



**Algoritmo genético híbrido con representación basada en
grupos para resolver problemas de cadenas de suministro
con un enfoque de optimización binivel**

*Tesis para obtener el grado de:
Maestra en Inteligencia Artificial*

*Presenta:
Rocío Salinas Guerra*

Supervisado por
Director:
Dr. Efrén Mezura Montes
Co-Directora:
Dra. Marcela Quiroz Castellanos

Agosto 2023

Agradecimientos

En primer lugar, quisiera agradecer a mi asesor el Dr. Efrén Mezura Montes por su apoyo personal y académico, sobre todo en su guía en este proyecto de investigación.

Quisiera agradecer a la Dra. Marcela Quiroz Castellanos por ser mi amiga y mentora. Gracias por mostrarme lo bella que es la ciencia y la divulgación.

Quisiera agradecer a mi familia por su constante apoyo en todas las decisiones que he tomado, aunque eso implique estar más lejos de casa. Gracias por todo el amor y paciencia.

Quisiera agradecer a la Mtra. Erandi, por todo el apoyo brindado desde antes de iniciar la MIA. Ha sido parte fundamental en mi estadía.

Quisiera agradecer al Dr. Sergio Hernández, a la Mtra. Erandi Barrientos y al Dr. Saúl Domínguez por sus revisiones y sugerencias que me ayudaron a mejorar este trabajo.

Quisiera expresar mi agradecimiento a Jesús Mejía por estar conmigo en cada

etapa de mi vida académica desde licenciatura. Gracias por todo el amor, por darme las fuerzas para no rendirme y por confiar en mi trabajo cuando ni siquiera yo lo hacía.

Quisiera agradecer al Dr. Porfirio Toledo Hernández por sus consejos y por preocuparse por mí. Sin duda alguna sigue siendo parte fundamental de mi vida académica.

Quisiera agradecer a mis compañeros de generación de la MIA, si bien el confinamiento nos impidió conocernos, fueron parte fundamental en mis inicios.

Este proyecto de investigación ha sido posible debido al respaldo del CONAHCyT a través de una beca para llevar a cabo mis estudios de Maestría.

Resumen

La toma de decisiones jerárquica puede observarse en diferentes áreas de investigación en las que dos niveles anidados definen un problema de optimización. Por ejemplo, en un problema de producción y distribución de una cadena de suministro, dos responsables de la toma de decisiones controlan estos procesos respectivamente. Es decir, una empresa se dedica únicamente a la distribución de bienes o productos, y otra organización tiene como objetivo a la producción óptima de los bienes demandados. Este trabajo trata entonces sobre la solución del Problema Ruteo de vehículos con múltiples depósitos binivel (Bi-MDVRP, por sus siglas en inglés), en el cual múltiples depósitos necesitan entregar productos para cubrir la demanda de múltiples minoristas sujetos a que las plantas de producción satisfagan la demanda de cada depósito de manera óptima. Este problema modelado como un problema binivel resulta ser altamente complejo de resolver desde el punto de vista teórico y computacional. En este trabajo se propone un algoritmo genético híbrido para aproximar soluciones del problema de planificación binivel. Utilizamos la información disponible sobre dicha problemática para implementar mecanismos heurísticos que mejoren los resultados obtenidos por los algoritmos más avanzados. Proponemos una representación basada en grupos para la configuración de cada depósito con el fin de satisfacer las restricciones relacionadas la capacidad de los vehículos, tiempos de carga y otras restricciones en este nivel.

Luego, para resolver el problema de producción, adoptamos un algoritmo exacto para resolver óptimamente este proceso. Los resultados experimentales muestran que la codificación basada en grupos reporta rápidamente soluciones de alta calidad para el conjunto de instancias utilizadas en nuestra experimentación.

Índice general

Resumen	IV
1. Introducción	3
1.1. Planteamiento del problema	5
1.2. Propuesta	6
1.3. Justificación	6
1.4. Hipótesis de la investigación	7
1.5. Objetivos generales	7
1.6. Objetivos específicos	7
1.7. Metodología	8
1.8. Estructura de la Tesis	9
1.9. Contribución	10
1.10. Publicaciones	10
2. Optimización Binivel	13
2.1. Introducción	13
2.2. Definición	15
2.3. Métodos de solución	17

3. Descripción del problema de cadenas de suministro binivel	19
3.1. Bi-MDVRP: Problema de Ruteo de vehículos con múltiples depósitos binivel.	19
3.2. Trabajo relacionado	23
4. Algoritmo Propuesto para Bi-MDVRP	27
4.1. Componentes Clave	28
4.2. Codificación de la solución	30
4.3. Inicialización de la Población	32
4.3.1. Primera variante: Inicialización aleatoria sin sesgo	32
4.3.2. Segunda variante: Inicialización aleatoria con sesgo	33
4.4. Cruza y Mutación en el Nivel Superior	33
4.5. Enrutamiento óptimo	36
4.6. Optimización de Nivel inferior	37
4.7. Complejidad computacional	38
5. Experimentación y Resultados	41
5.1. Instancias del problema	41
5.2. Algoritmos bajo comparación	45
5.2.1. Algoritmo coevolutivo con esquema de migración: CEMBA	45
5.2.2. Algoritmo coevolutivo con esquema de migración híbrido: CEMBA-H	45
5.3. Configuración de parámetros y metodología de prueba estadística	46
5.4. Resultados	48
5.4.1. Resultados obtenidos por ByGA-S	48
5.4.2. Resultados Obtenidos por ByGA-C	51
5.4.3. Número de evaluaciones y tiempo de ejecución	53
5.4.4. Comparación por número de generaciones	54
5.4.5. Visualización de la sub-rutas obtenidas	67
5.5. Discusión	72
5.5.1. Discusión sobre ByGA-S	72
5.5.2. Discusión sobre ByGA-C	72
5.5.3. Discusión sobre ambos algoritmos	73

6. Conclusiones y Trabajo a Futuro	75
6.1. Conclusiones	75
6.2. Trabajo a Futuro	77
Bibliografía	89

Índice de figuras

2.1. Publicaciones de Optimización Binivel [65].	14
3.1. Ilustración del problema Bi-MDVRP [15] e indicación de la solución del problema de nivel superior (delimitada por la línea continua). Obsérvese que el nivel inferior consiste en conectar de forma óptima las plantas (triángulos) con los depósitos (círculos).	22
3.2. Publicaciones por año sobre Bi-MDVRP.	24
4.1. Representación de la solución. Se observan dos depósitos (grupos) con dos vehículos cada uno. Note que cada vehículo tiene asociado una sub-ruta que inicia y termina en el depósito correspondiente.	31
4.2. Operador de Cruza. Observe que se eliminan minoristas en el primer y segundo padre, y se generan dos descendientes, incorporando los minoristas faltantes del segundo padre al primer hijo usando la inserción de costo mínimo. Note que se procede de manera análoga en el segundo hijo.	35

4.3. En el operador de mutación se eliminan y se insertan minoristas. Aquí se observa que se eliminó el minorista 4 y luego se insertó en el segundo depósito, en su segunda sub-ruta, usando la inserción de costo mínimo.	36
4.4. Enrutamiento óptimo. Observe que cada sub-ruta (sub-tour) se representa como una permutación de los minoristas asociados al depósito dado.	38
5.1. Conjunto de instancias 1: Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas.	56
5.2. Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas (conjunto de instancias 2).	57
5.3. Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas (conjunto de instancias 3).	58
5.4. Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas (conjunto de instancias 1).	60
5.5. Conjunto de instancias 2: Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas.	61
5.6. Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas (conjunto de instancias 3).	62
5.7. Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 1).	64
5.8. Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 2).	65
5.9. Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 3).	66
5.10. Visualizando los resultados cuando ByGA-S resolvió la instancia pr1.	67
5.11. Visualizando los resultados cuando ByGA-S resolvió la instancia pr3.	68
5.12. Visualizando los resultados cuando ByGA-S resolvió la instancia pr5.	69
5.13. Visualizando los resultados cuando ByGA-S resolvió la instancia pr7.	70
5.14. Visualizando los resultados cuando ByGA-S resolvió la instancia pr9.	71

Índice de tablas

5.1. Descripción de instancias de Bi-MDVRP para los conjuntos de instancias 1 (<i>pr</i>) - 2 (<i>bipr</i>) [15, 55]	43
5.2. Instancias Bi-MDVRP 2022: conjunto de instancias 3 (<i>p</i>).	44
5.3. Promedio de la función objetivo ByGA-S y valor <i>GAP</i> con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 1).	49
5.4. Promedio de la función objetivo ByGA-S y valor <i>GAP</i> con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 2).	49
5.5. Promedio de la función objetivo ByGA-S y valor <i>GAP</i> con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 3).	50
5.6. Promedio de la función objetivo ByGA-C y valor <i>GAP</i> con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 1).	51
5.7. Promedio de la función objetivo ByGA-C y valor <i>GAP</i> con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 2).	51

5.8. Promedio de la función objetivo ByGA-C y valor <i>GAP</i> con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 3).	52
5.9. Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1).	53
5.10. Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2).	53
5.11. Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3).	54
5.12. Parámetros obtenidos por IRACE con un número máximo de 40 generaciones.	55
5.13. Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.	56
5.14. Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.	57
5.15. Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.	58
5.16. Parámetros obtenidos por IRACE con un número máximo de 100 generaciones.	59
5.17. Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.	60

5.18.Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.	61
5.19.Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.	62
5.20.Parámetros obtenidos por IRACE con un número máximo de 40 generaciones.	63
5.21.Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.	64
5.22.Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.	65
5.23.Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.	66

Lista de Abreviaturas

Note que las abreviaturas están dadas por sus sigas en inglés.

Abreviación	Definición
BO	Optimizacion Binivel
BLOP	Problema de Optimización Binivel
BLOPs	Problemas de Optimización Binivel
VRP	Problema de Ruteo de Vehículos
MDVRP	Problema de Ruteo de Vehículos con Múltiples Depósitos
Bi-MDVRP	Problema de Ruteo de Vehículos con Múltiples Depósitos Binivel
ByGA	Algoritmo Genético Híbrido Binivel (Bilevel Hybrid Genetic Algorithm)
CEMBA	Algoritmo Binivel Coevolutivo con Esquema de Migración

Capítulo 1

Introducción

En optimización y teoría de juegos, es frecuente encontrar situaciones en las que agentes en conflicto toman acciones de acuerdo con una secuencia predefinida (tipo líder-seguidor) de Stackelberg [76]. Al extender este concepto a un par de problemas de optimización arbitrarios, se obtiene la clase de problemas de conocidos como optimización binivel, que permiten modelar muchos procesos de decisión jerárquicos. El término “programación binivel” apareció por primera vez en un artículo de Candler y Norton [19], quienes consideraron una formulación multinivel en el contexto de la economía medioambiental. A principios de los años ochenta se introdujo la programación matemática para el modelado de procesos de decisión jerárquica [64]. Dado que estos problemas de programación binivel, se definen como un problema de optimización que contiene a otro problema de optimización anidado, son altamente costosos computacionalmente debido a su alta dificultad. En dicho trabajo de investigación se centró en resolver problemas con una función objetivo con propiedades lineales, convexas y bilineales en las

restricciones. Así, se ha demostrado que estos problemas son del tipo NP-Duro [51, 39].

Se ha observado un creciente interés tanto desde el punto de vista teórico como práctico en el uso de conceptos de optimización binivel, lo que se refleja en el aumento de problemas modelados de esta manera. Estos problemas tienen diversas aplicaciones en situaciones de la vida real, tales como Economía y Finanzas [82, 85, 76, 81], Economía medioambiental [19], Transporte [82, 75, 35], entre otras áreas [67, 83, 14]. Muchas de estas aplicaciones se desarrollan en espacios discretos, como los problemas de producción y distribución en cadenas de suministro binivel que involucran plantas, centros de distribución y minoristas. Estas aplicaciones se dividen en:

- Producción: se encarga de la fabricación y envío de productos para satisfacer la demanda de los depósitos.
- Distribución y logística: donde se determina el envío de productos desde el depósito a los minoristas.

Un factor importante para el manejo de cadenas de suministros son los costos de transporte, y juegan un papel fundamental en dichos problemas. Estos se clasifican como problemas de ruteo de vehículos (VRP). Tememos que los VRP son problemas de optimización combinatoria y de programación entera que buscan dar servicio a un número de clientes con una flota de vehículos. Este problema inicialmente fue propuesto en 1959 por Dantzig y Ramser [37]. Debido a la complejidad inherente y las restricciones involucradas en la resolución de tales problemas, los VRP y sus variantes entran en la categoría de problemas NP-Duros [57].

Una de las variantes del VRP y que es fundamental en este trabajo es el problema de ruteo de vehículos con múltiples depósitos (MDVRP, por sus siglas en inglés) [54]. Esta versión se caracteriza en asignar minoristas a distintos depósitos y asignarlos a una ruta para satisfacer las demandas de los clientes. Otra de las

variantes de los problemas de ruteo de vehículos, son los que involucran vehículos aéreos no tripulados (UAVs), ya que han demostrado un potencial significativo en la resolución de problemas de optimización binivel [61, 36]. En estos escenarios, los UAVs se han utilizado para abordar diversos desafíos en logística y gestión de cadenas de suministro [58, 86, 58]. Centrándonos en los procesos de cadenas de suministro, el VRP y sus variantes se consideraban como sistemas independientes, es decir, no se involucraban los procesos de producción. En las últimas décadas, los procesos de producción y distribución se fueron integrando en un mismo problema [45, 44], siendo modelados como problemas multi-objetivos y resueltos mediante métodos exactos [8] y heurísticos [32, 12],

Recientemente, el problema de planificación de una cadena de suministros se ha modelado a través de una estructura jerárquica, es decir, utilizando la definición de optimización binivel. Específicamente el problema de producción y distribución binivel es conocido como Bi-MDVRP, por sus siglas en inglés [15]. Dentro del estudio de la literatura en optimización binivel discreta se reporta en los últimos años un incremento en el interés por resolver esta clase de problemas [74, 28]. Cabe mencionar que la resolución del problema Bi-MDVRP en su mayoría son abordados por métodos aproximados (metaheurísticas) [74, 22, 24] y métodos híbridos conocidos como mateheurísticas para obtener mejores resultados [17, 18].

1.1 Planteamiento del problema

Los problemas de producción y distribución de una cadena de suministro generalmente involucran distintos tomadores de decisiones en los que incluyen la ubicación de las plantas o depósitos, la adquisición de productos, el costo de producción, la entrega de productos, entre otros aspectos. Dichas decisiones suelen estar vinculadas, pero en situaciones prácticas se suelen resolver por separado, utilizando métodos de optimización mono-objetivo y multi-objetivo. Sin embargo, estas tareas involucran tomadores de decisiones en dos niveles distintos, con

una relación jerárquica entre ellos. Así, en el nivel superior del proceso de decisión controla los centros de distribución (conjunto de depósitos) que buscan minimizar los costos de distribución de productos a los minoristas que los demandan y el costo de adquisición de los productos. En el nivel inferior, cada planta de fabricación, al recibir el pedido de los centros de distribución, busca minimizar sus costos de operación y de envío de los productos a los depósitos.

1.2 Propuesta

En este trabajo se propone un algoritmo genético híbrido binivel para la resolución del problema Bi-MDVRP, en donde el nivel superior se resuelve mediante un algoritmo genético y el nivel inferior a través de un método exacto aplicable.

1.3 Justificación

Los problemas de optimización binivel en espacios discretos están teniendo un papel fundamental y desafiante para el cómputo evolutivo debido a su estructura jerárquica y por ser altamente costosos para encontrar una solución óptima. En el problema Bi-MDVRP nos encontramos con la complejidad propia del problema MDVRP y al anidar un segundo problema se vuelve aún más complejo.

Si bien la literatura especializada reporta algoritmos enfocados en resolver el problema Bi-MDVRP, se tiene la particularidad que utilizan métodos exactos para resolver diferentes partes del problema, lo que los hace limitados a instancias de problemas pequeños [18]. Además, los algoritmos metaheurísticos utilizan representaciones de solución que no toman en cuenta la estructura binivel ni la información conocida del problema [74, 28]. Por este motivo, se busca proponer un algoritmo genético híbrido binivel que tenga resultados de mejor calidad en la función objetivo en los conjuntos de instancias de prueba reportados en la literatura.

1.4 Hipótesis de la investigación

El diseño de un Algoritmo Genético Híbrido con una representación basada en grupos puede ofrecer resultados competitivos en comparación con algoritmos reportados en la literatura especializada (métodos metaheurísticos e híbridos) en el Problema Bi-MDVRP. La comparación se hará en términos de la precisión (cercanía al valor óptimo o mejor solución conocida) y del indicador de desempeño conocido como GAP.

Note que en este trabajo las variables independientes del estudio son los: (1) componentes del algoritmo: representación de soluciones basada en grupos, operadores de selección, método exacto y (2) parámetros del algoritmo. Las variables dependientes están dadas por: (1) cercanía a la mejor solución conocida e (2) indicador de desempeño GAP.

1.5 Objetivos generales

Desarrollar una versión del algoritmo genético híbrido (usando el esquema anidado) con una representación eficiente para resolver el problema Bi-MDVRP con resultados competitivos con un menor número de evaluaciones que algoritmos evolutivos del estado del arte.

1.6 Objetivos específicos

- Estudiar las propiedades de los problemas de optimización binivel discreta.
- Estudiar las diferentes representaciones de soluciones, por ejemplo, aquellas basadas en representación de grupos para el problema Bi-MDVRP. Debido a que no se encontró evidencia de estas representaciones se estén usando en dicho problema.

- Identificar algoritmos propuestos por otros autores que han mostrado un buen desempeño al resolver problemas Bi-MDVRP.
- Elegir conjuntos de instancias de prueba en la literatura especializada.
- Creación de instancias de prueba.
- Implementar un nuevo algoritmo que permita tener un desempeño competitivo.
- Identificar el costo computacional del algoritmo en términos de complejidad computacional.
- Probar la eficiencia del algoritmo a desarrollar en dicho conjunto de instancias de prueba.

1.7 Metodología

En esta sección se describe la metodología para la propuesta e implementación de un algoritmo genético híbrido binivel con representación basada en grupos para resolver el problema Bi-MDVRP y analizar su desempeño en términos del promedio del valor de la función objetivo, el número de la función de evaluación y tiempo de ejecución.

- Revisión de la literatura sobre el problema de ruteo de vehículos con múltiples depósitos binivel.
- Selección de conjuntos de instancias, las cuales consisten en términos del tamaño de vértice, estructura topológica de las plantas y depósitos, y capacidad de los vehículos y plantas. Para el problema a atacar, se eligen dos conjuntos de instancias, el conjunto **pr** y **bipr** que se encuentran reportados en la literatura por Herminia I. Calvete et al. [15] y Francois Legillon et. al. [55] respectivamente. Estos dos conjuntos de instancias son creados a partir del conjunto de instancia **pr** proporcionados por Cordeau [33] siguiendo

el enfoque binivel de Herminia I. Calvete et al. [15]. En la Sección 5.1 se muestra las diferencias entre el conjunto de instancias **pr** y **bipr**.

- Creación de instancias de prueba utilizando conjunto p creados a partir de los proporcionados por Cordeau [33] siguiendo el enfoque binivel de Herminia I. Calvete et al. [15] con los parámetros de Francois Legillon et. al. [55]. En la Sección 5.1 se muestran las diferencias entre los tres conjuntos de instancias a utilizar (instancias reportadas en la literatura (pr y $bipr$) y creadas (p)).
- Identificación del conjunto de técnicas con buen desempeño para Bi-MDVRP.
- Creación del diseño del Algoritmo híbrido binivel con una representación de soluciones basada en grupos.
- Aplicación del algoritmo a las instancias de prueba para obtener los siguientes objetivos: (1) promedio de la función objetivo, (2) número mínimo de evaluaciones, y (3) reducir el tiempo de ejecución.
- Análisis del rendimiento mediante experimentos numéricos.

1.8 Estructura de la Tesis

El documento se encuentra organizado de la siguiente manera: En el Capítulo 2 se describen conceptos de optimización binivel y sus métodos de solución. En el Capítulo 3 se describe el modelado matemático y el trabajo relacionado del Problema Bi-MDVRP. En el Capítulo 4 se describe el algoritmo genético híbrido de optimización binivel propuesto. El Capítulo 5 se muestra la experimentación computacional y se describe el conjunto de instancias de prueba. Finalmente, se presenta el análisis de los resultados obtenidos. Por último, en el Capítulo 6 se presentan las conclusiones del trabajo de investigación y trabajo futuro.

1.9 Contribución

Propuesta de un algoritmo genético híbrido para resolver el problema de planificación y producción binivel (Bi-MDVRP). Esta propuesta es la primera que emplea una representación de la solución basada en grupos para establecer una configuración factible para cada depósito. En consecuencia, se satisfacen las restricciones relacionadas con un depósito.

1.10 Publicaciones

Esta sección contiene las presentaciones y los productos de investigación derivados de este trabajo. Las publicaciones se dividen en dos partes, (1) artículo de conferencia, (2) capítulo de libro. También se agregan los eventos y foros en los que se participó.

Artículo en Congreso Internacional

1. Rocio Salinas-Guerra, Efrén Mezura-Montes, Marcela Quiroz-Castellanos, Jesus-Adolfo Mejia-de-Dios, and Slim Bechikh, **A Hybrid Evolutionary Approach with Group-based Solution Encoding for Solving the Constrained Bilevel Multi-Depot Vehicle Routing Problem**, in Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Chicago, USA, IEEE Press, 2023 (accepted).

Capítulo de libro

1. Salinas-Guerra, R., Mejía-Dios, JA., Mezura-Montes, E., Márquez-Grajales, A. (2023). **An Evolutionary Bilevel Optimization Approach for Neuroevolution**. In: Castillo, O., Melin, P. (eds) Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics.

Studies in Computational Intelligence, vol 1096. Springer, Cham. https://doi.org/10.1007/978-3-031-28999-6_25.

Presentaciones en foros especializados y de divulgación

1. Rocío Salinas-Guerra, Efrén Mezura-Montes, Marcela Quiroz-Castellanos, Héctor-Gabriel Acosta-Mesa, **Optimización Binivel Discreta y sus Aplicaciones**, Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2021), Ciudad de Mexico (online).
2. Rocío Salinas-Guerra, Efrén Mezura-Montes, **Un enfoque binivel para la optimización de redes neuronales convolucionales**, Annual Meeting of the Society for Industrial and Applied Mathematics Mexico Section (MexSIAM 2022), Saltillo Coahuila.
3. Rocío Salinas-Guerra, Efrén Mezura-Montes, **Producción y distribución como un modelado Binivel**, Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2022), Facultad de Matemáticas UADY Merida Yucatan.
4. Rocío Salinas-Guerra, Efrén Mezura-Montes, **An Evolutionary Bilevel Optimization Approach for Neuroevolution**, International Seminar of Computational Intelligence (ISCI 2022), Tijuana, Mexico (online).

Capítulo 2

Optimización Binivel

La Optimización Binivel (OB) es un problema de optimización en el que algunas variables están restringidas para ser la solución óptima de otro problema de optimización. De esta forma, la optimización binivel se puede utilizar para modelar relaciones jerárquicas entre dos tomadores de decisiones. En este capítulo se presenta una descripción general sobre OB. Además, se describen algunas aplicaciones de este tipo de problemas en situaciones de la vida real, donde las variables están definidas sobre espacios discretos.

2.1 Introducción

Un problema binivel es un caso particular de problema de múltiples niveles. La optimización multinivel aborda problemas en los que varios (dos o más) tomadores de decisiones interactúan jerárquicamente y en cada subnivel hay un toma-

dor de decisiones que observa las decisiones de los niveles superiores, y actúan en función a sus propios intereses. A diferencia de los problemas de optimización multinivel, en los problemas de optimización binivel (BLOP, por sus siglas en inglés), hay dos tomadores de decisiones, el líder y seguidor; y se optimiza la estrategia del líder, minimizando (maximizando) la función objetivo del líder sobre el conjunto de estrategias factibles, teniendo en cuenta que el seguidor siempre reacciona de manera óptima a la decisión del líder.

Desde un punto de vista histórico, la OB está relacionada con los juegos de Stalckelberg en el campo de teoría de juegos. El cual fue introducido por Von Stalckelberg con el problema en economía [76] que se utilizó para modelar la interacción entre dos empresas que compiten secuencialmente en la cantidad de producción. Bracken y McGill introducen formalmente la definición de optimización binivel en 1973 [13] y en 1977 las denominaciones binivel y multinivel son adoptadas [19].

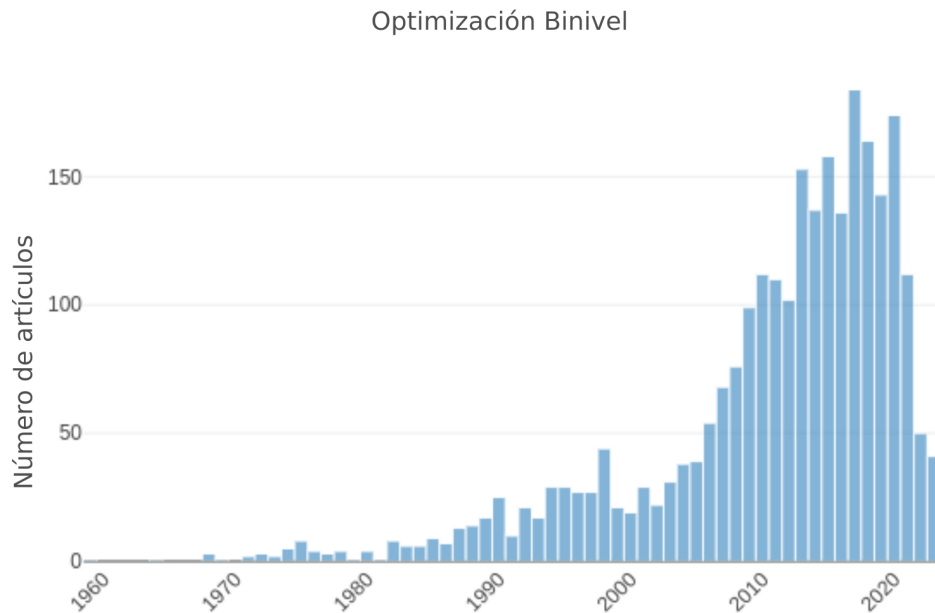


Figura 2.1: Publicaciones de Optimización Binivel [65].

Al paso de los años, la investigación de los problemas de optimización binivel se ha incrementado tal y como se muestra en la Figura 2.1, habiendo más de 2000 artículos publicados hasta la fecha [65]. Se ha observado que los problemas de optimización binivel en espacios continuos han sido muy estudiados en comparación con problemas binivel que involucran variables discretas [66]. Por lo que esto motiva aún más realizar trabajos en este tipo de problemas binivel. Para obtener una descripción general de OB, existen trabajos relacionados en sus distintos enfoques mono-objetivo y multi-objetivo se recomiendan los siguientes artículos [66, 64, 63, 42] y los libros del área encontramos los de Talbi [77], Bard [10] y Dempe [40, 41].

2.2 Definición

Como se mencionó anteriormente, un problema de optimización binivel es un problema de optimización en el cual un subconjunto de variables está sujeto a tomar el valor de una solución óptima de otro problema de optimización dado y parametrizado por las variables restantes. En términos matemáticos, un problema de optimización binivel puede ser definido de la siguiente manera [40].

Definition 2.2.1 (Problema de optimización binivel). Supongamos que $F : X \times Y \rightarrow \mathbb{R}$ y $f : X \times Y \rightarrow \mathbb{R}$ son la función objetivo del nivel superior (líder) e inferior (seguidor), respectivamente.

$$\text{mín}\{F(x, y) : x \in X, y \in Y\} \quad (2.1)$$

Sujeto a:

$$y \in \arg \text{mín}\{f(x, y) : g_j(x, y) \leq 0, j = 1, \dots, J\} \quad (2.2)$$

$$G_k(x, y) \leq 0, k = 1, \dots, K \quad (2.3)$$

Donde, $g_j : X \times Y \rightarrow \mathbb{R}$ y $G_k : X \times Y \rightarrow \mathbb{R}$ denotan la restricción del nivel inferior y superior, respectivamente.

En optimización binivel existen diferentes tipos de problemas según el número y tipo de objetivos, tipos de restricciones, entre otras propiedades. Anteriormente, se describió el caso continuo para funciones mono-objetivo.

- **El número de objetivos manejados por ambos niveles de optimización**

Esta clasificación se realiza en función del criterio del número de funciones objetivo, se derivan dos tipos de BLOPs: (1) optimización binivel mono-objetivo, estos problemas en ambos niveles únicamente tienen un objetivo y (2) optimización binivel multi-objetivo, este tipo de problemas aumenta significativamente la complejidad, ya que las funciones del nivel superior o inferior podrían representar un problema multi-objetivo el cual induce complejidad tanto teórica como computacional. Muchos problemas reales modelan de esta manera [84].

- **El tipo de variables codificadas**

Esta clasificación utiliza como criterio el tipo de variables codificadas, (1) BLOP continuo en el que las funciones objetivo F y f se definen sobre $X \subseteq \mathbb{R}^n$ y $Y \subseteq \mathbb{R}^m$ con su respectiva imagen en \mathbb{R} [9], es decir, las variables del nivel superior e inferior se codifican con valores reales, (2) BLOP discreto, en este tipo de problema F y f están definidas sobre $X \subseteq \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ y $Y \subseteq \mathbb{R}^{m_1} \times \mathbb{Z}^{m_2}$ con su respectiva imagen en \mathbb{R} [80], donde sus variables son discretas en ambos niveles. Además, se pueden clasificar en BLOP mixtos que combinan variables de decisión tanto continuas como discretas.

- **La naturaleza de las funciones objetivo y las restricciones**

Esta clasificación tiene en cuenta la naturaleza de las funciones objetivo y restricciones, es decir, las propiedades matemáticas de las restricciones y las funciones objetivo. Tales como linealidad, convexidad, si las funciones son cuadráticas, entre otras propiedades [11, 77].

- **La relación entre los dos niveles**

Su clasificación se basa en la relación entre el nivel superior y el inferior. Esta relación está determinada por el comportamiento de nivel inferior. (1)

Optimista, el seguidor le da al líder aquellas soluciones del nivel inferior que benefician el problema del nivel superior, es decir, si el nivel inferior contiene múltiples soluciones óptimas, le brinda al líder aquellas soluciones que minimizan la función objetivo del nivel superior [68, 7], (2) Pesimista, el problema de nivel superior se optimiza en función del peor caso en el que el problema de nivel inferior elige una solución que conduce al peor valor de la función del nivel superior. Es importante señalar que la posición pesimista es, en general, menos tratable que la optimista, ya que es un problema que en varios casos no se comporta bien porque introduce funciones no tratables durante las reformulaciones [59, 60].

Respecto a la complejidad computacional de los problemas binivel, se ha estudiado desde los orígenes [47]. La literatura especializada reporta que los problemas binivel pertenecen a la clase de problemas NP-Duros, ya que no es posible resolver este tipo de problemas mediante una máquina de Turing no determinista en tiempo polinomial [79].

2.3 Métodos de solución

Los BLOP representan una clase conocida de problemas desafiantes con dos problemas de optimización anidados [40]. El objetivo principal consiste en optimizar el problema de nivel superior que tiene como restricción otro problema de optimización. Por lo tanto, evaluar cada solución de nivel superior requiere encontrar una solución óptima del correspondiente problema de nivel inferior, lo cual implica un alto costo computacional. En el campo de la optimización binivel, se han desarrollado varios métodos que operan utilizando esta estructura anidada [77]. Además, se han propuesto métodos de solución no anidados que convierten el problema binivel en un problema de un solo nivel [78]. Sin embargo, la mayoría de estos enfoques están limitados a una clase de problemas altamente restrictiva o son computacionalmente costosos, lo que dificulta su aplicación a BLOPs más complicados. Esta situación ha llevado a la comunidad de la computación evolu-

tiva a abordar este tipo de problemas, y recientemente se han propuesto varios trabajos en este ámbito. Se han identificado los siguientes enfoques para la resolución de problemas bi-nivel mediante el uso de enfoques metaheurísticos [66], y a continuación se enumeran los más relevantes:

1. **Reducción a un nivel:** Transforman el BLOP en un problema de un solo nivel para neutralizar el efecto binivel. Estos métodos utilizan una metodología aproximada para reemplazar el problema de optimización de nivel inferior [29, 72].
2. **Enfoques anidados:** Son uno de los métodos conocidos propuestos para resolver problemas binivel, donde para cada solución de nivel superior se ejecuta un optimizador de nivel inferior. En esta categoría, los algoritmos de optimización jerárquica intentan resolver los dos niveles secuencialmente, mejorando las soluciones en cada nivel para obtener una buena solución general en ambos niveles [53].
3. **Enfoques no anidados:** Los enfoques no anidados abordan el problema de optimización binivel tratándolo como un único nivel y se resuelve utilizando metaheurísticas, sin considerar la estructura anidada. Estas técnicas emplean métodos de optimización de un solo nivel y utilizan el conocimiento del problema de nivel inferior para evitar optimizaciones explícitas. Se logra esto al tener en cuenta la relación entre los dos niveles e incorporar restricciones o información adicional en el problema de nivel superior [49, 50].
4. **Enfoques asistidos por subrogados:** Estas aproximaciones utilizan un esquema anidado como forma de solucionar problemas. No obstante, el problema del nivel inferior no se aborda a través de otra metaheurística, sino que se utiliza en su lugar una estrategia aproximada para deducir soluciones viables durante el proceso de optimización. Esta estrategia aproximada resulta especialmente valiosa para reducir el número de evaluaciones de la función objetivo [46, 38].

Capítulo 3

Descripción del problema de cadenas de suministro binivel

En esta sección, se plantea el problema de estudio y se describe su modelado matemático binivel y la revisión de la literatura del problema de ruteo de vehículos con múltiples depósitos binivel (Bi-MDVRP, por sus silgas en inglés).

3.1 Bi-MDVRP: Problema de Ruteo de vehículos con múltiples depósitos binivel.

El problema planteado considera una situación en la que dos de empresas (depósitos y plantas) interactúan entre sí de forma jerárquica dentro de una cadena de suministro, lo cual funciona de la siguiente manera. El depósito adquiere y distribuye diferentes tipos de productos sobre un subconjunto seleccionado

de minoristas; mientras que la planta fabrica los productos demandados por los depósitos. En este problema, se asume que los depósitos son los que distribuyen los productos y diseñan rutas para satisfacer al subconjunto seleccionado de minoristas con el objetivo de minimizar el costo de entrega y las plantas son las encargadas de fabricar los productos buscan minimizar el costo de producción y envío de los productos a los depósitos.

Consideremos los siguientes parámetros, K , L , R , S son el conjunto de plantas, depósitos, minoristas y vehículos terrestres con sus índices k , l , r y s respectivamente. Sea E el conjunto de vértices que conectan los depósitos y los minoristas. Además, el parámetro R_l representa si el depósito l sirve a los minoristas y R_s determina si el vehículo s los visita. Tenemos que S_l representa el conjunto de vehículos utilizados por cada depósito l , b_r la demanda del minorista r y A_k la capacidad de producción de la planta k .

Los parámetros asociados en las funciones objetivo de nivel superior e inferior vienen dados por: c_{jl}^{11} que es el costo de enviar un artículo de i a j siendo $i \in E$, c_{kl}^{12} el costo de adquirir un artículo más el costo de descargarlo de la planta k al almacén l y c_{kl}^{22} el costo de producción de un artículo, que se produce en la planta k y se envía al almacén l . También necesitamos denotar la capacidad Q de cada vehículo y los tiempos, de tal forma que, t_{ij} es el tiempo de traslado de ir del origen i al destino j , t_r el tiempo de descarga de un vehículo en el minorista r y T_s el tiempo límite del vehículo s .

La siguiente formulación plantea el problema Bi-MDVRP [15, 18].

Nivel Superior

$$\min \sum_{s \in S} \sum_{(ij) \in E} c_{ij}^{11} x_{ij}^s + \sum_{k \in K} \sum_{l \in L} c_{k,l}^{12} y_{kl} \quad (3.1)$$

$$s.t. \{x_{ij}^s\} \text{ satisface las restricciones del MDVRP} \quad (3.2)$$

$$x_{ij}^s \in \{0, 1\} \quad (ij) \in E, s \in S \quad (3.3)$$

$$\sum_{s \in S} \sum_{j \in R \cup L} x_{ij}^s = 1 \quad i \in R \quad (3.4)$$

$$\sum_{s \in S} \sum_{i \in R \cup L} x_{ij}^s = 1 \quad j \in R \quad (3.5)$$

$$\sum_{j \in R_l} x_{ij}^s \leq 1 \quad s \in S_l, i, l \in L \quad (3.6)$$

$$(3.7)$$

$$\sum_{i \in R_l} x_{ij}^s \leq 1 \quad s \in S_l, j, l \in L \quad (3.8)$$

$$\sum_{j \in L} x_{ij}^s = 0 \quad s \in S_l, i, l \in L \quad (3.9)$$

$$\sum_{i \in L} x_{ij}^s = 0 \quad s \in S_l, j, l \in L \quad (3.10)$$

$$\sum_{i \in W} \sum_{j \in W} x_{ij}^s \leq |W| - 1 \quad W \subseteq R, 2 \leq |W| \leq |R|, s \in S \quad (3.11)$$

$$\sum_{i \in R_s} b_i \sum_{j \in R_s} x_{i,j}^s \leq Q \quad s \in S \quad (3.12)$$

$$\sum_{i \in R_s} \sum_{j \in R_s} (t_{ij} + \tau_i) x_{ij}^s \leq T_s \quad s \in S \quad (3.13)$$

Las Ecuaciones (3.1-3.13) son el problema líder y las ecuaciones definidas por las Ecuaciones (3.15-3.17) son el problema seguidor. (3.1) Representa la función objetivo del líder, el cual busca minimizar el costo de distribución de los productos de los depósitos a los minoristas más el costo de adquirir un producto desde una planta hacia los depósitos y el costo de descarga de un producto, (3.2) describe la variable binaria del líder, (3.3) representa que solo puede salir un vehículo del minorista l , (3.4) solo puede llegar un vehículo al minorista l , (3.5) y (3.6) establecen que los depósitos pueden ser o no utilizados, (3.9) nos dice que no puede salir ningún vehículo de un depósito para conectar con otro depósito, (3.10) indica que no puede llegar un vehículo a un depósito de otro depósito, es decir, que no existe conexión entre depósitos. La restricción (3.11) no permite sub-tours dentro

de un vehículo, i.e., cada vehículo le corresponde a un depósito. En (3.12) nos indica la capacidad de cada vehículo y en (3.13) el tiempo que no debe exceder el vehículo en realizar las rutas en donde se debe considerar el tiempo de descarga y el traslado de un origen a destino.

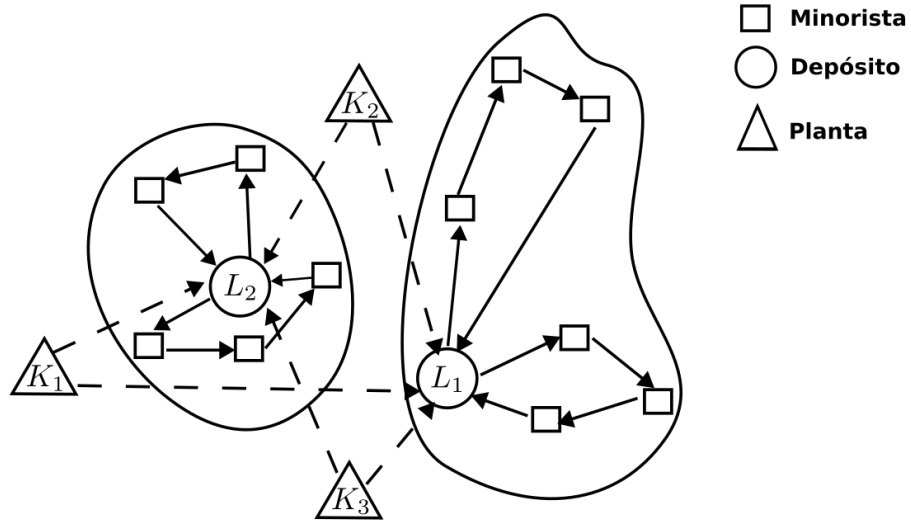


Figura 3.1: Ilustración del problema Bi-MDVRP [15] e indicación de la solución del problema de nivel superior (delimitada por la línea continua). Obsérvese que el nivel inferior consiste en conectar de forma óptima las plantas (triángulos) con los depósitos (círculos).

Nivel Inferior

$$y \in \arg \min \sum_{k \in K} \sum_{l \in L} c_{k,l}^{22} \bar{y}_{k,l} \quad (3.14)$$

$$s.t. \sum_{l \in L} \bar{y}_{kl} \leq A_k \quad k \in K \quad (3.15)$$

$$\sum_{k \in K} \bar{y}_{kl} = \sum_{s \in S_l} \sum_{r \in R_s} b_r \quad l \in L \quad (3.16)$$

$$\bar{y}_{kl} \geq 0 \quad k \in K, l \in L \quad (3.17)$$

La restricción (3.14) representa la función objetivo del seguidor, donde se busca minimizar el costo de producción y envío de los productos fabricados a los depósitos, (3.15) indica la capacidad de producción de la planta y la restricción (3.16) es donde se involucra el nivel inferior con el nivel superior, en esta restricción se busca satisfacer las demandas de los minoristas atendidos por cada depósito y la restricción (3.17) establece la no negatividad de las variables de nivel inferior.

Para ilustrar el problema Bi-MDVRP, se visualiza en la Figura 3.1, que incluye la solución del Bi-MDVRP con dos depósitos, once minoristas y tres plantas.

3.2 Trabajo relacionado

En los últimos años, la optimización binivel discreta ha emergido como un área importante de investigación en optimización. Así, en esta sección se revisan los trabajos relacionados con la solución de la optimización bilevel discreta, centrándose en la solución del problema de ruteo de vehículos con múltiples depósitos binivel (Bi-MDVRP).

En la revisión de la literatura se encuentran distintos enfoques heurísticos para resolver el problema Bi-MDVRP, cabe destacar que cada algoritmo resuelve un conjunto de instancias distinto [15, 20]. Por otro lado, los conjuntos de instancias contribuyen al análisis de las bondades y desventajas de los distintos algoritmos y con esta información la creación de nuevas estrategias que permiten la mejora de los algoritmos para obtener soluciones de mayor calidad. Si nos centramos en las publicaciones de los distintos conjuntos de instancias inspirados con el modelado binivel creado por Hermina et al [15] notemos que desde el 2011 hasta la fecha ha ido incrementando la investigación sobre el problema Bi-MDVRP y extendido a problemas multiobjetivo como se muestra en la Figura 3.2. Los cuales se clasifican de la siguiente manera, siendo un total de 22 publicaciones:

1. Congreso: 11

- Mono-Objetivo: 9 [48, 16, 22, 20, 23, 24, 25, 27, 3]
- Multi-Objetivo: 3 [55, 5, 2]

2. Revista: 8

- Mono-Objetivo: 7 [15, 18, 74, 26, 28, 21, 4]
- Multi-Objetivo: 2 [73, 1]

3. Capítulo de libro: 1

- Mono-Objetivo: 0
- Multi-Objetivo: 1 [56]

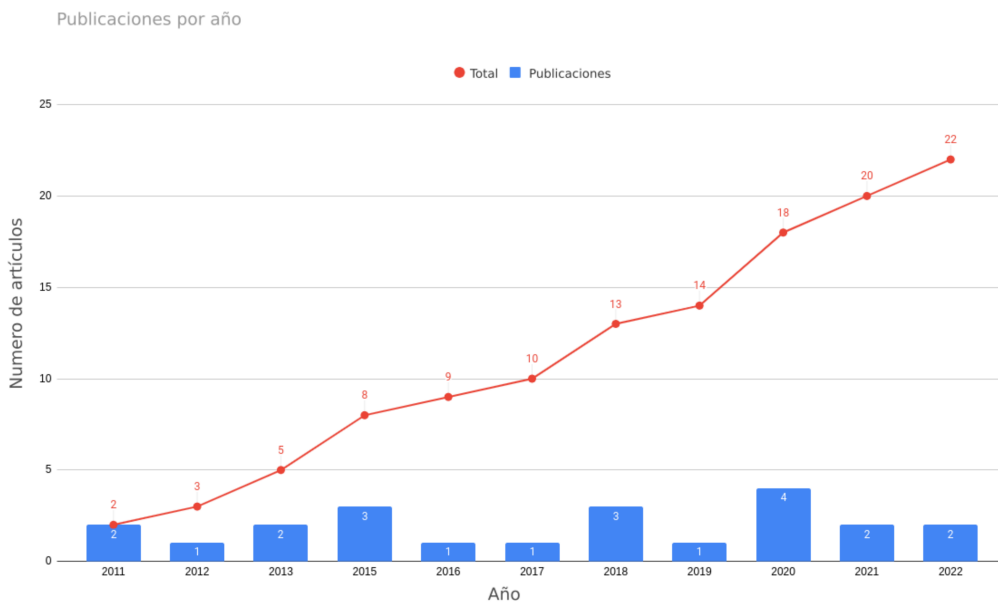


Figura 3.2: Publicaciones por año sobre Bi-MDVRP.

Uno de los primeros algoritmos para el problema Bi-MDVRP fue propuesto en [15], donde se introdujo el Bilevel Ant Colony System (BACS). BACS se basa en la estrategia de optimización de colonias de hormigas, pero utiliza un enfoque Stackelberg para manejar la estructura binivel. La representación de la solución

utilizada en el enfoque BACS se basa principalmente en cadenas binarias para representar las conexiones entre minoristas, depósitos y plantas.

Otro trabajo relacionado se presentó como un algoritmo híbrido basado en el procedimiento de búsqueda dispersa (HSS) [18]. En esta propuesta de solución, los autores consideran un cierto grado de aleatoriedad en la búsqueda para generar soluciones factibles. HSS es una de las propuestas más sofisticadas porque incorpora técnicas clásicas para generar sub-rutas óptimas. El software comercial CPLEX resuelve el nivel inferior mediante programación lineal.

En cuanto a los algoritmos evolutivos, en [55] se propuso un método coevolutivo de dos niveles que utiliza algoritmos repetidos (denominado COBRA). Este algoritmo coevolutivo se basa en la proximidad de los valores óptimos de las variables de nivel superior al valor óptimo de las variables de nivel inferior. COBRA representa las soluciones de nivel superior mediante una matriz de permutación que concatena las rutas para cada vehículo y para cada depósito. Esta representación es útil porque se satisfacen muchas restricciones (por ejemplo, evita visitar a minoristas repetidos).

Posteriormente, en [22] se propuso un algoritmo coevolutivo diferente basado en la descomposición (COBDA). COBDA pretendía generar una solución inicial prometedora aplicando una estrategia de descomposición en el espacio de búsqueda. Se presentó una versión actualizada de COBDA con soluciones mejoradas mediante aprendizaje por transferencia como Algoritmo Evolutivo Memético (MCOBDA) [24] mejorando las soluciones (representadas como permutaciones) mediante aprendizaje por transferencia para reducir el esfuerzo computacional. Otra versión de COBDA fue presentada, introduciendo conceptos sobre reacciones químicas [28]. En ese último trabajo, los autores descompusieron la población de moléculas en varias subpoblaciones distribuidas evaluadas en una arquitectura computacional multihilo.

Uno de los algoritmos coevolutivos más recientes (denominado CEMBA) fue propuesto en [74]. CEMBA utiliza la descomposición binaria de la población en ambos niveles para reducir el número de evaluaciones de funciones de nivel su-

terior. Cabe mencionar que CEMBA utiliza dos algoritmos coevolutivos anidados. Posteriormente, se propuso una versión mejorada de CEMBA para resolver problemas combinatorios multiobjetivo [73]. Los vectores de solución de nivel superior en esos algoritmos basados en CEMBA se codifican como permutaciones como en COBRA.

En resumen, la mayoría de las propuestas utilizan una representación basada en permutaciones para codificar las soluciones del nivel superior en un marco coevolutivo, principalmente para reducir el costo computacional relacionado con la evaluación del problema de optimización del nivel inferior.

A diferencia de la propuesta anterior, proponemos un algoritmo genético híbrido para resolver el mencionado problema de planificación y producción binivel (Bi-MDVRP). Utilizamos la información disponible sobre el problema para implementar mecanismos heurísticos que mejoren los resultados obtenidos por los algoritmos más avanzados. En la literatura no se ha reportado el uso de una representación de solución basada en grupos. Por lo tanto, esta propuesta es la primera en utilizar dicha representación para la configuración factible de cada depósito. Gracias a esta codificación, se satisfacen las restricciones relacionadas con cada depósito. Los resultados experimentales muestran que la codificación basada en grupos obtiene rápidamente soluciones de alta calidad para los conjuntos de instancias.

Capítulo 4

Algoritmo Propuesto para Bi-MDVRP

Antes de presentar nuestra propuesta llamada “Bilevel Hybrid Genetic Algorithm” (ByGA), es importante mencionar que en este capítulo se propone un marco de trabajo y del mismo se establecen dos variantes denotadas por ByGA-S y ByGA-C. La primera variante ByGA-S del algoritmo, es más sencilla que ByGA-C, pues tiene una inicialización simple que genera soluciones aleatorias sin sesgo. Mientras que ByGA-C implementa un proceso de inicialización más sofisticado e implementa un mecanismo para enrutar óptimamente minoristas asignados a un depósito.

Antes de describir la propuesta, describamos los componentes del Bi-MDVRP donde se pueden aplicar estrategias heurísticas considerando la información que se conoce acerca del problema.

- **Asignación de minoristas a depósitos:** La asignación de los minoristas a un depósito es una de las tareas más importantes porque puede repercutir en los costos de distribución o adquisición. Es decir, un minorista puede aumentar el costo de distribución si se encuentra muy lejos del depósito asignado, implicando un alto valor en la función objetivo.
- **Generación de rutas (rutas óptimas) para cada depósito:** Una vez que un minorista se asigna de forma óptima a un depósito, es necesario insertarlo en la ruta óptima teniendo en cuenta la capacidad y el tiempo del vehículo asignado.
- **Optimizador de nivel inferior:** Por último, una vez que todos los minoristas han sido asignados a los depósitos, y han sido enrutados de manera óptima, tenemos que enviar las demandas de los depósitos al optimizador de nivel inferior, que tiene que minimizar el costo de producción y el costo relacionado con la entrega a los depósitos, considerando la demanda requerida.

Tomando en cuenta lo anterior, en la siguiente sección se describe cada componente clave de la propuesta donde se abordan los puntos descritos.

4.1 Componentes Clave

Teniendo en cuenta los puntos anteriores, podemos implementar mecanismos heurísticos para manejar cada componente mencionado.

- En primer lugar, para asignar los minoristas al depósito, proponemos un algoritmo genético de agrupación (GGA, por sus siglas en inglés) considerando cada depósito como un grupo, y utilizando operadores genéticos de agrupación para generar nuevas soluciones [69].
- En segundo lugar, necesitamos generar de forma óptima cada ruta para minimizar el costo de distribución (esto no afecta al costo de adquisición).

Aquí, en una segunda versión del algoritmo, sugerimos utilizar un enrutador basado en un algoritmo genético anidado para generar las rutas óptimas para cada depósito.

- Una de las partes más importantes es elegir el optimizador de nivel inferior para satisfacer la viabilidad relacionada con el problema de nivel inferior. Una vez que el algoritmo genético del nivel superior sugiere una solución (depósitos con rutas de costo mínimo), el nivel inferior toma las demandas como restricciones para minimizar el costo de producción. Dado que el espacio de búsqueda del nivel inferior tiene un tamaño igual al número de depósitos multiplicado por el número de plantas y que la función objetivo pertenece a un problema de programación lineal, podemos utilizar un optimizador exacto. Es importante mencionar que si el problema de nivel inferior no tuviera sus características matemáticas conocidas, entonces sería necesario utilizar una búsqueda heurística para aproximar soluciones en ese nivel.

Considerando los elementos mencionados arriba, proponemos un marco de trabajo denotado por ByGA y dos variantes con las siguientes características.

1. ByGA-S: Esta versión del algoritmo se caracteriza por tener la primera propuesta de inicialización descrita en el apartado 4.3.1.
2. ByGA-C: Esta versión del algoritmo es una mejora del ByGA-S la cual se destaca por tener como inicialización la segunda propuesta descrita en el apartado 4.3.2 y el algoritmo genético local incorporado para el enrutamiento óptimo descrito en el apartado 4.5.

Note que ByGA-S y ByGA-C comparten los mismos componentes a excepción de la inicialización y el generador de sub-rutas óptimas. El marco de trabajo de nuestra propuesta se describen en el Algoritmo 1 resume cada componente de la propuesta y cada componente de nuestra propuesta se detalla las siguientes secciones.

Algoritmo 1: ByGA

- 1 Generar población inicial P .
 - 2 **mientras** no se cumplan los criterios de detención **hacer**
 - 3 Seleccione dos padre usando un torneo binario (con respecto la función del nivel superior).
 - 4 Cree un conjunto vacío $Q = \emptyset$.
 - 5 Cree soluciones candidatas (depósitos configurados) e insértelas en Q .
 - 6 Para cada solución de nivel superior, optimice el problema de nivel inferior correspondiente.
 - 7 Coloca S como resultado de unir P y Q .
 - 8 Realice un enrutamiento óptimo a una solución aleatoria en S .
 - 9 Establezca P como las mejores soluciones en S .
-

Algoritmo 2: Inserción de costo mínimo.

- 1 **Entrada:** $L_s = \{L_1, L_2, \dots, L_{|L|}\}$ un conjunto de $|L|$ depósitos, y minoristas r a asignar.
 - 2 **Salida:** L_s que contiene al minorista r
 - 3 **mientras** r no esté asignado a un depósito en L_s **hacer**
 - 4 Seleccione aleatoriamente un depósito L_i de L_s .
 - 5 **para** cada vehículo v en el depósito L_i **hacer**
 - 6 **si** insertar r en v no viola ninguna restricción del vehículo. **entonces**
 - 7 Inserta r en la mejor posición (costo mínimo de distribución) de la ruta del vehículo v que se encuentra en L_i .
 - 8 **Reporta** L_s con el depósito L_i que ahora contiene al minorista r en el vehículo v .
 - 9 **Termina y reporta** L_s .
-

4.2 Codificación de la solución

La codificación de soluciones adoptada aquí se basa en grupos debido a que Bi-MDVRP puede verse como un problema de agrupación [70, 69]. Así, la representación de soluciones adoptada se describe a continuación.

Dado que un depósito está asociado a varios minoristas, cada depósito se representa como un grupo que “contiene” minoristas. Una vez asignados los minoristas a un grupo, podemos aplicar una heurística adecuada para generar la ruta

correspondiente para cada vehículo disponible. La Figura 4.1 ilustra dos depósitos (o grupos) que contienen dos vehículos cada uno.

Es necesario asignar cada minorista a un depósito adecuado de forma que el costo relacionado con la distribución sea mínimo. Para resolver este problema, proponemos la inserción de un minorista en un depósito con un costo mínimo. La idea principal es elegir un depósito aleatorio y, a continuación, insertar el minorista en una ruta que satisfaga las restricciones de los vehículos. Es decir, a partir del conjunto de depósitos L_s se actualizará un depósito aleatorio L_i insertando el minorista r en la mejor posición de ruta de un vehículo aleatorio elegido de L_i . El Algoritmo 2 describe esta inserción de costo mínimo.

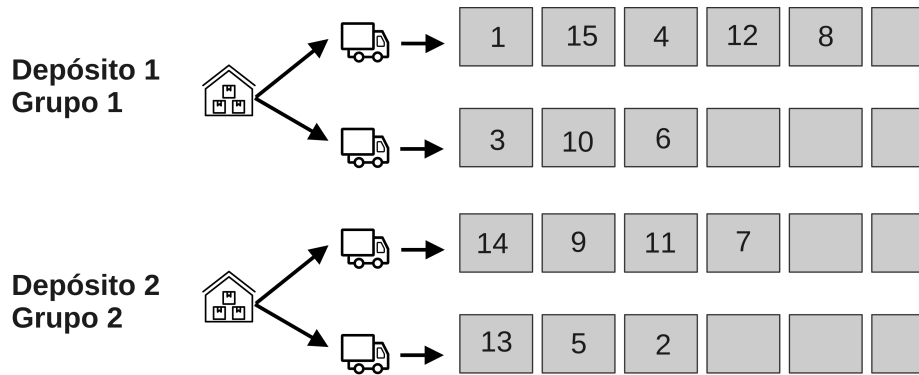


Figura 4.1: Representación de la solución. Se observan dos depósitos (grupos) con dos vehículos cada uno. Note que cada vehículo tiene asociado una sub-ruta que inicia y termina en el depósito correspondiente.

La Figura 4.1, nos muestra la representación basada en grupos de una solución, la cual se representa como un conjunto de centros de distribución con minoristas asignados y rutas de menor costo y se observa de la siguiente manera:

$$[[1, 15, 4, 12, 8, -], [3, 10, 6], [14, 9, 11, 7, -], [13, 5, 2]]$$

4.3 Inicialización de la Población

Se espera que la población inicial sea lo más diversa posible con soluciones factibles, a continuación, adoptamos una estrategia de muestreo aleatorio para generar soluciones factibles aleatorias.

Este procedimiento de inicialización se puede utilizar para generar N soluciones factibles al azar. Así, la población de soluciones se denota como $\{(\mathbf{L}_s^{(i)}, \mathbf{y}_i) : i = 1, 2, \dots, N\}$, donde $\mathbf{L}_s^{(i)}$ es un conjunto que contiene sub-rutas para el i -ésimo depósito, mientras que \mathbf{y}_i la solución óptima correspondiente al nivel inferior.

4.3.1 Primera variante: Inicialización aleatoria sin sesgo

Inspirándonos en la revisión bibliográfica sobre problemas de agrupación [70], sugerimos lo siguiente para generar una solución aleatoria.

1. Para $i = 1, 2, \dots, N$, hacer:
2. Barajar la lista de minoristas y asignarlos a un depósito aleatorio utilizando la inserción de costo mínimo descrita en el Algoritmo 2.
3. Establecer $\mathbf{L}_s^{(i)}$ como el grupo (i -ésimo depósito) que contiene minoristas enrutados en una sub-ruta factible.
4. Calcular la demanda de cada depósito en $\mathbf{L}_s^{(i)}$.
5. Utilizar las demandas calculadas para obtener la solución óptima de nivel inferior \mathbf{y}_i utilizando el optimizador de nivel inferior descrito en la Sección 4.6.

En esta propuesta, los minoristas son seleccionados de manera aleatoria y asignados al depósito más cercano, teniendo en cuenta la capacidad y el tiempo máximo del vehículo para no violar las restricciones del problema.

4.3.2 Segunda variante: Inicialización aleatoria con sesgo

En el segundo procedimiento de inicialización, se considera igualmente un orden aleatorio de los minoristas a ser asignados a un depósito [18]. Sin embargo, esta vez no se insertan al depósito más cercano, sino “probablemente” al depósito más cercano, es decir, se perturba de manera aleatoria la distancia entre el minorista a ser insertado y un depósito, esto ayuda a insertarlo a depósitos no necesariamente más cercanos a los minoristas en general, pero en promedio ese sería el caso esperado.

1. Para $i = 1, 2, \dots, N$, hacer:
2. Barajar la lista de minoristas y al depósito con la “distancia aleatoria” mínima definida, es decir, se asigna un minorista r al siguiente depósito l donde

$$l \in \arg \min_l \{d(l, r) + r \cdot d(l, r)\}, \text{ con } r \in U([-0.5, 0.5]).$$

Se puede observar que al minorista se le puede asignar el depósito más cercano; sin embargo, probablemente se pueda insertar en otro depósito en la vecindad, introduciendo así diversidad a la población.

3. Establecer $L_s^{(i)}$ como el grupo (i -ésimo depósito) que contiene minoristas enrutados en una sub-ruta factible.
4. Calcular la demanda de cada depósito en $L_s^{(i)}$.
5. Utilizar las demandas calculadas para obtener la solución óptima de nivel inferior y_i utilizando el optimizador de nivel inferior descrito en la Sección 4.6.

4.4 Cruza y Mutación en el Nivel Superior

El operador de cruza es un componente crucial del algoritmo genético, ya que debe ayudar a explorar y explotar el espacio de búsqueda en el nivel superior.

Aquí, seleccionamos N número de padres (solo considerando la solución de nivel superior denotada como L_s) a través de un torneo binario, que se cruzarán para generar N soluciones con distribución similar a la de los padres correspondientes. En este trabajo, proponemos el operador de cruce descrito en los siguientes pasos y en la Figura 4.2.

1. Para cada minorista en el segundo padre, si $\gamma < p_{cr}$ (donde γ es un número aleatorio uniforme en $[0, 1]$), entonces ese minorista se elimina del primer padre.
2. De forma similar, para cada minorista del primer padre, si $\gamma < p_{cr}$, entonces ese minorista se elimina del segundo padre.
3. Una vez realizado el proceso de eliminación de minoristas, se insertan los minoristas que faltan mediante la inserción de costo mínimo para garantizar la viabilidad de las soluciones. El primer descendiente se obtiene insertando en el segundo padre los minoristas eliminados del primero. Del mismo modo, la segunda descendencia se genera mediante la inserción en el primer padre de los minoristas eliminados del segundo padre.

El operador de variación anterior produce dos soluciones similares con respecto al padre seleccionado, pero la información de ambos padres está presente en las soluciones candidatas. Una vez generados los dos descendientes, procedemos a aplicar el operador de mutación de la siguiente manera:

1. Dada una solución candidata, seleccionar los minoristas con probabilidad p_m y eliminarlos de la solución.
2. Utilizar el Algoritmo 2 (inserción de costo mínimo) para insertar los minoristas eliminados.

Se puede observar que N nuevas soluciones se crean después de los operadores de cruce y mutación. Sugerimos insertar las nuevas soluciones en la población

actual y reducir el tamaño de la población mediante la eliminación de las peores soluciones (respecto a los valores de la función objetivo de nivel superior) de tal manera que el tamaño original de la población se mantiene, es decir, se ordenan

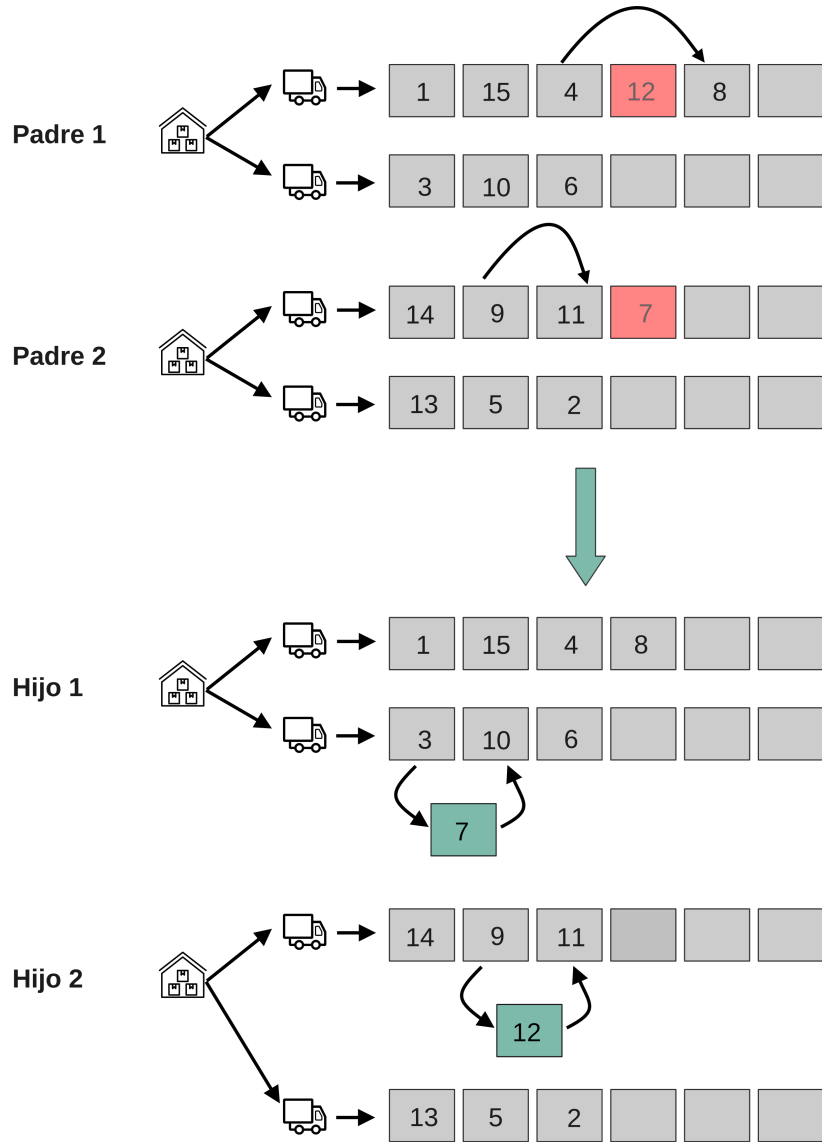


Figura 4.2: Operador de Cruza. Observe que se eliminan minoristas en el primer y segundo padre, y se generan dos descendientes, incorporando los minoristas faltantes del segundo padre al primer hijo usando la inserción de costo mínimo. Note que se procede de manera análoga en el segundo hijo.

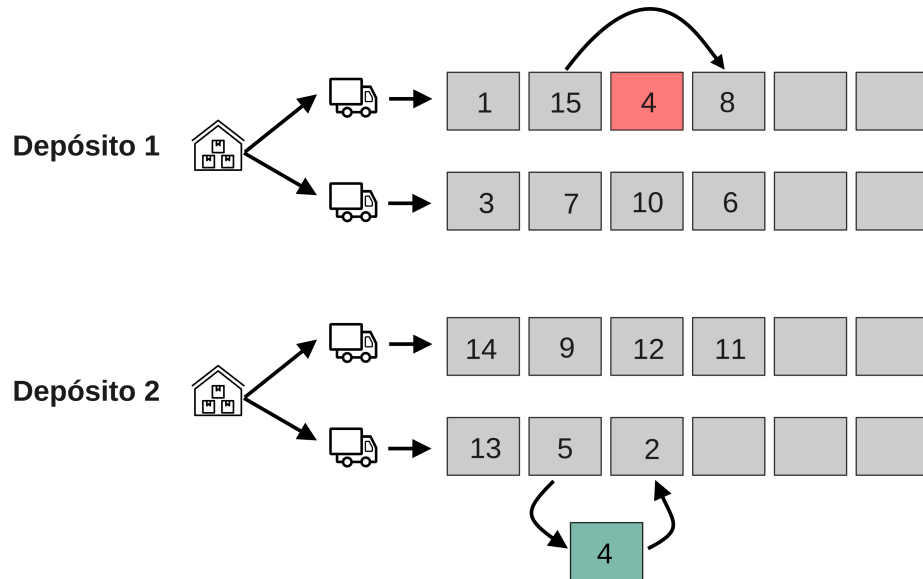


Figura 4.3: En el operador de mutación se eliminan y se insertan minoristas. Aquí se observa que se eliminó el minorista 4 y luego se insertó en el segundo depósito, en su segunda sub-ruta, usando la inserción de costo mínimo.

las soluciones de menor a mayor valor de la función objetivo del nivel superior, y se eliminan las peores en este ordenamiento.

4.5 Enrutamiento óptimo

Una vez que se asigna un conjunto de minoristas a un depósito, se usa un algoritmo genético local sin parámetros para este propósito [71], es decir, se optimiza la sub-ruta usando una representación basada en permutaciones para minimizar el costo de distribución. El algoritmo genético local se ejecuta durante un número limitado de generaciones (50 generaciones) y se aplica solo a una solución aleatoria antes de realizar el entorno de selección del algoritmo genético binivel. Esto se debe a que al agregar un mecanismo local en el nivel superior, el costo computacional aumenta en el tiempo de ejecución.

A continuación, se detalla este procedimiento:

1. Seleccione una solución de la población (de manera aleatoria).
2. Por cada depósito L en la solución seleccionada hacer:
 - a) Cree una población de permutaciones aleatorias de minoristas asignados al depósito L . Tome como referencia la Figura 4.4 para generar dichas permutaciones.
 - b) Use dicha población para ejecutar el algoritmo genético local durante un número específico de generaciones. Para generar nuevas soluciones, utilice la cruce “Order Crossover” y mutación basada en el intercambio de dos minoristas seleccionados de manera aleatoria [71]. Use como función de aptitud el costo de la ruta.
3. Reemplazar la solución seleccionada por la que se modificó usando el algoritmo genético local.

Sugerimos aplicar este mecanismo solo a una solución para evitar aumentar el costo computacional.

Cabe mencionar que el enrutamiento óptimo es agregado solo para la variante que ByGA-C.

4.6 Optimización de Nivel inferior

Este optimizador se aplica en el nivel inferior para hallar la cantidad de producción que debe entregarse a los depósitos. Dado que el problema de optimización de nivel inferior es en realidad un problema de programación lineal, se adopta el algoritmo símplex para resolver dicho problema lineal utilizando la biblioteca de código abierto GLPK¹ (GNU linear programming kit). Es importante señalar que este optimizador de nivel inferior puede cambiarse por otro si cambian las condiciones del problema en este nivel.

¹Disponible en www.gnu.org/software/glpk

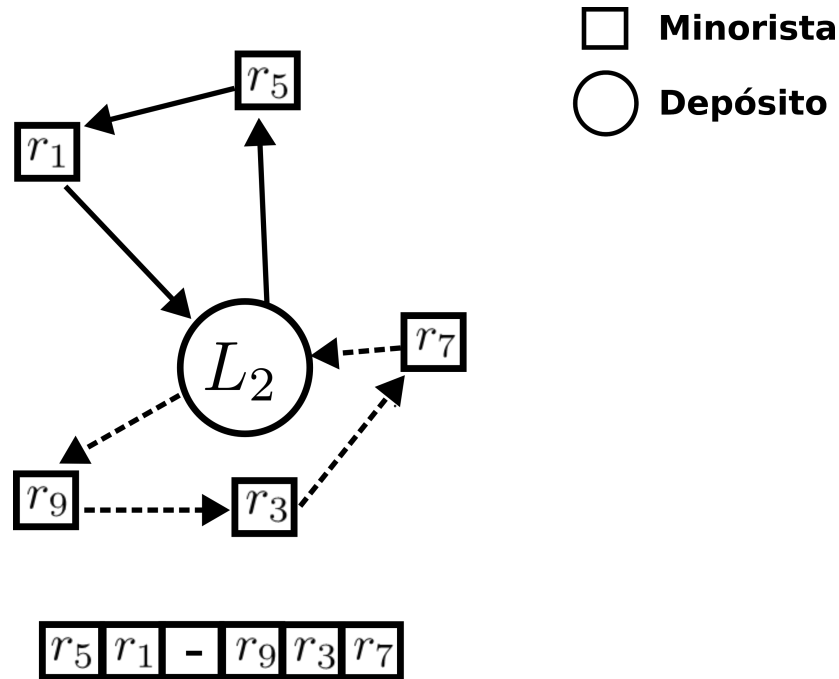


Figura 4.4: Enrutamiento óptimo. Observe que cada sub-ruta (sub-tour) se representa como una permutación de los minoristas asociados al depósito dado.

4.7 Complejidad computacional

La complejidad computacional de un algoritmo genético depende de sus componentes tales como el tamaño de la población, la dimensión del problema, el número de generaciones y los operadores genéticos del algoritmo (selección, cruce y mutación) [30].

En nuestro caso, analizaremos la complejidad computacional del algoritmo ByGA-C debido a que es la variante que tiene más elementos en su diseño. Así, que lo analizaremos para una generación o iteración de ByGA-C.

- Costo de la inicialización: Como se generarán N soluciones y se considerarán $n_r = |R|$ minoristas para ser insertados en los depósitos utilizando la inserción de costo mínimo (Algoritmo 2). Este algoritmo inserta cada minorista a cada depósito, por lo que se necesitan $O(n_r n_L)$ con n_L el número de

depósitos. Así, para generar N soluciones iniciales se requieren $O(Nn_r n_L)$ en el peor de los casos.

- La cruce propuesta requiere a lo más n_r minoristas, luego se utiliza el algoritmo de costo mínimo dada por el Algoritmo 2 que requiere $O(n_r n_L)$ en el peor de los casos. Así, para generar N descendientes, se necesita $O(Nn_r n_L)$. Note que la mutación requiere el mismo número de operaciones en el peor de los casos.
- Optimización del nivel inferior requiere tantas operaciones como el optimizador del nivel inferior. Debido a que adoptamos el algoritmo simplex, sabemos que el número de operaciones requeridas es $O(2^{n_L \cdot n_k})$ [52] debido a que el nivel de abajo está en función de las demandas de cada depósito y el número de plantas denotado por n_k .
- Otro componente incorporado en ByGA-C es el enrutador óptimo que, para cada depósito, minimiza el costo de cada sub-ruta. Así, el costo de aplicar este mecanismo a una solución es de $O(N_{\text{loc}} G_{\text{loc}}^{n_r})$, donde N_{loc} es el tamaño de la población, G_{loc} el número de generaciones de este algoritmo genético local.

Considerando lo anterior, la complejidad computacional para una iteración de ByGA-C que se obtiene es $O(N(n_r n_L + 2^{n_L \cdot n_k}))$.

Note que la complejidad descrita es una aproximación, ya que puede variar debido a los múltiples elementos que involucran a nuestra propuesta. Sin embargo, en la sección de resultados y discusión (Capítulo 5) analizamos el tiempo computacional.

Capítulo 5

Experimentación y Resultados

Para la validación del rendimiento de ByGA-S y ByGA-C, llevamos a cabo un estudio experimental entre las versiones de nuestro algoritmo propuesto y los algoritmos CEMBA y CEMBA-H descritos en la Sección 5.2. El entorno computacional utilizado para la experimentación de los algoritmos constó de un equipo Intel Xeon E5-2603 v3 de dos procesadores con una frecuencia de 1,6 GHz. Además, de una memoria interna de 64 GB de tipo DDR4-SDRAM con una velocidad de memoria de 2133 MHz. Este entorno permitió realizar las pruebas y ejecuciones de los algoritmos en un ambiente de alto rendimiento.

5.1 Instancias del problema

Para la experimentación computacional, los algoritmos se probaron utilizando los siguientes tres conjuntos de instancias, (1) el conjunto *pr* [15], (2) conjunto *bipr* [55], y (3) el conjunto *p* (generadas) en donde el nivel superior involucra datos del Problema de Ruteo de Vehículos con Múltiples Depósitos (MDVRP),

obtenidos del banco de instancias de [6]. En las Tablas 5.1- 5.2, $|K|$ denota el número de plantas para el seguidor, $|s|$ es el número de vehículos, $|L|$ es el número de depósitos, $|R|$ representa el número de minoristas, T_s es el tiempo límite de la duración de las rutas y $|Q|$ denota la carga máxima de cada vehículo. La diferencia entre los tres conjuntos de instancias (*pr*, *bipr* y *p*) radica en que en el conjunto de instancias 3 (*p*), representado en la Tabla 5.2, no se tiene la restricción del tiempo límite para la duración de las rutas ($|T_s|$).

Ahora, para el problema del nivel inferior se tiene un conjunto de datos que son la ubicación de las plantas y su capacidad de producción. La ubicación de las plantas se encuentra aleatoriamente en el intervalo de $[-200, 200] \times [-200, 200]$ mientras que la capacidad de las plantas se obtiene con el cociente de $\left[\frac{D}{|K|}, D\right]$, donde D es la Demanda total y $|K|$ número de plantas según una distribución uniforme. Por otro lado, se supone que los costos c_{kl}^{12} no dependen de la planta y se generan mediante una distribución uniforme entre $[0.5, 1]$ para el conjunto de instancias 1 (Tabla 5.1) y $[0,36, 5.36]$ para el conjunto de instancias 2 y 3 (Tabla 5.1 y Tabla 5.2 respectivamente), y c_{kl}^{22} se obtiene con el producto de dos términos: el primero es aleatorio entre $[2, 5]$ para el conjunto de instancias 1 y $[1, 4]$ para el conjunto de instancias 2 y 3. Es importante tener en cuenta que las instancias se crearon agregando tantas plantas como depósitos que se encuentran ubicados aleatoriamente en el mapa.

Tabla 5.1: Descripción de instancias de Bi-MDVRP para los conjuntos de instancias 1 (*pr*) - 2 (*bipr*) [15, 55]

Instancia	$ K $	$ s $	$ L $	$ R $	T_s	Q
1	4	1	4	48	500	200
2	4	3	4	96	480	195
3	4	3	4	144	460	190
4	4	4	4	192	440	185
5	4	5	4	240	420	180
6	4	6	4	288	400	175
7	6	1	6	72	500	200
8	6	2	6	144	475	190
9	6	3	6	216	450	180
10	6	4	6	288	425	170

Tabla 5.2: Instancias Bi-MDVRP 2022: conjunto de instancias 3 (p).

Instancia	$ K $	$ s $	$ L $	$ R $	Q
p01	4	4	4	50	80
p02	4	2	4	50	160
p03	5	3	5	75	140
p04	2	8	2	100	100
p05	2	5	2	100	200
p06	3	6	3	100	100
p07	4	4	4	100	100
p08	2	14	2	249	500
p09	3	12	3	249	500
p10	4	8	4	249	500
p11	5	6	5	249	500
p12	2	5	2	80	60
p13	2	5	2	80	60
p14	2	5	2	80	60
p15	4	5	4	160	60
p16	4	5	4	160	60
p17	4	5	4	160	60
p18	6	5	6	240	60
p19	6	5	6	240	60
p20	6	5	6	240	60
p21	9	5	9	360	60
p22	9	5	9	360	60
p23	9	5	9	360	60

5.2 Algoritmos bajo comparación

Para la validación del rendimiento de ByGA-S y ByGA-C, llevamos a cabo un estudio experimental entre las versiones de nuestro algoritmo propuesto y otros dos algoritmos que se propusieron para resolver los conjuntos de instancias utilizadas. Los algoritmos utilizados se describen como:

5.2.1 Algoritmo coevolutivo con esquema de migración: CEMBA

El algoritmo CEMBA propuesto por Rihab Said [74] es un algoritmo coevolutivo que utiliza dos poblaciones en cada nivel y un esquema de migración; con el objetivo de minimizar considerablemente el número de Evaluaciones de la Función objetivo (FE, por sus siglas en inglés) mientras se asegura una buena convergencia hacia el óptimo global del nivel superior. Este algoritmo se destaca en su inicialización en ambos niveles utilizando el Método de descomposición de espacio discreto (DSDM por sus siglas en inglés) propuesto por Chaabani et al [22].

5.2.2 Algoritmo coevolutivo con esquema de migración híbrido: CEMBA-H

Se realizó una modificación en el algoritmo CEMBA incorporando un optimizador exacto en el nivel inferior. El algoritmo CEMBA es conocido por ser coevolutivo, utilizando dos poblaciones y un esquema de migración en ambos niveles (nivel superior e inferior). Sin embargo, en esta propuesta de CEMBA-H, se decidió mantener únicamente el esquema coevolutivo con esquema de migración en el nivel superior. Mientras que en el nivel inferior se resuelve de manera exacta utilizando el optimizador mencionado en la Sección 4.6.

5.3 Configuración de parámetros y metodología de prueba estadística

Para proporcionar una comparación justa, los algoritmos han sido codificados en el lenguaje de programación Julia en su versión 1.7.2 y se ajustaron los respectivos parámetros utilizando la herramienta de ajuste de parámetros IRACE [62]. Para cada algoritmo se utilizó un tiempo de parada de 168 horas con un número máximo de 100 generaciones y se seleccionaron 19 instancias para el ajuste de parámetros (lo mismo para configurar los cuatro enfoques). A continuación se muestra la mejor configuración obtenida para cada algoritmo.

1. ByGA-S

- Población P: 100
- Cruza: 1
- Mutación: 0.0199
- Generaciones: 100

2. ByGA-C

- Población P: 40
- Cruza: 0.6
- Mutación: 0.0343
- Parámetros para el enrutamiento óptimos:
 - Población: 100
 - Criterio de paro: 50 generaciones
 - Cruza: Order crossover (no requiere parámetros)
 - Mutación: Mutación simple (swap de dos minoristas, no requiere parámetros)
- Generaciones: 100

3. CEMBA

- Población Superior: $UP_1 = 100, UP_2 = 100,$
- Población Inferior: $LP_1 = 60, LP_2 = 60,$
- Generación superior e inferior: $UG = 100, LG = 100.$
- Cruza RBX: 0.3
- Mutación uniforme: 0.2388
- Esquema de migración α : 10

4. CEMBA-H

- Población Superior: $UP_1 = 100, UP_2 = 100,$
- Generación superior: $UG = 100.$
- Cruza RBX: 0.9
- Mutación uniforme: 0.0121
- Esquema de migración α : 6

Indicador de Desempeño

Como indicador de desempeño se utiliza el valor del GAP [15, 18], calculado con la solución reportada por nuestras propuestas (ByGA-S y ByGA-C) en las instancias inspiradas en el mundo real descritas anteriormente. Sea F^{kn} el valor conocido y F^{fv} el valor encontrado, se define como en la Ecuación 5.1

$$\text{GAP} = \frac{F^{kn} - F^{fv}}{F^{kn}} \cdot 100 \quad (5.1)$$

El GAP indica el porcentaje de error con respecto a algún valor conocido del valor de la función objetivo de nivel superior. En este caso, los valores conocidos son los obtenidos por los algoritmos en comparación a la propuesta. Obsérvese que se obtiene un GAP negativo si nuestra propuesta informa de un mejor valor de la función objetivo de nivel superior con respecto al algoritmo comparado.

5.4 Resultados

Se tiene que una prueba estadística permite la toma de decisiones cuantitativas con respecto a un proceso al determinar si hay suficiente evidencia para rechazar una hipótesis sobre el proceso. La prueba de rangos Wilcoxon es un análisis que compara y evalúa dos muestras en busca de diferencias significativas [43]. En este estudio, utilizamos la prueba de Wilcoxon por pares para determinar las estadísticas entre los resultados. Para esto, es necesario determinar un tamaño de muestra aceptable. En la literatura especializada se ha ido estudiando el mínimo requerido y se sugiere un tamaño de muestra de 30 para pruebas estadísticas [31]. Los resultados mostrados en las Tablas y Figuras siguientes son las obtenidas del promedio de las 31 ejecuciones independientes.

Esta sección muestra los resultados obtenidos por la propuesta ByGA-S, ByGA-C y la comparación contra los algoritmos CEMBA y CEMBA-H.

5.4.1 Resultados obtenidos por ByGA-S

Las Tablas 5.3-5.5 muestran los resultados estadísticos obtenidos de 31 ejecuciones independientes de los algoritmos dados en cada columna, en los cuales se utilizó la prueba de suma de rangos de Wilcoxon (nivel de confianza de 95 %) para determinar las diferencias significativas y se encontró que ByGA-S obtuvo mejores resultados que CEMBA y CEMBA-H en los 3 conjuntos de instancias. Además, el GAP se da como un indicador para comparar el desempeño de ByGA-S respecto a CEMBA y CEMBA-H. En las Tablas 5.6-5.8, el GAP negativo nos indica que ByGA-S obtuvo un mejor valor en la función objetivo en el nivel superior respecto a los algoritmos en comparación.

Tabla 5.3: Promedio de la función objetivo ByGA-S y valor GAP con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 1).

Instancias	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-S vs CEMBA	GAP: ByGA-S vs CEMBA-H
pr1	1,424.12	3,087.69	2,403.13	-116.81	-68.75
pr2	2,427.07	6,556.70	4,940.90	-170.15	-103.57
pr3	3,970.31	12,806.36	10,019.35	-222.55	-152.36
pr4	5,636.37	16,123.74	13,880.13	-186.07	-146.26
pr5	6,585.38	17,554.02	17,038.17	-166.56	-158.73
pr6	7,590.11	23,475.98	22,281.11	-209.30	-193.55
pr7	1,851.95	8,563.35	3,967.63	-362.40	-114.24
pr8	3,782.94	14,921.06	10,168.96	-294.43	-168.81
pr9	5,061.91	21,902.65	16,363.79	-332.70	-223.27
pr10	8,843.87	27,952.52	24,239.99	-216.07	-174.09

Tabla 5.4: Promedio de la función objetivo ByGA-S y valor GAP con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 2).

Instancias	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-S vs CEMBA	GAP: ByGA-S vs CEMBA-H
bipr1	1,401.63	3,116.06	2,350.70	-122.32	-67.71
bipr2	2,353.76	6,732.26	4,853.61	-186.02	-106.21
bipr3	3,848.00	12,848.86	9,854.99	-233.91	-156.11
bipr4	4,889.27	15,595.73	13,006.95	-218.98	-166.03
bipr5	6,454.91	17,731.17	16,680.43	-174.69	-158.41
bipr6	8,294.76	22,974.69	23,160.09	-176.98	-179.21
bipr7	1,841.41	9,151.67	3,956.84	-396.99	-114.88
bipr8	3,911.46	16,447.90	10,234.04	-320.51	-161.64
bipr9	5,298.08	24,781.15	16,722.80	-367.74	-215.64
bipr10	8,439.10	30,140.57	23,974.68	-257.15	-184.09

Tabla 5.5: Promedio de la función objetivo ByGA-S y valor *GAP* con respecto a CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 3).

Instancias	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-S vs CEMBA	GAP: ByGA-S vs CEMBA-H
p1	1,396.48	2,576.92	1,891.25	-84.53	-35.43
p2	1,061.86	2,151.15	1,513.18	-102.58	-42.50
p3	1,987.88	4,617.63	2,993.77	-132.29	-50.60
p4	2,834.84	5,098.71	4,641.25	-79.86	-63.72
p5	2,397.03	4,461.35	3,615.55	-86.12	-50.83
p6	2,332.29	4,932.95	4,192.44	-111.51	-79.76
p7	2,460.70	5,120.19	4,679.09	-108.08	-90.15
p8	18,560.89	36,151.51	35,979.67	-94.77	-93.85
p9	15,982.04	37,122.03	33,575.74	-132.27	-110.08
p10	16,974.39	38,080.10	36,266.92	-124.34	-113.66
p11	16,607.50	38,700.37	37,359.72	-133.03	-124.96
p12	2,391.90	6,671.39	4,695.17	-178.92	-96.29
p13	2,506.79	6,722.93	4,814.23	-168.19	-92.05
p14	2,410.16	6,624.82	4,714.06	-174.87	-95.59
p15	5,212.70	20,744.48	16,068.51	-297.96	-208.26
p16	5,419.16	20,657.75	16,274.97	-281.20	-200.32
p17	5,304.33	20,597.17	16,160.14	-288.31	-204.66
p18	8,159.37	40,696.68	34,801.37	-398.77	-326.52
p19	8,341.87	40,901.91	34,979.58	-390.32	-319.33
p20	8,340.48	41,159.12	34,978.20	-393.49	-319.38
p21	12,452.85	100,731.74	67,722.79	-708.91	-443.83
p22	13,012.29	99,488.38	68,282.24	-664.57	-424.75
p23	12,491.81	97,672.70	67,761.76	-681.89	-442.45

5.4.2 Resultados Obtenidos por ByGA-C

Las Tablas 5.6-5.8 muestran los resultados estadísticos obtenidos de 31 ejecuciones independientes de los algoritmos dados en cada columna, en los cuales se utilizó la prueba de suma de rangos de Wilcoxon para determinar las diferencias significativas y se encontró que ByGA-C obtuvo mejores resultados que ByGA-S, CEMBA y CEMBA-H en los 3 conjuntos de instancias. Además, el GAP se da como un indicador para comparar el desempeño de ByGA-C respecto a ByGA-S, CEMBA y CEMBA-H. En las Tablas 5.6-5.8, el GAP negativo nos indica que ByGA-C obtuvo un mejor valor en la función objetivo en el nivel superior respecto a los algoritmos en comparación.

Tabla 5.6: Promedio de la función objetivo ByGA-C y valor *GAP* con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 1).

Instancias	ByGA-C	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-C vs ByGA-S	GAP: ByGA-S vs CEMBA	GAP: ByGA-S vs CEMBA-H
pr1	1,466.72	1,424.12	3,087.69	2,403.13	2.90	-110.52	-63.84
pr2	2,351.35	2,427.07	6,556.70	4,940.90	-3.22	-178.85	-110.13
pr3	3,354.79	3,970.31	12,806.36	10,019.35	-18.35	-281.73	-198.66
pr4	5,048.45	5,636.37	16,123.74	13,880.13	-11.65	-219.38	-174.94
pr5	5,811.11	6,585.38	17,554.02	17,038.17	-13.32	-202.08	-193.20
pr6	6,383.43	7,590.11	23,475.98	22,281.11	-18.90	-267.76	-249.05
pr7	1,911.20	1,851.95	8,563.35	3,967.63	3.10	-348.06	-107.60
pr8	3,640.41	3,782.94	14,921.06	10,168.96	-3.92	-309.87	-179.34
pr9	4,818.80	5,061.91	21,902.65	16,363.79	-5.05	-354.53	-239.58
pr10	8,369.80	8,843.87	27,952.52	24,239.99	-5.66	-233.97	-189.61

Tabla 5.7: Promedio de la función objetivo ByGA-C y valor *GAP* con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 2).

Instancias	ByGA-C	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-C – ByGA-S	GAP: ByGA-S – CEMBA	GAP: ByGA-S – CEMBA-H
bipr1	1,449.64	1,401.63	3,116.06	2,350.70	3.31	-114.95	-62.16
bipr2	2,245.31	2,353.76	6,732.26	4,853.61	-4.83	-199.84	-116.17
bipr3	3,252.34	3,848.00	12,848.86	9,854.99	-18.31	-295.06	-203.01
bipr4	4,277.37	4,889.27	15,595.73	13,006.95	-14.31	-264.61	-204.09
bipr5	5,615.46	6,454.91	17,731.17	16,680.43	-14.95	-215.76	-197.05
bipr6	7,165.17	8,294.76	22,974.69	23,160.09	-15.77	-220.64	-223.23
bipr7	1,911.46	1,841.41	9,151.67	3,956.84	3.66	-378.78	-107.01
bipr8	3,786.60	3,911.46	16,447.90	10,234.04	-3.30	-334.37	-170.27
bipr9	5,038.45	5,298.08	24,781.15	16,722.80	-5.15	-391.84	-231.90
bipr10	7,960.56	8,439.10	30,140.57	23,974.68	-6.01	-278.62	-201.17

Tabla 5.8: Promedio de la función objetivo ByGA-C y valor *GAP* con respecto a ByGA-S, CEMBA y CEMBA-H de 31 ejecuciones independientes (conjunto de instancias 3).

Instancias	ByGA-C	ByGA-S	CEMBA	CEMBA-H	GAP: ByGA-C vs ByGA-S	GAP: ByGA-S vs CEMBA	GAP: ByGA-S vs CEMBA-H
p1	1,333.62	1,396.48	2,576.92	1,891.25	-4.71	-93.23	-41.81
p2	1,016.56	1,061.86	2,151.15	1,513.18	-4.46	-111.61	-48.85
p3	1,894.24	1,987.88	4,617.63	2,993.77	-4.94	-143.77	-58.05
p4	2,543.89	2,834.84	5,098.71	4,641.25	-11.44	-100.43	-82.45
p5	2,041.89	2,397.03	4,461.35	3,615.55	-17.39	-118.49	-77.07
p6	2,066.05	2,332.29	4,932.95	4,192.44	-12.89	-138.76	-102.92
p7	2,326.12	2,460.70	5,120.19	4,679.09	-5.79	-120.12	-101.15
p8	15,423.91	18,560.89	36,151.51	35,979.67	-20.34	-134.39	-133.27
p9	12,456.29	15,982.04	37,122.03	33,575.74	-28.30	-198.02	-169.55
p10	14,059.58	16,974.39	38,080.10	36,266.92	-20.73	-170.85	-157.95
p11	14,544.58	16,607.50	38,700.37	37,359.72	-14.18	-166.08	-156.86
p12	1,711.26	2,391.90	6,671.39	4,695.17	-39.77	-289.85	-174.37
p13	1,830.33	2,506.79	6,722.93	4,814.23	-36.96	-267.31	-163.03
p14	1,730.15	2,410.16	6,624.82	4,714.06	-39.30	-282.90	-172.46
p15	3,521.49	5,212.70	20,744.48	16,068.51	-48.03	-489.08	-356.30
p16	3,727.95	5,419.16	20,657.75	16,274.97	-45.37	-454.13	-336.57
p17	3,613.12	5,304.33	20,597.17	16,160.14	-46.81	-470.07	-347.26
p18	5,409.52	8,159.37	40,696.68	34,801.37	-50.83	-652.32	-543.34
p19	5,587.73	8,341.87	40,901.91	34,979.58	-49.29	-632.00	-526.01
p20	5,586.35	8,340.48	41,159.12	34,978.20	-49.30	-636.78	-526.14
p21	7,976.76	12,452.85	100,731.74	67,722.79	-56.11	-1,162.81	-749.00
p22	8,536.20	13,012.29	99,488.38	68,282.24	-52.44	-1,065.49	-699.91
p23	8,015.73	12,491.81	97,672.70	67,761.76	-55.84	-1,118.51	-745.36

5.4.3 Número de evaluaciones y tiempo de ejecución

En este apartado mostramos los valores del promedio de 31 corridas independientes respecto al número de evaluaciones de la función objetivo del nivel superior e inferior, y el tiempo de ejecución de cada propuesta analizada en este trabajo en las Tablas 5.9-5.11. Los resultados en **negritas** nos indica el algoritmo que obtuvo mejores resultados en dicha instancia.

Tabla 5.9: Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1).

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
pr1	1.75E+04	3.51E+04	72.726737	9.80E+03	1.91E+04	49.186653	3.20E+05	7.68E+08	2,085.660565	1.60E+06	3.20E+06	385.105840
pr2	4.99E+04	9.98E+04	288.394857	1.98E+04	3.85E+04	975.612322	3.20E+05	7.68E+08	2,345.605512	1.60E+06	3.20E+06	399.978968
pr3	5.01E+04	1.00E+05	344.908034	2.05E+04	4.01E+04	1,832.967452	3.20E+05	7.68E+08	2,682.502779	1.60E+06	3.20E+06	412.462628
pr4	5.01E+04	1.00E+05	394.265138	2.05E+04	4.01E+04	3,101.952792	3.20E+05	7.68E+08	2,952.499167	1.60E+06	3.20E+06	422.075867
pr5	5.01E+04	1.00E+05	441.135210	2.05E+04	4.01E+04	4,152.038785	3.20E+05	7.68E+08	3,111.263141	1.60E+06	3.20E+06	449.568176
pr6	5.01E+04	1.00E+05	490.437395	2.05E+04	4.01E+04	5,454.436522	3.20E+05	7.68E+08	3,340.186985	1.60E+06	3.20E+06	460.459482
pr7	3.18E+04	6.36E+04	194.970417	1.62E+04	3.15E+04	109.364362	3.20E+05	7.68E+08	2,689.717595	1.60E+06	3.20E+06	464.077207
pr8	5.01E+04	1.00E+05	387.714855	2.05E+04	4.01E+04	1,336.270476	3.20E+05	7.68E+08	3,072.830692	1.60E+06	3.20E+06	486.771284
pr9	5.01E+04	1.00E+05	459.956516	2.05E+04	4.01E+04	2,708.712664	3.20E+05	7.68E+08	3,387.539281	1.60E+06	3.20E+06	501.943280
pr10	5.01E+04	1.00E+05	516.329736	2.05E+04	4.00E+04	4,089.847188	3.20E+05	7.68E+08	3,696.422287	1.60E+06	3.20E+06	514.810424

Tabla 5.10: Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2).

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
bipr1	1.78E+04	3.55E+04	53.880001	1.01E+04	1.97E+04	36.456936	3.20E+05	7.68E+08	2,353.960655	1.60E+06	3.20E+06	451.341742
bipr2	4.97E+04	9.95E+04	237.790826	1.98E+04	3.86E+04	1,235.975444	3.20E+05	7.68E+08	2,632.789538	1.60E+06	3.20E+06	347.749937
bipr3	5.01E+04	1.00E+05	293.532766	2.05E+04	4.01E+04	2,098.049879	3.20E+05	7.68E+08	2,916.088579	1.60E+06	3.20E+06	371.766211
bipr4	5.01E+04	1.00E+05	342.487776	2.05E+04	4.01E+04	3,398.419011	3.20E+05	7.68E+08	3,134.739228	1.60E+06	3.20E+06	381.895823
bipr5	5.01E+04	1.00E+05	395.496187	2.05E+04	4.01E+04	4,421.978438	3.20E+05	7.68E+08	3,400.588175	1.60E+06	3.20E+06	407.248788
bipr6	5.01E+04	1.00E+05	426.503885	2.05E+04	4.01E+04	5,750.485889	3.20E+05	7.68E+08	3,705.045277	1.60E+06	3.20E+06	403.442698
bipr7	3.57E+04	7.13E+04	171.602574	1.50E+04	2.92E+04	79.181129	3.20E+05	7.68E+08	2,976.971277	1.60E+06	3.20E+06	417.383520
bipr8	5.01E+04	1.00E+05	334.180320	2.05E+04	4.01E+04	1,668.399847	3.20E+05	7.68E+08	3,284.374987	1.60E+06	3.20E+06	422.457849
bipr9	5.01E+04	1.00E+05	404.394975	2.05E+04	4.01E+04	2,988.912285	3.20E+05	7.68E+08	3,602.193734	1.60E+06	3.20E+06	443.319946
bipr10	5.01E+04	1.00E+05	458.914348	2.05E+04	4.01E+04	4,400.667007	3.20E+05	7.68E+08	3,903.079160	1.60E+06	3.20E+06	459.761404

Tabla 5.11: Número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3).

Instancia	ByGA-C			ByGA-S			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
p1	4.18E+04	8.36E+04	147.487586	1.38E+04	2.70E+04	663.348315	3.20E+05	7.68E+08	7,074.792603	1.60E+06	3.20E+06	304.742824
p2	3.84E+04	7.69E+04	130.602113	1.40E+04	2.73E+04	449.521823	3.20E+05	7.68E+08	4,322.335686	1.60E+06	3.20E+06	473.513452
p3	4.90E+04	9.79E+04	196.625755	2.02E+04	3.94E+04	1,130.850505	3.20E+05	7.68E+08	4,650.370201	1.60E+06	3.20E+06	487.557001
p4	4.96E+04	9.93E+04	153.722101	1.98E+04	3.87E+04	3,763.262468	3.20E+05	7.01E+08	2,615.398663	1.60E+06	3.20E+06	342.830150
p5	5.01E+04	1.00E+05	136.880137	2.05E+04	4.01E+04	3,111.758906	3.20E+05	7.34E+08	2,879.541179	1.60E+06	3.20E+06	343.911649
p6	5.01E+04	1.00E+05	137.463618	2.05E+04	4.01E+04	2,442.437300	3.20E+05	7.68E+08	4,410.625490	1.60E+06	3.20E+06	397.940158
p7	4.98E+04	9.96E+04	213.697704	1.99E+04	3.87E+04	2,223.317256	3.20E+05	7.68E+08	5,856.498836	1.60E+06	3.20E+06	322.833977
p8	5.01E+04	1.00E+05	298.146842	2.05E+04	4.01E+04	13,353.093454	3.20E+05	7.52E+08	6,097.942068	1.60E+06	3.20E+06	490.804706
p9	5.01E+04	1.00E+05	294.528882	2.05E+04	4.01E+04	13,913.148212	3.20E+05	7.68E+08	8,924.213736	1.60E+06	3.20E+06	352.978206
p10	5.01E+04	1.00E+05	286.143934	2.05E+04	4.01E+04	9,359.872966	3.20E+05	7.68E+08	6,686.682374	1.60E+06	3.20E+06	574.874437
p11	5.01E+04	1.00E+05	339.566193	2.05E+04	4.01E+04	9,334.064459	3.20E+05	7.68E+08	8,858.448373	1.60E+06	3.20E+06	433.081417
p12	5.01E+04	1.00E+05	166.026189	1.99E+04	3.89E+04	1,526.959114	3.20E+05	7.17E+08	3,185.192969	1.60E+06	3.20E+06	407.408653
p13	5.01E+04	1.00E+05	163.712280	1.99E+04	3.89E+04	1,788.879687	3.20E+05	6.88E+08	3,898.410600	1.60E+06	3.20E+06	258.943832
p14	5.01E+04	1.00E+05	166.035618	1.99E+04	3.89E+04	1,644.868184	3.20E+05	6.98E+08	3,566.104186	1.60E+06	3.20E+06	379.615072
p15	5.01E+04	1.00E+05	259.410535	2.05E+04	4.01E+04	3,753.716048	3.20E+05	7.68E+08	6,620.626937	1.60E+06	3.20E+06	336.739950
p16	5.01E+04	1.00E+05	248.634833	2.05E+04	4.01E+04	3,570.646515	3.20E+05	7.68E+08	5,750.833458	1.60E+06	3.20E+06	522.294919
p17	5.01E+04	1.00E+05	243.315235	2.05E+04	4.01E+04	3,745.682423	3.20E+05	7.68E+08	6,810.544320	1.60E+06	3.20E+06	336.960630
p18	5.01E+04	1.00E+05	314.445513	2.05E+04	4.01E+04	5,265.510065	3.20E+05	7.68E+08	8,120.196037	1.60E+06	3.20E+06	653.921853
p19	5.01E+04	1.00E+05	347.633648	2.05E+04	4.01E+04	5,502.564954	3.20E+05	7.68E+08	9,673.618252	1.60E+06	3.20E+06	475.422408
p20	5.01E+04	1.00E+05	298.467208	2.05E+04	4.01E+04	5,271.477029	3.20E+05	7.68E+08	7,047.893672	1.60E+06	3.20E+06	657.678029
p21	5.01E+04	1.00E+05	488.290929	2.05E+04	4.01E+04	8,400.814669	3.20E+05	7.68E+08	14,602.310914	1.60E+06	3.20E+06	843.920170
p22	5.01E+04	1.00E+05	372.947026	2.05E+04	4.01E+04	6,949.624929	3.20E+05	7.68E+08	11,950.977247	1.60E+06	3.20E+06	929.005119
p23	5.01E+04	1.00E+05	507.536106	2.05E+04	4.01E+04	8,414.589014	3.20E+05	7.68E+08	15,362.391153	1.60E+06	3.20E+06	844.803838

5.4.4 Comparación por número de generaciones

Ahora, nos centraremos en el comportamiento de los algoritmos ByGA-S, ByGA-C, CEMBA y CEMBA-H con distintos números de generaciones.

Este estudio se realizó para la evaluación del rendimiento de los algoritmos en función al número de generaciones, ya que permite una comparación justa y estandarizada entre las diferentes instancias del problema. Esto nos permite analizar el comportamiento de los algoritmos respecto al número evaluaciones de la función objetivo y el tiempo requerido para encontrar el óptimo o una solución factible. Al centrarnos en el número de generaciones, nos enfocamos en la eficiencia y eficacia de los algoritmos para encontrar resultados satisfactorios en un número limitado de iteraciones y con un costo computacional menor.

Para estas experimentaciones, se realizó un ajuste de parámetros automático, y fue realizado por IRACE [62], Para cada algoritmo se utilizó un tiempo de paro de 168 horas y se seleccionaron 19 instancias para el ajuste de parámetros (lo

mismo para configurar los cuatro enfoques) con un número máximo de 40, 100, 500 y 1000 generaciones en 31 corridas independientes.

Experimentación con 40 generaciones

Tabla 5.12: Parámetros obtenidos por IRACE con un número máximo de 40 generaciones.

Algoritmo	Parámetros
	Población P: 100
ByGA-S	Cruza: 1.0
	Mutación: 0.0311
	Generaciones: 40
	Población P: 80
	Cruza: 0.6
	Mutación: 0.0089
ByGA-C	Enrutamiento óptimo:
	- Población: 100
	- Criterio de paro: 50
	- Cruza: Order crossover (no requiere parametros)
	- Mutación: Mutacion simple (swap de dos minoristas, no requiere parametros)
	Generaciones: 40
	Población Superior size: $UP_1 = 100$, $UP_2 = 100$
	Población Inferior size: $LP_1 = 60$, $LP_2 = 60$
	Generación superior e inferior: UG = 40, LG = 40
CEMBA	Cruza RBX: 0.2
	Mutación uniforme: 0.2351
	Esquema de migración α : 4
	Población Superior size: $UP_1 = 100$, $UP_2 = 100$
	Generación superior: UG = 40
CEMBA-H	Cruza RBX: 0.9
	Mutación uniforme: 0.0274
	Esquema de migración α : 10

Conjunto de instancias 1

40 Generaciones

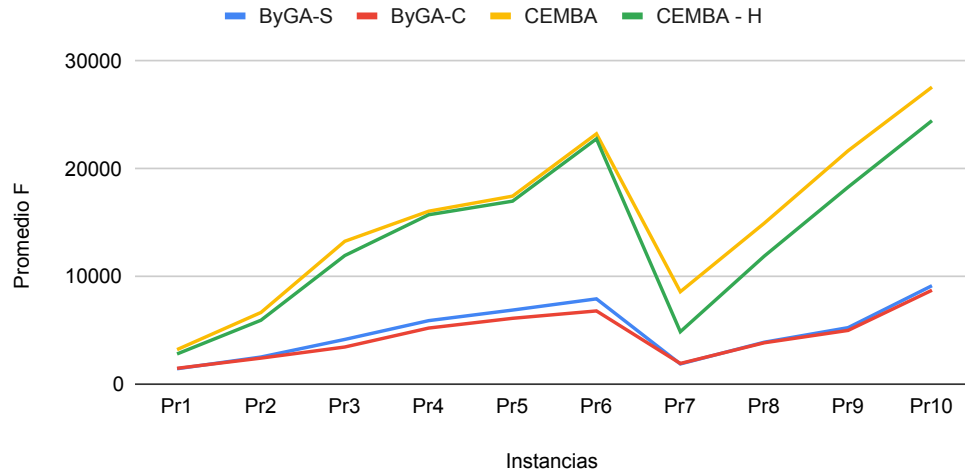


Figura 5.1: Conjunto de instancias 1: Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas.

Tabla 5.13: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
pr1	4.10E+03	8.20E+03	20.27226863	3.32E+03	6.56E+03	21.04275887	1.28E+05	3.07E+08	1560.57736	1.28E+05	2.56E+05	11.58698756
pr2	4.10E+03	8.20E+03	25.60281423	3.32E+03	6.56E+03	230.8036867	1.28E+05	3.07E+08	1884.535252	1.28E+05	2.56E+05	25.5270135
pr3	4.10E+03	8.20E+03	29.28500794	3.32E+03	6.56E+03	557.9956063	1.28E+05	3.07E+08	2219.72528	1.28E+05	2.56E+05	27.44052127
pr4	4.10E+03	8.20E+03	34.53940919	3.32E+03	6.56E+03	815.125557	1.28E+05	3.07E+08	2551.587757	1.28E+05	2.56E+05	28.05595643
pr5	4.10E+03	8.20E+03	40.64238601	3.32E+03	6.56E+03	928.6183642	1.28E+05	3.07E+08	2769.243667	1.28E+05	2.56E+05	31.44320532
pr6	4.10E+03	8.20E+03	43.39756571	3.32E+03	6.56E+03	1065.955407	1.28E+05	3.07E+08	2998.510947	1.28E+05	2.56E+05	33.29242833
pr7	4.10E+03	8.20E+03	26.94463388	3.32E+03	6.56E+03	26.12883685	1.28E+05	3.07E+08	2201.019044	1.28E+05	2.56E+05	33.31327708
pr8	4.10E+03	8.20E+03	33.8773682	3.32E+03	6.56E+03	316.1395419	1.28E+05	3.07E+08	2618.872412	1.28E+05	2.56E+05	36.17396299
pr9	4.10E+03	8.20E+03	39.49339652	3.32E+03	6.56E+03	798.775704	1.28E+05	3.07E+08	3035.314402	1.28E+05	2.56E+05	38.36857935
pr10	4.10E+03	8.20E+03	44.74672671	3.32E+03	6.55E+03	1057.760713	1.28E+05	3.07E+08	3418.259768	1.28E+05	2.56E+05	41.12298205

Conjunto de instancias 2

40 Generaciones

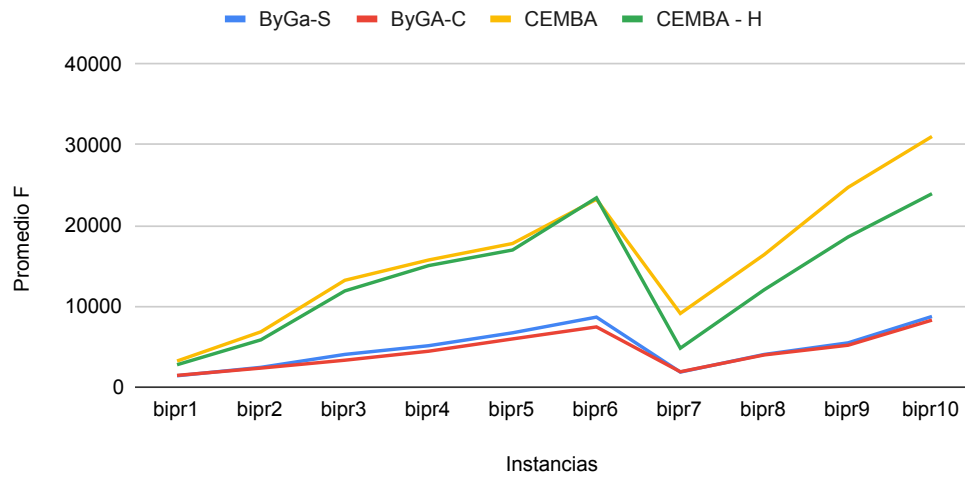


Figura 5.2: Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas (conjunto de instancias 2).

Tabla 5.14: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
bipr1	4.10E+03	8.20E+03	21.42115839	3.32E+03	6.56E+03	14.19349204	1.28E+05	3.07E+08	1587.795795	1.28E+05	2.56E+05	32.49163141
bipr2	4.10E+03	8.20E+03	27.16974052	3.32E+03	6.56E+03	189.447731	1.28E+05	3.07E+08	1915.115741	1.28E+05	2.56E+05	34.99023747
bipr3	4.10E+03	8.20E+03	31.27299586	3.32E+03	6.56E+03	433.8154183	1.28E+05	3.07E+08	2236.747072	1.28E+05	2.56E+05	35.92127234
bipr4	4.10E+03	8.20E+03	35.65150565	3.32E+03	6.56E+03	669.3400954	1.28E+05	3.07E+08	2548.04959	1.28E+05	2.56E+05	37.72371681
bipr5	4.10E+03	8.20E+03	40.9280571	3.32E+03	6.56E+03	772.0003774	1.28E+05	3.07E+08	2775.392814	1.28E+05	2.56E+05	41.87629466
bipr6	4.10E+03	8.20E+03	47.2218498	3.32E+03	6.56E+03	896.8734331	1.28E+05	3.07E+08	2996.646715	1.28E+05	2.56E+05	40.03518357
bipr7	4.10E+03	8.20E+03	28.22396	3.32E+03	6.56E+03	20.1453263	1.28E+05	3.07E+08	2169.634322	1.28E+05	2.56E+05	40.20397115
bipr8	4.10E+03	8.20E+03	34.05030042	3.32E+03	6.56E+03	252.157585	1.28E+05	3.07E+08	2665.072514	1.28E+05	2.56E+05	43.62082256
bipr9	4.10E+03	8.20E+03	41.57225018	3.32E+03	6.56E+03	619.5941157	1.28E+05	3.07E+08	2941.821711	1.28E+05	2.56E+05	48.96125726
bipr10	4.10E+03	8.20E+03	46.40650327	3.32E+03	6.55E+03	908.193741	1.28E+05	3.07E+08	3333.427157	1.28E+05	2.56E+05	48.36579907

Conjunto de instancias 3

40 Generaciones

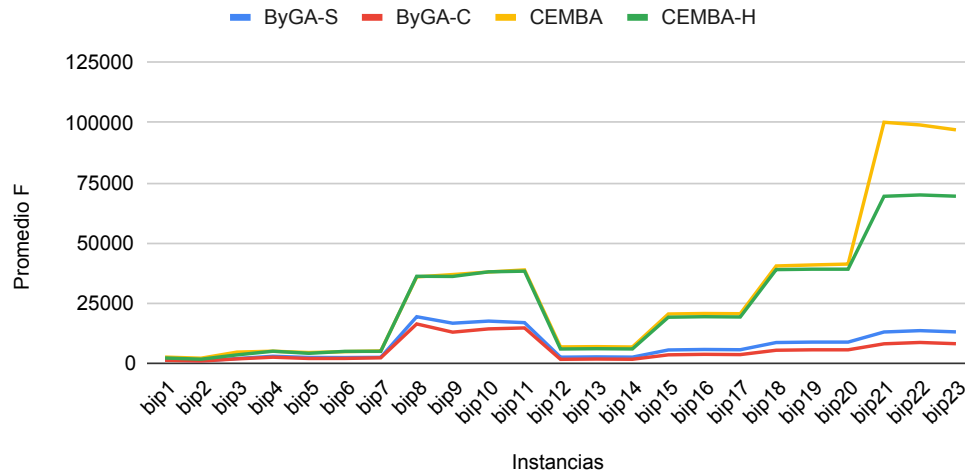


Figura 5.3: Promedio de la función objetivo del nivel superior en 40 generaciones con 31 corridas (conjunto de instancias 3).

Tabla 5.15: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
p1	4.10E+03	8.20E+03	22.39079892	3.32E+03	6.56E+03	80.83070542	1.28E+05	3.07E+08	2729.988952	1.28E+05	2.56E+05	33.22773963
p2	4.10E+03	8.20E+03	17.27686049	3.32E+03	6.56E+03	55.14847987	1.28E+05	3.07E+08	1117.809033	1.28E+05	2.56E+05	37.21473534
p3	4.10E+03	8.20E+03	17.39740019	3.32E+03	6.56E+03	111.3095866	1.28E+05	3.07E+08	1192.880119	1.28E+05	2.56E+05	39.11133712
p4	4.01E+03	8.02E+03	11.73043559	2.89E+03	5.69E+03	390.9914315	1.28E+05	2.86E+08	711.4877436	1.28E+05	2.56E+05	27.57947273
p5	4.10E+03	8.20E+03	11.98162176	3.32E+03	6.56E+03	287.6358101	1.28E+05	2.96E+08	612.8598906	1.28E+05	2.56E+05	26.82540591
p6	4.10E+03	8.20E+03	12.0116216	3.32E+03	6.56E+03	217.357819	1.28E+05	3.07E+08	824.678215	1.28E+05	2.56E+05	30.68175981
p7	4.01E+03	8.03E+03	25.95569498	2.92E+03	5.76E+03	277.3778194	1.28E+05	3.07E+08	2541.450548	1.28E+05	2.56E+05	38.00591874
p8	4.10E+03	8.20E+03	29.97414118	3.32E+03	6.56E+03	678.3113322	1.28E+05	3.02E+08	1648.551228	1.28E+05	2.56E+05	42.01548553
p9	4.10E+03	8.20E+03	35.97720875	3.32E+03	6.56E+03	862.2725516	1.28E+05	3.07E+08	3294.574416	1.28E+05	2.56E+05	38.17196345
p10	4.10E+03	8.20E+03	26.52364509	3.32E+03	6.56E+03	504.4030476	1.28E+05	3.07E+08	1754.410055	1.28E+05	2.56E+05	46.06099963
p11	4.10E+03	8.20E+03	38.98034963	3.32E+03	6.56E+03	886.2950671	1.28E+05	3.07E+08	3407.311276	1.28E+05	2.56E+05	48.39168976
p12	4.10E+03	8.20E+03	14.95513283	3.32E+03	6.56E+03	165.2802532	1.28E+05	2.90E+08	811.8078574	1.28E+05	2.56E+05	34.15917895
p13	4.10E+03	8.20E+03	22.22631658	3.32E+03	6.56E+03	279.1071408	1.28E+05	2.80E+08	1571.090945	1.28E+05	2.56E+05	30.64270005
p14	4.10E+03	8.20E+03	17.66482299	3.32E+03	6.56E+03	244.5900959	1.28E+05	2.84E+08	1327.550137	1.28E+05	2.56E+05	34.12221193
p15	4.10E+03	8.20E+03	29.62082232	3.32E+03	6.56E+03	516.9863777	1.28E+05	3.07E+08	2740.613379	1.28E+05	2.56E+05	36.33309753
p16	4.10E+03	8.20E+03	21.9960441	3.32E+03	6.56E+03	406.4301092	1.28E+05	3.07E+08	1551.302189	1.28E+05	2.56E+05	41.41209976
p17	4.10E+03	8.20E+03	29.77305697	3.32E+03	6.56E+03	499.1316343	1.28E+05	3.07E+08	2763.505542	1.28E+05	2.56E+05	40.53409648
p18	4.10E+03	8.20E+03	28.46973835	3.32E+03	6.56E+03	583.6394105	1.28E+05	3.07E+08	2110.441075	1.28E+05	2.56E+05	54.35676799
p19	4.10E+03	8.20E+03	39.09582866	3.32E+03	6.56E+03	759.5122479	1.28E+05	3.07E+08	3523.38867	1.28E+05	2.56E+05	50.48606795
p20	4.10E+03	8.20E+03	26.60203064	3.32E+03	6.56E+03	465.4419149	1.28E+05	3.07E+08	1864.025538	1.28E+05	2.56E+05	52.89828244
p21	4.10E+03	8.20E+03	54.66344728	3.32E+03	6.56E+03	1032.865388	1.28E+05	3.07E+08	4794.6907	1.28E+05	2.56E+05	83.09598331
p22	4.10E+03	8.20E+03	33.68767669	3.32E+03	6.56E+03	558.0745595	1.28E+05	3.07E+08	2218.6834	1.28E+05	2.56E+05	72.26589848
p23	4.10E+03	8.20E+03	54.6086271	3.32E+03	6.56E+03	1019.16839	1.28E+05	3.07E+08	4126.199621	1.28E+05	2.56E+05	78.12212823

Experimentación con 100 generaciones

Tabla 5.16: Parámetros obtenidos por IRACE con un número máximo de 100 generaciones.

Algoritmo	Parámetros
ByGA-S	Población P: 100
	Cruza: 1
	Mutación: 0.0199
	Generaciones: 100
ByGA-C	Población P: 40
	Cruza: 0.6
	Mutación: 0.0343
ByGA-C	Enrutamiento óptimo:
	- Población: 100
	- Criterio de paro: 50
	- Cruza: Order crossover (no requiere parametros)
CEMBA	- Mutación: Mutacion simple (swap de dos minoristas, no requiere parametros)
	Población Superior size: $UP_1 = 100, UP_2 = 100$
	Población Inferior size: $LP_1 = 60, LP_2 = 60$
	Generación superior e inferior: $UG = 100, LG = 100$
CEMBA	Cruza RBX: 0.3
	Mutación uniforme: 0.2388
	Esquema de migración α : 10
CEMBA	Población Superior size: $UP_1 = 100, UP_2 = 100$
	Generación superior: $UG = 100$
CEMBA-H	Cruza RBX: 0.9
	Mutación uniforme: 0.0121
	Esquema de migración α : 6

Conjunto de instancias 1

100 Generaciones

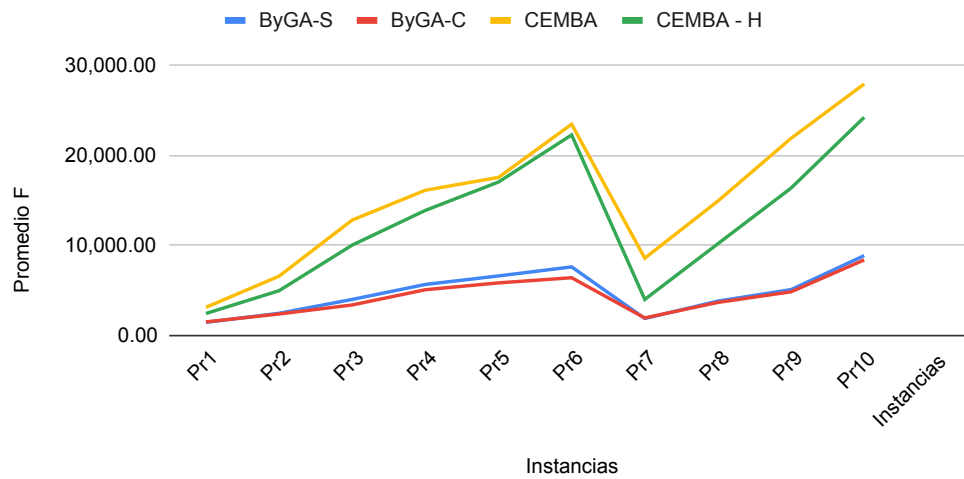


Figura 5.4: Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas (conjunto de instancias 1).

Tabla 5.17: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
pr1	1.75E+04	3.51E+04	72.726737	9.80E+03	1.91E+04	49.186653	3.20E+05	7.68E+08	2,085.660565	1.60E+06	3.20E+06	385.105840
pr2	4.99E+04	9.98E+04	288.394857	1.98E+04	3.85E+04	975.612322	3.20E+05	7.68E+08	2,345.605512	1.60E+06	3.20E+06	399.978968
pr3	5.01E+04	1.00E+05	344.908034	2.05E+04	4.01E+04	1,832.967452	3.20E+05	7.68E+08	2,682.502779	1.60E+06	3.20E+06	412.462628
pr4	5.01E+04	1.00E+05	394.265138	2.05E+04	4.01E+04	3,101.952792	3.20E+05	7.68E+08	2,952.499167	1.60E+06	3.20E+06	422.075867
pr5	5.01E+04	1.00E+05	441.135210	2.05E+04	4.01E+04	4,152.038785	3.20E+05	7.68E+08	3,111.263141	1.60E+06	3.20E+06	449.568176
pr6	5.01E+04	1.00E+05	490.437395	2.05E+04	4.01E+04	5,454.436522	3.20E+05	7.68E+08	3,340.186985	1.60E+06	3.20E+06	460.459482
pr7	3.18E+04	6.36E+04	194.970417	1.62E+04	3.15E+04	109.364362	3.20E+05	7.68E+08	2,689.717595	1.60E+06	3.20E+06	464.077207
pr8	5.01E+04	1.00E+05	387.714855	2.05E+04	4.01E+04	1,336.270476	3.20E+05	7.68E+08	3,072.830692	1.60E+06	3.20E+06	486.771284
pr9	5.01E+04	1.00E+05	459.956516	2.05E+04	4.01E+04	2,708.712664	3.20E+05	7.68E+08	3,387.539281	1.60E+06	3.20E+06	501.943280
pr10	5.01E+04	1.00E+05	516.329736	2.05E+04	4.00E+04	4,089.847188	3.20E+05	7.68E+08	3,696.422287	1.60E+06	3.20E+06	514.810424

Conjunto de instancias 2

100 Generaciones

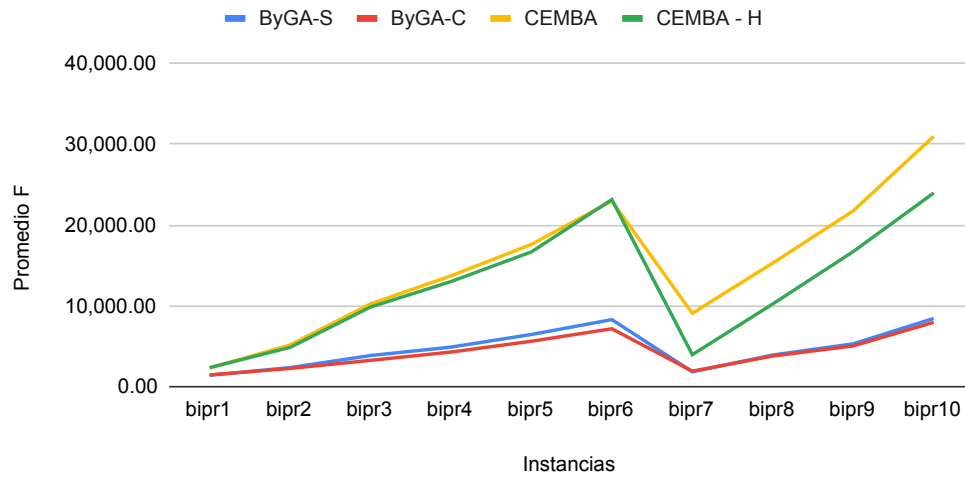


Figura 5.5: Conjunto de instancias 2: Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas.

Tabla 5.18: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
bipr1	1.78E+04	3.55E+04	53.880001	1.01E+04	1.97E+04	36.456936	3.20E+05	7.68E+08	2,353.960655	1.60E+06	3.20E+06	451.341742
bipr2	4.97E+04	9.95E+04	237.790826	1.98E+04	3.86E+04	1,235.975444	3.20E+05	7.68E+08	2,632.789538	1.60E+06	3.20E+06	347.749937
bipr3	5.01E+04	1.00E+05	293.532766	2.05E+04	4.01E+04	2,098.049879	3.20E+05	7.68E+08	2,916.088579	1.60E+06	3.20E+06	371.766211
bipr4	5.01E+04	1.00E+05	342.487776	2.05E+04	4.01E+04	3,398.419011	3.20E+05	7.68E+08	3,134.739228	1.60E+06	3.20E+06	381.895823
bipr5	5.01E+04	1.00E+05	395.496187	2.05E+04	4.01E+04	4,421.978438	3.20E+05	7.68E+08	3,400.588175	1.60E+06	3.20E+06	407.248788
bipr6	5.01E+04	1.00E+05	426.503885	2.05E+04	4.01E+04	5,750.485889	3.20E+05	7.68E+08	3,705.045277	1.60E+06	3.20E+06	403.442698
bipr7	3.57E+04	7.13E+04	171.602574	1.50E+04	2.92E+04	79.181129	3.20E+05	7.68E+08	2,976.971277	1.60E+06	3.20E+06	417.383520
bipr8	5.01E+04	1.00E+05	334.180320	2.05E+04	4.01E+04	1,668.399847	3.20E+05	7.68E+08	3,284.374987	1.60E+06	3.20E+06	422.457849
bipr9	5.01E+04	1.00E+05	404.394975	2.05E+04	4.01E+04	2,988.912285	3.20E+05	7.68E+08	3,602.193734	1.60E+06	3.20E+06	443.319946
bipr10	5.01E+04	1.00E+05	458.914348	2.05E+04	4.01E+04	4,400.667007	3.20E+05	7.68E+08	3,903.079160	1.60E+06	3.20E+06	459.761404

Conjunto de instancias 3

100 Generaciones

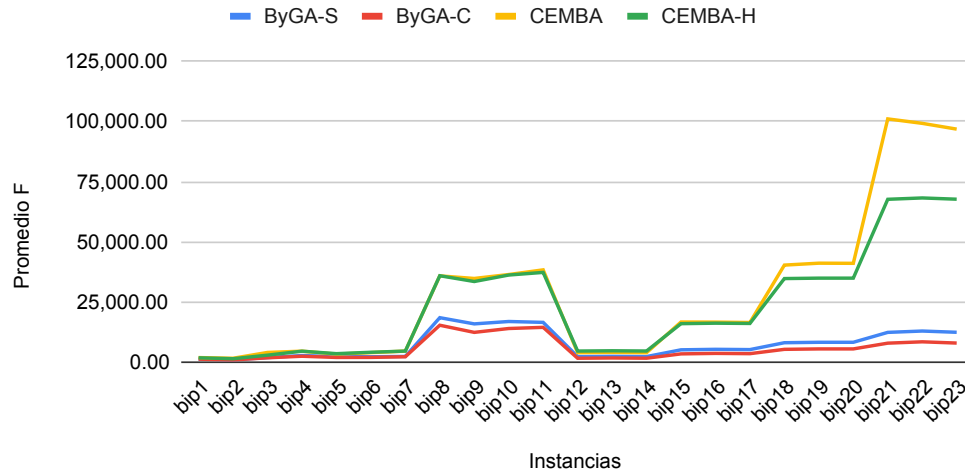


Figura 5.6: Promedio de la función objetivo del nivel superior en 100 generaciones con 31 corridas (conjunto de instancias 3).

Tabla 5.19: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
p1	4.18E+04	8.36E+04	147.487586	1.38E+04	2.70E+04	663.348315	3.20E+05	7.68E+08	7,074.792603	1.60E+06	3.20E+06	304.742824
p2	3.84E+04	7.69E+04	130.602113	1.40E+04	2.73E+04	449.521823	3.20E+05	7.68E+08	4,322.335686	1.60E+06	3.20E+06	473.513452
p3	4.90E+04	9.79E+04	196.625755	2.02E+04	3.94E+04	1,130.850505	3.20E+05	7.68E+08	4,650.370201	1.60E+06	3.20E+06	487.557001
p4	4.96E+04	9.93E+04	153.722101	1.98E+04	3.87E+04	3,763.262468	3.20E+05	7.01E+08	2,615.398663	1.60E+06	3.20E+06	342.830150
p5	5.01E+04	1.00E+05	136.880137	2.05E+04	4.01E+04	3,111.758906	3.20E+05	7.34E+08	2,879.541179	1.60E+06	3.20E+06	343.911649
p6	5.01E+04	1.00E+05	137.463618	2.05E+04	4.01E+04	2,442.437300	3.20E+05	7.68E+08	4,410.625490	1.60E+06	3.20E+06	397.940158
p7	4.98E+04	9.96E+04	213.697704	1.99E+04	3.87E+04	2,223.317256	3.20E+05	7.68E+08	5,856.498836	1.60E+06	3.20E+06	322.833977
p8	5.01E+04	1.00E+05	298.146842	2.05E+04	4.01E+04	13,353.093454	3.20E+05	7.52E+08	6,097.942068	1.60E+06	3.20E+06	490.804706
p9	5.01E+04	1.00E+05	294.528882	2.05E+04	4.01E+04	13,913.148212	3.20E+05	7.68E+08	8,924.213736	1.60E+06	3.20E+06	352.978206
p10	5.01E+04	1.00E+05	286.143934	2.05E+04	4.01E+04	9,359.872966	3.20E+05	7.68E+08	6,686.682374	1.60E+06	3.20E+06	574.874437
p11	5.01E+04	1.00E+05	339.566193	2.05E+04	4.01E+04	9,334.064459	3.20E+05	7.68E+08	8,858.448373	1.60E+06	3.20E+06	433.081417
p12	5.01E+04	1.00E+05	166.026189	1.99E+04	3.89E+04	1,526.959114	3.20E+05	7.17E+08	3,185.192969	1.60E+06	3.20E+06	407.408653
p13	5.01E+04	1.00E+05	163.712280	1.99E+04	3.89E+04	1,788.879687	3.20E+05	6.88E+08	3,898.410600	1.60E+06	3.20E+06	258.943832
p14	5.01E+04	1.00E+05	166.035618	1.99E+04	3.89E+04	1,644.868184	3.20E+05	6.98E+08	3,566.104186	1.60E+06	3.20E+06	379.615072
p15	5.01E+04	1.00E+05	259.410535	2.05E+04	4.01E+04	3,753.716048	3.20E+05	7.68E+08	6,620.626937	1.60E+06	3.20E+06	336.739950
p16	5.01E+04	1.00E+05	248.634833	2.05E+04	4.01E+04	3,570.646515	3.20E+05	7.68E+08	5,750.833458	1.60E+06	3.20E+06	522.294919
p17	5.01E+04	1.00E+05	243.315235	2.05E+04	4.01E+04	3,745.682423	3.20E+05	7.68E+08	6,810.544320	1.60E+06	3.20E+06	336.960630
p18	5.01E+04	1.00E+05	314.445513	2.05E+04	4.01E+04	5,265.510065	3.20E+05	7.68E+08	8,120.196037	1.60E+06	3.20E+06	653.921853
p19	5.01E+04	1.00E+05	347.633648	2.05E+04	4.01E+04	5,502.564954	3.20E+05	7.68E+08	9,673.618252	1.60E+06	3.20E+06	475.422408
p20	5.01E+04	1.00E+05	298.467208	2.05E+04	4.01E+04	5,271.477029	3.20E+05	7.68E+08	7,047.893672	1.60E+06	3.20E+06	657.678029
p21	5.01E+04	1.00E+05	488.290929	2.05E+04	4.01E+04	8,400.814669	3.20E+05	7.68E+08	14,602.310914	1.60E+06	3.20E+06	843.920170
p22	5.01E+04	1.00E+05	372.947026	2.05E+04	4.01E+04	6,949.624929	3.20E+05	7.68E+08	11,950.977247	1.60E+06	3.20E+06	929.005119
p23	5.01E+04	1.00E+05	507.536106	2.05E+04	4.01E+04	8,414.589014	3.20E+05	7.68E+08	15,362.391153	1.60E+06	3.20E+06	844.803838

Experimentación con 500 generaciones

Tabla 5.20: Parámetros obtenidos por IRACE con un número máximo de 40 generaciones.

Algoritmo	Parámetros
ByGA-S	Población P: 100
	Cruza: 1.0
	Mutación: 0.0215
	Generaciones: 500
ByGA-C	Población P: 40
	Cruza: 1.0
	Mutación: 0.0581
	Enrutamiento óptimo: <ul style="list-style-type: none"> - Población: 100 - Criterio de paro: 50 - Cruza: Order crossover (no requiere parametros) - Mutación: Mutacion simple (swap de dos minoristas, no requiere parametros)
CEMBA	Población Superior size: $UP_1 = 20$, $UP_2 = 20$
	Población Inferior size: $LP_1 = 60$, $LP_2 = 60$
	Generación superior e inferior: UG = 500, LG = 500
	Cruza RBX: 0.3
CEMBA-H	Mutación uniforme: 0.0446
	Esquema de migración α : 10
	Población Superior size: $UP_1 = 100$, $UP_2 = 100$
	Generación superior: UG = 500
CEMBA-H	Cruza RBX: 0.1
	Mutación uniforme: 0.0060
	Esquema de migración α : 9

Conjunto de instancias 1

500 Generaciones

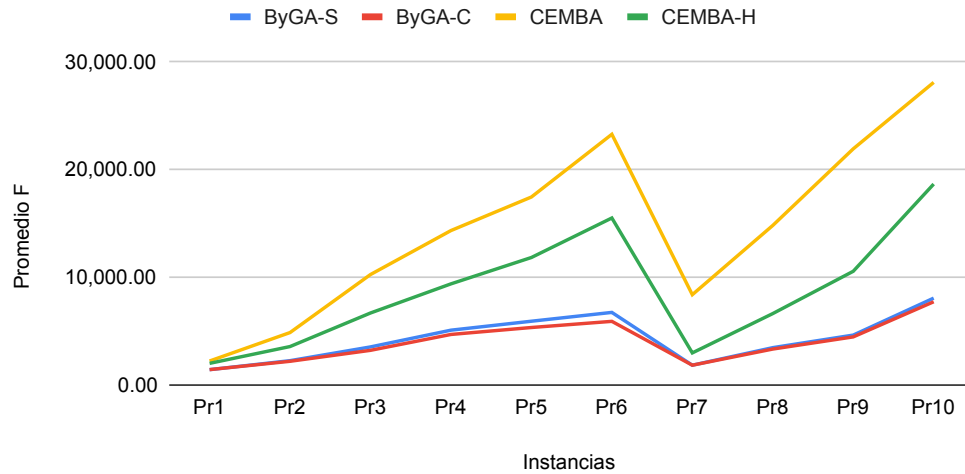


Figura 5.7: Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 1).

Tabla 5.21: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 1) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
pr1	1.75E+04	3.51E+04	72.726737	9.80E+03	1.91E+04	49.186653	3.20E+05	7.68E+08	2,085.660565	1.60E+06	3.20E+06	385.105840
pr2	4.99E+04	9.98E+04	288.394857	1.98E+04	3.85E+04	975.612322	3.20E+05	7.68E+08	2,345.605512	1.60E+06	3.20E+06	399.978968
pr3	5.01E+04	1.00E+05	344.908034	2.05E+04	4.01E+04	1,832.967452	3.20E+05	7.68E+08	2,682.502779	1.60E+06	3.20E+06	412.462628
pr4	5.01E+04	1.00E+05	394.265138	2.05E+04	4.01E+04	3,101.952792	3.20E+05	7.68E+08	2,952.499167	1.60E+06	3.20E+06	422.075867
pr5	5.01E+04	1.00E+05	441.135210	2.05E+04	4.01E+04	4,152.038785	3.20E+05	7.68E+08	3,111.263141	1.60E+06	3.20E+06	449.568176
pr6	5.01E+04	1.00E+05	490.437395	2.05E+04	4.01E+04	5,454.436522	3.20E+05	7.68E+08	3,340.186985	1.60E+06	3.20E+06	460.459482
pr7	3.18E+04	6.36E+04	194.970417	1.62E+04	3.15E+04	109.364362	3.20E+05	7.68E+08	2,689.717595	1.60E+06	3.20E+06	464.077207
pr8	5.01E+04	1.00E+05	387.714855	2.05E+04	4.01E+04	1,336.270476	3.20E+05	7.68E+08	3,072.830692	1.60E+06	3.20E+06	486.771284
pr9	5.01E+04	1.00E+05	459.956516	2.05E+04	4.01E+04	2,708.712664	3.20E+05	7.68E+08	3,387.539281	1.60E+06	3.20E+06	501.943280
pr10	5.01E+04	1.00E+05	516.329736	2.05E+04	4.00E+04	4,089.847188	3.20E+05	7.68E+08	3,696.422287	1.60E+06	3.20E+06	514.810424

Conjunto de instancias 2

500 Generaciones

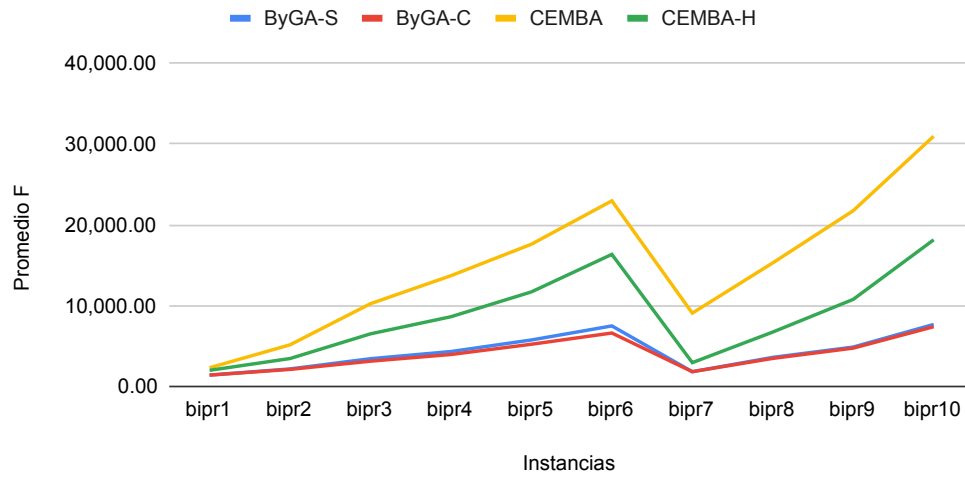


Figura 5.8: Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 2).

Tabla 5.22: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 2) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
bipr1	1.78E+04	3.55E+04	53.880001	1.01E+04	1.97E+04	36.456936	3.20E+05	7.68E+08	2,353.960655	1.60E+06	3.20E+06	451.341742
bipr2	4.97E+04	9.95E+04	237.790826	1.98E+04	3.86E+04	1,235.975444	3.20E+05	7.68E+08	2,632.789538	1.60E+06	3.20E+06	347.749937
bipr3	5.01E+04	1.00E+05	293.532766	2.05E+04	4.01E+04	2,098.049879	3.20E+05	7.68E+08	2,916.088579	1.60E+06	3.20E+06	371.766211
bipr4	5.01E+04	1.00E+05	342.487776	2.05E+04	4.01E+04	3,398.419011	3.20E+05	7.68E+08	3,134.739228	1.60E+06	3.20E+06	381.895823
bipr5	5.01E+04	1.00E+05	395.496187	2.05E+04	4.01E+04	4,421.978438	3.20E+05	7.68E+08	3,400.588175	1.60E+06	3.20E+06	407.248788
bipr6	5.01E+04	1.00E+05	426.503885	2.05E+04	4.01E+04	5,750.485889	3.20E+05	7.68E+08	3,705.045277	1.60E+06	3.20E+06	403.442698
bipr7	3.57E+04	7.13E+04	171.602574	1.50E+04	2.92E+04	79.181129	3.20E+05	7.68E+08	2,976.971277	1.60E+06	3.20E+06	417.383520
bipr8	5.01E+04	1.00E+05	334.180320	2.05E+04	4.01E+04	1,668.399847	3.20E+05	7.68E+08	3,284.374987	1.60E+06	3.20E+06	422.457849
bipr9	5.01E+04	1.00E+05	404.394975	2.05E+04	4.01E+04	2,988.912285	3.20E+05	7.68E+08	3,602.193734	1.60E+06	3.20E+06	443.319946
bipr10	5.01E+04	1.00E+05	458.914348	2.05E+04	4.01E+04	4,400.667007	3.20E+05	7.68E+08	3,903.079160	1.60E+06	3.20E+06	459.761404

Conjunto de instancias 3

500 Generaciones

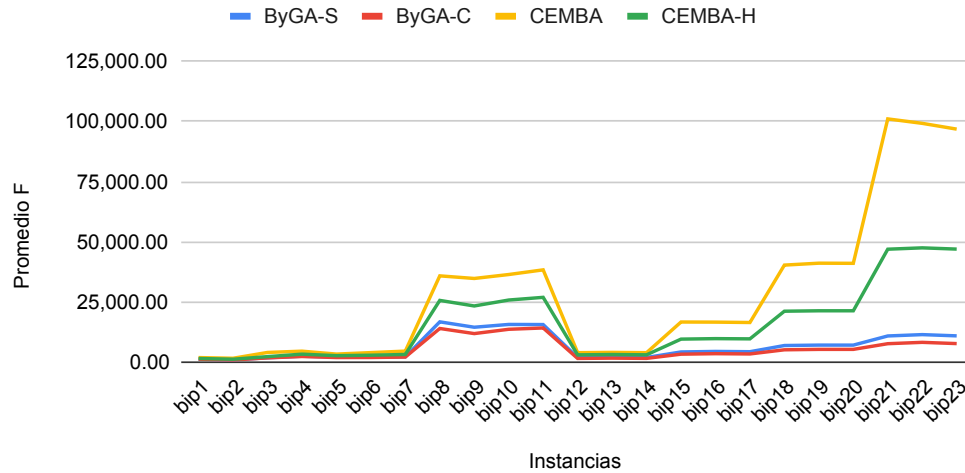


Figura 5.9: Promedio de la función objetivo del nivel superior en 500 generaciones con 31 corridas (conjunto de instancias 3).

Tabla 5.23: Promedio del número de FEs (nivel superior e inferior) y tiempo computacional (segundos) requerido por los algoritmos comparados en Bi-MDVRP (conjunto de instancias 3) en 31 corridas independientes.

Instancia	ByGA-S			ByGA-C			CEMBA			CEMBA-H		
	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T	FE_U	FE_L	T
p1	4.18E+04	8.36E+04	147.487586	1.38E+04	2.70E+04	663.348315	3.20E+05	7.68E+08	7,074.792603	1.60E+06	3.20E+06	304.742824
p2	3.84E+04	7.69E+04	130.602113	1.40E+04	2.73E+04	449.521823	3.20E+05	7.68E+08	4,322.335686	1.60E+06	3.20E+06	473.513452
p3	4.90E+04	9.79E+04	196.625755	2.02E+04	3.94E+04	1,130.850505	3.20E+05	7.68E+08	4,650.370201	1.60E+06	3.20E+06	487.557001
p4	4.96E+04	9.93E+04	153.722101	1.98E+04	3.87E+04	3,763.262468	3.20E+05	7.01E+08	2,615.398663	1.60E+06	3.20E+06	342.830150
p5	5.01E+04	1.00E+05	136.880137	2.05E+04	4.01E+04	3,111.758906	3.20E+05	7.34E+08	2,879.541179	1.60E+06	3.20E+06	343.911649
p6	5.01E+04	1.00E+05	137.463618	2.05E+04	4.01E+04	2,442.437300	3.20E+05	7.68E+08	4,410.625490	1.60E+06	3.20E+06	397.940158
p7	4.98E+04	9.96E+04	213.697704	1.99E+04	3.87E+04	2,223.317256	3.20E+05	7.68E+08	5,856.498836	1.60E+06	3.20E+06	322.833977
p8	5.01E+04	1.00E+05	298.146842	2.05E+04	4.01E+04	13,353.093454	3.20E+05	7.52E+08	6,097.942068	1.60E+06	3.20E+06	490.804706
p9	5.01E+04	1.00E+05	294.528882	2.05E+04	4.01E+04	13,913.148212	3.20E+05	7.68E+08	8,924.213736	1.60E+06	3.20E+06	352.978206
p10	5.01E+04	1.00E+05	286.143934	2.05E+04	4.01E+04	9,359.872966	3.20E+05	7.68E+08	6,686.682374	1.60E+06	3.20E+06	574.874437
p11	5.01E+04	1.00E+05	339.566193	2.05E+04	4.01E+04	9,334.064459	3.20E+05	7.68E+08	8,858.448373	1.60E+06	3.20E+06	433.081417
p12	5.01E+04	1.00E+05	166.026189	1.99E+04	3.89E+04	1,526.959114	3.20E+05	7.17E+08	3,185.192969	1.60E+06	3.20E+06	407.408653
p13	5.01E+04	1.00E+05	163.712280	1.99E+04	3.89E+04	1,788.879687	3.20E+05	6.88E+08	3,898.410600	1.60E+06	3.20E+06	258.943832
p14	5.01E+04	1.00E+05	166.035618	1.99E+04	3.89E+04	1,644.868184	3.20E+05	6.98E+08	3,566.104186	1.60E+06	3.20E+06	379.615072
p15	5.01E+04	1.00E+05	259.410535	2.05E+04	4.01E+04	3,753.716048	3.20E+05	7.68E+08	6,620.626937	1.60E+06	3.20E+06	336.739950
p16	5.01E+04	1.00E+05	248.634833	2.05E+04	4.01E+04	3,570.646515	3.20E+05	7.68E+08	5,750.833458	1.60E+06	3.20E+06	522.294919
p17	5.01E+04	1.00E+05	243.315235	2.05E+04	4.01E+04	3,745.682423	3.20E+05	7.68E+08	6,810.544320	1.60E+06	3.20E+06	336.960630
p18	5.01E+04	1.00E+05	314.445513	2.05E+04	4.01E+04	5,265.510065	3.20E+05	7.68E+08	8,120.196037	1.60E+06	3.20E+06	653.921853
p19	5.01E+04	1.00E+05	347.633648	2.05E+04	4.01E+04	5,502.564954	3.20E+05	7.68E+08	9,673.618252	1.60E+06	3.20E+06	475.422408
p20	5.01E+04	1.00E+05	298.467208	2.05E+04	4.01E+04	5,271.477029	3.20E+05	7.68E+08	7,047.893672	1.60E+06	3.20E+06	657.678029
p21	5.01E+04	1.00E+05	488.290929	2.05E+04	4.01E+04	8,400.814669	3.20E+05	7.68E+08	14,602.310914	1.60E+06	3.20E+06	843.920170
p22	5.01E+04	1.00E+05	372.947026	2.05E+04	4.01E+04	6,949.624929	3.20E+05	7.68E+08	11,950.977247	1.60E+06	3.20E+06	929.005119
p23	5.01E+04	1.00E+05	507.536106	2.05E+04	4.01E+04	8,414.589014	3.20E+05	7.68E+08	15,362.391153	1.60E+06	3.20E+06	844.803838

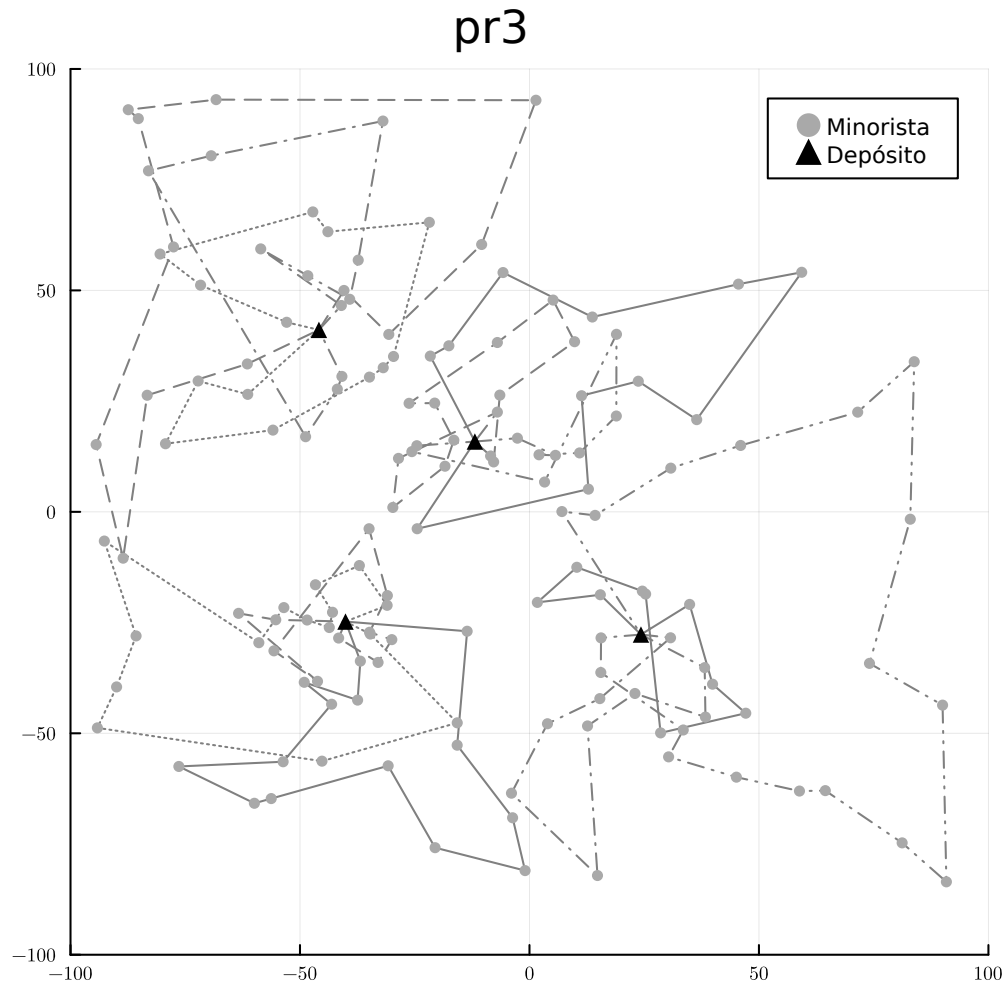


Figura 5.11: Visualizando los resultados cuando ByGA-S resolvió la instancia pr3.

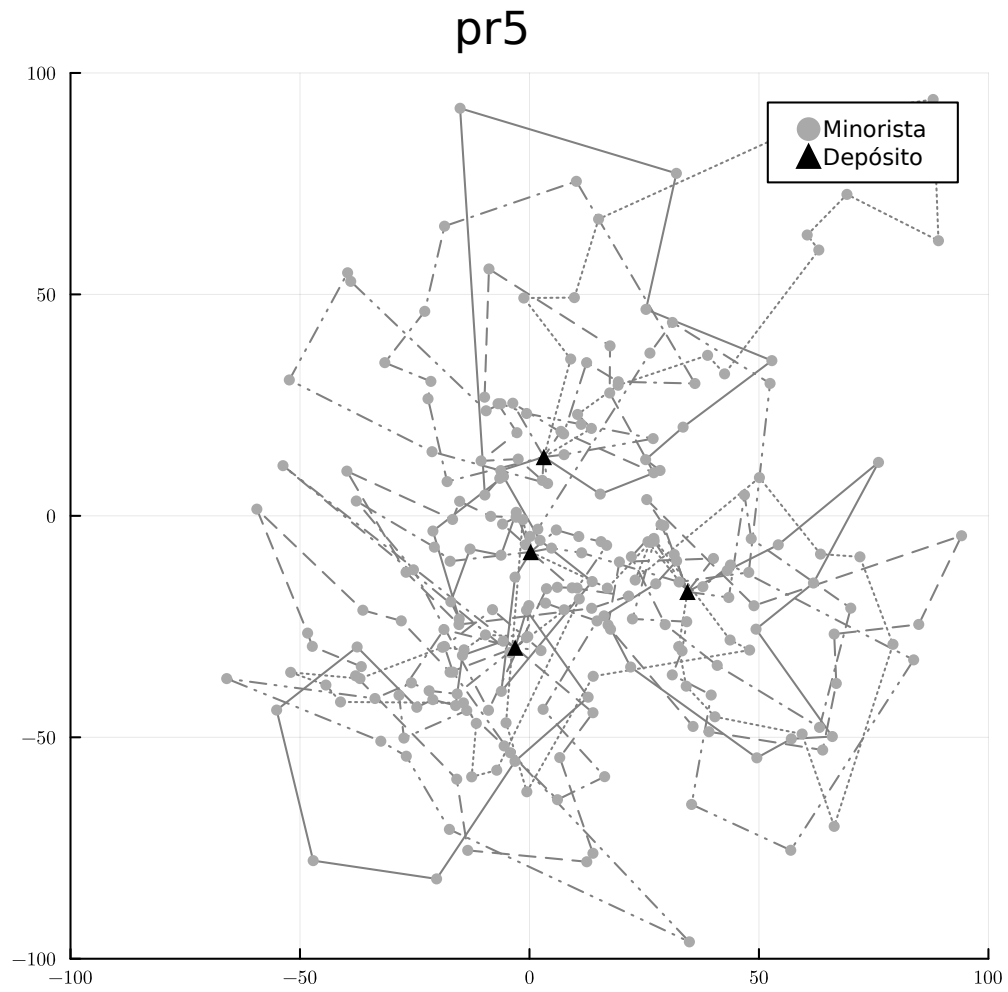


Figura 5.12: Visualizando los resultados cuando ByGA-S resolvió la instancia pr5.

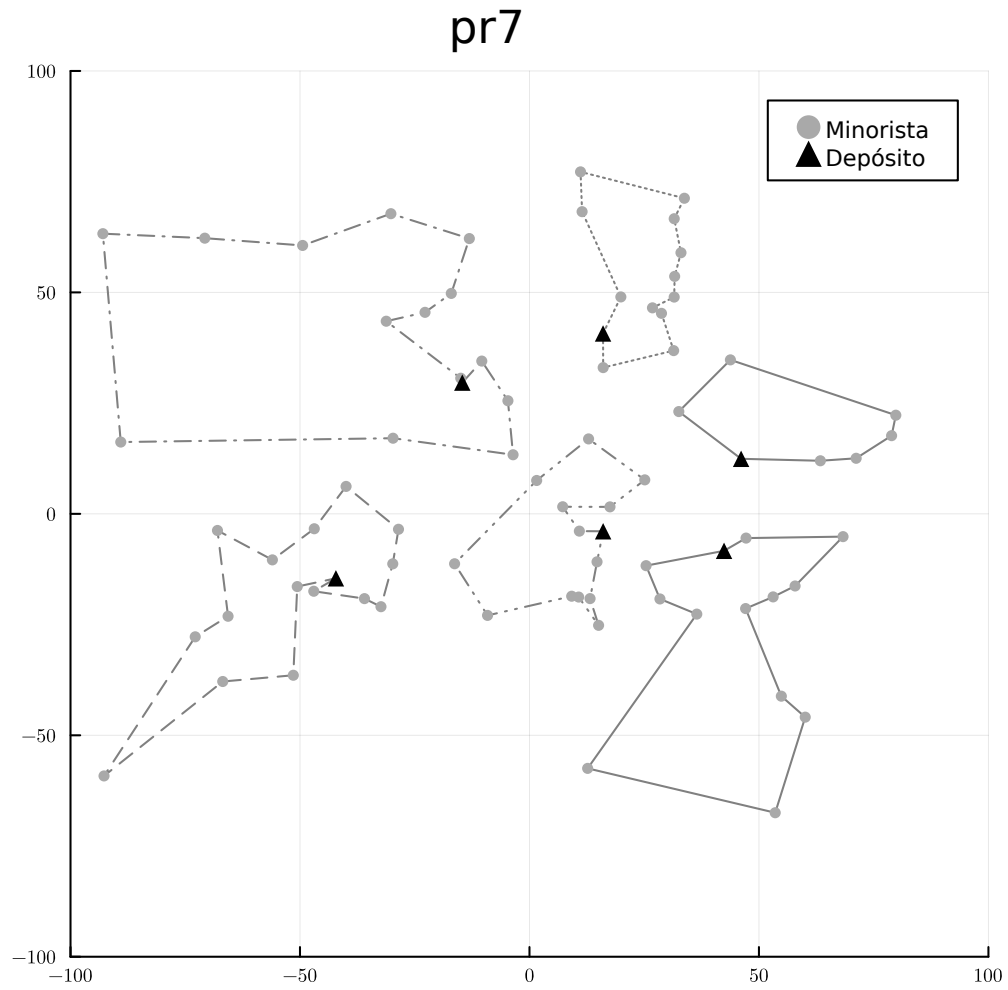


Figura 5.13: Visualizando los resultados cuando ByGA-S resolvió la instancia pr7.

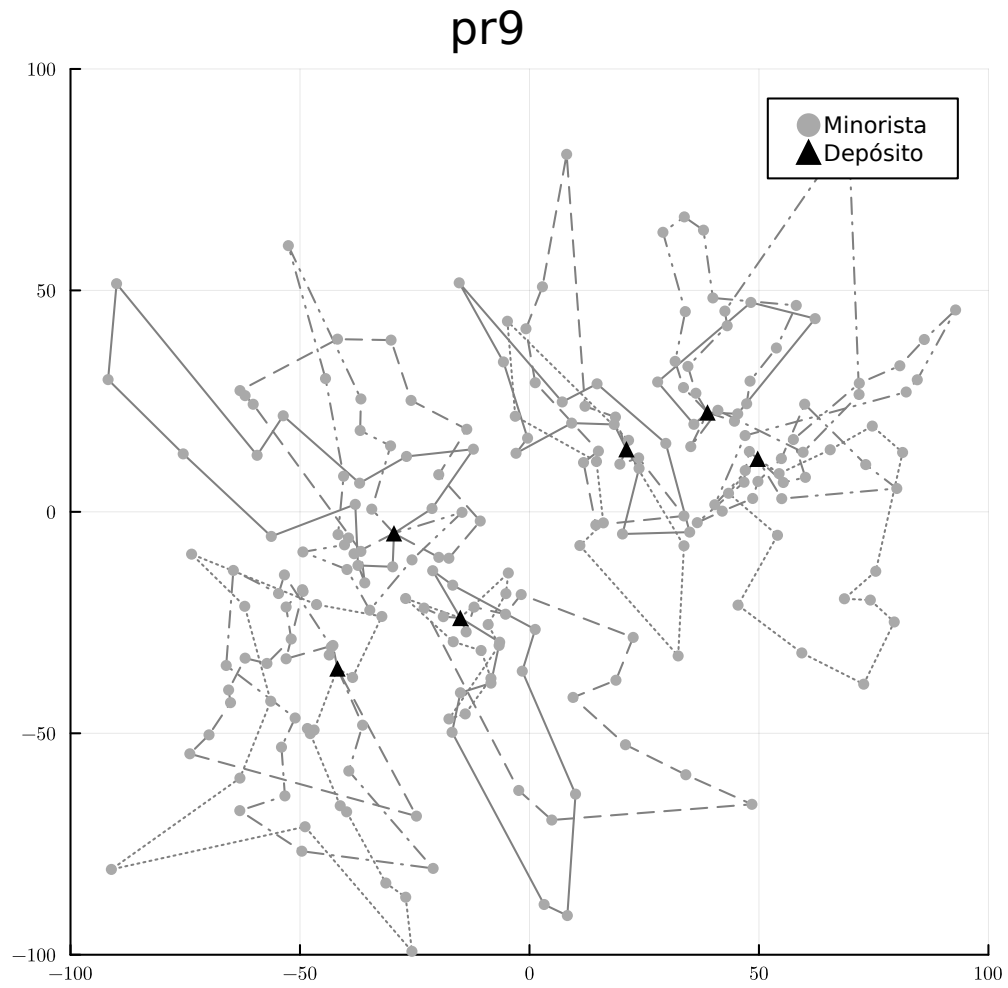


Figura 5.14: Visualizando los resultados cuando ByGA-S resolvió la instancia pr9.

5.5 Discusión

De la experimentación detallada anteriormente, podemos discutir los resultados en tres partes. La primera parte discute los resultados de la primera propuesta ByGA-S. La segunda parte discute los resultados obtenidos por la propuesta ByGA-C. Finalmente se discuten los resultados comparando ByGA-S y ByGA-C.

5.5.1 Discusión sobre ByGA-S

De los resultados obtenidos (Tablas 5.3-5.8) podemos observar que el GAP obtenido por nuestra propuesta es significativamente mejor que CEMBA y CEMBA-H. Además, cuando CEMBA adopta un optimizador exacto en el nivel inferior, entonces muestra un mejor desempeño al reportar valores más altos de GAP. Sin embargo, es suficiente para reportar mejores soluciones que las reportadas por ByGA-S. ByGA-S muestra mejor desempeño debido a que la representación de las soluciones ayuda a generar soluciones factibles, permitiendo a los operadores de mutación y cruza explorar y explotar mejores regiones del espacio de búsqueda. Note que el desempeño de esta ByGA-S es superior en los conjuntos de instancias adoptados en la experimentación.

Respecto al costo computacional, podemos observar que ByGA-S mantiene valores bajos en el número de evaluaciones de la función objetivo en comparación con CEMBA y CEMBA-H. Lo cual nos indica que, en la mayoría de los casos, en los tres conjuntos de instancias; nuestra propuesta menos costosa en este punto. Lo anterior particularmente útil cuando se quiera resolver algún problema de este tipo, pero donde la función objetivo sea más costosa de computar.

5.5.2 Discusión sobre ByGA-C

Notemos que ByGA-C, al ser una variante de ByGA-S, obtiene valores del GAP superiores a los reportados por CEMBA y su variante híbrida, CEMBA-H. Por lo que nos centraremos en discutir los resultados cuando se compara respecto a la

variante ByGA-S.

La experimentación observamos que ByGA-C es significativamente mejor en el 80% de los casos que ByGA-S para el conjunto de instancias *pr*. Las instancias con un peor valor de GAP fueron *pr1* y *pr7*. Estas instancias tienen una cantidad no tan grande de minoristas que deben ser asignados, es decir, el espacio de búsqueda del nivel superior no es tan grande con el resto de instancias para el conjunto de instancias *pr*. Lo que no podría ocurrir es que el nivel de arriba puede tener varios óptimos locales, induciendo a la propuesta converger a dichos valores. Además, esta segunda propuesta mantiene valores superiores de GAP a los producidos por CEMBA y CEMBA-H. Note que este fenómeno se observa en el conjunto de instancias 3.

Finalmente, para el Conjunto de instancias 3 se aprecia que ByGA-C es superior a ByGA-S en todo el conjunto de instancias. Lo cual no indica que evidentemente ByGA-C mejora de desempeño de la primera versión de nuestro algoritmo genético binivel.

5.5.3 Discusión sobre ambos algoritmos

Notemos que cada algoritmo tiene sus ventajas particulares. En primer lugar, destaca que el algoritmo ByGA-C muestra un rendimiento superior en términos del promedio de la función objetivo en el nivel superior, así como en el número de evaluaciones en comparación con los otros algoritmos. Por otro lado, si nos centramos en el algoritmo ByGA-S, observamos una mejora progresiva a medida que avanzan las generaciones, lo que conlleva a una disminución del porcentaje del GAP en comparación con ByGA-C. Las principales ventajas de ByGA-S radican en su menor costo en relación con el tiempo requerido, mientras que ByGA-C requiere un mayor tiempo de ejecución, pero sus resultados son superiores a los reportados por ByGA-S como se puede visualizar en las Tablas 5.13 - 5.23 y las Figuras 5.1 - 5.9.

Dado lo anterior, podemos asegurar que el diseño de nuestras variantes que

utilizan una representación basada en grupos ofrecen resultados competitivos y con significancia estadística en comparación con los algoritmos CEMBA y CEMBA-H cuando se resolvieron los tres conjuntos de instancias. Por lo que se acepta la hipótesis de investigación dada en la Subsección 1.4.

Capítulo 6

Conclusiones y Trabajo a Futuro

Después de presentar los resultados de la experimentación, las conclusiones y el trabajo futuro se presentan en este capítulo.

6.1 Conclusiones

En este trabajo propusimos un algoritmo genético binivel identificado como ByGA para resolver el problema Bi-MDVRP, mismo que es un problema combinatorio binivel que tiene como objetivo planificar de manera óptima la producción y distribución de bienes o productos.

Nuestra propuesta de solución ByGA cuenta con dos variantes con las siguientes características diferenciadoras.

1. ByGA-S: Este algoritmo se caracteriza por tener un mecanismo de iniciali-

zación de soluciones que funciona de la siguiente manera: una solución se genera ingresando de manera aleatoria minoristas a depósitos y luego se ingresa a la sub-ruta asegurándose de que cumpla con las restricciones del problema. La cruce consiste en transmitir minoristas de un depósito a otro de otra solución, seleccionados a partir de un torneo binario. El operador de mutación elimina una cantidad pequeña de minoristas en sus respectivas rutas, y luego se ingresan en otras rutas de manera aleatoria.

2. ByGA-C: Esta versión del algoritmo es una mejora del ByGA-S la cual se destaca por tener como inicialización una estrategia que permite generar una mayor diversidad de soluciones. Además, los operadores genéticos incorporan mecanismos de enrutamiento heurísticos para mejorar los costos de distribución por cada ruta en cada depósito.

Se realizó una extensa experimentación para determinar el desempeño de nuestras propuestas de solución para resolver el problema Bi-MDVRP. Se consideraron dos conjuntos de instancias del estado del arte y un conjunto de instancias generadas que en total suman 43 instancias inspiradas en problemas del mundo real. Comparamos nuestros resultados con un algoritmo del estado del arte competitivo identificado como CEMBA y su variante híbrida CEMBA-H.

Los resultados obtenidos sugieren que el GAP obtenido por nuestra primer variante ByGA-S es significativamente mejor que CEMBA y CEMBA-H. Además, cuando CEMBA adopta un optimizador exacto en el nivel inferior (CEMBA-H), entonces muestra un mejor desempeño al reportar valores más altos de GAP. Sin embargo, CEMBA-H, no es suficiente para reportar mejores soluciones que las reportadas por ByGA-S. Se observó que ByGA-S ha sido superior respecto a CEMBA y CEMBA-H. Adicionalmente, encontramos que ByGA-C, al ser una variante de ByGA-S, obtuvo valores competitivos en comparación con CEMBA, CEMBA-H y ByGA-S.

Respecto al costo computacional, podemos observar que ByGA-C mantiene valores bajos en el número de evaluaciones de la función objetivo en comparación con ByGA-S, CEMBA y CEMBA-H. Esto indica que, en la mayoría de los casos y

para los tres conjuntos de instancias, nuestra propuesta es la menos costosa en este punto. Lo anterior es particularmente útil cuando se quiera resolver algún problema de este tipo, pero donde la función objetivo sea más costosa de computar.

Los principales hallazgos respecto al diseño algorítmico se describe a continuación. La representación de una solución es parte muy importante para que un algoritmo heurístico pueda explorar mejor el espacio de búsqueda factible, una vez que la representación adecuada es adoptada, el proceso de inicialización debe proveer de soluciones diversas entre sí para evitar la convergencia en óptimos locales. Además, la cruza y mutación deben mantener un buen compromiso entre exploración y mutación para evitar converger en óptimos locales.

Finalmente, conocer información acerca del problema de optimización binivel que se está resolviendo es muy importante para proponer mecanismos heurísticos que permitan encontrar soluciones factibles de manera rápida.

6.2 Trabajo a Futuro

- Extensión de nuestras propuestas de algoritmos al caso dinámico del problema Bi-MDVRP, ya que el transporte y la producción suelen ser procesos/entornos que dependen del tiempo. En este caso se generarían un nuevo conjunto de instancias utilizando el conjunto de datos de Problema de Ruteo de Vehículos con Múltiples Depósitos y Ventanas de Tiempo [34].
- Adaptación de las variantes de ByGA a través de Aprendizaje por Transferencia e incorporación de Memes para satisfacer las restricciones y la aceleración de búsqueda en el nivel superior.
- Aplicar los algoritmos propuestos en las rutas de drones podría potenciar la eficiencia y efectividad de las entregas, al considerar restricciones logísticas y técnicas, como la duración de la batería y la altitud máxima permitida para los drones.

Bibliografía

- [1] Malek Abbassi, Abir Chaabani, Nabil Absi, and Lamjed Ben Said. An elitist cooperative evolutionary bi-level multi-objective decomposition-based algorithm for sustainable supply chain. *International Journal of Production Research*, 60(23):7013–7032, 2022.
- [2] Malek Abbassi, Abir Chaabani, and Lamjed Ben Said. An investigation of a bi-level non-dominated sorting algorithm for production-distribution planning system. In Franz Wotawa, Gerhard Friedrich, Ingo Pill, Roxane Koitz-Hristov, and Moonis Ali, editors, *Advances and Trends in Artificial Intelligence. From Theory to Practice*, pages 819–826, Cham, 2019. Springer International Publishing.
- [3] Malek Abbassi, Abir Chaabani, and Lamjed Ben Said. An improved bi-level multi-objective evolutionary algorithm for the production-distribution planning system. In Vicenç Torra, Yasuo Narukawa, Jordi Nin, and Núria Agell, editors, *Modeling Decisions for Artificial Intelligence*, pages 218–229, Cham, 2020. Springer International Publishing.
- [4] Malek Abbassi, Abir Chaabani, and Lamjed Ben Said. An efficient chemical

- reaction algorithm for multi-objective combinatorial bi-level optimization. *Engineering Optimization*, 54(4):665–686, 2022.
- [5] Malek Abbassi, Abir Chaabani, Lamjed Ben Said, and Nabil Absi. Bi-level multi-objective combinatorial optimization using reference approximation of the lower level reaction. *Procedia Computer Science*, 176:2098–2107, 2020. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.
- [6] Federico Alonso, M Jesús Alvarez, and John E Beasley. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, 59:963–976, 2008.
- [7] Maria João Alves and Carlos Henggeler Antunes. A differential evolution algorithm to semivectorial bilevel problems. In Giuseppe Nicosia, Panos Pardalos, Giovanni Giuffrida, and Renato Umeton, editors, *Machine Learning, Optimization, and Big Data*, pages 172–185, Cham, 2018. Springer International Publishing.
- [8] Gülay Barbarosoğlu and Demet Özgür. Hierarchical design of an integrated production and 2-echelon distribution system. *European Journal of Operational Research*, 118(3):464–484, 1999.
- [9] J.F. Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.
- [10] Jonathan F Bard. Some properties of the bilevel programming problem. *Journal of optimization theory and applications*, 68(2):371–378, 1991.
- [11] Arpan Biswas and Christopher Hoyle. A literature review: solving constrained non-linear bi-level optimization problems with classical methods. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 59193, page V02BT03A025. American Society of Mechanical Engineers, 2019.

- [12] Mourad Boudia, Mohamed Aly Ould Louly, and Christian Prins. A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research*, 34(11):3402–3419, 2007.
- [13] Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations research*, 21(1):37–44, 1973.
- [14] Anthony P Burgard, Priti Pharkya, and Costas D Maranas. Optknock: a bi-level programming framework for identifying gene knockout strategies for microbial strain optimization. *Biotechnology and bioengineering*, 84(6):647–657, 2003.
- [15] Herminia I Calvete, Carmen Galé, and María-José Oliveros. Bilevel model for production–distribution planning solved by using ant colony optimization. *Computers & operations research*, 38(1):320–327, 2011.
- [16] Herminia I Calvete, Carmen Galé, and María-José Oliveros. A hybrid algorithm for solving a bilevel production-distribution planning problem. In *International Conference on Modeling and Simulation in Engineering, Economics and Management*, pages 138–144. Springer, 2013.
- [17] Herminia I Calvete, Carmen Galé, José A Iranzo, José-Fernando Camacho-Vallejo, and Martha-Selene Casas-Ramírez. A matheuristic for solving the bilevel approach of the facility location problem with cardinality constraints and preferences. *Computers & Operations Research*, 124:105066, 2020.
- [18] José-Fernando Camacho-Vallejo, Rafael Muñoz-Sánchez, and José Luis González-Velarde. A heuristic algorithm for a supply chain’s production-distribution planning. *Computers & Operations Research*, 61:110–121, 2015.
- [19] Wilfred Candler and Roger Norton. *Multi-level programming and development policy*. The World Bank, 1977.
- [20] Abir Chaabani, Slim Bechikh, and Lamjed Ben Said. *A co-evolutionary decomposition-based algorithm for Bi-Level combinatorial optimization*. 2015.

- [21] Abir Chaabani, Slim Bechikh, and Lamjed Ben Said. A co-evolutionary hybrid decomposition-based algorithm for bi-level combinatorial optimization problems. *Soft Computing*, 24(10):7211–7229, 2020.
- [22] Abir Chaabani, Slim Bechikh, Lamjed Ben Said, and Radhia Azzouz. An improved co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1363–1364, 2015.
- [23] Abir Chaabani, Slim Bechikh, Lamjed Ben Said, and Radhia Azzouz. An improved co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, page 1363–1364, New York, NY, USA, 2015. Association for Computing Machinery.
- [24] Abir Chaabani, Slim Bechikh, and Lamjed Ben Said. A memetic evolutionary algorithm for bi-level combinatorial optimization: a realization between bi-mdvrp and bi-cvrp. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1666–1673. IEEE, 2016.
- [25] Abir Chaabani, Slim Bechikh, and Lamjed Ben Said. A co-evolutionary decomposition-based chemical reaction algorithm for bi-level combinatorial optimization problems. *Procedia Computer Science*, 112:780–789, 2017. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France.
- [26] Abir Chaabani, Slim Bechikh, and Lamjed Ben Said. A new co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization. *Applied Intelligence*, 48(9):2847–2872, 2018.
- [27] Abir Chaabani and Lamjed Ben said. Hybrid codba-ii algorithm coupling a co-evolutionary decomposition-based algorithm with local search method to

- solve bi-level combinatorial optimization. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 506–513, 2018.
- [28] Abir Chaabani and Lamjed Ben Said. Transfer of learning with the co-evolutionary decomposition-based algorithm-ii: a realization on the bi-level production-distribution planning system. *Applied Intelligence*, 49:963–982, 2019.
- [29] Chi-Bin Cheng, Hsu-Shih Shih, and Boris Chen. Subsidy rate decisions for the printer recycling industry by bi-level optimization techniques. *Operational Research*, 17:901–919, 2017.
- [30] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007.
- [31] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. Academic press, 2013.
- [32] Morris A Cohen and Hau L Lee. Strategic analysis of integrated production-distribution systems: models and methods. *Operations research*, 36(2):216–228, 1988.
- [33] Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks: An International Journal*, 30(2):105–119, 1997.
- [34] Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8):928–936, 2001.
- [35] Philip Tobianto Daely, Yohana Jayanti Aruan, Jae Min Lee, and Dong-Seong Kim. Dynamic vrp optimization using discrete pso in edge computing environment. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 654–656, 2021.

- [36] Egidio D’Amato, Immacolata Notaro, Francesco Silvestre, and Massimiliano Mattei. Bi-level flight path optimization for uav formations. In *2017 international conference on unmanned aircraft systems (ICUAS)*, pages 690–697. IEEE, 2017.
- [37] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.
- [38] Jesús-Adolfo Mejía de Dios, Efrén Mezura-Montes, and Marcela Quiroz-Castellanos. Automated parameter tuning as a bilevel optimization problem solved by a surrogate-assisted population-based approach. *Applied Intelligence*, 51(8):5978–6000, January 2021.
- [39] Kalyanmoy Deb and Ankur Sinha. An efficient and accurate solution methodology for bilevel multi-objective programming problems using a hybrid evolutionary-local-search algorithm. *Evolutionary computation*, 18(3):403–449, 2010.
- [40] S. Dempe. *Foundations of Bilevel Programming*. Springer Science & Business Media, 2002.
- [41] S. Dempe and J. Dutta. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Mathematical Programming*, 131(1-2):37–48, February 2010.
- [42] Stephan Dempe. Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *Optimization*, 52(3):333–359, 2003.
- [43] Joaquín Derrac, Salvador García, Daniel Molina, and Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.
- [44] Behnam Fahimnia, Reza Zanjirani Farahani, Romeo Marian, and Lee Luong. A review and critique on integrated production–distribution planning models and techniques. *Journal of Manufacturing Systems*, 32(1):1–19, 2013.

- [45] Marc Goetschalckx, Carlos J Vidal, and Koray Dogan. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European journal of operational research*, 143(1):1–18, 2002.
- [46] Zhong-Hua Han, Ke-Shi Zhang, et al. Surrogate-based optimization. *Real-world applications of genetic algorithms*, 343:343–362, 2012.
- [47] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on scientific and Statistical Computing*, 13(5):1194–1217, 1992.
- [48] H. I. Calvete., C. Galé., and M. J. Oliveros. Evolutive and aco strategies for solving the multi-depot vehicle routing problem. In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications - ECTA, (IJCCI 2011)*, pages 73–79. INSTICC, SciTePress, 2011.
- [49] Md Monjurul Islam, Hemant Kumar Singh, and Tapabrata Ray. Use of a non-nested formulation to improve search for bilevel optimization. In *AI 2017: Advances in Artificial Intelligence: 30th Australasian Joint Conference, Melbourne, VIC, Australia, August 19–20, 2017, Proceedings 30*, pages 106–118. Springer, 2017.
- [50] Md Monjurul Islam, Hemant Kumar Singh, Tapabrata Ray, and Ankur Sinha. An enhanced memetic algorithm for single-objective bilevel optimization problems. *Evolutionary computation*, 25(4):607–642, 2017.
- [51] Robert G Jeroslow. The polynomial hierarchy and a simple model for competitive analysis. *Mathematical programming*, 32(2):146–164, 1985.
- [52] Victor Klee and George J Minty. How good is the simplex algorithm. *Inequalities*, 3(3):159–175, 1972.
- [53] Hande Küçükaydın, Necati Aras, and I Kuban Altinel. A hybrid tabu search heuristic for a bilevel competitive facility location model. In *Hybrid Metaheuristics: 7th International Workshop, HM 2010, Vienna, Austria, October 1-2, 2010. Proceedings 7*, pages 31–45. Springer, 2010.

- [54] Gilbert Laporte, Yves Nobert, and Serge Taillefer. Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation science*, 22(3):161–172, 1988.
- [55] François Legillon, Arnaud Liefoghe, and El-Ghazali Talbi. Cobra: A cooperative coevolutionary algorithm for bi-level optimization. In *2012 IEEE Congress on evolutionary computation*, pages 1–8. IEEE, 2012.
- [56] François Legillon, Arnaud Liefoghe, and El-Ghazali Talbi. *CoBRA: A Coevolutionary Metaheuristic for Bi-level Optimization*, pages 95–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [57] Jan Karel Lenstra and AHG Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.
- [58] Song Li and Ling-hui Chen. Optimization of the vrp with single depot based on vehicle coordination strategy. In *2010 International Conference on Intelligent Computation Technology and Automation*, volume 2, pages 862–865, 2010.
- [59] Bingbing Liu, Zhongping Wan, Jiawei Chen, and Guangmin Wang. Optimality conditions for pessimistic semivectorial bilevel programming problems. *Journal of Inequalities and Applications*, 2014(1):1–26, 2014.
- [60] June Liu, Yuxin Fan, Zhong Chen, and Yue Zheng. Pessimistic bilevel optimization: a survey. *International Journal of Computational Intelligence Systems*, 11(1):725–736, 2018.
- [61] Wei Liu, Zheng Zheng, and Kai-Yuan Cai. Bi-level programming based real-time path planning for unmanned aerial vehicles. *Knowledge-Based Systems*, 44:34–47, 2013.
- [62] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez-Cáceres, M. Birattari, and T. Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

- [63] P. Marcotte, G. Savard, and D. L. Zhu. A trust region algorithm for nonlinear bilevel programming. *Operations research letters*, 29(4):171–179, 2001.
- [64] Patrice Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical programming*, 34(2):142–162, 1986.
- [65] Jesús Mejía. jmejia8/bilevel-optimization-references: Bibliographic References on Bilevel Optimization, June 2023.
- [66] Jesús-Adolfo Mejía-de Dios, Alejandro Rodríguez-Molina, and Efrén Mezura-Montes. Multiobjective bilevel optimization: A survey of the state-of-the-art. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–0, 2023.
- [67] Miles G Nicholls. The application of non-linear bi-level programming to the aluminium industry. *Journal of Global Optimization*, 8:245–261, 1996.
- [68] Calice Olivier Pieume, Patrice Marcotte, Laure Pauline Fotso, Patrick Siarry, et al. Generating efficient solutions in bilevel multi-objective programming problems. *American Journal of Operations Research*, 3(02):289, 2013.
- [69] Marcela Quiroz-Castellanos, Laura Cruz-Reyes, Jose Torres-Jimenez, Claudia Gómez S., Héctor J. Fraire Huacuja, and Adriana C.F. Alvim. A grouping genetic algorithm with controlled gene transmission for the bin packing problem. *Computers & Operations Research*, 55:52–64, 2015.
- [70] Octavio Ramos-Figueroa, Marcela Quiroz-Castellanos, Efrén Mezura-Montes, and Oliver Schütze. Metaheuristics to solve grouping problems: A review and a case study. *Swarm and Evolutionary Computation*, 53:100643, 2020.
- [71] Gerhard Reinelt. *The traveling salesman: computational solutions for TSP applications*, volume 840. Springer, 2003.
- [72] Kemal H Sahin and Amy R Ciric. A dual temperature simulated annealing approach for solving bilevel programming problems. *Computers & chemical engineering*, 23(1):11–25, 1998.

- [73] Rihab Said, Slim Bechikh, Ali Louati, Abdulaziz Aldaej, and Lamjed Ben Said. Solving combinatorial multi-objective bi-level optimization problems using multiple populations and migration schemes. *IEEE Access*, 8:141674–141695, 2020.
- [74] Rihab Said, Maha Elarbi, Slim Bechikh, and Lamjed Ben Said. Solving combinatorial bi-level optimization problems using multiple populations and migration schemes. *Operational Research*, pages 1–39, 2021.
- [75] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. Transportation policy formulation as a multi-objective bilevel optimization problem. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 1651–1658. IEEE, 2015.
- [76] Heinrich von Stackelberg et al. *Theory of the market economy*. 1952.
- [77] E. G. Talbi. A taxonomy of metaheuristics for bi-level optimization. In *Metaheuristics for bi-level optimization*, pages 1–39. Springer, 2013.
- [78] Rajiv R Thareja and Raphael T Haftka. Efficient single-level solution of hierarchical problems in structural optimization. *AIAA journal*, 28(3):506–514, 1990.
- [79] L. Vicente, G. Savard, and J. Júdice. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications*, 81(2):379–399, 1994.
- [80] L. N Vicente and P. H. Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994.
- [81] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*, 2nd rev. 1947.
- [82] Judith YT Wang, Matthias Ehrgott, Kim N Dirks, and Abhishek Gupta. A bilevel multi-objective road pricing model for economic, environmental and health sustainability. *Transportation Research Procedia*, 3:393–402, 2014.
- [83] Wenxing Xiao, Gang Du, Yingying Zhang, and Xiaojie Liu. Coordinated optimization of low-carbon product family and its manufacturing process design

- by a bilevel game-theoretic model. *Journal of cleaner production*, 184:754–773, 2018.
- [84] Y. Yin. Multiobjective bilevel optimization for transportation planning and management problems. *Journal of advanced transportation*, 36(1):93–105, 2002.
- [85] Yafeng Yin. Genetic-algorithms-based approach for bilevel programming models. *Journal of transportation engineering*, 126(2):115–120, 2000.
- [86] Theopisti Zaza and Arthur Richards. Ant colony optimization for routing and tasking problems for teams of uavs. In *2014 UKACC International Conference on Control (CONTROL)*, pages 652–655, 2014.