciales valores dentro del rango de 0.2 a 0.3.

3.1.6. Determinación de valores de "r" para el algoritmo VMC_2OptBI_r: Al tener r ∈ [-0.3; 0,3], siendo r ∈ R, se tiene una cantidad infinita de valores para "r", por ello, se decidió elegir una cantidad que permita encontrar una solución en un tiempo razonable (minutos), por lo cual, se optó por seleccionar valores igualmente espaciados por una amplitud, calculada así:

$$Amplitud = \frac{LSuperior - LInferior}{NIntervalos} \qquad (7)$$

En la Ecuación 7, se tiene que el Límite Inferior (LInferior) es igual a -0.3 y el Limite Superior (LSuperior) igual a 0.3. Dichos valores pueden modificarse para realizar una búsqueda en intervalos más pequeños según criterio del investigador, pero con ese rango fue suficiente para obtener soluciones aceptables en un tiempo razonable. Para la presente investigación una solución es aceptable cuando presenta un *error relativo porcentual* menor a 7% con respecto al óptimo y un tiempo es razonable cuando implica menos de 60 minutos para resolver el problema. También se determinaron distintos valores de "r" a tomar para la ejecución del algoritmo VMC_2OptBI_r, los cuales presentaron la siguiente forma:

$$r_n = LInferior + (n)x(Amplitud), n = 0, \dots, NIntervalos \quad (8)$$

En la ecuación 8, se observa que "r" toma en total "NIntervalos+1" valores, donde el primer valor es igual a LInferior y el último valor es igual a LSuperior debido a que la variable "n" toma valores desde "0" hasta NIntervalos (en total NIntervalos+1 valores).

Al realizarse distintas ejecuciones sobre el algoritmo VMC_2OptBI_r se consideró como NIntervalos=60 una cantidad adecuada para obtener soluciones aceptables, debido a que con NIntervalos=80 se obtuvo casi los mismos resultados con la diferencia de que con la primera cantidad definida se consumen menos recursos computacionales, por ello, una cantidad mayor para el parámetro NIntervalos solo provoca mayor consumo de recursos computacionales y las soluciones no presentaron mejoría en la mayoría de instancia revisadas.

3.2. Descripción del algoritmo

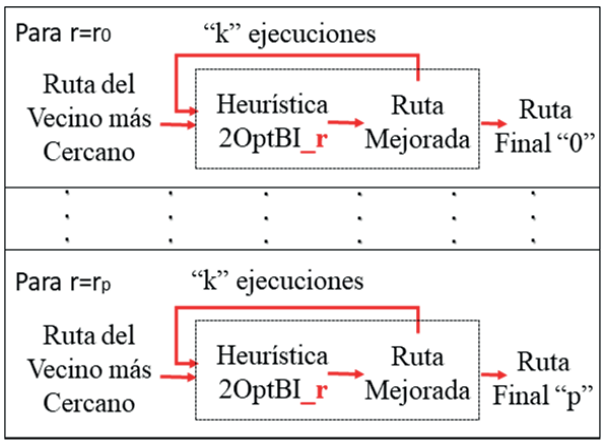
Después de la realización de los pasos indicados en la Metodología, se procedió a la descripción del algoritmo híbrido desarrollado, el cual fue denominado VMC_2OptBI_r, cuya nomenclatura proviene de las siguientes partes:

- El término VMC, proviene del hecho de haber utilizado el algoritmo heurístico del Vecino más Cercano para obtener una solución inicial, que es mejorada en una etapa posterior de nuestro algoritmo.
- El término 2Opt, proviene de utilizar el Intercambio 2Opt, que a partir de la ruta construida por el VMC busca mejorar la solución actual.
- El termino BI proviene de haber utilizado la política de mejoramiento: Best Improvement, indicado por Hansen et. al. [5] para realizar los intercambios 2Opt.
- El término "r" proviene del factor (1+r) que fue introducido al criterio para realizar el intercambio 2Opt, con el objetivo de modificar la trayectoria de búsqueda de nuestro algoritmo.

En la figura 5 se presenta el Diagrama de Bloques del funcionamiento del VMC_2OptBI_r.

Figura 5

Diagrama de Bloques del VMC_2OptBI_r



Se observa en la Figura 5 que la heurística 2OptBI_r realiza una cantidad "k" de ejecuciones para cada valor de "r", y si se tiene "p+1" valores para "r" se tendrá en total "p+1" rutas finales, de las cuales se escoge la ruta asociada al menor costo de recorrido. Con lo detallado, al producir tantas rutas como valores toma la variable "r" estamos ante un algoritmo heurístico tipo poblacional, donde cada individuo corresponde a la ruta generada por el respectivo valor de "r".

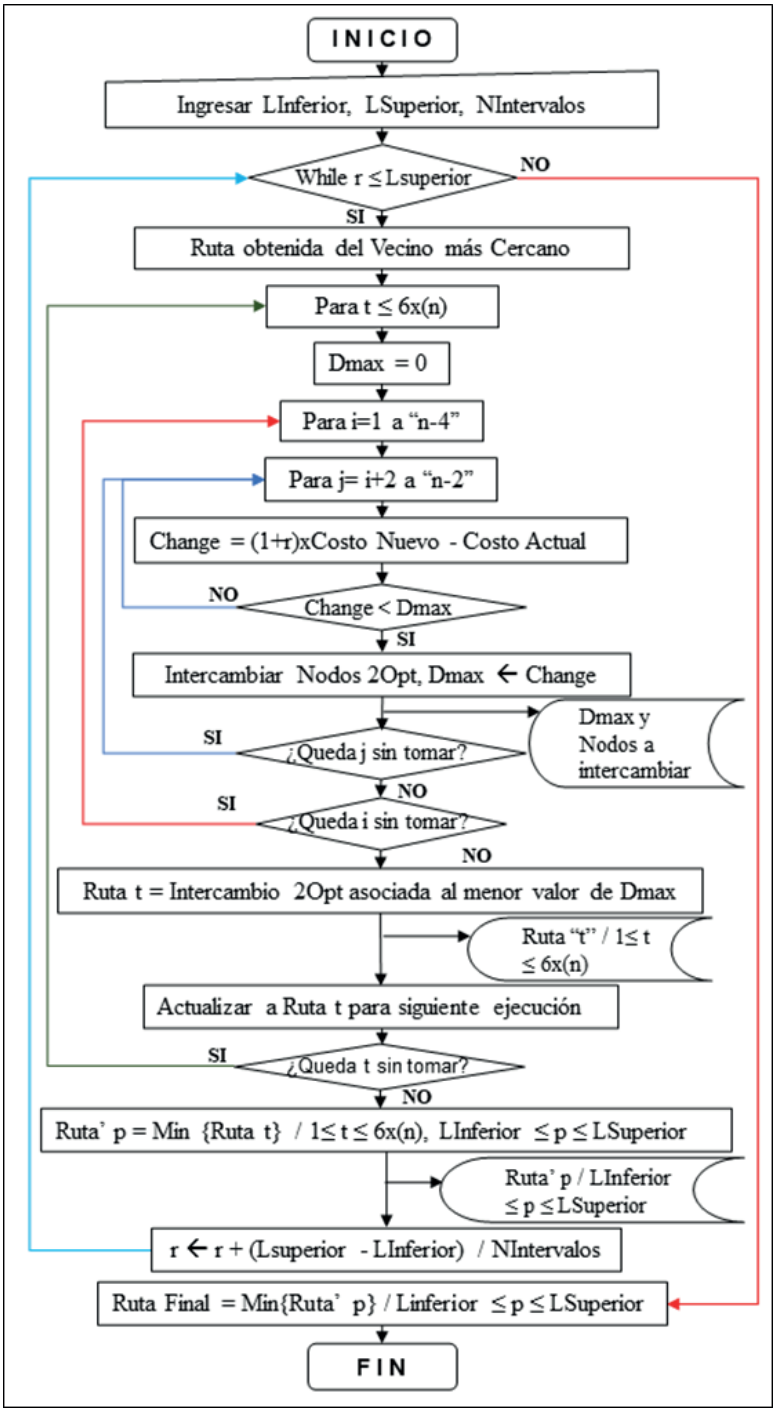
A partir del diagrama de bloques, presentado en la Figura 5, se formalizó el funcionamiento del algoritmo por medio del Diagrama de Flujo del VMC_2OptBI_r, el cual se muestra en la Figura 6.

3.3. Descripción del Proceso de Búsqueda de soluciones

Durante las distintas pruebas basadas en ensayo y error para la búsqueda de rangos de valores de "r" se observó la importancia de realizar una búsqueda minuciosa de valores de "r" no tan distanciados entre ellos. Por lo tanto, una primera propuesta fue determinar un valor

Figura 6

Diagrama de Flujo del VMC_2OptBI_r



de Amplitud (Véase Ecuación 7) igual a N con lo cual se tomarían en total N+1 valores de "r", en razón de ello, conforme crecía el tamaño de instancia lo único que producía eran mayor tiempo en encontrar una solución, por ello, se planteó realizar la búsqueda en dos etapas:

– La primera búsqueda se realizó con r ∈ [-0.3;0.3], NIntervalos = 60 para encontrar un r_0 asociado al mejor costo disponible, con el objetivo de probar distintos valores equidistantes dentro del intervalo y determinar una buena solución.

– La segunda fue una búsqueda, a partir de r_0, resultado de la primera búsqueda, con NIntervalos = 60 y un rango de valores r ∈ [r_0-0.05; r_0+0.05].

Se encontraron valores de "r que determinaron soluciones de alta calidad sin comprometer de manera excesiva el tiempo necesario para su determinación. Para ver los resultados finales en la determinación de valores de "r", ver la Tabla 5 del apéndice B que muestra valores de r ∈ [-0.3;0.3] para las 63 instancias seleccionadas.

4. Validación y Resultados

4.1. Validación

Para esta sección se planteó la división en tres etapas:

– En la *primera etapa* se realizó la evaluación de la mejoría de la solución obtenida por el VMC_2OptBI_r con respecto a las soluciones del VMC y VMC_2OptBI, es decir, comparación de nuestras soluciones con la versión sin el factor (1+r) y el algoritmo del Vecino más Cercano, para ello se realizó lo siguiente:

  • Medición de la dispersión de los errores relativos aplicando la desviación estándar según el algoritmo: VMC, VMC_2OptBI y VMC_2OptBI_r

  • Mejoría medida con el Nivel de Eficacia del VMC_2OptBI_r con respecto a sus versiones básicas: VMC y VMC_2OptBI.

  Para ello, se tomó las 63 instancias seleccionadas en el ítem "3.1.2.Selección de instancias simétricas TSPLIB", el detalle de las instancias se presenta en el Apéndice B en la Tabla 5.

– En la *segunda etapa* se realizó la evaluación de la mejoría de nuestras soluciones con respecto a las obtenidas por las versiones del Sacrificio Cortoplacista Adaptativo de Pérez y Jaramillo [4], para ello se realizó lo siguiente:

  • Medición de la dispersión de los errores relativos usando la desviación estándar según el algoritmo: SCA_2Opt, SCA_2Opt_r y el VMC_2OptBI_r.

  • Mejoría medida con el Nivel de Eficacia del VMC_2OptBI_r con respecto al SCA_2Opt y el SCA_2Opt_r.

  Con ese fin, se tomó 11 instancias que se utilizaron en la investigación de Pérez y Jaramillo [4] cuyo método de cálculo de la matriz de pesos fue el Euclidiano.

– En la *tercera etapa* se realizó la evaluación de la mejora de nuestras soluciones con respecto a las obtenidas por los algoritmos metaheurísticos de investigaciones recientes (2019-2022) que hubiesen trabajado con instancias simétricas de TSPLIB, para ello se realizó lo siguiente:

  • Medición de la dispersión de los errores relativos a través de la desviación estándar según el algoritmo metaheurístico y VMC_2OptBI_r.

  • Mejoría medida con el Nivel de Eficacia del VMC_2OptBI_r con respecto a versiones de algoritmos metaheurísticos revisados.

Para realizar esto se tomaron las instancias simétricas del problema del agente viajero de TSPLIB, cuyas matrices de pesos hayan sido calculadas por el método euclidiano (EUC_2D). A continuación, se indican los algoritmos y los autores:

– La versión del Bee Algorithm implementada por Ismail, Hartono, Zeybek y Pham [24], quienes la utilizaron para solucionar instancias simétricas de TSPLIB entre 100 hasta 200 nodos.

– La versión de Ant System implementada por Thirugnanasambandam et. al.[10], quienes la utilizaron para solucionar instancias simétricas de TSPLIB de tamaño hasta 100 nodos.

– La versión del SMACFA fue implementada por Liu, M., Li, Y., Li, A., Huo, Q., Zhang, N., Qu, N., Zhu, M. & Chen, L [25], quienes la utilizaron para solucionar instancias simétricas de TSPLIB de tamaño hasta 442 nodos. Ese algoritmo resultó de la fusión del algoritmo Slime Mold (SMA) que imita el proceso de alimentación del moho mucilaginoso, con el algoritmo Ant Colony (ACO), resultando el nuevo algoritmo SMACFA.

– La versión del AS-FA-LS fue implementada por Rokbani, N., Kromer, P., Twir, I., & Alimi, A. M. [26], quienes propusieron una hibridación entre los algoritmos Ant Colony Optimization (ACO), Firefly Algorithm (FA) y Local Search por medio del Intercambio 2Opt. De la combinación de los tres algoritmos mencionados se construyó al algoritmo

AS-FA-LS, donde el FA se utilizó para calcular los parámetros del ACO y el Intercambio 2Opt para evitar soluciones subóptimas. Este algoritmo fue utilizado para resolver instancias simétricas de TSPLIB de tamaño de hasta 200 nodos.

4.2. Resultados

4.2.1. Comparación del VMC, VMC_2OptBI y el VMC_2OptBI_r

a. Dispersión de los errores relativos porcentuales del VMC, VMC_2OptBI y el VMC_2OptBI_r. Los resultados se presentan en la Tabla 5 (Apéndice B) que muestra las soluciones según la instancia simétrica de TSPLIB revisada y su error relativo (porcentual), a partir del cual se construyó la Figura 7 y la Figura 8.

En la Figura 8, se omitió la instancia brg180 para el VMC al presentar un error relativo máximo atípico (3466.7%). Además, se pudo concluir que:

- El rango de errores relativos (Mínimo y Máximo) del VMC_2OptBI_r fue menor en comparación al VMC y el VMC_2OptBI
- La desviación estándar resultó ser menor para el VMC_2OptBI_r, es decir, datos menos dispersos con respecto a la media.

b. Nivel de Eficacia del VMC_2OptBI_r comparado con el VMC_2OptBI y el VMC: A partir la Tabla 5 del Apéndice B, que muestra las soluciones según la instancia simétrica de TSPLIB y sus error relativo porcentual se concluyó que:

– El Nivel de Eficacia del VMC_2OptBI_r alcanzó un valor de 100% con respecto al Vecino más Cercano (VMC) en las 63 instancias revisadas.

– El Nivel de Eficacia del VMC_2OptBI_r alcanzó un valor de 100% con respecto al VMC_2OptBI en las 63 instancias revisadas.

Figura 7

Error relativo porcentual de soluciones obtenidas según algoritmo

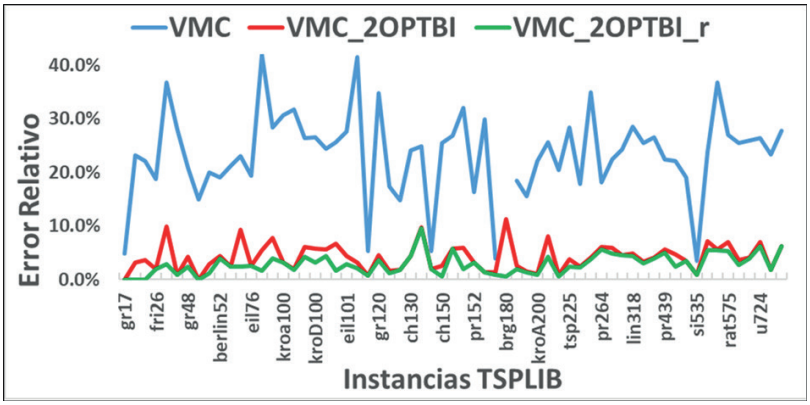
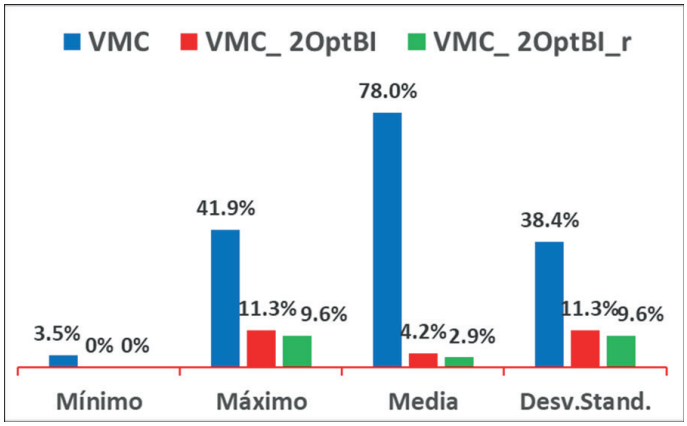


Figura 8

Estadísticos para error relativo porcentual de soluciones obtenidas según algoritmos VMC, VMC_2OptBI y VMC_2OptBI_r



4.2.2. Comparación del SCA_2Opt, SCA_2Opt_r y el VMC_2OptBI_r

a. *Dispersión de los errores relativos del SCA_2Opt, SCA_2Opt_r y el VMC_2OptBI_r.* Los resultados del error relativo (porcentual) de las soluciones obtenidas se consolidaron en la Tabla 2, a partir de la cual se construyó la Figura 9.

   A partir de la Figura 9, se pudo concluir que:

   – El rango de errores relativos porcentuales (Mínimo y Máximo) del VMC_2OptBI_r resultó muy similar al del SCA_2Opt_r.

   – La desviación estándar resultó ser menor para el SCA_2Opt_r, seguida del VMC_2OptBI_r, y la media de los Errores Relativos para el VMC_2OptBI_r resultó ser la menor.

b. *Nivel de Eficacia del VMC_2OptBI_r comparado con el SCA_2Opt y SCA_2Opt_r.* A partir de la Tabla 2, se realizó el cálculo para el Nivel de Eficacia, tal que:

   - El Nivel de Eficacia del VMC_2OptBI_r en comparación con el SCA_2Opt fue de 100% con respecto a las instancias revisadas.

   - El Nivel de Eficacia del VMC_2OptBI_r con respecto al SCA_2Opt_r alcanzó un valor de 81.8% con respecto a las instancias revisadas.

4.2.3. Comparación del VMC_2OptBI_r con las metaheurísticas seleccionadas.

a. *Dispersión de los errores relativos del VMC_2OptBI_r con respecto a las metaheurísticas seleccionadas.* Los resultados se consolidaron en la Tabla 3, la cual muestra el error relativo (porcentual) de las soluciones obtenidas según el algoritmo metaheurístico según instancia TSPLIB, donde:

Tabla 2

Error relativo porcentual de soluciones obtenidas por el SCA_2Opt, SCA_2Opt_r y el VMC_2OptBI_r

Instancia TSPLIB	Óptimo TSPLIB	Soluciones obtenidas			Error relativo porcentual		
		SCA_2Op	SCA_2Opt_r	VMC_2OPT_r	SCA_2Opt	SCA_2Opt_r	VMC_2OptBI_r
1. Swiss42	1273	1375	1285	1284	8.01%	0.94%	0.86%
2. eil51	426	449.85	436.08	431	5.60%	2.37%	1.17%
3. st70	675	713.98	690.62	691	5.77%	2.31%	2.37%
4. eil76	538	567.01	559.61	552	5.39%	4.02%	2.60%
5. rat99	1211	1268.95	1259.51	1258	4.79%	4.01%	3.88%
6. kroa100	21282	22379.4	22219.3	21946	5.16%	4.40%	3.12%
7. kroA150	26524	29076.9	27999.7	28053	9.62%	5.56%	5.76%
8. KroB150	26130	28136.3	26904.48	26625	7.68%	2.96%	1.89%
9. d198	15780	16212	16180.53	15992	2.74%	2.54%	1.34%
10. tsp225	3919	4071	4054.77	4012	3.88%	3.46%	2.37%
11. Pcb442	50778	55133.8	53287.29	52024	8.58%	4.94%	2.45%

Figura 9

Estadísticos para error relativo porcentual de soluciones obtenidas según algoritmos SCA_2opt, SCA_2Opt_r y VMC_2OptBI_r

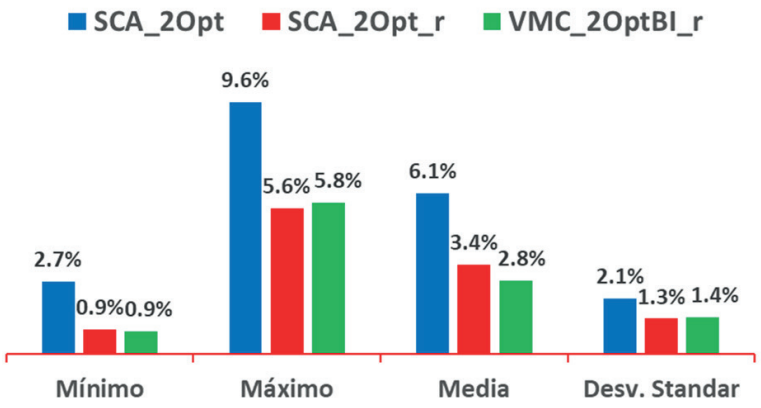


Tabla 3

Error relativo porcentual de soluciones obtenidas según algoritmo metaheurístico utilizado e instancia TSPLIB

Caso TSPLIB	Óptimo	Bees Algorithm	Ant System	SMA CFA	AS-FA -LS	VMC_2Opt BI_r	Bees Algorithm	Ant System	SMA _CFA	AS-FA -LS	VMC_2Opt BI_r
eil51	426	-	456.9127	-	428	431	-	7.3%	-	0.5%	1.2%
berlin52	7542	-	7681.454	7572	7542	7842	-	1.8%	0.4%	0.0%	4.0%
st70	675	-	721.2423	-	675	691	-	6.9%	-	0.0%	2.4%
eil76	538	-	563.1793	-	541	552	-	4.7%	-	0.6%	2.6%
pr76	108159	-	118693.7	114668	-	109970	-	9.7%	6.0%	-	1.7%
rat99	1211	-	-	-	1213	1258	-	-	-	0.2%	3.9%
kroa100	21282	21469	-	-	21282	21946	0.9%	-	-	0.0%	3.1%
kroB100	22141	22618	-	-	-	22534	2.2%	-	-	-	1.8%
kroC100	20749	20969	-	-	-	21621	1.1%	-	-	-	4.2%
kroD100	21294	21392	-	-	-	21968	0.5%	-	-	-	3.2%
kroE100	22068	22266	-	-	-	23026	0.9%	-	-	-	4.3%
eil101	629	-	-	-	639	647	-	-	-	1.6%	2.9%
lin105	14379	-	-	15055	-	14692	-	-	4.7%	-	2.2%
ch130	6110	-	-	6403	-	6375	-	-	4.8%	-	4.3%
kroA150	26524	27527	-	-	-	28053	3.8%	-	-	-	5.8%
kroB150	26130	27095	-	-	-	26625	3.7%	-	-	-	1.9%
ch150	6528	-	-	6776	6571	6569	-	-	3.8%	0.7%	0.6%
kroA200	29368	31550	-	-	29532	29637	7.4%	-	-	0.6%	0.9%
kroB200	29437	32024	-	-	-	30686	8.8%	-	-	-	4.2%
tsp225	3919	-	-	4239	-	4012	-	-	8.2%	-	2.4%
a280	2579	-	-	2884	-	2706	-	-	11.8%	-	4.9%
Pcb442	50778	-	-	60677	-	52024	-	-	19.5%	-	2.5%
					Mínimo		0.5%	1.8%	0.4%	0%	0.6%
					Máximo		8.8%	9.7%	19.5%	1.6%	5.8%
					Media		3.3%	6.1%	7.4%	0.5%	3.0%
					Desv. Standar		8.8%	9.7%	19.5%	1.6%	1.4%

- El rango del error relativo porcentual (Mínimo y Máximo) del VMC_2OptBI_r resultó estar entre el 0.6% al 5.8%, siendo solo superado por la metaheurística AS-FA-LS que tiene un rango comprendido entre 0% a 1.6%.
- La desviación estándar resultó ser menor para el VMC_2OptBI_r, seguida del AS-FA-LS, pero al analizar la media respectiva de los errores relativos, el AS-FA-LS presentó un valor menor en comparación al VMC_2OptBI_r.

A partir de la Tabla 3 se construyó la Figura 10, que muestra el valor mínimo, máximo, media y desviación estándar del error relativo porcentual de la solución obtenida según las instancias revisadas.

De la Figura 10, el VMC_2OptBI_r fue muy competitivo con respecto a 3 de las 4 metaheurísticas revisadas, solo siendo superada por el AS-FA-LS.

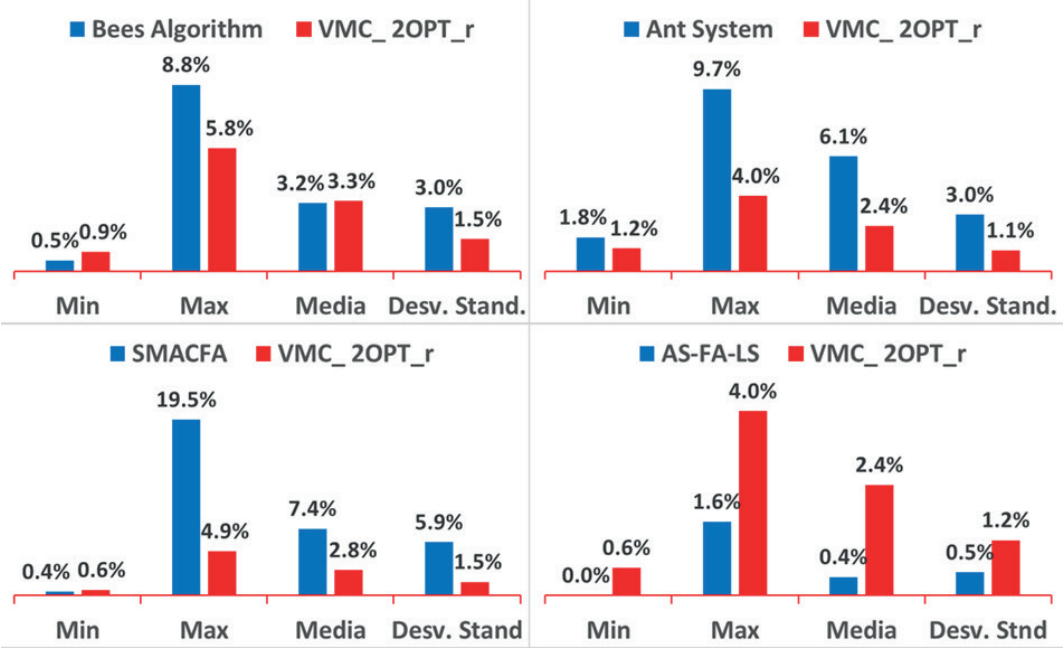
b. Nivel de Eficacia del VMC_2OptBI_r comparado con otras metaheurísticas: A partir de los resultados consolidados en la Tabla 3 se construyó la Tabla 4 que muestra el Nivel de Eficacia del algoritmo VMC_2OptBI_r con respecto a los cuatro algoritmos metaheurísticos seleccionados.

En la Tabla 4, se puede observar que el VMC_2OptBI_r alcanzó los siguientes niveles de eficacia:

- Por encima del 80% al compararse con las soluciones del Ant System y el SMACFA.
- Del 11% al compararse con el algoritmo AS-FA-LS, el cual también usa la heurística del Intercambio 2Opt y dos metaheurísticas que hacen más complejo al algoritmo.
- Un Nivel de Eficacia de 44% al compararse con el Bee Algorithm.

Figura 10

Estadísticos para error relativo porcentual de soluciones obtenidas según metaheurísticas e instancias revisadas



Los resultados mostraron que por medio de una modificación del algoritmo 2Opt se pudo obtener rutas para el problema del agente viajero simétrico cuyas soluciones fueron competitivas con las obtenidas por otros algoritmos metaheurísticos (ver Tabla 4), donde la ventaja del VMC_2OptBI_r radica en su sencilla implementación en comparación con metaheurísticas tales como: algoritmos genéticos, recocido simulado, algoritmo de enjambre de abejas, colina de hormigas, entre otras.

Tabla 4

Nivel de eficacia del VMC_2OptBI_r vs Otras Metaheurísticas

Detalle	Bees Algorithm	Ant System	SMACFA	AS-FA-LS
VMC_2OptBI_r	44%	80%	88%	11%

5. Conclusiones y Recomendaciones

5.1. Conclusiones

- El algoritmo del VMC_2OptBI_r logró alcanzar un Nivel de Eficacia del 100% para las instancias revisadas al compararse con el algoritmo del Vecino más Cercano (VMC) y el VMC_2OptBI, superando enormemente al VMC.
- El algoritmo del VMC_2OptBI_r, logró alcanzar un Nivel de Eficacia de 100% al compararse con el SCA_2Opt_r en las instancias revisadas.
- Si bien el Nivel de Eficacia del VMC_2OptBI_r no alcanzó el valor de 100% al compararse con modificaciones recientes de metaheurísticas (2019-2022), alcanzó valores del 11% al 88%, los cuales lo hacen competitivo frente a las metaheurísticas revisadas, en particular por la sencillez de su implementación.
- Para los valores de "r", se notó un predominio de valores negativos, en un 60% de las 63 instancias revisadas, para encontrar la mejor solución disponible.
- Por medio de pequeñas pero significativas modificaciones se logró la construcción de una heurística híbrida con buena capacidad de obtener soluciones competitivas respecto a las de los algoritmos metaheurísticos.

5.2. Recomendaciones

- Para la implementación de este algoritmo se recomienda una búsqueda en dos etapas donde NIntervalos = 60 permanece constante: en la primera etapa para determinar un $r_0 \in [-0.3;0.3]$ y en la segunda etapa para determinar un $r \in [r_0-0.05; r_0+0.05]$.
- Durante las pruebas de ensayo y error, si bien es cierto que presentar una mayor cantidad de ejecuciones del algoritmo 2OptBI_r ayuda a construir mejores rutas, estas también pueden encontrarse si se elige un buen intervalo para seleccionar valores de "r".

6. Trabajos Futuros

Para futuros trabajos se propone construir y analizar nuevas estructuras que modifiquen el criterio de decisión para realizar el intercambio 2Opt, tales como:

- La implementación de la política de intercambio 2Opt First Improvement.
- Una combinación del criterio de adición del factor (1+r) de Pérez y Jaramillo [4] con el introducido por Levin y Yovel [3] donde la función de costo se reemplazó por $f(c(x))= c(x)^r$.
- Adicionar un criterio procedente de un algoritmo heurístico o metaheurístico para combinarlo con nuestro VMC_2OptBI_r.
- Reemplazar la solución inicial obtenida por el algoritmo del Vecino más Cercano (VMC) por otra hallada por algún otro algoritmo heurístico o metaheurístico.
- Estudiar la posibilidad de implementar la versión del Intercambio 3Opt con la política Best Improvement (BI) y el factor (1+r), es decir, un algoritmo denominado VMC_3OptBI_r.
- Estudiar si el factor de perturbación (1+r) puede adicionarse al criterio de decisión de otro algoritmo heurístico o metaheurístico.

Referencias

[1] Moroyoqui Olan M.A., Orozco-Rosas U., & Picos K., "Implementación de un Algoritmo Genético modificado para la solución al Problema del Agente Viajero," Revista Aristas: Investigación Básica y Aplicada, vol. 8, no. 17, pp. 190-196, 2022, https://repositorio.cetys.mx/bitstream/60000/1426/1/171-Texto%20del%20art%c3%adculo-405-1-10-20220531.pdf (accessed 06 March 2023)

[2] López E., Salas Ó., & Murillo Á., "El problema del agente viajero: un algoritmo determinístico usando Búsqueda Tabú," Revista de Matemática: teoría y aplicaciones, vol. 21, no. 1, pp. 127-144, 2014, https://www.redalyc.org/pdf/453/45331281008.pdf (accessed 06 March 2023)

[3] Levin, A., & Yovel, U., "Nonoblivious 2-Opt heuristics for the traveling salesman problem," Networks, vol. 62, no. 3, pp. 201-219, 2013, https://doi.org/10.1002/net.21512 (accessed 22 Februry 2023)

[4] Pérez Rave, J., & Jaramillo Álvarez, P., "Sacrificio cortoplacista adaptativo 2opt (SCA_2opt): Una heurística inspirada en el pensamiento sistémico," Production. vol. 24, no. 1, pp. 26-40, 2014, https://doi.org/10.1590/S0103-65132013005000033, (accessed 22 February 2023)

[5] Hansen P., Mladenovic N., Todosijevic N., y Hanafi, S. "Variable neighborhood search: basics and variants," EURO Journal on Computational Optimization, vol. 5, no. 3, pp. 423-454, 2017, https://www.researchgate.net/publication/306084902_Variable_neighborhood_search_basics_and_variants (accessed 07 March 2023)

[6] Da Costa, Rhuggenaath, Zhang y Akcay, "Learning 2-opt Heuristics for the Traveling Salesman Problem via Deep Reinforcement Learning," In Proceedings of Machine Learning Research of Asian Conference on Machine Learning., pp. 465-480, 2020,a. http://proceedings.mlr.press/v129/costa20a/costa20a.pdf (accessed 24 February 2023)

[7] Zuñiga Vargas, L., "Un algoritmo heurístico para el coloreo de grafos," Benemérita Universidad Autónoma de Puebla (Tesis de Pregrado), Puebla, México, 2016. https://hdl.handle.net/20.500.12371/12969 (accessed 27 April 2023)

[8] Anaya-Fuentes, "El problema del agente viajero resuelto mediante agrupación en clústeres y algoritmos genéticos," Pädi Boletín Científico De Ciencias Básicas E Ingenierías Del ICBI, vol. 9, no. 17, pp. 88-97, 2021, http://portal.amelica.org/ameli/jatsRepo/595/5952727017/5952727017.pdf (accessed 24 February 2023).

[9] Guo Ping, Hou Mengliang, Ye Lian, "MEATSP: a membrane evolutionary algorithm for solving TSP," IEEE Access, vol. 8, pp. 199081-199096, 2020, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9245472 (accessed 07 March 2023)

[10] Thirugnanasambandam K. y Raghav R. S., "Experimental analysis of ant system on travelling salesman problem dataset TSPLIB," EAI Endorsed Transactions on Pervasive Health and Technology, vol. 5, no. 19, 2019, https://eudl.eu/pdf/10.4108/eai.13-7-2018.163092 (Accessed 3 March 2023)

[11] Astoquillca-Yaranga D.J., "Reducción de costos de recolección y transporte de residuos sólidos en contenedores soterrados aplicando algoritmos genéticos para determinación de rutas Caso: distrito Bellavista-Callao," Universidad Nacional Mayor de San Marcos (Tesis de Pregrado), Lima, Perú, 2022.

[12] Pérez de la Cruz, C., "Un algoritmo Híbrido para un problema de horarios con restricciones especiales," Universidad Nacional Autónoma de México (Tesis de Maestría), Ciudad de México, México, 2011, http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/4723/Tesis.pdf?sequence=1 (accessed 25 February 2023)

[13] Glover, Fred, "Future paths for integer programming and links to artificial intelligence," Computers & operations research, vol 13, no 5, pp. 533-549, 1986. https://leeds-faculty.colorado.edu/glover/fred%20pubs/174%20-%20Future%20Paths%20for%20Integer%20Programming%20TS.pdf (accessed 27 April 2023)

[14] Cockbaine Ojeda J. y Silva Urrea R. , "Perfeccionando algoritmos heurísticos para el problema NP-C E-TSP," Ingeniare. Revista chilena de ingeniería, vol, 21, no. 2, pp. 196-204, 2013, https://www.scielo.cl/pdf/ingeniare/v21n2/art04.pdf, (accessed 06 March 2023)

[15] López Guevara, R., "Minimización del ancho de banda de matrices simétricas dispersas aplicando algoritmos genéticos," Universidad Nacional Mayor de San Marcos (Tesis de Maestría), Lima, Perú, 2009.

[16] Maguiña Agurto, L., "Implantación de VRP - Solver aplicando la heurística de Clarke Wright para el ruteo del transporte terrestre en el área de distribución caso de estudio: industrias alimentarias," Universidad Nacional Mayor de San Marcos (Tesis de Pregrado), Lima, Perú, 2016.

[17] Du P., Liu N., Zhang H., Lu J., "An improved ant colony optimization based on an adaptive heuristic factor for the traveling salesman problem," Journal of Advanced Transportation, vol. 2021, pp. 1-16, 2021, https://doi.org/10.1155/2021/6642009 (accessed 7 March 2023)

[18] Hoto R. S., Bressan G. M. y Rodrigues M. O., “Minimizing the preparation time of a tubes machine: Exact Solution and Heuristics,” Pesquisa Operacional, vol. 38, no. 1, pp. 135-152, 2018, https://www.scielo.br/j/pope/a/qf3ngtJFydnb9JmqVYBPFWx/?lang=en (accessed 26 February 2023)

[19] Lima S. J. A., Araújo S. A. y Schimit, P. H. T., “A hybrid approach based on genetic algorithm and nearest neighbor heuristic for solving the capacitated vehicle routing problema,” Revista Boaciencia. Negocios e Tecnologías, vol. 2, no. 1, pp. 18-33, 2022, https://boaciencia.org/index.php/tecnologiaynegocios/article/view/56/52 (accessed 26 February 2023)

[20] Hougardy S., Zaiser F., Zhong X., “The approximation ratio of the 2-Opt Heuristic for the metric Traveling Salesman Problem,” Operations Research Letters, vol. 48, no. 4, pp. 401-404, 2020, https://doi.org/10.1016/j.orl.2020.05.007 (accessed 06 March 2023)

[21] Schulz A.A.H., Scarpin C., & de Souza A.S., “Uma abordagem meta heurística para o problema do VRPTW”. In IX Congresso Brasileiro de Engenharia de Produção, Dic. 2019, https://aprepro.org.br/conbrepro/2019/anais/arquivos/10202019_231036_5dad11bc08ab3.pdf (accessed 06 March 2023)

[22] Gutiérrez E. I., “Uso de algoritmos heurísticos para el diseño de rutas de recolección de residuos sólidos urbanos,” Universidad Politécnica de Tulancingo (Tesis de Maestría), Hidalgo, México, 2018, http://www.upt.edu.mx/Contenido/Investigacion/Contenido/TESIS/MOP/2018/MOP_T_2018_01_IGE.pdf (accessed 10 March 2023)

[23] Esteban Nieto, N., “Tipos de Investigación,” 2018. https://core.ac.uk/download/pdf/250080756.pdf (accessed 22 Febreary 2023)

[24] Ismail A.H., Hartono N., Zeybek S., Pham D.T., “Using the Bees Algorithm to solve combinatorial optimisation problems for TSPLIB”. In IOP Conference Series: Materials Science and Engineering, vol. 847, no. 1, pp. 012027, 2020, https://iopscience.iop.org/article/10.1088/1757-899X/847/1/012027/pdf (Accessed 3 March 2023)

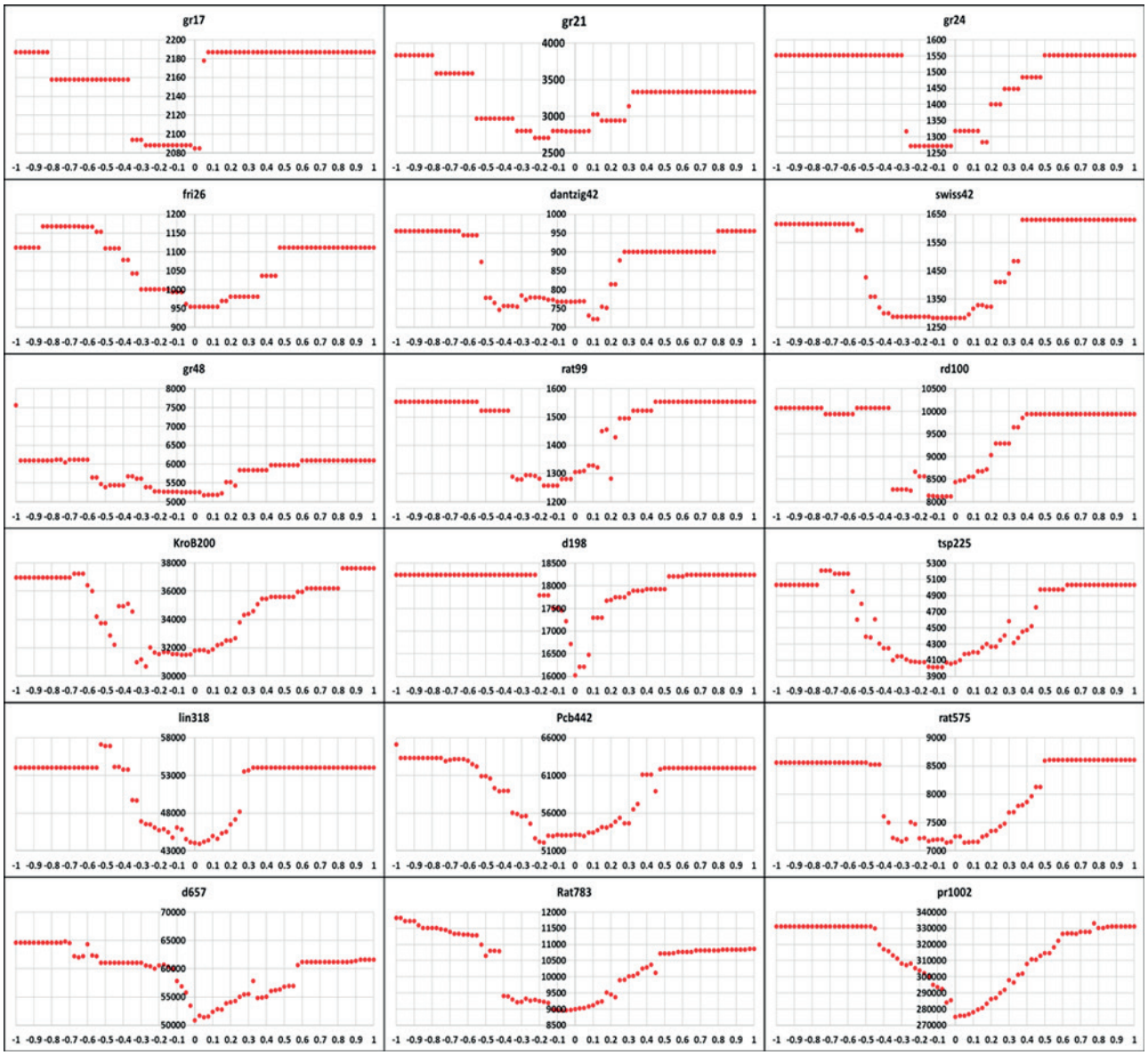
[25] Liu M., Li Y., Li A., Huo Q., Zhang N., Qu N., Zhu M. y Chen L. “A slime mold-ant colony fusion algorithm for solving traveling salesman problem”. IEEE Access, vol. 8, pp. 202508-202521, 2020, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9247179 (Accessed 3 March 2023)

[26] Rokbani N., Kromer P., Twir I., & Alimi, A. M. “A Hybrid Hierarchical Heuristic-ACO With Local Search Applied to Travelling Salesman Problem, AS-FA-Ls,” International Journal of System Dynamics Applications (IJSDA), vol. 9, no. 3, pp. 58-73, 2020, https://www.igi-global.com/pdf.aspx?tid=257243&ptid=229848&ctid=4&oa=true&isxn=9781799807957 (Accessed 3 March 2023)

Apéndice

Apéndice A

Comportamiento de la mejor solución disponible para distintos valores de "r" según instancia simétrica TSPLIB



Apéndice B.

Tabla 5

Error relativo porcentual de soluciones obtenidas por el VMC, VMC_2OptBI y el VMC_2OptBI_r

Instancia	Óptimo	VMC	VMC_ 2OptBI	r	VMC_ 2OptBI_r	Error Relativo VMC	Error Relativo VMC_2OptBI	Error Relativo VMC_2OptBI_r
1.gr17	2085	2187	2085	-0.01	2085	4.9%	0.0%	0.0%
2.gr21	2707	3333	2795	-0.24	2707	23.1%	3.3%	0.0%
3.gr24	1272	1553	1318	-0.25	1272	22.1%	3.6%	0.0%
4.fri26	937	1112	955	-0.04	955	18.7%	1.9%	1.9%
5.dantzig42	699	956	768	0.08	719	36.8%	9.9%	2.9%
6.Swiss42	1273	1630	1284	-0.14	1284	28.0%	0.9%	0.9%
7.gr48	5046	6098	5259	0.0317	5168	20.8%	4.2%	2.4%
8.hk48	11461	13181	11461	-0.05	11461	15.0%	0.0%	0.0%
9.eil51	426	511	438	-0.01	431	20.0%	2.8%	1.2%
10.berlin52	7542	8980	7877	-0.01	7842	19.1%	4.4%	4.0%
11.Brazil58	25395	30774	26008	$1.4225x10^{-16}$	26008	21.2%	2.4%	2.4%
12.st70	675	830	738	-0.18	691	23.0%	9.3%	2.4%
13.eil76	538	642	552	-0.14	552	19.3%	2.6%	2.6%
14.pr76	108159	153462	114099	-0.21	109970	41.9%	5.5%	1.7%
15.rat99	1211	1554	1305	-0.19	1258	28.3%	7.8%	3.9%
16.kroa100	21282	27807	21946	-0.04	21946	30.7%	3.1%	3.1%
17.kroB100	22141	29158	22571	-0.11	22534	31.7%	1.9%	1.8%
18.kroC100	20749	26227	21997	-0.1467	21621	26.4%	6.0%	4.2%
19.kroD100	21294	26947	22535	-0.035	21968	26.5%	5.8%	3.2%
20.kroE100	22068	27460	23305	-0.02	23026	24.4%	5.6%	4.3%
21.rd100	7910	9938	8439	-0.0067	8041	25.6%	6.7%	1.7%
22.eil101	629	803	657	-0.07	647	27.7%	4.5%	2.9%
23.lin105	14379	20356	14826	-0.08	14692	41.6%	3.1%	2.2%
24.pr107	44303	46680	44648	-0.01	44625	5.4%	0.8%	0.7%
25.gr120	6942	9351	7253	-0.07	7198	34.7%	4.5%	3.7%
26.pr124	59030	69297	59990	-0.07	59764	17.4%	1.6%	1.2%
27.bier127	118282	135737	120460	-0.02	120460	14.8%	1.8%	1.8%
28.ch130	6110	7579	6375	$1.4225x10^{-16}$	6375	24.0%	4.3%	4.3%
29.pr136	96772	120769	106162	0.11	106045	24.8%	9.7%	9.6%
30.pr144	58537	61652	59702	-0.01	59702	5.3%	2.0%	2.0%
31.ch150	6528	8191	6695	-0.253	6569	25.5%	2.6%	0.6%
32.kroA150	26524	33633	28053	-0.01	28053	26.8%	5.8%	5.8%
33.kroB150	26130	34499	27679	-0.24	26625	32.0%	5.9%	1.9%
34.pr152	73682	85699	76053	-0.02	76053	16.3%	3.2%	3.2%
35.u159	42080	54675	42626	-0.01	42626	29.9%	1.3%	1.3%
36.si175	21407	22263	21705	-0.09	21613	4.0%	1.4%	1.0%
37.brg180(*)	1950	69550	2170	-0.005	1960(*)	3466.7%	11.3%	0.5%
38.rat195	2323	2752	2382	-0.23	2369	18.5%	2.5%	2.0%
39.d198	15780	18240	16021	-0.00125	15992	15.6%	1.5%	1.3%
40.kroA200	29368	35859	29654	0.0017	29637	22.1%	1.0%	0.9%
41.kroB200	29437	36980	31799	-0.27	30686	25.6%	8.0%	4.2%
42.ts225	126643	152493	127831	-0.14	127413	20.4%	0.9%	0.6%
43.tsp225	3919	5033	4069	-0.13	4012	28.4%	3.8%	2.4%
44.pr226	80369	94683	82361	-0.015	82241	17.8%	2.5%	2.3%
45.gil262	2378	3208	2477	-0.05	2467	34.9%	4.2%	3.7%
46.pr264	49135	58023	52138	-0.02	51902	18.1%	6.1%	5.6%
47.a280	2579	3157	2733	-0.02	2706	22.4%	6.0%	4.9%
48.pr299	48191	59890	50400	$1.4225x10^{-16}$	50400	24.3%	4.6%	4.6%
49.lin318	42029	54019	44055	0.02	43909	28.5%	4.8%	4.5%
50.rd400	15281	19183	15788	-0.09	15749	25.5%	3.3%	3.1%
51.fl417	11861	15013	12339	0.01	12323	26.6%	4.0%	3.9%
52.pr439	107217	131281	113207	0.07	112657	22.4%	5.6%	5.1%
53.Pcb442	50778	61979	53162	-0.1733	52024	22.1%	4.7%	2.5%
54.d493	35002	41665	36245	$1.4225x10^{-16}$	36245	19.0%	3.6%	3.6%
55.si535	48450	50144	48910	-0.0033	48907	3.5%	0.9%	0.9%
56.pa561	2763	3422	2959	$1.4225x10^{-16}$	2916	23.9%	7.1%	5.5%
57.u574	36905	50459	38973	-0.0167	38925	36.7%	5.6%	5.5%
58.rat575	6773	8605	7242	0.0267	7138	27.0%	6.9%	5.4%
59.p654	34643	43457	35895	-0.0033	35597	25.4%	3.6%	2.8%
60.d657	48912	61627	50896	$1.4225x10^{-16}$	50873	26.0%	4.1%	4.0%
61.u724	41910	52943	44835	-0.0117	44498	26.3%	7.0%	6.2%
62.Rat783	8806	10867	8964	-0.05	8963	23.4%	1.8%	1.8%
63.Pr1002	259045	331103	275309	$1.4225x10^{-16}$	275309	27.8%	6.3%	6.3%

(*) El caso 37.brg180 presentó la particularidad de ser la única instancia en necesitar un mayor número de ejecuciones para alcanzar la mejor respuesta disponible, en este caso se necesitó un mínimo de 13x(180)= 2340 ejecuciones para alcanzar la solución 1960.