



Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

COLONIA DE HORMIGAS PARA EL
PROBLEMA CLÁSICO DE RUTAS DE
VEHÍCULOS

T E S I S

QUE PARA OBTENER POR EL GRADO DE:
LICENCIADA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

TERESA BECERRIL TORRES

TUTOR:

DRA. MARÍA DE LUZ GASCA SOTO



CIUDAD UNIVERSITARIA, CD. MX., 2023



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

1. Datos del alumno

Becerril

Torres

Teresa

5532405909

Universidad Nacional Autónoma de México

Facultad de Ciencias

Ciencias de la Computación

315045132

2. Datos del tutor

Dra.

María de Luz

Gasca

Soto

3. Datos del sinodal 1

Dra.

Katya

Rodríguez

Vázquez

4. Datos del sinodal 2

Dr.

José de Jesús

Galaviz

Casas

5. Datos del sinodal 3

Dr.

Canek

Peláez

Valdés

6. Datos del sinodal 4

M. en C.

Oscar

Hernández

Constantino

7. Datos del trabajo escrito.

Colonia de hormigas para el Problema Clásico de Rutas de Vehículos

83 p.

2023

A mi esfuerzo, a mi familia y a mi estrella favorita, Lucy.

Agradecimientos

A mi hermana por su amor, comprensión y apoyo incondicional.

A mis padres por su amor, confianza, motivación y por acompañarme en cada paso.

A mis abuelitos por su cariño y por todas sus enseñanzas.

A la Doctora Luz por su guía, apoyo y confianza.

A la Facultad de Ciencias por brindarme su conocimiento y guía.

A la UNAM por permitirme cumplir mi sueño y permitirme conocer a grandes personas.

Índice general

Índice de figuras	VII
Índice de tablas	IX
Introducción	1
1. Optimización por Colonias de Hormigas	3
1.1. Antecedentes	3
1.2. Comportamiento de las hormigas para buscar alimento	4
1.3. Metaheurística ACO	6
1.3.1. Aplicaciones	7
1.4. Similitudes y diferencias entre las hormigas reales y artificiales	8
2. Problema de Rutas de Vehículos	9
2.1. Variantes del VRP	10
2.2. Definición formal del CVRP	14
2.3. Ejemplo del CVRP	16
3. La metaheurística ACO para el CVRP	19
3.1. Metaheurística ACO para el CVRP	19
3.2. Construcción de las rutas	23
3.3. Actualización del rastro feromonas	24
3.4. Múltiples colonias de hormigas	24
3.5. Estrategias de mejora	25
3.5.1. Heurística 2-opt	25
3.5.2. Listas de candidatos	26
4. Implementación del programa	27
4.1. Detalles de la implementación	27
4.2. Implementación	27
4.3. Instrucciones de ejecución	30
5. Experimentos y Resultados	35
5.1. Objetivo	35
5.2. Detalles de la ejecución	35

5.3. Detalles del experimento	35
5.3.1. Parámetros de configuración	36
5.3.2. Ejemplar de 15 ciudades	36
5.3.3. Ejemplar de 20 ciudades	37
5.3.4. Ejemplar de 30 ciudades	38
5.3.5. Tamaños de las listas de candidatos	40
5.4. Resultados	40
5.4.1. Ejemplar de 15 ciudades	40
5.4.2. Ejemplar de 20 ciudades	44
5.4.3. Ejemplar de 30 ciudades	48
5.5. Comparación de resultados	52
5.5.1. Ejemplar de 20 ciudades	52
5.5.2. Ejemplar de 30 ciudades	54
Conclusiones	57
Apéndices	59
A. Teoría de Gráficas	59
A.1. Conceptos básicos	59
A.2. Notación para gráficas	61
B. Complejidad Computacional	63
B.1. Conceptos básicos	63
C. Heurística 2-opt	67
D. Recursos de ejecución	71
E. Detalles de las ciudades	73
E.1. Lista de Ciudades	73
E.2. Tabla de distancias	76
Bibliografía	81
Índice alfabético	83

Índice de figuras

1.1. Las hormigas siguen una ruta entre el nido y la fuente de alimento.	5
1.2. Aparece un obstáculo en la ruta.	5
1.3. Las hormigas eligen si giran a la izquierda o a la derecha con la misma probabilidad. El rastro de feromonas se acumula más rápidamente en la ruta más corta.	5
1.4. Todas las hormigas han elegido la ruta más corta.	6
2.1. Representación gráfica de un ejemplar del CVRP.	11
2.2. Representación gráfica de un ejemplar del HFVRP.	11
2.3. Representación gráfica de un ejemplar del MDVRP.	12
2.4. Representación gráfica de un ejemplar del VRPPD.	13
2.5. Representación gráfica de un ejemplar del VRPTW.	14
2.6. Ejemplar del CVRP.	16
2.7. Solución R para el ejemplar del CVRP.	17
2.8. Solución R' para el ejemplar del CVRP.	17
5.1. Distancias de las mejores soluciones de E_{20} en este trabajo.	52
5.2. Distancias de las mejores soluciones de E_{20} en la Tesis de Estudillo.	53
5.3. Demandas de las mejores soluciones de E_{20} en ambos trabajos.	53
5.4. Distancias de las mejores soluciones de E_{30} en este trabajo.	54
5.5. Distancias de las mejores soluciones de E_{30} en la Tesis de J. Estudillo.	55
5.6. Demandas de las mejores soluciones de E_{30} en ambos trabajos.	56
C.1. Inicio del proceso de la heurística 2-opt.	68
C.2. Posibles intercambios de las aristas (a, g) y (b, c)	69
C.3. Eliminar las aristas (b, g) y (d, e)	69
C.4. Intercambiar las aristas (b, g) y (d, e) por las aristas (b, e) y (g, d)	70
C.5. Eliminar las aristas (e, h) y (g, f)	70
C.6. Intercambiar las aristas (e, h) y (g, f) por las aristas (e, f) y (h, g)	70

Índice de tablas

1.1. Aplicaciones de la ACO.	7
2.1. Rutas de la solución R para el ejemplar del CVRP.	16
2.2. Rutas de la solución R' para el ejemplar del CVRP.	18
5.1. Parámetros de configuración de los ejemplares.	36
5.2. Ciudades y depósito de E_{15}	36
5.3. Ciudades y depósito de E_{20}	37
5.4. Ciudades y depósito de E_{30}	38
5.5. Tamaños de las listas de candidatos para cada ejemplar.	40
5.6. Solución de E_{15} con CHU sin estrategias de mejora.	40
5.7. Solución de E_{15} con CHU y heurística 2-Opt.	41
5.8. Solución de E_{15} con CHU y listas de candidatos de longitud 4.	41
5.9. Solución de E_{15} con CHU y listas de candidatos de longitud 8.	41
5.10. Solución de E_{15} con CHM sin estrategias de mejora.	42
5.11. Solución de E_{15} con CHM y heurística 2-Opt.	42
5.12. Solución de E_{15} con CHM y listas de candidatos de longitud 4.	42
5.13. Solución de E_{15} con CHM y listas de candidatos de longitud 8.	43
5.14. Solución de E_{20} con CHU sin estrategias de mejora.	44
5.15. Solución de E_{20} con CHU y heurística 2-Opt.	44
5.16. Solución de E_{20} con CHU y listas de candidatos de longitud 5.	44
5.17. Solución de E_{20} con CHU y listas de candidatos de longitud 10.	45
5.18. Solución de E_{20} con CHM sin estrategia de mejora.	45
5.19. Solución de E_{20} con CHM y heurística 2-Opt.	46
5.20. Solución de E_{20} con CHM y listas de candidatos de longitud 5.	46
5.21. Solución de E_{20} con CHM y listas de candidatos de longitud 10.	46
5.22. Solución de E_{30} con CHU sin estrategia de mejora.	48
5.23. Solución de E_{30} con CHU y heurística 2-Opt.	48
5.24. Solución de E_{30} con CHU y listas de candidatos de longitud 8.	49
5.25. Solución de E_{30} con CHU y listas de candidatos de longitud 15.	49
5.26. Solución de E_{30} con CHM sin estrategias de mejora.	50
5.27. Solución de E_{30} con CHM y heurística 2-Opt.	50
5.28. Solución de E_{30} con CHM y listas de candidatos de longitud 8.	50
5.29. Solución de E_{30} con CHM y listas de candidatos de longitud 15.	51
5.30. Variación de las distancias de las soluciones de E_{20}	53

ÍNDICE DE TABLAS

5.31. Variación de las demandas de las soluciones de E_{20}	54
5.32. Variación de las distancias de las soluciones de E_{30}	55
5.33. Variación de las demandas de las soluciones de E_{30}	56
A.1. Notación para gráficas	61
D.1. Promedio de los recursos utilizados por cada método para E_{20}	71
D.2. Promedio de los recursos utilizados por cada método para E_{30}	72

Introducción

El **Problema de Rutas de Vehículos** consiste en encontrar el conjunto óptimo o cercano al óptimo de rutas para una flota, de tal forma que satisfagan las demandas de un conjunto de clientes dispersos geográficamente alrededor de un depósito central, minimizando costos. Sin embargo, al ser un problema **NP-difícil**, no se conocen métodos exactos y eficientes que resuelvan este problema, por ello en este trabajo se propone un enfoque heurístico para tratar el problema.

El enfoque de **Optimización de Colonia de Hormigas** simula el comportamiento de las hormigas en la naturaleza cuando buscan alimento, ya que las hormigas son capaces tanto de encontrar la ruta más corta desde su nido hasta las fuentes de alimento como de adaptarse a los cambios del entorno. El enfoque permite ver a las hormigas como agentes independientes que cooperan entre ellas para encontrar soluciones a diversos problemas de optimización; en particular, al Problema de Rutas de Vehículos.

Este trabajo tiene por objetivo principal analizar y comparar la calidad de las soluciones generadas por dos métodos de optimización diferentes (colonia de hormigas única y múltiples), así como evaluar el impacto del uso de estrategias de mejora (listas de candidatos y heurística 2-opt) en la calidad de la solución construida por cada uno de estos métodos para el Problema de Rutas de Vehículos Clásico. Para alcanzar este objetivo, se realizaron las siguientes tareas:

1. Exponer el marco teórico tanto del enfoque de Optimización de Colonia de Hormigas como del Problema de Rutas de Vehículos.
2. Explicar e implementar los métodos de optimización (colonia de hormigas única y múltiples) y las estrategias de mejora (listas de candidatos y heurística 2-opt).
3. Analizar el desempeño computacional de cada uno de los métodos de optimización sobre ejemplares a través de una serie de ejecuciones. Así como evaluar el impacto del uso de diferentes tamaños de listas de candidatos y el uso de la heurística 2-opt en la calidad de la solución construida por cada uno de estos.
4. Comparar los resultados obtenidos con otros trabajos que tratan el mismo problema utilizando diferentes estrategias.

Estructura de la tesis

- Capítulo 1 **Optimización por Colonias de Hormigas**. Se dan los antecedentes, iniciando con la descripción del comportamiento de las hormigas para buscar alimento; continuando con una breve descripción del funcionamiento de la heurística; finalmente, se dan las similitudes y diferencias entre las hormigas reales y las artificiales.
- Capítulo 2 **Problema de Rutas de Vehículos**. Se presenta la descripción general del problema, las variantes más utilizadas y la definición formal de la variante con la que se va a trabajar.
- Capítulo 3 **La metaheurística ACO para el CVRP**. Se dan los detalles de los métodos de optimización colonia de hormiga única y múltiples, así como de las estrategias de mejora (lista de candidatos y heurística 2-opt).
- Capítulo 4 **Implementación del programa**. Se explica brevemente tanto la implementación de los métodos de optimización y las estrategias de mejora como su ejecución.
- Capítulo 5 **Experimentos y Resultados**. Se describe el objetivo de los experimentos, los detalles de la implementación, los detalles de los experimentos realizados: los ejemplares y su configuración, así como los resultados obtenidos.
- Apéndice A **Teoría de Gráficas**. Se presentan algunas definiciones básicas sobre Teoría de Gráficas utilizadas en este trabajo, para establecer la notación.
- Apéndice B **Complejidad Computacional**. Se dan algunas definiciones básicas sobre Complejidad Computacional usadas en este trabajo.
- Apéndice C **Heurística 2-opt**. Se presenta una breve descripción de la heurística 2-opt y de su funcionamiento.
- Apéndice D **Recursos de ejecución**. Se dan los recursos utilizados por cada método para construir las soluciones de los ejemplares.
- Apéndice E **Detalles de las ciudades**. Se presenta la lista de ciudades que se utilizaron en los experimentos y las distancias entre cada par de ciudades.

Capítulo 1

Optimización por Colonias de Hormigas

La **Optimización por Colonias de Hormigas**, *ACO*¹, forma parte de la **Inteligencia de Enjambres**, la cual estudia los patrones del comportamiento colectivo de las abejas, termitas, hormigas, entre otros insectos sociales para simular procesos. Debido a la capacidad de los enjambres para prosperar en la naturaleza y resolver tareas complejas de supervivencia.

La ACO es una **metaheurística** (técnica heurística de alto nivel) que simula el comportamiento de las colonias de hormigas en la naturaleza para buscar alimento y posee habilidades mejoradas como: conocer las distancias a otros lugares y guardar las acciones pasadas en una memoria. Los principios de autoorganización que permiten el comportamiento altamente coordinado de las hormigas reales pueden aprovecharse para coordinar poblaciones de agentes independientes (hormigas) que cooperan entre ellos para encontrar soluciones a los problemas de optimización combinatoria.

1.1. Antecedentes

Marco Dorigo publicó en 1992 la primera implementación de ACO en su Tesis Doctoral [1] y la llamó **Sistema de Hormigas**, *AS*², esta implementación consistió en tres algoritmos: *ant-quantity*, *ant-density* y *ant-cycle*. La diferencia entre estos tres algoritmos era que en *ant-quantity* y *ant-density* la cantidad de feromonas se actualizaban cada vez que las hormigas se movían de una ciudad a otra, mientras que en *ant-cycle* la cantidad de feromonas se actualizaban una vez que todas las hormigas habían terminado de construir el camino y la calidad de dicho camino se utilizaba para actualizar la cantidad de feromonas. El algoritmo *ant-cycle* tenía un mejor rendimiento, por lo que la investigación de los otros dos algoritmos se detuvo y el algoritmo *ant-cycle* se utilizó para comparar sus resultados con otros algoritmos ya establecidos.

El Sistema de Hormigas se aplicó por primera vez al Problema del Agente Viajero y al Problema de la Asignación Cuadrática, desde entonces se crearon diferentes variantes

¹Siglas en inglés de *Ant Colony Optimization*. Se utilizará el acrónimo ACO para referirse a esta metaheurística.

²Siglas en inglés de *Ant System*.

que intentaron mejorar su rendimiento. En 1999 la metaheurística ACO fue definida formalmente por Dorigo, Di Carlo y Gambardella [2, 3], para establecer un marco teórico para todas las metaheurísticas basadas en colonias de hormigas.

1.2. Comportamiento de las hormigas para buscar alimento

En la naturaleza, una hormiga no puede buscar comida de manera eficaz individualmente, ya que sus capacidades son muy limitadas. En cambio, las hormigas en grupo poseen la capacidad de resolver problemas complejos como: encontrar las rutas más cortas desde su nido hasta las fuentes de alimento, trasladar eficazmente el alimento a su colonia y adaptarse a los cambios del entorno. Las hormigas se coordinan a través de la **estigmergia**, la cual es una forma de comunicación indirecta entre agentes y se basa en la secreción de **feromonas**³ en el ambiente.

Cada hormiga se mueve de forma aleatoria y va depositando una cantidad constante de feromonas, cuando encuentra un rastro de feromonas dejado por otras hormigas debe decidir si lo sigue o no. En caso de que decida seguir dicho rastro, las feromonas de la hormiga refuerzan el rastro actual, provocando que la ruta se vuelva más atractiva para las siguientes hormigas.

A medida que más hormigas son capaces de completar las rutas más cortas, las feromonas se acumulan más rápido en dichas rutas y las rutas más largas se refuerzan menos. La evaporación de las feromonas hace que las rutas menos deseables sean más difíciles de detectar y disminuye aún más su uso.

Sin embargo, siempre existe la posibilidad de que una hormiga no siga una ruta con un alto nivel de feromonas. Esto permite descubrir rutas más cortas, alternativas o nuevas fuentes de alimento. Lo cual garantiza una navegación exitosa alrededor de los obstáculos que interrumpen una ruta. La selección de rutas por hormigas es un proceso pseudoaleatorio y es un elemento clave de la metaheurística ACO [4].

Ejemplo

Las hormigas se mueven por una ruta que conecta su nido con una fuente de alimento, como se muestra en la figura 1.1. Cada hormiga va depositando una cierta cantidad de feromonas mientras se mueve y es probable que prefiera seguir una ruta con un alto nivel de feromonas. El rastro de feromonas puede ser interrumpido por la aparición repentina de un obstáculo en la ruta, como se ilustra en la figura 1.2 provocando que las hormigas que se encuentran delante del obstáculo, no puedan seguir el rastro de feromonas y, por tanto, tienen que elegir entre girar a la derecha o a la izquierda. En promedio, una mitad de las hormigas van a elegir girar a la derecha y la otra mitad a la izquierda, como se muestra en la figura 1.3. Las hormigas que eligen la ruta más corta alrededor del obstáculo reconstruirán más rápido el rastro de feromonas en comparación con las que eligen la ruta más larga. A pesar de que todas las hormigas se mueven aproximadamente a la misma velocidad y depositan un rastro de feromonas aproximadamente al mismo

³Las feromonas son sustancias químicas que estimulan determinados comportamientos en otros individuos, generalmente de la misma especie, pero también pueden emplearse contra otras especies.

ritmo, es un hecho que se van a tardar más en recorrer el obstáculo por su lado más largo que por su lado más corto, lo que hace que el rastro de feromonas se acumule más rápidamente en el lado más corto, ya que las hormigas tienen una preferencia por estas rutas. Por lo que después de un tiempo, la mayoría de las hormigas elegirán la ruta más corta, como se ilustra en la figura 1.4, mientras que las hormigas restantes al no seguir el rastro de feromonas pueden descubrir rutas alternativas o nuevas fuentes de alimentos.

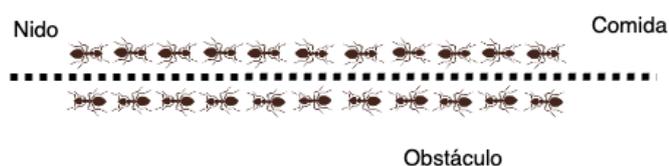


Figura 1.1: Las hormigas siguen una ruta entre el nido y la fuente de alimento.

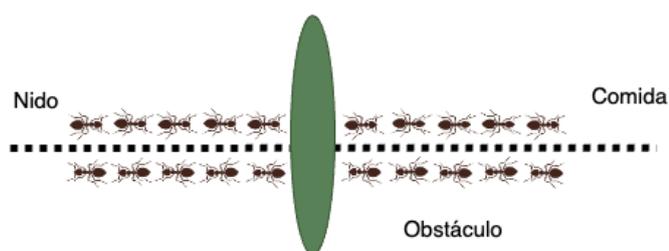


Figura 1.2: Aparece un obstáculo en la ruta.

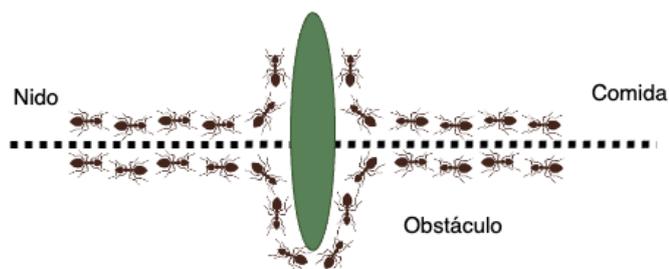


Figura 1.3: Las hormigas eligen si giran a la izquierda o a la derecha con la misma probabilidad. El rastro de feromonas se acumula más rápidamente en la ruta más corta.

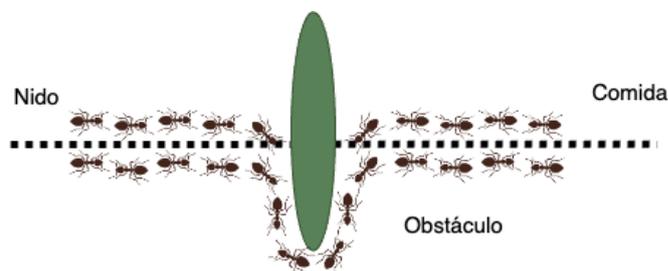


Figura 1.4: Todas las hormigas han elegido la ruta más corta.

1.3. Metaheurística ACO

Una metaheurística es una heurística de alto nivel que utiliza conceptos algorítmicos para guiar y modificar a otras heurísticas con el objetivo de evitar que dichas heurísticas queden atrapados en óptimos locales [5]. La metaheurística ACO consiste en un conjunto finito de agentes independientes (hormigas artificiales) que cooperan entre ellos para encontrar soluciones de alta calidad para problemas de optimización combinatoria en un tiempo razonable.

Cada agente construye una solución moviéndose a través de un conjunto finito de estados vecinos. Los movimientos se eligen aplicando una fórmula probabilística que utiliza la memoria del agente, el rastro de feromonas y la información local del problema. La memoria del agente almacena sus acciones pasadas y puede utilizarse para calcular el valor de la solución generada o la aportación de cada movimiento realizado. Además, puede aprovecharse para manejar la viabilidad de las soluciones de tal forma que los agentes puedan construir soluciones factibles, utilizando sólo el conocimiento sobre el estado local y los efectos de las acciones que se pueden realizar en ese estado.

Los rastros de feromonas depositados por todos los agentes se guardan en una memoria local compartida, la cual influye en las decisiones de los agentes. Cada agente deposita una cantidad constante de feromonas mientras construye una solución, después de haber construido dicha solución vuelve a recorrer todos los estados visitados reforzando el rastro de feromonas. Cuantos más agentes escojan un estado, éste se recompensa añadiendo feromonas en él y se vuelve más atractivo para los siguientes agentes. La cantidad de feromonas depositada es proporcional a la calidad de la solución encontrada. De tal forma que si un estado ha contribuido a generar una solución de alta calidad, la cantidad de feromonas de este estado aumentará proporcionalmente a su contribución.

Los valores heurísticos y la cantidad de feromonas son utilizados por la fórmula probabilística para dirigir la búsqueda hacia las áreas más atractivas del espacio de búsqueda. La evaporación de las feromonas evita que todos los agentes vayan hacia una región subóptima del espacio de búsqueda, favoreciendo así la exploración de nuevas áreas del espacio de búsqueda. Una vez que un agente haya terminado de construir una solución, se elimina del sistema.

1.3.1. Aplicaciones

En esta sección se muestran algunas aplicaciones de la ACO según el libro *Ant Colony Optimization* de Dorigo y Stützle [6]:

Tipo de problema	Nombre del problema
	Agente Viajero
Enrutamiento	Rutas de Vehículos
	Ordenamiento secuencial
Asignación	Asignación Cuadrática
	Coloración de Gráficas
Calendarización	Sistemas de producción <i>Job Shop</i>
	Sistemas de producción <i>Open Shop</i>
Subconjunto	Mochila Múltiple
	Conjunto Independiente
	Clan

Tabla 1.1: Aplicaciones de la ACO.

A continuación, se presenta una breve descripción de cada tipo de problema:

- **Problemas de enrutamiento:** consisten en encontrar o construir caminos de costo mínimo en una red determinada.
- **Problemas de asignación:** consisten en asignar un conjunto de elementos a un número determinado de recursos, minimizando el número o costo de las asignaciones realizadas.
- **Problemas de calendarización:** consisten en asignar máquinas y recursos a las tareas con el fin de completar cada una de ellas bajo ciertas restricciones impuestas.
- **Problemas de subconjunto:** consisten en encontrar un subconjunto del conjunto de elementos disponibles sujetos a restricciones específicas del problema.

1.4. Similitudes y diferencias entre las hormigas reales y artificiales

En esta sección se describen las similitudes y diferencias entre las hormigas artificiales y las hormigas reales, las cuales son:

Similitudes

- Son agentes independientes que cooperan entre ellos para encontrar buenas soluciones a cualquier problema.
- Encuentran la ruta más corta que conecta los lugares de origen con los de destino.
- Se mueven al azar por los estados adyacentes y depositan una cantidad constante de feromonas.
- Utilizan los rastros de feromonas para comunicarse entre ellas.

Diferencias

- Las hormigas artificiales viven en un mundo discreto y sus movimientos consisten en transiciones entre estados discretos.
- Cada hormiga, después de haber construido una ruta, vuelve a recorrer todos los estados visitados agregando una cantidad de feromonas proporcional a la calidad de la solución encontrada.
- Las hormigas artificiales tienen memoria de sus acciones pasadas y conocen la distancia a otros lugares.

Las similitudes entre ambos tipos de hormigas se debe a que las hormigas artificiales simulan el mismo comportamiento de las hormigas reales para encontrar buenas soluciones a los problemas que se les presentan, mientras que las diferencias se deben a que las hormigas artificiales se pueden enriquecer con habilidades adicionales de tal forma que aumente la calidad de las soluciones encontradas.

Capítulo 2

Problema de Rutas de Vehículos

El **Problema de Rutas de Vehículos**, VRP^4 , consiste en encontrar el conjunto óptimo o cercano al óptimo de rutas para una flota, de tal forma que satisfagan las demandas de un conjunto de clientes dispersos geográficamente alrededor de un depósito, minimizando los costos asociados a recorrerlas de tal forma que se maximice el servicio al cliente.

Característica del VRP:

- Se conocen las demandas de todos los clientes y las distancias entre cada par de clientes. También se conoce la distancia entre cada cliente y el depósito.
- Cada vehículo tiene una capacidad de carga y únicamente se utiliza en una ruta.
- Todas las rutas deben de comenzar y terminar en el depósito.
- La demanda total atendida por cada vehículo no puede exceder su capacidad de carga.
- Si las distancias son simétricas, es decir, si la distancia de viajar entre dos lugares no depende de la dirección del viaje, tenemos un VRP simétrico, de lo contrario tenemos un VRP asimétrico.

Los primeros autores en publicar un modelo del VRP fueron Dantzig y Ramser en 1959 [7], aunque ellos trataban de solucionar el problema de surtir combustible a una gran cantidad de estaciones de servicio desde un depósito, minimizando la distancia recorrida por los camiones que transportan el combustible. El VRP es una generalización del **Problema del Agente Viajero**, TSP^5 , debido a la necesidad de realizar varias rutas ajenas con diferentes distancias. Por ejemplo, crear diferentes recorridos en una agencia de viajes para visitar los destinos más populares de una ciudad o país.

La relevancia de este problema radica en su importancia económica para las empresas, en la logística y la distribución [8], ya que nos indica como optimizar la manera de transportar productos, personas y/o servicios que requieren llevarse a varios lugares

⁴Siglas en inglés de *Vehicle Routing Problem*, este problema también se traduce como *Problema de Ruteo de Vehículos*. Se utilizará el acrónimo VRP para referirse a este problema.

⁵Siglas en inglés de *Travelling salesman problem*.

que se encuentran dispersos geográficamente, de modo el costo asociado sea el mínimo posible. Algunas aplicaciones del VRP podrían ser: la distribución de mercancía, la recolección de residuos, las visitas domiciliarias de un médico, la entrega de mensajería y paquetería, la distribución de gasolina, la planificación de las rutas de los sistemas de transporte público y los sistemas de navegación GPS.

El VRP es un problema de optimización combinatoria, ya que el proceso de selección de las rutas de los vehículos permite seleccionar cualquier combinación de clientes al determinar la ruta de entrega de cada vehículo, por lo que el número de soluciones factibles para el problema incrementa exponencialmente con el aumento del número de clientes a los que hay que prestar servicio [9]. Por lo tanto, encontrar una solución óptima para el VRP es NP-Difícil [10, 11]. De manera que el uso de heurísticas en la búsqueda de soluciones se considera un enfoque razonable, sobre todo para ejemplares muy grandes del VRP.

Se han realizado una gran cantidad de investigaciones sobre el VRP, incluyendo enfoques metaheurísticos avanzados como los algoritmos genéticos [12, 13], la búsqueda tabú [14, 15] y el recocido simulado [15]. Una cantidad limitada de investigaciones ha utilizado el enfoque de optimización por colonias de hormigas y estrategias de mejora (listas de candidatos y heurística 2-opt).

2.1. Variantes del VRP

En esta sección se describen algunas de las variantes del VRP más importantes según el estudio de De Jaegere et al. [16].

VRP Clásico

El **VRP Clásico**, $CVRP^6$, es la variante más común y básica del VRP. El CVRP consiste en encontrar una serie de m rutas con un costo mínimo, este conjunto de rutas debe satisfacer las demandas de n clientes que están dispersos geográficamente.

Las restricciones del CVRP son:

- La flota es homogénea y tiene m vehículos, todos los vehículos tienen la misma capacidad de carga Q .
- Cada vehículo debe comenzar y terminar su ruta en el depósito central.
- La demanda total atendida por cada vehículo no puede exceder Q .
- Cada cliente es visitado una sola vez por un vehículo y tiene una demanda menor o igual que Q .

En la figura 2.1 se muestra la representación gráfica de un ejemplar del CVRP donde el vértice naranja es el depósito y los demás vértices representan a los clientes, los cuales

⁶Siglas en inglés de *Capacitated Vehicle Routing Problem*, este problema generalmente se traduce como *VRP Clásico*. Se utilizará el acrónimo CVRP para referirse a este problema.

tienen una demanda y el color de la ruta a la que pertenecen. Cada vehículo tiene una capacidad de carga $Q = 12$ y atienden una demanda total menor o igual que Q .

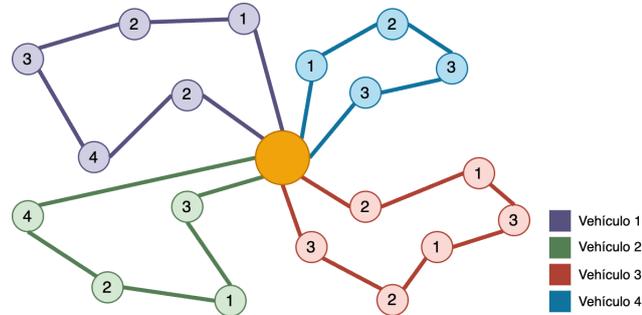


Figura 2.1: Representación gráfica de un ejemplar del CVRP.

VRP con Flota Heterogénea

El **VRP con Flota Heterogénea**, *HFVRP*⁷, variante del CVRP que hace uso de una flota heterogénea de m vehículos que tienen diferentes características como la capacidad de carga, la velocidad y el tipo de combustible que utiliza. Este problema consiste en determinar qué vehículo es más conveniente utilizar en cada ruta, de tal forma que se minimicen los costos asociados a recorrerla. Esta variante también es llamada **VRP con Flota Mixta**.

Las restricciones adicionales de este problema son:

- La flota es heterogénea y tiene m vehículos, los cuales poseen diferentes características, especialmente cada vehículo cuenta con una capacidad de carga Q_i distinta.
- Cada cliente es visitado una sola vez por un vehículo y tiene una demanda menor o igual a la capacidad de carga de alguno de los vehículos.

En la figura 2.2 se ilustra la representación gráfica de un ejemplar del HFVRP donde el vértice naranja es el depósito y los demás vértices representan a los clientes, los cuales tienen una demanda y el color de la ruta a la que pertenecen. El vehículo 1 atiende una demanda de 16, el vehículo 2 satisface una demanda de 24 y el vehículo 3 atiende una demanda de 8.

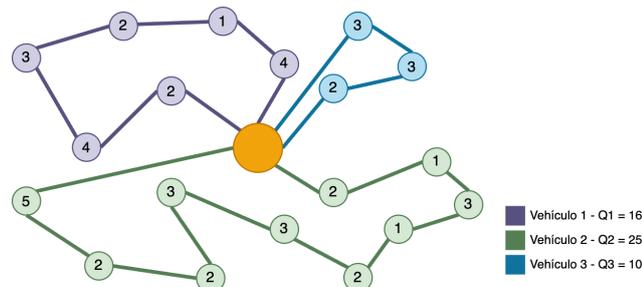


Figura 2.2: Representación gráfica de un ejemplar del HFVRP.

⁷Siglas en inglés de *Heterogeneous Fleet Vehicle Routing Problem*.

VRP con Múltiples Depósitos

El **VRP con Múltiples Depósitos**, *MDVRP*⁸, variante del CVRP que se caracteriza por tener más de un depósito en la misma área geográfica que los clientes, por lo que la distancia entre un cliente y cada depósito varía. Este problema consiste en asignar a los clientes al depósito más cercano y encontrar las rutas con un costo mínimo que satisfagan las demandas de los clientes asignados a cada depósito.

Las restricciones adicionales de este problema son:

- A cada depósito se le asigna una flota de k vehículos, $k < m$.
- Cada cliente es asignado al depósito más cercano y es visitado una sola vez por un vehículo que pertenece a ese depósito.

En la figura 2.3 se muestra la representación gráfica de un ejemplar del MDVRP donde los vértices naranjas son los depósitos y los demás vértices representan a los clientes, los cuales tienen una demanda y el color de la ruta a la que pertenecen. Todos los vehículos tienen una capacidad de carga $Q = 10$ y atienden una demanda total menor o igual que Q .

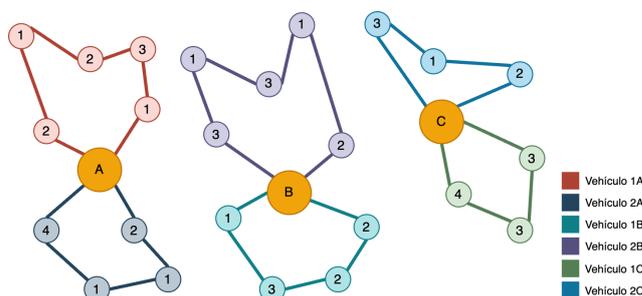


Figura 2.3: Representación gráfica de un ejemplar del MDVRP.

VRP con Recolección y Entrega

El **VRP con Recolección y Entrega**, *VRPPD*⁹, variante del CVRP en la que algunos clientes de la ruta son para recolección y otros son para entrega. Este problema consiste en generar un conjunto de rutas con un costo mínimo, de tal forma que se visiten primero al cliente de recolección antes que al cliente de entrega correspondiente.

Las restricciones adicionales de este problema son:

- Cada cliente tiene asociadas dos cantidades, la primera hace referencia a la demanda que debe ser entregada y la segunda representa la demanda que debe ser recolectada.
- Todos los vehículos deben visitar primero al cliente de recolección antes que al cliente de entrega correspondiente.

⁸Siglas en inglés de *Multi Depot Vehicle Routing Problem*.

⁹Siglas en inglés de *Vehicle Routing Problem with Pickup and Delivery*.

- La carga final de cada vehículo va a ser igual a la carga inicial menos todas las demandas entregadas más todas las demandas recogidas, esta carga final no puede superar Q .

En la figura 2.4 se ilustra la representación gráfica de un ejemplar del VRPPD donde el vértice naranja es el depósito, los vértices de color verde son los clientes de entrega y los vértices morados son clientes de recolección, cada cliente tiene una demanda de entrega y una demanda de recolección. Cada vehículo tiene una capacidad de carga $Q = 10$ y atienden una demanda total menor o igual que Q .

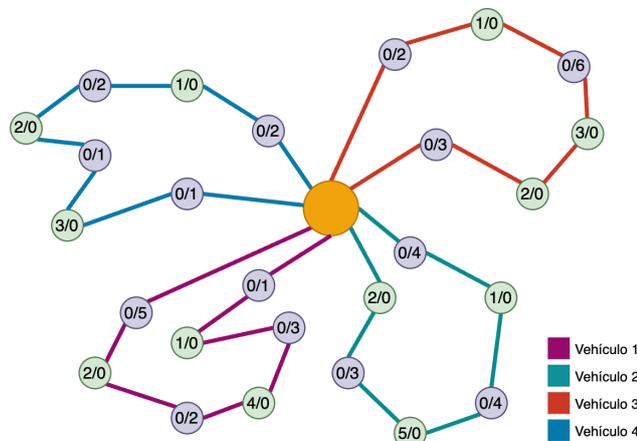


Figura 2.4: Representación gráfica de un ejemplar del VRPPD.

VRP con Ventanas de Tiempo

En el **VRP con Ventanas de Tiempo**, $VRPTW^{10}$, variante del CVRP en la que se definen en qué horarios es posible visitar a un cliente. Este problema consiste en encontrar el conjunto de rutas con un costo mínimo, satisfagan las demandas y respeten los horarios de visitas de todos clientes. En algunas soluciones se recorre una distancia más larga en un tiempo menor o una distancia más corta en un tiempo mayor.

Las restricciones adicionales de este problema son:

- Todos los vehículos cuentan con un tiempo de viaje, el cual abarca desde que el vehículo sale del depósito hasta que visita a todos los clientes de la ruta y regresa al depósito.
- Cada cliente tiene asociado un intervalo de tiempo $[a_i, b_i]$ en el que debe ser atendido y se denomina ventana de tiempo.
- El tiempo de servicio para cada cliente es el tiempo durante el vehículo se detiene para atender al cliente.

¹⁰Siglas en inglés de *Vehicle Routing Problem with Time Windows*.

2. PROBLEMA DE RUTAS DE VEHÍCULOS

Las ventanas de tiempo pueden ser estrictas, si no es posible atender al cliente fuera del intervalo de tiempo; o flexibles, si es posible atender al cliente fuera de los límites de tiempo a cambio de un costo o penalización. Además, todas las variantes anteriores se pueden plantear con ventanas de tiempo.

En la figura 2.5 se muestra la representación gráfica de un ejemplar del VRPTW donde el vértice naranja es el depósito y los demás vértices representan a los clientes, el tono de azul indica la ventana de tiempo del cliente. Todos los vehículos tienen una capacidad de carga $Q = 12$ y atienden una demanda total menor o igual que Q .

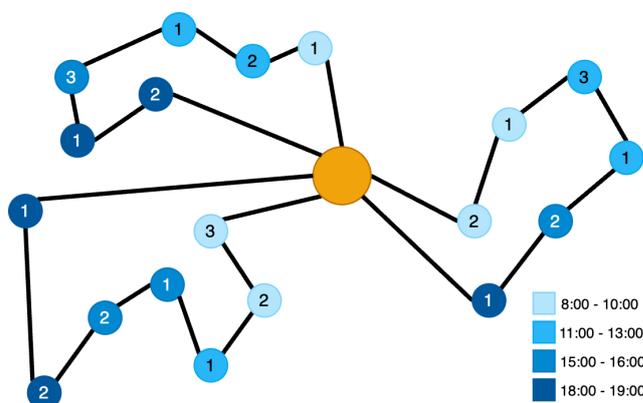


Figura 2.5: Representación gráfica de un ejemplar del VRPTW.

2.2. Definición formal del CVRP

En esta sección se presenta una definición formal del CVRP, los conceptos utilizados de Teoría de Gráficas y de Complejidad Computacional se encuentran en el Apéndice A y en el Apéndice B, respectivamente.

Sea $G = (V, E)$ una gráfica completa, ponderada y no dirigida, con un atributo demanda en cada vértice, donde:

- m es el número de vehículos.
- Q es la capacidad de carga de los vehículos.
- $V = \{v_0, v_1, v_2, \dots, v_n\}$ es el conjunto de n clientes y un depósito v_0 , con $n \in \mathbb{N}$.
- $E = \{(v_i, v_j) \mid i \neq j, \forall i, j, \text{ con } i, j = 0, 1, \dots, n\}$ es el conjunto de aristas que representan los tramos entre los clientes.
- $d : E \rightarrow \mathbb{Z}$ es la función de distancias, donde $d(v_i, v_j)$ es la distancia de ir del vértice v_i al vértice v_j con $i \neq j$ y $d(v_i, v_j) > 0$ para todo $v_i, v_j \in V$. Además, $d(v_i, v_i) = 0, \forall i, i = 0, 1, \dots, n$.
- $s : V \rightarrow \mathbb{Z}$ es la función de demandas, donde $s(v_i)$ es la demanda del cliente v_i con $1 \leq s(v_i) \leq Q, \forall i, i = 1, \dots, n$ y $s(v_0) = 0$.

Una solución del VRP es un conjunto $R = \{r_1, r_2, \dots, r_m\}$ de rutas, donde cada $r_i = [c_{i0}, c_{i1}, \dots, c_{i\ell}]$ es la ruta del vehículo $i \ \forall i, i = 1, \dots, m$, con ℓ igual a la longitud de la ruta r_i menos uno y $c_{ij} \in V, \ \forall j, j = 1, \dots, \ell$.

La distancia recorrida D_i por la ruta r_i se define como la suma de las distancias entre cada par de clientes visitados en r_i :

$$D_i = \sum_{j=1}^{\ell} d(c_{i(j-1)}, c_{ij}).$$

La demanda atendida S_i por la ruta r_i es igual a la suma de las demandas de los clientes visitados en r_i :

$$S_i = \sum_{j=1}^{\ell-1} s(c_j).$$

La distancia total recorrida D_T por la solución R se define como la suma de las distancias de todas las rutas:

$$D_T = \sum_{i=1}^m D_i.$$

La demanda total atendida S_T por la solución R se define como la suma de las demandas de todas las rutas:

$$S_T = \sum_{i=1}^m S_i.$$

La **función objetivo** $f : R^* \rightarrow \mathbb{Z}$ consiste en minimizar D_T . Donde R^* es el conjunto de todas las rutas de G . Está sujeta a las siguientes restricciones:

- Dada una solución R , cada cliente es visitado una sola vez por un único vehículo:

$$\exists r_i \in R \text{ tal que } c_{ij} \in r_i, \ \forall c_{ij} \in V - \{v_0\},$$

$$\forall c_{ij} \in V - \{v_0\}, \text{ si } c_{ij} \in r_i \text{ y } c_{ij} \in r_l \text{ entonces } r_i = r_l, \ \forall r_i, r_l \in R.$$

- Todas las rutas deben comenzar y terminar en el depósito central:

$$c_{i0} = v_0 = c_{i\ell} \ \forall r_i \in R.$$

- La demanda total atendida por cada vehículo no puede exceder Q :

$$S_i \leq Q \ \forall r_i \in R.$$

Durante el desarrollo de este trabajo se estudia el CVRP simétrico, es decir, $d(c_i, c_j) = d(c_j, c_i), \ \forall i, j$ con $i, j = 1, \dots, n$.

2.3. Ejemplo del CVRP

Consideremos la gráfica de la figura 2.6, los vértices representan al depósito v_0 y a los clientes desde v_1 hasta v_5 , todos los clientes tienen una demanda s y cada arista indica la distancia d asociada al par de vértices adyacentes. El número de clientes es $n = 5$ y la capacidad de carga Q es igual a 7. La demanda total de este ejemplar es 13 y como la capacidad de carga de los vehículos es igual a 7 entonces sólo se necesitan $m = 2$ vehículos para satisfacer las demandas de todos los clientes. A continuación se muestran dos posibles soluciones a este ejemplar:

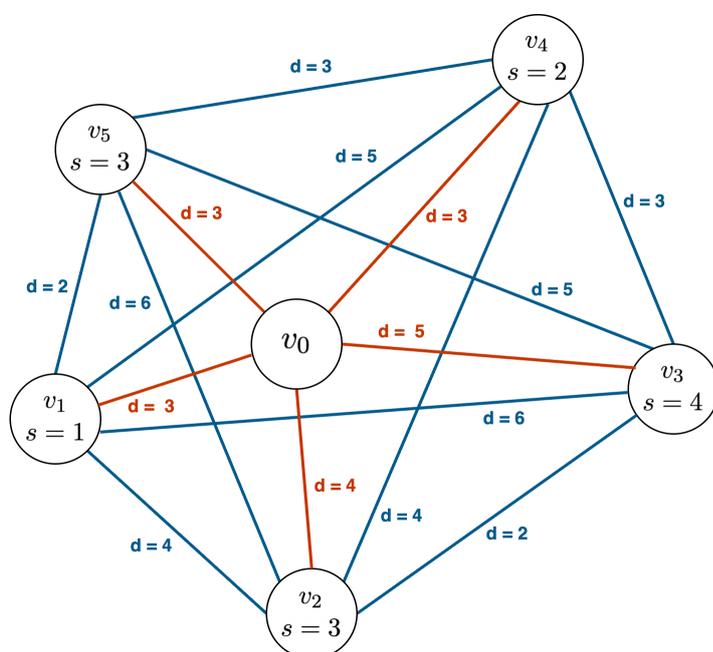


Figura 2.6: Ejemplar del CVRP.

En la primera solución que se muestra en la figura 2.7, el vehículo 1 satisface la mayor demanda posible (ruta marcada de morado) y el segundo vehículo atiende a los clientes que faltaron (ruta marcada de verde). Las rutas de los vehículos son:

Vehículo	Ruta	Demanda	Distancia
r_1	$[v_0, v_5, v_3, v_0]$	7	13
r_2	$[v_0, v_1, v_4, v_2, v_0]$	6	16

Tabla 2.1: Rutas de la solución R para el ejemplar del CVRP.

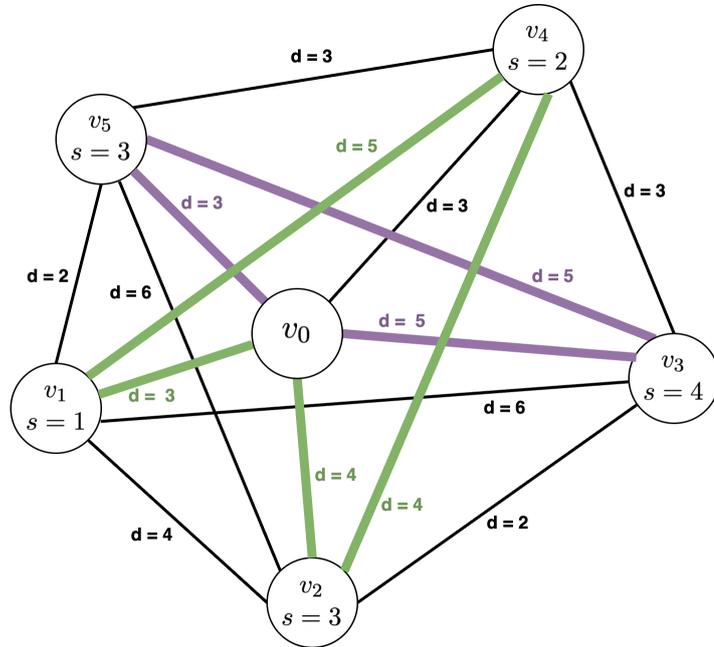


Figura 2.7: Solución R para el ejemplar del CVRP.

Otra posible solución se muestra en la figura 2.8, el vehículo 1 satisface la mayor demanda posible (ruta marcada de verde) y el segundo vehículo atiende a los clientes que faltaron (ruta marcada de morado). Las rutas de los vehículos son:

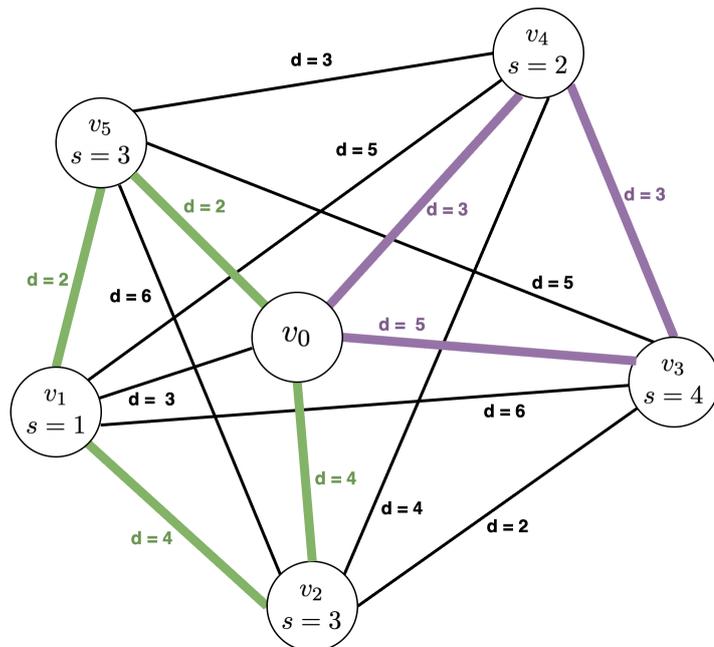


Figura 2.8: Solución R' para el ejemplar del CVRP.

2. PROBLEMA DE RUTAS DE VEHÍCULOS

Vehículo	Ruta	Demanda	Distancia
r'_1	$[v_0, v_5, v_1, v_2, v_0]$	7	12
r'_2	$[v_0, v_4, v_3, v_0]$	6	11

Tabla 2.2: Rutas de la solución R' para el ejemplar del CVRP.

Las soluciones R y R' cumplen con las restricciones del CVRP, la solución R recorre una distancia total de 29 mientras que la solución R' viaja una distancia total de 23. Por lo que la solución R' es una mejor solución que R , ya que recorre una menor distancia al satisfacer las demandas de todos los clientes.

Capítulo 3

La metaheurística ACO para el CVRP

En este capítulo se expone como utilizar la metaheurística ACO para atacar el CVRP. Para ello vamos a considerar a cada hormiga como un vehículo y el nido como el depósito. El material expuesto en este capítulo, está basado en el artículo *Ant colony optimization techniques for the vehicle routing problem* de Bell y McMullen [4].

3.1. Metaheurística ACO para el CVRP

El esquema general de la metaheurística **ACO** se muestra en el pseudocódigo 1. A continuación se da una breve descripción de los parámetros y funciones complementarias utilizadas por la metaheurística.

- n es el número de clientes.
- m es el número de hormigas en la colonia.
- Q es la capacidad de carga de las hormigas.
- *depot* es el depósito en el que las hormigas deben iniciar y terminar sus rutas.
- *clients* es el arreglo de tamaño n que guarda los clientes. El valor de *clients*[j] es el vértice v_j de G , para cualquier $j = 1, \dots, n$.
- *beta* es la importancia de la distancia en comparación con la cantidad de feromonas.
- q_0 es la probabilidad de seleccionar al mejor cliente posible a visitar.
- *alpha* es la velocidad de evaporación del rastro de feromonas.
- *tau0* es la cantidad inicial del rastro de feromonas en todas las aristas de G .
- *step* es el número de pasos de la construcción.
- *stoppingCriterion* es el criterio de término de la ejecución de la metaheurística.

3. LA METAHEURÍSTICA ACO PARA EL CVRP

Pseudocódigo 1 Metaheurística ACO

Entrada: Un ejemplar I del CVRP; es decir, una gráfica $G = (V, E)$ completa, ponderada y no dirigida con $V = \{v_0, \dots, v_n\}$ y $E = \{(v_0, v_1), \dots, (v_{n-1}, v_n)\}$, un arreglo S de demandas, una matriz D de distancias, un entero $m > 0$ y un entero $Q > 0$.

Salida: Un arreglo bidimensional de tamaño $m \times n$ que contiene las rutas de una solución factible para I muy cercana a cualquier solución óptima de I .

```
1: /* Inicializar datos */
2: depot ← V[0]
3: clients ← V - {v0}
4: n ← |V| - 1
5: beta, q0, alpha, tau0 ← InicializarConfiguracion()
6: stoppingCriterion ← false
7: bestRoutes ← ∅
8: bestSolution ← ∅
9: visited ← InicializarVisitas(n)
10: pheromones ← InicializarRastro(n, tau0)
11: ants ← InicializarColonia(m, Q)
12: /* Construcción de las rutas */
13: while (stoppingCriterion != true) do
14:   ReiniciarHormigas(ants, Q, depot)
15:   step ← 0
16:   while (step < n) do
17:     step ← step + 1
18:     for i = 1 to m do
19:       if (ants[i].finish != true) then
20:         SeleccionarCliente(ant[i], step - 1, beta, q0, pheromones, visited, D)
21:         k ← ants[i].memory[step - 1]
22:         j ← ants[i].memory[step]
23:         unvisited ← VerificarNoVisitados(ants[i], visited, S)
24:         /* Actualización local de feromonas */
25:         ActualizarFL(pheromones, alpha, tau0, k, j)
26:         if (ants[i].Qi == 0 or unvisited != true) then
27:           ants[i].memory[step + 1] ← depot
28:           ants[i].rDistance ← CalcularDistanciaRuta(ants[i].memory, D)
29:           ants[i].finish ← true
30:         end if
31:       end if
32:     end for
33:   end while
34:   bestRoutes ← ObtenerMejoresRutas(ants)
35:   bestSolution ← AgregarMejorRuta(bestRoutes)
36:   MarcarClientesVisitados(visited, bestSolution)
37:   /* Actualización global de feromonas */
38:   ActualizarFG(pheromones, alpha, bestRoutes)
39: end while
40: return bestSolution
```

- *bestRoutes* es la matriz de tamaño $m \times n$ que almacena las mejores rutas encontradas hasta el momento.
- *bestSolution* es la matriz de tamaño $m \times n$ que guarda las rutas de la mejor solución factible construida por la colonia de hormigas.
- *visited* es un arreglo de tamaño n que marca a los clientes visitados. El valor de *visited*[j] es *true* si el cliente ha sido visitado y *false* en otro caso, para cualquier $j = 1, \dots, n$.
- *pheromones* es la matriz de tamaño $n \times n$ que almacena el rastro de feromonas en todas las aristas. El valor de *pheromones*[k][j] es la cantidad de feromonas en la arista (v_k, v_j) de G , para cualesquiera $k, j = 0, \dots, n$.
- *unvisited* es la bandera que indica si hay clientes que no han sido visitados y tienen una demanda menor o igual que la capacidad de carga actual de la hormiga *Ant*[i], para cualquier $i = 1, \dots, m$.
- S es el arreglo de tamaño n que guarda las demandas de los clientes. El valor de S [j] es la demanda del cliente v_j de G , para cualquier $j = 1, \dots, n$.
- D es la matriz de tamaño $n \times n$ que almacena las distancias entre cada par de clientes y entre cada cliente y el depósito. El valor de D [k][j] es la distancia entre los vértices v_k y v_j de G , para cualesquiera $k, j = 0, \dots, n$.
- *ants* es el arreglo de tamaño m que guarda las hormigas de la colonia. Para cualquier $i = 1, \dots, m$, el valor de *ants*[i] tiene los siguientes parámetros:
 - *memory* es el arreglo de tamaño $k < n$ que guarda la ruta construida, la cual empieza y termina en el depósito de G .
 - Q_i es la capacidad de carga actual.
 - *rDistance* es la distancia que recorre la ruta que ha construido.
 - *finish* es una bandera que indica si ya terminó de construir su ruta.

La estructura de las hormigas se muestra en el pseudocódigo 2.

Pseudocódigo 2 Hormiga

Entrada: Un entero positivo $Q \geq 0$.

Salida: Una hormiga de la colonia.

- 1: *memory* $\leftarrow []$
 - 2: $Q_i \leftarrow Q$
 - 3: *rDistance* $\leftarrow 0$
 - 4: *finish* $\leftarrow false$
-

- **InicializarConfiguracion()**: establece el valor más adecuado para *beta*, *q0*, *alpha* y *tau0* según la definición particular de I .
-

3. LA METAHEURÍSTICA ACO PARA EL CVRP

- **InicializarVisitas(n)**: regresa el arreglo *visited* de tamaño n con todos los elementos establecidos como *false*, es decir, marca a todos los clientes como no visitados.
- **InicializarRastro(n, τ_0)**: regresa la matriz *pheromones* de tamaño $n \times n$, cada celda de la matriz representa una arista de G y contiene la cantidad inicial τ_0 del rastro de feromonas.
- **InicializarColonia(m, Q)**: regresa el arreglo *ants* de tamaño m , el cual contiene las hormigas de la colonia con capacidad de carga Q .
- **ReiniciarHormigas($ants, Q, depot$)**: reinicia la memoria de todas las hormigas en el arreglo *ants*, para ello primero la vacía y luego le agrega el depósito para inicializar la ruta. Además, restablece la capacidad de carga actual igual a Q .
- **SeleccionarCliente($ant[i], k, \beta, q_0, pheromones, visited, D$)**: regresa el siguiente cliente j a visitar aplicando una estrategia de selección basada en probabilidades. La fórmula 3.1 describe más detalladamente la estrategia de selección utilizada para elegir al próximo cliente a visitar.
- **VerificarNoVisitados($ants[i], visited, S$)**: regresa *true* si hay clientes que no han sido visitados por ninguna hormiga (marcados con *false* en *visited* y que no estén en la memoria de $Ant[i]$) y tienen una demanda en S menor o igual que la capacidad de carga actual de la hormiga $Ant[i]$.
- **ActualizarFL($pheromones, \alpha, \tau_0, k, j$)**: realiza la actualización local del rastro de feromonas de la arista (v_k, v_j) . La fórmula 3.3 describe más detalladamente como se realiza la actualización local del rastro de feromonas.
- **CalcularDistanciaRuta($ants[i].memory, D$)**: regresa la distancia recorrida por la ruta $ants[i].memory$ utilizando la matriz de distancias D .
- **ObtenerMejoresRutas($ants$)**: regresa las mejores rutas construidas hasta el momento por algunas de las m hormigas de la colonia, estas rutas recorren la menor distancia posible y al mismo tiempo atienden la mayor demanda posible.
- **AgregarMejorRuta($bestRoutes$)**: agrega la mejor ruta en *bestRoutes* al arreglo bidimensional *bestSolution*.
- **MarcarClientesVisitados($visited, bestSolution$)**: establece como *true* a los clientes visitados en las rutas de *bestSolution* en el arreglo *visited*.
- **ActualizarFG($pheromones, \alpha, bestRoutes$)**: realiza la actualización global del rastro de feromonas de las aristas incluidas en cada una de las rutas de *bestRoutes*. La fórmula 3.4 describe más detalladamente como se realiza la actualización global del rastro de feromonas.

3.2. Construcción de las rutas

Inicialmente, cada hormiga se encuentra en el depósito, por lo que su capacidad de carga actual es igual a Q y su memoria sólo contiene al depósito. Posteriormente, la hormiga i selecciona de manera aleatoria el primer cliente j a visitar de los clientes que no han sido marcados como visitados, lo agrega a su memoria y actualiza su capacidad de carga. Si la capacidad de carga de la hormiga i es igual a 0, regresa al depósito, lo agrega a su memoria y calcula la distancia que recorre la ruta que ha construido. En caso contrario, la hormiga i elige el siguiente cliente j a visitar a partir del último cliente k que ha visitado de acuerdo con la siguiente estrategia de selección basada en probabilidades [17]:

$$j = \begin{cases} \operatorname{argmax}\{(\tau_{ku})(\eta_{ku})^\beta\} \text{ para } u \notin M_i \text{ y } u \in C_i & \text{si } q \leq q_0, \\ p_{kj} & \text{en otro caso.} \end{cases} \quad (3.1)$$

Donde:

- τ_{ku} es la cantidad de feromonas entre el cliente k y el cliente u .
- η_{ku} es la inversa de la distancia entre el cliente k y el cliente u .
- β es el exponente positivo que establece la importancia de la distancia en comparación con la cantidad de feromonas.
- M_i es la memoria de la hormiga i , en la cual se almacena la ruta construye.
- C_i son los clientes que no han sido visitados con una demanda menor o igual que la capacidad de carga actual de la hormiga i .
- q es una variable aleatoria continua en $[0, 1]$ que representa la probabilidad de seleccionar un cliente j que se encuentra cerca del cliente k .
- q_0 es un parámetro que representa la probabilidad de seleccionar al mejor cliente j posible, con $0 \leq q_0 \leq 1$.
- p_{jk} es la probabilidad con la que elige moverse del cliente j al cliente k .

La hormiga i selecciona al siguiente cliente j con el valor más alto en la ecuación 3.1; excepto si $q > q_0$. En este caso, elige al siguiente cliente a visitar en función de la distribución de probabilidad de p_{ij} , que favorece a los clientes más cercanos con altos niveles de feromonas:

$$p_{kj} = \frac{(\tau_{kj})(\eta_{jk})^\beta}{\sum_{u \notin M_i} (\tau_{ku})(\eta_{ku})^\beta} \text{ si } j \notin M_i \text{ y } j, u \in C_i; \text{ en otro caso } 0. \quad (3.2)$$

La hormiga i selecciona al siguiente cliente j si p_{ij} es diferente de 0, lo agrega a su memoria y actualiza su capacidad de carga. Si la capacidad de carga de la hormiga i es

igual a 0, regresa al depósito, lo agrega a su memoria y calcula la distancia que recorre la ruta que ha construido. En caso contrario, la hormiga i utiliza la fórmula 3.1 para seleccionar al próximo cliente que va a visitar. Si p_{ij} es igual a 0 entonces la hormiga i regresa al depósito, lo agrega a su memoria, calcula la distancia que recorre la ruta que ha construido y establece que ha terminado de construir su ruta. Este proceso continúa hasta que todos los clientes hayan sido visitados.

3.3. Actualización del rastro feromonas

Los rastros de feromonas deben actualizarse para reflejar el desempeño y la calidad de las soluciones encontradas, a fin de mejorar las soluciones futuras. Esta actualización consta de la actualización local y global de los rastros de feromonas.

La actualización local es aplicada después de que una hormiga haya seleccionado al siguiente cliente j a visitar a partir del cliente k durante la construcción de su ruta, esta actualización reduce el rastro de feromonas en la arista (v_k, v_j) de G de tal forma que la arista se vuelve menos deseable para las siguientes hormigas y así se garantiza que ninguna arista se vuelve demasiado dominante. Para ello, se utiliza la siguiente fórmula:

$$\tau_{jk} = (1 - \alpha)\tau_{jk} + (\alpha)\tau_0. \quad (3.3)$$

Donde:

- α es la velocidad de evaporación del rastro de feromonas.
- τ_{jk} es el rastro de feromonas en la arista (v_j, v_k) de G .
- τ_0 es la cantidad inicial del rastro de feromonas asignado a todas las aristas de G .

La actualización global es aplicada después de que l hormigas hayan construido las mejores rutas hasta el momento, esta actualización agrega feromonas a todas las aristas incluidas en cada una de estas rutas respectivamente. La actualización global se realiza de acuerdo con la siguiente ecuación:

$$\tau_{jk} = (1 - \alpha)\tau_{jk} + \alpha(L_i)^{-1}. \quad (3.4)$$

Donde L_i es la distancia de la mejor ruta encontrada por la hormiga i correspondiente. Esta actualización fomenta el uso de rutas más cortas y aumenta la probabilidad de que las rutas futuras utilicen las aristas contenidas en las rutas de la mejor solución hasta el momento.

3.4. Múltiples colonias de hormigas

Se ha propuesto en investigaciones pasadas [17] el uso de grupos especializados de hormigas, debido a que las hormigas en la naturaleza tienen diferentes trabajos como

construir el nido, buscar alimento, defender la colonia y nutrir a las larvas. Este método de optimización es eficaz para encontrar soluciones de alta calidad para ejemplares del CVRP de gran tamaño.

En este método se utiliza una colonia de hormigas para cada vehículo, por lo que todos los vehículos tienen un rastro de feromonas único. De esta forma se diferencian las rutas que utiliza el vehículo i de las rutas que utiliza el vehículo j con el fin de que las hormigas de la colonia i no se distraigan con los rastros de feromonas de las hormigas de la colonia j , $\forall i, j, i, j = 1, \dots, m$ y $i \neq j$.

Inicialmente, las hormigas de cada colonia se encuentran en el depósito con una carga actual igual a Q y su memoria sólo contiene al depósito. Posteriormente, la hormiga i selecciona de manera aleatoria el primer cliente j a visitar y si este no ha sido visitado por alguna hormiga de otra colonia, lo agrega a su memoria, lo marca como visitado y actualiza su capacidad de carga. En caso contrario, la hormiga i elige de forma aleatoria un cliente j de los clientes que no han sido visitados por ninguna de las hormigas de otra colonia o bien escoge un cliente j visitado por alguna de las hormigas de su colonia, marca el cliente j como visitado, lo agrega a su memoria y actualiza su capacidad de carga. Si la capacidad de carga de la hormiga i es igual a 0, regresa al depósito, lo agrega a su memoria y calcula la distancia que recorre la ruta que ha construido. En otro caso, la hormiga i elige el siguiente cliente j utilizando la fórmula 3.1 de los clientes que no han sido visitados por ninguna de las hormigas de otra colonia o bien selecciona un cliente j visitado por alguna de las hormigas de su colonia con una demanda menor o igual que su capacidad de carga que favorezca su ruta. Agrega el cliente j a su memoria, lo marca como visitado y actualiza su capacidad de carga. Este proceso continúa hasta que todos los clientes hayan sido visitados. Del mismo modo, los rastros de feromonas son actualizados usando las ecuaciones 3.3 y 3.4.

Este método limita la comunicación entre las hormigas de diferentes colonias, pero aumenta la probabilidad de que un vehículo siga utilizando las rutas que ha encontrado eficaces y no se distraiga con los rastros de feromonas de las hormigas asignadas a otros vehículos, de tal forma que se hallan simultáneamente múltiples rutas eficaces.

3.5. Estrategias de mejora

En esta sección se describen brevemente las dos estrategias de mejora utilizadas para aumentar la calidad de las rutas construidas por las hormigas.

3.5.1. Heurística 2-opt

Esta estrategia consiste en utilizar la heurística 2-opt¹¹ para tratar de mejorar las soluciones construidas por las hormigas. Recibe una solución $R = \{r_1, r_2, \dots, r_m\}$, $\forall i, i = 1, \dots, m$, para cada ruta r_i se va a probar todos los posibles intercambios 2-opt, los cuales generan una nueva ruta r'_i . Si r'_i recorre una distancia menor que r_i se actualiza la ruta r_i con r'_i y se vuelven a probar todos los posibles intercambios 2-opt. En caso contrario

¹¹En el apéndice C se describe brevemente la heurística 2-opt.

se descarta r'_i y se trata de buscar un posible intercambio 2-opt que mejore a r_i . Cuando la ruta r_i ya no se pueda mejorar, se denomina 2-óptima.

El número de intercambios 2-opt que se exploran en la búsqueda aumenta en gran medida conforme el tamaño de las rutas de entrada incrementa, así que se puede considerar como la solución de varios TSP [18].

3.5.2. Listas de candidatos

Esta estrategia consiste en utilizar listas de candidatos para restringir el número de clientes que se pueden considerar en cada paso de la construcción de la ruta de cada hormiga. A cada cliente v_j se le asigna una lista de candidatos LC_j de longitud ℓ :

$$LC_j = [a_{j1}, \dots, a_{j\ell}] \text{ tal que } d(v_j, a_{jk}) \geq d_{min}, \forall k, k = 1, \dots, \ell$$

Donde:

- $a_{jk} \in V - \{v_0\}$ es el cliente k -ésimo que se encuentra a una distancia pequeña del cliente v_j .
- $d_{min} = \arg \min \{d(v_j, a_{jk})\}$ es la distancia entre el cliente v_j y el cliente a_{jk} más cercano.

Inicialmente, cada hormiga se encuentra en el depósito, por lo que su capacidad de carga actual es igual a Q y su memoria sólo contiene al depósito. Posteriormente, la hormiga i selecciona de manera aleatoria el primer cliente l a visitar de los clientes que no han sido marcados como visitados de la lista de candidatos del depósito y en caso de que todos los posibles candidatos ya han sido marcados como visitados, elige aleatoriamente un cliente l que no lo haya sido, lo agrega a su memoria y actualiza su capacidad de carga. Si la capacidad de carga de la hormiga i es igual a 0, regresa al depósito, lo agrega a su memoria y calcula la distancia que recorre la ruta que ha construido. En caso contrario, la hormiga i elige el siguiente cliente l a visitar de los clientes que aún no han sido marcados como visitados de la lista de candidatos de j y en caso de que todos los posibles candidatos ya han sido visitados, escoge aleatoriamente un cliente l que no lo haya sido y cumpla con la restricción de carga de la hormiga, lo agrega a su memoria y actualiza su capacidad de carga. Si la capacidad de carga de la hormiga i es igual a 0, regresa al depósito, lo agrega a su memoria y calcula la distancia que recorre la ruta que ha construido. En caso contrario, la hormiga elige el próximo cliente utilizando la lista de candidatos del cliente l . Este proceso continúa hasta que todos los clientes hayan sido visitados.

La longitud de las listas debe de ser igual a una fracción del número de clientes n . Según Bullnheimer la longitud de la lista de candidatos debe ser igual a una cuarta parte del número de clientes $n/4$, independientemente del número de clientes [19]. Esta restricción evita que el algoritmo desperdicie recursos considerando movimientos hacia clientes que se encuentran a una gran distancia del cliente actual y tienen muy pocas posibilidades de crear una mejor solución para el ejemplar. Por lo que en este trabajo se explora el impacto de utilizar dos diferentes tamaños de listas de candidatos y, al mismo tiempo, explorar el efecto que tiene esta selección en diferentes tamaños de ejemplares.

Capítulo 4

Implementación del programa

En este capítulo se describen los detalles de la implementación del programa y las instrucciones de ejecución del mismo.

4.1. Detalles de la implementación

El programa fue implementado en el lenguaje de programación **Java 17.0.5** y el IDE **Eclipse 2022-09 (4.25.0)**, también se utilizó **MySQL 8.0.31** y **Apache Maven 4.0.0**. El nombre y estado de cada ciudad se almacenó en una base de datos, así como la distancia entre cada par de clientes para facilitar el acceso a los datos. La cantidad de feromonas entre cada par de clientes se modeló en una matriz simétrica.

4.2. Implementación

En esta sección se describe brevemente cada una de las clases del programa¹² y se presenta un esquema general de los métodos más importantes.

- **Solution:**

Esta es la clase principal del programa, lee los identificadores y demandas de los clientes y el depósito del ejemplar del archivo **Orders.txt**, además obtiene los datos de la base de datos y genera los clientes correspondientes. Busca una solución factible para un ejemplar del CVRP utilizando el número del método que se le pasa como parámetro.

- **-s 1:** Método de optimización colonia de hormiga única (CHU¹³) sin estrategias de mejora.
- **-s 2:** CHU y heurística 2-opt como estrategia de mejora.

¹²La implementación de este programa se puede consultar en el siguiente repositorio de GitHub: https://github.com/Terebece/Programa_Tesis

¹³Se utilizará el acrónimo CHU para referirse al método de optimización colonia de hormigas única.

4. IMPLEMENTACIÓN DEL PROGRAMA

- `-s 3 -c n`: CHU y listas de candidatos como estrategia de mejora, donde n es la longitud de las listas de candidatos.
- `-s 4`: Método de optimización colonias de hormigas múltiples (CHM¹⁴) sin estrategias de mejora.
- `-s 5`: CHM y heurística 2-opt como estrategia de mejora.
- `-s 6 -c n`: CHM y listas de candidatos como estrategia de mejora, donde n es la longitud de las listas de candidatos.

El argumento `-s` nos permite identificar el número de la estrategia que se quiere usar y el argumento `-c` nos permite localizar la longitud de las listas de candidatos. En las secciones 3.2 y 3.4 se describe como se utilizan los métodos de optimización colonia de hormiga única y colonias de hormigas múltiples, respectivamente. En la sección 3.5 se expone como se usan las estrategias de mejora: heurística 2-opt y listas de candidatos.

- **VRP:**

Clase que utiliza la metaheurística ACO y las estrategias de mejora para construir una solución factible para un ejemplar del CVRP. Lee los parámetros de configuración del archivo `Configurations.txt`. Los principales métodos de esta clase son:

- `macoTwoOpt()`: aplica la heurística 2-opt para mejorar las rutas construidas utilizando CHM y regresa la mejor solución, la cual contiene la mejor ruta construida por cada una de las colonias.
- `acoTwoOpt(colony)`: aplica la heurística 2-opt para mejorar las rutas construidas usando el método CHU y regresa la mejor solución, la cual contiene las mejores rutas construidas por la colonia `colony` que satisfacen la demanda de todos los clientes.
- `macoVRP(lengthLC)`: Regresa la mejor solución, la cual contiene la mejor ruta construida por cada una de las colonias. Si `lengthLC` es igual a 0, utiliza el método CHM sin estrategias de mejora para construir las rutas. En otro caso, usa el método CHM y las listas de candidatos como estrategia de mejora.
- `acoVRP(colony, lengthLC)`: Regresa la mejor solución, la cual contiene las mejores rutas construidas por la colonia `colony` que satisfacen la demanda de todos los clientes. Si `lengthLC` es diferente de 0, utiliza el método CHM y las listas de candidatos como estrategia de mejora para construir las rutas. En otro caso, usa el método CHU sin estrategias de mejora.
- `vrp(strategy, lengthLC)`: regresa la mejor solución construida utilizando la estrategia `strategy` correspondiente.

- **ACO:**

Esta clase simula una **Colonia de hormigas**, la cual nos permite crear y utilizar las hormigas de la colonia para construir las rutas, guardar las mejores rutas

¹⁴Se usará el acrónimo CHM para referirse al método de optimización colonias de hormigas múltiples.

construidas por las hormigas, realizar la actualización local y global del rastro de feromonas, también nos ayuda a evaporar una cantidad constante de feromonas. Sus principales métodos son:

- `updateLP(k,j)`: realiza la actualización local del rastro de feromonas, la cual reduce la cantidad de feromonas en la arista que incide en el cliente k y en el cliente j .
- `updateGP(route, dist)`: realiza la actualización global del rastro de feromonas, la cual agrega feromonas a todas las aristas incluidas en la ruta $route$.

- **Candidates:**

Clase que se encarga de buscar las **listas de candidatos** para todos los clientes y seleccionar al siguiente cliente a visitar utilizando las listas como estrategia de mejora. Los principales métodos de esta clase son:

- `searchCandidatesLists()`: busca la lista de candidatos para cada uno de los clientes.
- `chooseBestClient(ant, k, visited, pheromones)`: selecciona el mejor cliente posible de la lista de candidatos del cliente k , que no haya sido visitado y con una demanda menor o igual que la capacidad de carga actual de la hormiga ant .
- `chooseNextClient(ant, k, visited, pheromones)`: elige al siguiente cliente que va a visitar la hormiga ant utilizando el método `chooseBestClient` y en caso de que todos los posibles candidatos ya han sido visitados elige aleatoriamente un cliente que no lo haya sido y cumpla con la restricción de carga de la hormiga.
- `chooseRandomClient(ant, visited)`: selecciona de manera aleatoria el primer cliente que va a visitar la hormiga ant de los clientes que no han sido visitados de la lista de candidatos del depósito y en caso de que todos los posibles candidatos ya han sido visitados elige aleatoriamente un cliente que no lo haya sido.

- **TwoOpt:**

Esta clase se encarga de mejorar las rutas utilizando la **heurística 2-opt** e intercambios 2-opt válidos. Sus principales métodos son:

- `improveRoute(route, dist)`: trata de mejorar la ruta $route$ probando todos los posibles intercambios 2-opt hasta encontrar una ruta r' 2-óptima que recorra una menor distancia que $dist$.
- `swap(route, i, j)`: realiza el intercambio 2-opt entre los clientes i y j de la ruta $route$.

- **Ant:**

Clase que simula una **hormiga** y tiene los siguientes atributos: una memoria en la que guarda la ruta que construye, una capacidad de carga actual, un número

4. IMPLEMENTACIÓN DEL PROGRAMA

con la distancia que recorre la ruta en la memoria y una bandera que indica si ya terminó de construir su ruta.

- **Client:**

Esta clase simula un **cliente** y tiene los siguientes atributos: un identificador, un nombre de ciudad, un nombre de estado, un índice con la posición del cliente en la lista de clientes, una demanda que debe satisfacerse, una bandera que nos indica si es el depósito y una lista de candidatos.

- **Connector:**

Clase que se encarga de establecer la **conexión** con la base de datos.

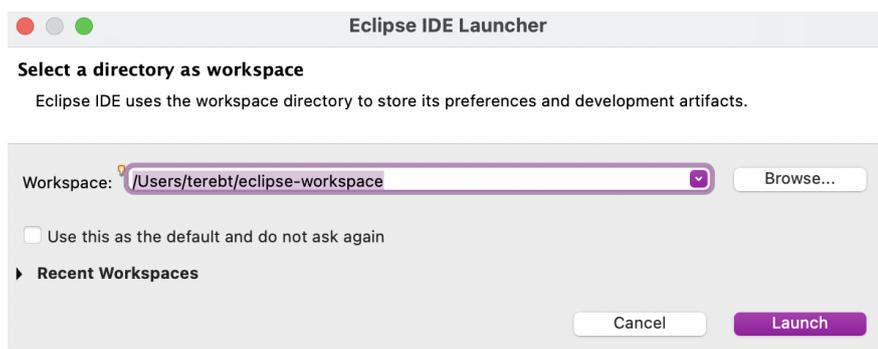
- **SearchBD:**

Esta clase se conecta con la base de datos **cities** para **obtener** los datos de los clientes y las distancias entre cada par de clientes, así como la distancia entre cada cliente y el depósito.

4.3. Instrucciones de ejecución

En esta sección se describe cómo ejecutar el programa¹⁵ desde Eclipse; en cada paso se presenta una captura de pantalla para ilustrar lo descrito.

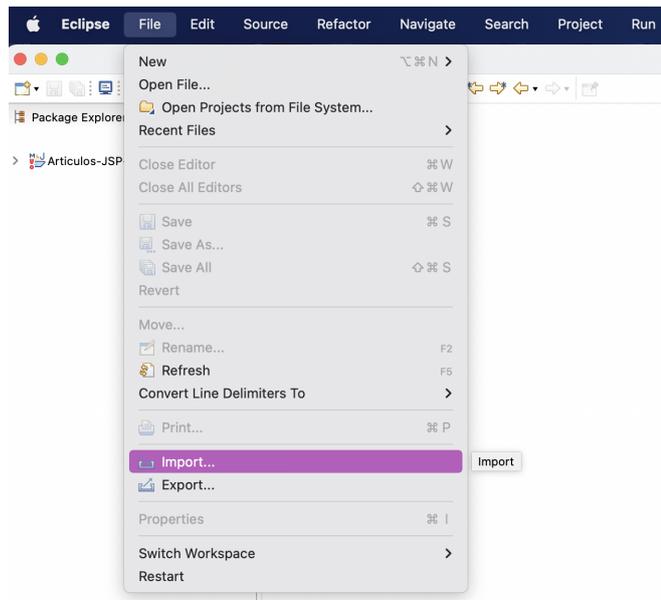
1. Abrir Eclipse IDE y elegir el directorio de trabajo.



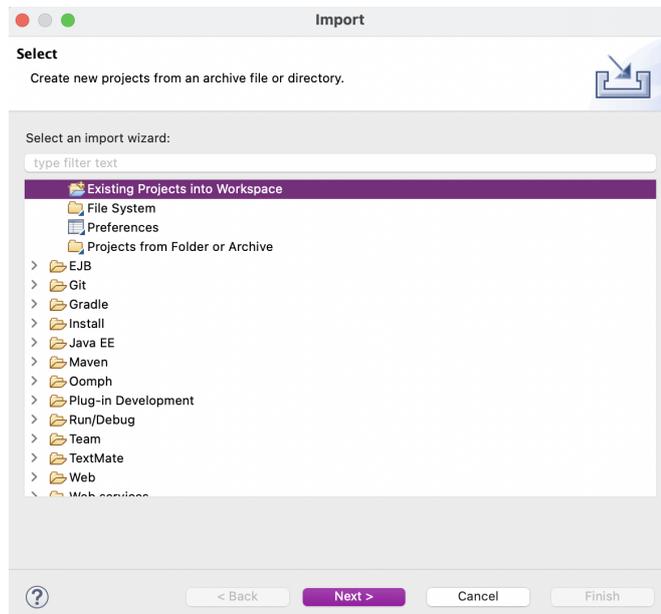
¹⁵Para la ejecución del programa se utilizaron los archivos `files/Orders.txt` y `files/Configurations.txt`.

2. Importar proyecto _tesis, en caso de que ya se haya importado el proyecto saltar este paso.

2.1 Seleccionar Import... del menú File.

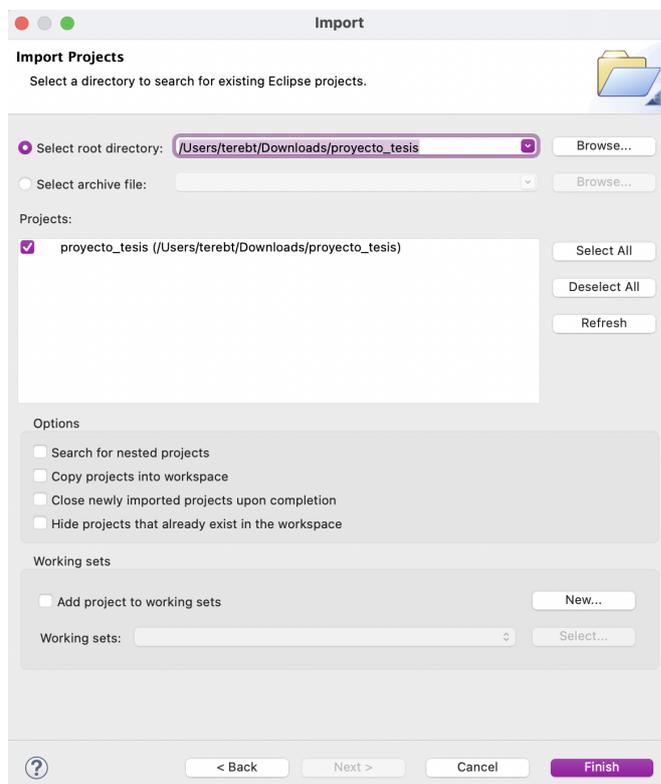


2.2 Elegir Existing Projects into Workspace y presionar Next.

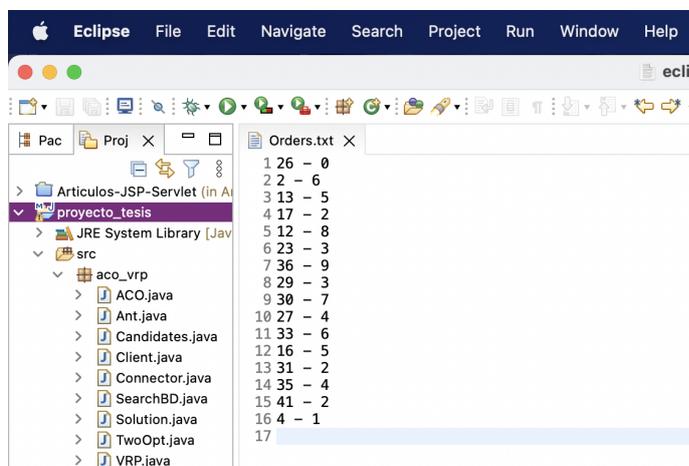


4. IMPLEMENTACIÓN DEL PROGRAMA

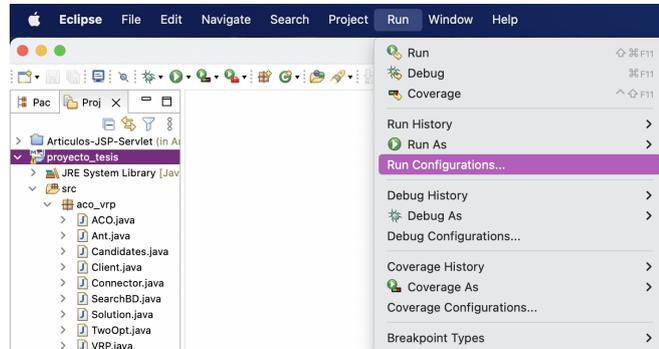
2.3 Buscar el directorio y el proyecto que se quiere importar. Presionar Finish para importar el proyecto.



3. Seleccionar `proyecto_tesis`, abrir el archivo `files/Orders.txt` e ingresar los identificadores y las demandas tanto del depósito como de los clientes del ejemplar con la siguiente estructura: `idCiudad – demanda`, el depósito debe tener una demanda igual a 0.

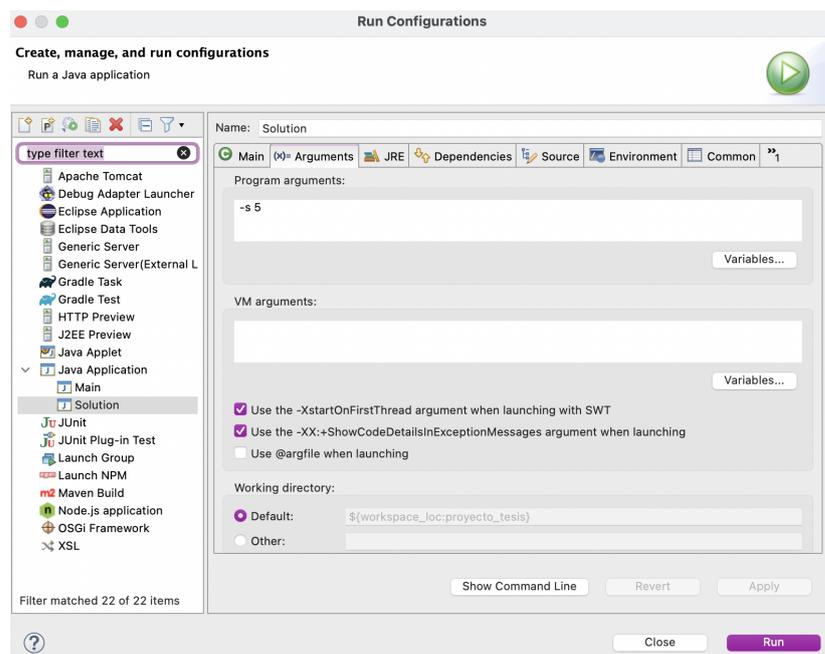


4. Seleccionar Run Configurations... del menú Run.



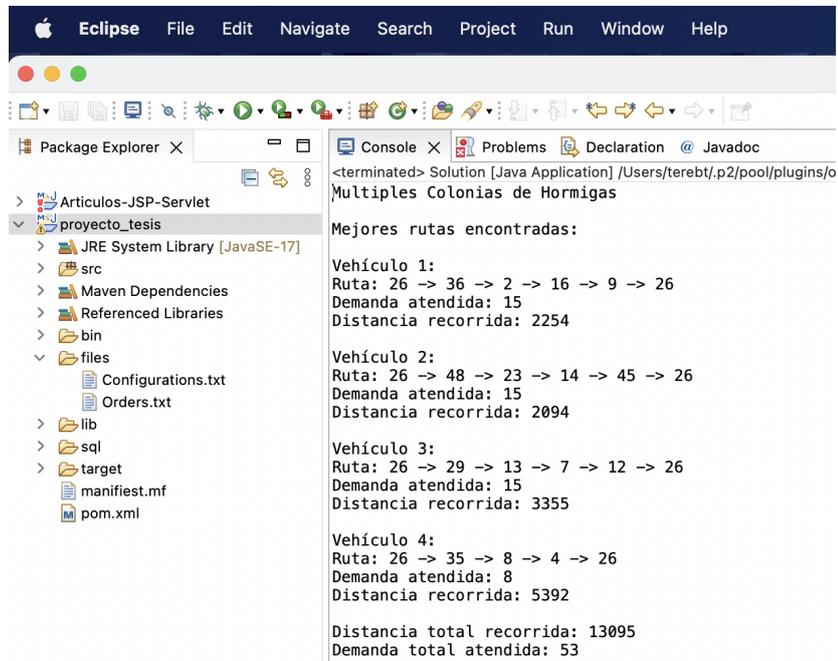
5. Seleccionar la pestaña Arguments y escribir los argumentos con los que se quiere ejecutar el programa en la sección Program arguments.

- -s 1 para utilizar CHU sin estrategias de mejora.
- -s 2 para usar CHU y heurística 2-opt como estrategia de mejora.
- -s 3 -c n para utilizar CHU y listas de candidatos como estrategia de mejora, donde n es la longitud de las listas de candidatos.
- -s 4 para usar CHM sin estrategias de mejora.
- -s 5 para utilizar CHM y heurística 2-opt como estrategia de mejora.
- -s 6 -c n para usar CHM y listas de candidatos como estrategia de mejora, donde n es la longitud de las listas de candidatos.



4. IMPLEMENTACIÓN DEL PROGRAMA

6. Presionar Run y se imprime la mejor solución encontrada para el ejemplar.



```
<terminated> Solution [Java Application] /Users/terebt/p2/pool/plugins/
Multiples Colonias de Hormigas

Mejores rutas encontradas:

Vehículo 1:
Ruta: 26 -> 36 -> 2 -> 16 -> 9 -> 26
Demanda atendida: 15
Distancia recorrida: 2254

Vehículo 2:
Ruta: 26 -> 48 -> 23 -> 14 -> 45 -> 26
Demanda atendida: 15
Distancia recorrida: 2094

Vehículo 3:
Ruta: 26 -> 29 -> 13 -> 7 -> 12 -> 26
Demanda atendida: 15
Distancia recorrida: 3355

Vehículo 4:
Ruta: 26 -> 35 -> 8 -> 4 -> 26
Demanda atendida: 8
Distancia recorrida: 5392

Distancia total recorrida: 13095
Demanda total atendida: 53
```

Capítulo 5

Experimentos y Resultados

En este capítulo se describen el objetivo, los detalles de ejecución, los detalles de los experimentos y los resultados, los cuales se comparan con un trabajo similar.

5.1. Objetivo

Analizar y comparar la calidad de las soluciones generadas por dos métodos de optimización diferentes (colonia de hormigas única y múltiples) sobre un conjunto de ejemplares, así como evaluar el impacto del uso de estrategias de mejora (listas de candidatos y heurística 2-opt) en la calidad de la solución construida por cada uno de estos métodos para el CVRP y comparar los resultados obtenidos.

5.2. Detalles de la ejecución

Las ejecuciones se llevaron a cabo en una laptop con las siguientes características:

- **Sistema Operativo:** macOS Monterey 12.5
- **Procesador:** Apple M2 3,49 GHz
- **Memoria RAM:** 8 GB
- **HDD:** 512 GB

5.3. Detalles del experimento

Se utilizaron tres ejemplares para las pruebas: el primero se usó para verificar que el programa funcionara y fue seleccionado aleatoriamente de un grupo de 51 ciudades de la República Mexicana, mientras que el segundo y el tercero se utilizaron para comparar

5. EXPERIMENTOS Y RESULTADOS

la calidad de las soluciones generadas por el programa y fueron propuestos en la sección *Ejemplares* de la Tesis de Licenciatura de Estudillo [20].

Para cada ejemplar se consideró una ciudad como el depósito central y a las demás ciudades como los clientes. Los parámetros de configuración son los mismos para los tres ejemplares y se utilizaron dos diferentes longitudes para las listas de candidatos.

5.3.1. Parámetros de configuración

Para los tres ejemplares se utilizaron los siguientes parámetros de configuración:

Significado	Variable	Valor
Número de vehículos	m	5
Capacidad de carga de los vehículos	Q	40
Importancia de la distancia	β	8.5
Probabilidad de seleccionar al mejor cliente	q_0	0.8
Velocidad de evaporación	α	0.1
Cantidad inicial de feromonas	τ_0	500

Tabla 5.1: Parámetros de configuración de los ejemplares.

5.3.2. Ejemplar de 15 ciudades

El depósito y los clientes del ejemplar de 15 ciudades, E_{15} ¹⁶, son los siguientes:

Identificador	Ciudad	Demanda
26	Monterrey	Depósito
2	Aguascalientes	6
13	Cuernavaca	5
17	Guanajuato	2
12	Colima	8
23	Mazatlán	3

Tabla 5.2: Ciudades y depósito de E_{15} .

¹⁶Se utilizará la nomenclatura E_{15} para referirse a este ejemplar.

Identificador	Ciudad	Demanda
36	Saltillo	9
29	Nuevo Laredo	3
30	Oaxaca de Juárez	7
27	Morelia	4
33	Puebla de Zaragoza	6
16	Guadalajara	5
31	Pachuca	2
35	Salina Cruz	4
41	Tepic	2
4	Cancún	1

Tabla 5.2 (Continuación)

5.3.3. Ejemplar de 20 ciudades

El depósito y los clientes del ejemplar de 20 ciudades, E_{20} ¹⁷, son los siguientes:

Identificador	Ciudad	Demanda
5	Chetumal	Depósito
46	Tuxtla Gutiérrez	1
44	Toluca de Lerdo	2
24	Mexicali	3
2	Aguascalientes	4
47	Veracruz	5
32	Piedras Negras	6
42	Tijuana	7

Tabla 5.3: Ciudades y depósito de E_{20} .

¹⁷Se utilizará la nomenclatura E_{20} para referirse a este ejemplar.

5. EXPERIMENTOS Y RESULTADOS

Identificador	Ciudad	Demanda
26	Monterrey	8
49	Villahermosa	9
8	Ciudad Cuauhtémoc	8
6	Chihuahua	7
22	Manzanillo	6
14	Culiacán Rosales	5
27	Morelia	4
3	Campeche	3
23	Mazatlán	2
13	Cuernavaca	1
4	Cancún	2
21	León	3
15	Ensenada	4

Tabla 5.3 (Continuación)

5.3.4. Ejemplar de 30 ciudades

El depósito y los clientes del ejemplar de 30 ciudades, E_{30} ¹⁸, son los siguientes:

Identificador	Ciudad	Demanda
9	Ciudad de México	Depósito
3	Campeche	1
15	Ensenada	2
36	Saltillo	3
1	Acapulco	4

Tabla 5.4: Ciudades y depósito de E_{30} .

¹⁸Se utilizará la nomenclatura E_{30} para referirse a este ejemplar.

Identificador	Ciudad	Demanda
26	Monterrey	5
16	Guadalajara	6
7	Chilpancingo de los Bravos	7
47	Veracruz	8
2	Aguascalientes	9
17	Guanajuato	8
24	Mexicali	7
13	Cuernavaca	6
25	Mérida	5
32	Piedras Negras	4
21	León	3
12	Colima	2
41	Tepic	1
20	La Paz	2
18	Hermosillo	3
29	Nuevo Laredo	4
31	Pachuca	5
8	Ciudad Cuauhtémoc	6
37	San Luis Potosí	7
48	Victoria de Durango	8
38	Santiago de Querétaro	9
49	Villahermosa	8
5	Chetumal	7
10	Ciudad Juárez	6
45	Torreón	5
22	Manzanillo	4

Tabla 5.4 (Continuación)

5.3.5. Tamaños de las listas de candidatos

Los tamaños de las listas de candidatos que se utilizaron son:

Ejemplar	Fracción	Tamaño
E_{15}	$n/4$	4
	$n/2$	8
E_{20}	$n/4$	5
	$n/2$	10
E_{30}	$n/4$	8
	$n/2$	15

Tabla 5.5: Tamaños de las listas de candidatos para cada ejemplar.

5.4. Resultados

En esta sección se muestran y analizan los resultados obtenidos de aplicar dos métodos de optimización diferentes y el impacto del uso de estrategias de mejora en la calidad de la solución. En el apéndice D se presenta el promedio de los recursos utilizados por cada método para los ejemplares E_{20} y E_{30} .

5.4.1. Ejemplar de 15 ciudades

La demanda total que se debe satisfacer en este ejemplar es de 67 unidades y como la capacidad de carga de los vehículos es igual a 40 unidades, entonces sólo se necesitan 2 vehículos para atender las demandas de todos los clientes. Las soluciones para este ejemplar son las siguientes:

- **CHU sin estrategias de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 13, 33, 17, 27, 12, 26]	40	3,392 km
2	[26, 29, 16, 23, 41, 31, 30, 35, 4, 26]	27	7,075 km
Total		67	10,467 km

Tabla 5.6: Solución de E_{15} con CHU sin estrategias de mejora.

- **CHU y heurística 2-Opt como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 13, 33, 17, 27, 12, 26]	40	3,392 km
2	[26, 29, 23, 41, 16, 31, 30, 35, 4, 26]	27	6,636 km
Total		67	10,028 km

Tabla 5.7: Solución de E_{15} con CHU y heurística 2-Opt.

- **CHU y listas de candidatos de longitud 4 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 13, 17, 12, 16, 41, 23, 26]	40	3,579 km
2	[26, 29, 27, 31, 33, 30, 35, 4, 26]	27	5,905 km
Total		67	9,484 km

Tabla 5.8: Solución de E_{15} con CHU y listas de candidatos de longitud 4.

- **CHU y listas de candidatos de longitud 8 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 2, 13, 33, 31, 27, 17, 16, 12, 41, 26]	40	3,743 km
2	[26, 36, 29, 23, 30, 35, 4, 26]	27	6,795 km
Total		67	10,538 km

Tabla 5.9: Solución de E_{15} con CHU y listas de candidatos de longitud 8.

La solución construida con el método CHU y heurística 2-opt es 4.37% mejor que la del método CHU sin estrategias de mejora (método 1). En la tabla 5.7 se puede ver que la ruta del vehículo 1 es 2-óptima, ya que la heurística no encontró una ruta que recorra una menor distancia que la ruta del vehículo 1 del método 1. Mientras que el vehículo 2 viaja una distancia 6.62% más corta que el vehículo 2 del método 1.

El método CHU y listas de candidatos de longitud 4 genera una solución 11.11% mejor que la del método CHU y listas de candidatos de longitud 8 (método 4). Como se puede ver en la tabla 5.8 la ruta del vehículo 1 recorre una distancia 4.58% menor que la ruta del vehículo 1 del método 4. Mientras que la ruta del vehículo 2 viaja una distancia 15.07% más corta que la del vehículo 2 del método 4. Además, es la mejor

5. EXPERIMENTOS Y RESULTADOS

solución construida utilizando el método de CHU, ya que recorre 10.36 % menos que la solución del método CHU sin estrategias de mejora y viaja 5.73 % menos que la solución del método CHU y heurística 2-opt.

- **CHM sin estrategias de mejora**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 17, 27, 13, 33, 12, 26]	40	3,299 km
2	[26, 29, 16, 23, 41, 31, 30, 35, 4, 26]	27	7,075 km
Total		67	10,374 km

Tabla 5.10: Solución de E_{15} con CHM sin estrategias de mejora.

- **CHM y heurística 2-Opt como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 17, 27, 13, 33, 12, 26]	40	3,299 km
2	[26, 29, 23, 41, 16, 31, 30, 35, 4, 26]	27	6,636 km
Total		67	9,935 km

Tabla 5.11: Solución de E_{15} con CHM y heurística 2-Opt.

- **CHM y listas de candidatos de longitud 4 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 17, 13, 12, 16, 41, 23, 26]	40	3,510 km
2	[26, 29, 27, 31, 33, 30, 35, 4, 26]	27	5,905 km
Total		67	9,415 km

Tabla 5.12: Solución de E_{15} con CHM y listas de candidatos de longitud 4.

- CHM y listas de candidatos de longitud 8 como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[26, 36, 2, 17, 13, 33, 31, 12, 41, 26]	40	3,570 km
2	[26, 29, 23, 16, 27, 30, 35, 4, 26]	27	6,683 km
Total		67	10,253 km

Tabla 5.13: Solución de E_{15} con CHM y listas de candidatos de longitud 8.

La solución generada con el método CHM y heurística 2-opt es 4.42% mejor que la del método CHM sin estrategias de mejora (método 5). Como se puede ver en la tabla 5.11 la ruta del vehículo 1 es 2-óptima, ya que la heurística no encontró una ruta que recorra una menor distancia que la ruta del vehículo 1 del método 5. Mientras que la ruta del vehículo 2 recorre una distancia 6.62% menor que la del vehículo 2 del método 5.

El método CHM y listas de candidatos de longitud 4 construye una solución 8.9% mejor que la del método CHM y listas de candidatos de longitud 8 (método 8). En la tabla 5.12 se puede ver que la ruta del vehículo 1 viaja una distancia 1.71% menor que la ruta del vehículo 1 del método 8. Mientras que la ruta del vehículo 2 recorre una distancia 13.18% más corta que la del vehículo 2 del método 8. Además, es la mejor solución generada utilizando el método CHM, ya que viaja 10.19% menos que la solución del método CHM sin estrategias de mejora y recorre 5.52% menos que la solución del método CHM y heurística 2-opt.

En cada uno de los métodos, el vehículo 1 atiende una demanda de 40 unidades y el vehículo 2 satisface una demanda de 27 unidades. La mejor solución para este ejemplar fue construida con el método CHM y listas de candidatos de longitud 4, ya que viaja 0.73% menos que la solución generada por el método CHU y listas de candidatos de longitud 4.

5.4.2. Ejemplar de 20 ciudades

La demanda total que se debe atender en este ejemplar es de 90 unidades y como los vehículos tienen una capacidad de carga igual a 40 unidades, entonces sólo se necesitan 3 vehículos para satisfacer las demandas de todos los clientes. Las soluciones para este ejemplar son las siguientes:

- **CHU sin estrategias de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[5, 46, 44, 13, 24, 42, 15, 6, 2, 21, 47, 3, 5]	40	9,172 km
2	[5, 4, 49, 32, 26, 27, 22, 14, 5]	40	8,352 km
3	[5, 8, 23, 5]	10	5,206 km
Total		90	22,730 km

Tabla 5.14: Solución de E_{20} con CHU sin estrategias de mejora.

- **CHU y heurística 2-Opt como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[5, 46, 13, 44, 15, 42, 24, 6, 2, 21, 47, 3, 5]	40	9,013 km
2	[5, 4, 49, 27, 22, 14, 32, 26, 5]	40	7,516 km
3	[5, 8, 23, 5]	10	5,206 km
Total		90	21,735 km

Tabla 5.15: Solución de E_{20} con CHU y heurística 2-Opt.

- **CHU y listas de candidatos de longitud 5 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[5, 46, 8, 49, 44, 13, 27, 21, 2, 22, 23, 5]	40	6,606 km
2	[5, 4, 3, 24, 42, 15, 6, 32, 26, 5]	40	9,474 km
3	[5, 47, 14, 5]	10	5,135 km
Total		90	21,215 km

Tabla 5.16: Solución de E_{20} con CHU y listas de candidatos de longitud 5.

- **CHU y listas de candidatos de longitud 10 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[5, 4, 46, 8, 49, 44, 13, 27, 21, 2, 22, 5]	40	6,410 km
2	[5, 24, 42, 15, 6, 32, 26, 23, 3, 5]	40	10,230 km
3	[5, 47, 14, 5]	10	5,135 km
Total		90	21,775 km

Tabla 5.17: Solución de E_{20} con CHU y listas de candidatos de longitud 10.

La solución generada con el método CHU y heurística 2-opt es 4.58 % mejor que la del método CHU sin estrategias de mejora (método 1). En la tabla 5.15 se puede ver que la ruta del vehículo 3 es 2-óptima, ya que la heurística no encontró una ruta que recorra una menor distancia que la ruta del vehículo 3 del método 1. Mientras que las rutas de los vehículos 1 y 2 recorren una distancia 9.85 % y 11.12 % menor que las rutas de los vehículos 1 y 2 del método 1, respectivamente.

El método CHU y listas de candidatos de longitud 5 construye una solución 2.64 % mejor que la del método CHU y listas de candidatos de longitud 10 (método 4). Como se puede ver en la tabla 5.16 la ruta del vehículo 1 viaja una distancia 2.96 % más larga que la del vehículo 1 del método 4 y el vehículo 2 recorre una distancia 7.98 % menor que el vehículo 2 del método 4. Mientras que la ruta del vehículo 3 viaja la misma distancia que la del vehículo 3 del método 4. Además, es la mejor solución generada utilizando el método CHU, ya que viaja 7.14 % menos que la solución del método CHU sin estrategias de mejora y recorre 7.16 % menos que la solución del método CHU y heurística 2-opt.

- **CHM sin estrategias de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[5, 4, 46, 8, 49, 44, 27, 13, 21, 2, 22, 5]	40	6,811 km
2	[5, 24, 42, 15, 6, 23, 26, 32, 3, 5]	40	10,336 km
3	[5, 47, 14, 5]	10	5,135 km
Total		90	22,282 km

Tabla 5.18: Solución de E_{20} con CHM sin estrategia de mejora.

5. EXPERIMENTOS Y RESULTADOS

- CHM y heurística 2-Opt como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[5, 4, 49, 13, 44, 27, 22, 2, 21, 46, 8, 5]	40	5,803 km
2	[5, 23, 24, 42, 15, 6, 32, 26, 3, 5]	40	9,041 km
3	[5, 47, 14, 5]	10	5,135 km
Total		90	19,979 km

Tabla 5.19: Solución de E_{20} con CHM y heurística 2-Opt.

- CHM y listas de candidatos de longitud 5 como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[5, 4, 49, 13, 44, 27, 22, 2, 21, 46, 8, 5]	40	5,803 km
2	[5, 26, 32, 6, 24, 42, 15, 23, 3, 5]	40	9,041 km
3	[5, 47, 14, 5]	10	5,135 km
Total		90	19,979 km

Tabla 5.20: Solución de E_{20} con CHM y listas de candidatos de longitud 5.

- CHM y listas de candidatos de longitud 10 como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[5, 47, 32, 26, 27, 22, 49, 4, 5]	40	7,009 km
2	[5, 46, 8, 44, 24, 42, 15, 6, 2, 21, 13, 5]	40	9,234 km
3	[5, 3, 14, 23, 5]	10	5,336 km
Total		90	21,579 km

Tabla 5.21: Solución de E_{20} con CHM y listas de candidatos de longitud 10.

La solución construida con el método CHM y heurística 2-opt es 11.53% mejor que la del método CHM sin estrategias de mejora (método 5). Como se puede ver en la tabla 5.19 la ruta del vehículo 3 es 2-óptima, ya que la heurística no encontró una ruta que recorra una menor distancia que la ruta del vehículo 3 del método 5. Mientras que las rutas de los vehículos 1 y 2 viajan una distancia 17.37% y 14.32% menor que las rutas de los vehículos 1 y 2 del método 5, respectivamente.

El método CHM y listas de candidatos de longitud 5 genera una solución 8.01% mejor que la del método CHM y listas de candidatos de longitud 10 (método 8). En la tabla 5.20 se puede ver que la ruta del vehículo 1 recorre una distancia 20.78% menor que la del vehículo 1 del método 8, el vehículo 2 viaja una distancia 2.13% más corta que el vehículo 2 del método 8 y la ruta del vehículo 3 viaja una distancia 3.91% menor que la del vehículo 3 del método 8. Además, es la mejor solución construida utilizando el método CHM, ya que recorre 11.53% menos que la solución del método CHM sin estrategias de mejora y viaja la misma distancia que la solución del método CHM y heurística 2-opt.

En cada uno de los métodos, los vehículos 1 y 2 satisfacen una demanda de 40 unidades, mientras que el vehículo 3 atiende una demanda de 27 unidades. La mejor solución para este ejemplar fue construida tanto con el método CHM y heurística 2-opt como con el método CHM y listas de candidatos de longitud 5, ya que viaja 6.18% menos que la solución generada por el método CHU y listas de candidatos de longitud 5.

5.4.3. Ejemplar de 30 ciudades

La demanda total que se debe satisfacer en este ejemplar es de 155 unidades y como la capacidad de carga de los vehículos es igual a 40 unidades, entonces sólo se necesitan 3 vehículos para atender las demandas de todos los clientes. Las soluciones para este ejemplar son las siguientes:

- **CHU sin estrategias de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[9, 31, 3, 25, 5, 49, 8, 47, 9]	40	3,926 km
2	[9, 13, 7, 1, 38, 15, 36, 26, 29, 9]	40	7,268 km
3	[9, 17, 21, 2, 37, 16, 12, 22, 41, 9]	40	2,551 km
4	[9, 48, 45, 32, 10, 24, 20, 18, 9]	35	8,789 km
Total		155	22,534 km

Tabla 5.22: Solución de E_{30} con CHU sin estrategia de mejora.

- **CHU y heurística 2-Opt como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[9, 31, 47, 8, 5, 25, 3, 49, 9]	40	3,697 km
2	[9, 1, 7, 13, 15, 29, 26, 36, 38, 9]	40	6,910 km
3	[9, 17, 21, 41, 12, 22, 16, 2, 37, 9]	40	2,394 km
4	[9, 32, 10, 24, 20, 18, 48, 45, 9]	35	8,778 km
Total		155	21,779 km

Tabla 5.23: Solución de E_{30} con CHU y heurística 2-Opt.

- **CHU y listas de candidatos de longitud 8 como estrategia de mejora:**

Vehículo	Ruta	Demanda	Distancia
1	[9, 49, 3, 25, 5, 8, 47, 31, 9]	40	3,697 km
2	[9, 37, 18, 24, 15, 10, 2, 16, 9]	40	6,256 km
3	[9, 38, 17, 21, 41, 12, 1, 7, 13, 9]	40	2,256 km
4	[9, 36, 32, 29, 26, 20, 45, 48, 22, 9]	35	7,385 km
Total		155	19,594 km

Tabla 5.24: Solución de E_{30} con CHU y listas de candidatos de longitud 8.

- **CHU y listas de candidatos de longitud 15:**

Vehículo	Ruta	Demanda	Distancia
1	[9, 3, 25, 5, 49, 8, 47, 31, 9]	40	3,901 km
2	[9, 15, 24, 18, 10, 45, 48, 2, 9]	40	6,495 km
3	[9, 36, 1, 7, 13, 38, 17, 21, 9]	40	3,226 km
4	[9, 37, 16, 12, 22, 41, 26, 32, 29, 20, 9]	35	10,292 km
Total		155	23,914 km

Tabla 5.25: Solución de E_{30} con CHU y listas de candidatos de longitud 15.

La solución construida con el método CHU y heurística 2-opt es 3.47% mejor que la del método CHU sin estrategias de mejora (método 1). Como se puede ver en la tabla 5.23 la ruta del vehículo 1 recorre una distancia 6.19% menor que la del vehículo 1 del método 1, el vehículo 2 viaja una distancia 5.18% más corta que el vehículo 2 del método 1, la ruta del vehículo 3 recorre una distancia 6.56% menor que la del vehículo 3 del método 1 y el vehículo 4 viaja una distancia 0.13% más corta que el vehículo 4 del método 1.

El método CHU y listas de candidatos de longitud 8 genera una solución 22.05% mejor que la del método CHU y listas de candidatos de longitud 15 (método 4). En la tabla 5.24 se puede ver que la ruta del vehículo 1 recorre una distancia 5.52% menor que la del vehículo 1 del método 4, el vehículo 2 viaja una distancia 3.82% más corta que el vehículo 2 del método 4, la ruta del vehículo 3 recorre una distancia 42.99% menor que la del vehículo 3 del método 4 y el vehículo 4 viaja una distancia 39.36% más corta que el vehículo 4 del método 4. Además, es la mejor solución construida utilizando el método CHU, ya que recorre 15.01% menos que la solución del método CHU sin estrategias de mejora y viaja 11.15% que la solución del método CHU y heurística 2-opt.

- CHM sin estrategias de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[9, 31, 3, 25, 5, 49, 8, 47, 9]	40	3,926 km
2	[9, 13, 7, 1, 38, 17, 21, 12, 41, 9]	40	2,712 km
3	[9, 15, 24, 18, 10, 32, 29, 26, 36, 16, 9]	40	7,028 km
4	[9, 37, 2, 48, 45, 20, 22, 9]	35	8,703 km
Total		155	22,369 km

Tabla 5.26: Solución de E_{30} con CHM sin estrategias de mejora.

- CHM y heurística 2-Opt como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[9, 31, 47, 8, 5, 25, 3, 49, 9]	40	3,697 km
2	[9, 13, 7, 1, 12, 41, 21, 17, 38, 9]	40	2,256 km
3	[9, 36, 26, 29, 32, 10, 24, 15, 18, 16, 9]	40	6,569 km
4	[9, 37, 2, 48, 45, 20, 22, 9]	35	8,703 km
Total		155	21,225 km

Tabla 5.27: Solución de E_{30} con CHM y heurística 2-Opt.

- CHM y listas de candidatos de longitud 8 como estrategia de mejora:

Vehículo	Ruta	Demanda	Distancia
1	[9, 49, 3, 25, 5, 8, 47, 31, 9]	40	3,697 km
2	[9, 45, 10, 24, 15, 18, 48, 2, 9]	40	6,037 km
3	[9, 13, 1, 7, 36, 21, 17, 38, 9]	40	2,631 km
4	[9, 37, 32, 29, 26, 20, 41, 12, 22, 16, 9]	35	6,834 km
Total		155	19,199 km

Tabla 5.28: Solución de E_{30} con CHM y listas de candidatos de longitud 8.

- CHM y listas de candidatos de longitud 15:

Vehículo	Ruta	Demanda	Distancia
1	[9, 3, 25, 5, 49, 8, 47, 31, 9]	40	3,901 km
2	[9, 37, 2, 16, 15, 24, 18, 10, 9]	40	6,588 km
3	[9, 13, 7, 1, 38, 17, 21, 41, 12, 9]	40	2,700 km
4	[9, 36, 26, 29, 32, 20, 45, 48, 22, 9]	35	9,457 km
Total		155	22,646 km

Tabla 5.29: Solución de E_{30} con CHM y listas de candidatos de longitud 15.

La solución generada con el método CHM y heurística 2-opt es 5.39 % mejor que la del método CHM sin estrategias de mejora (método 5). En la tabla 5.27 se puede ver que la ruta del vehículo 4 es 2-óptima, ya que la heurística no encontró una ruta que recorra una menor distancia que la ruta del vehículo 4 del método 5. Mientras que la ruta del vehículo 1 viaja una distancia 6.19 % más corta que la del vehículo 1 del método 5, el vehículo 2 recorre una distancia 20.21 % menor que el vehículo 2 del método 5 y la ruta del vehículo 3 viaja una distancia 6.99 % más corta que la del vehículo 3 del método 5.

El método CHM y listas de candidatos de longitud 8 genera una solución 17.95 % mejor que la del método CHM y listas de candidatos de longitud 15 (método 8). En la tabla 5.28 se puede ver que la ruta del vehículo 1 recorre una distancia 5.5 % menor que la del vehículo 1 del método 8, el vehículo 2 viaja una distancia 9.13 % más corta que el vehículo 2 del método 8, la ruta del vehículo 3 recorre una distancia 2.62 % menor que la del vehículo 3 del método 8 y el vehículo 4 viaja una distancia 38.38 % más corta que el vehículo 4 del método 8. Además, es la mejor solución construida utilizando el método CHM, ya que recorre 16.51 % menos que la solución del método CHM sin estrategias de mejora y viaja 10.55 % que la solución del método CHM y heurística 2-opt.

En cada uno de los métodos, los vehículos 1, 2 y 3 satisfacen una demanda de 40 unidades, mientras que el vehículo 3 atiende una demanda de 35 unidades. La mejor solución para este ejemplar fue construida con el método CHM y listas de candidatos de longitud 8, ya que recorre 2.06 % menos que la solución generada por el método CHU y listas de candidatos de longitud 8.

5.5. Comparación de resultados

En esta sección se comparan los resultados de los ejemplares E_{20} y E_{30} con los resultados de la sección *Resultados CVRP* de la Tesis de Licenciatura de Estudillo [20], respectivamente. Además, se utilizan los símbolos: \bar{x} para hacer referencia al promedio, σ para la desviación estándar y CV para el coeficiente de variación.

5.5.1. Ejemplar de 20 ciudades

En la figura 5.1 se muestran las distancias recorridas por cada una de las soluciones generadas con los métodos de optimización (Colonia de hormiga única y múltiples) sin utilizar y utilizando las estrategias de mejora (listas de candidatos y heurística 2-opt). Estas soluciones sólo hacen uso de los vehículos necesarios para atender la demanda total del ejemplar.

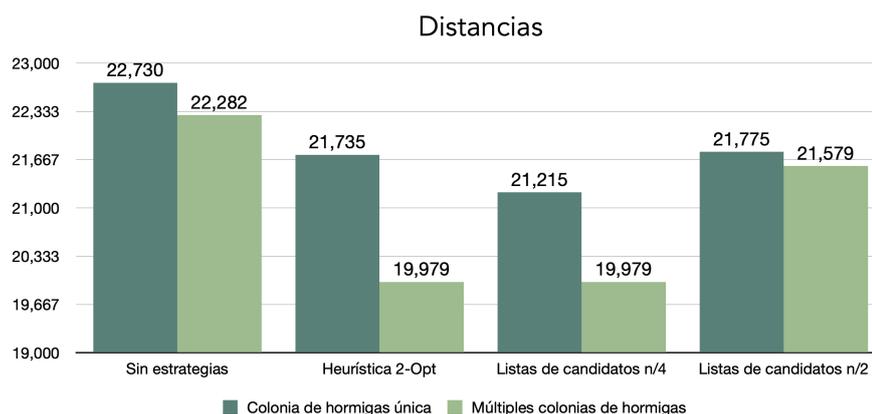


Figura 5.1: Distancias de las mejores soluciones de E_{20} en este trabajo.

En la figura 5.2 se presentan las distancias en kilómetros de las mejores soluciones halladas en la subsección *Ejemplar de 20 ciudades* de la sección *Resultados CVRP* de [20]. Los modelos tienen las siguientes variaciones:

- **Modelo 1:** La función de costo toma sólo la menor distancia recorrida y no permite que ningún vehículo supere su capacidad de carga.
- **Modelo 2:** La función de costo ajusta la carga a cada camión cercana al promedio y agrega un castigo al vehículo que no tenga $\pm 20\%$ de la carga promedio, en este caso es de 18 unidades por vehículo.
- **Modelo 3:** La función de costo toma sólo la menor distancia recorrida y permite anular algunas rutas. Además, permite que no se utilicen todos los camiones.

Las soluciones de este trabajo recorren una distancia promedio de 21,409.25 km con una desviación estándar de 928.96 km, mientras que las del trabajo de Estudillo viajan una distancia promedio de 19,882.33 km con una desviación estándar de 2,688.94 km.

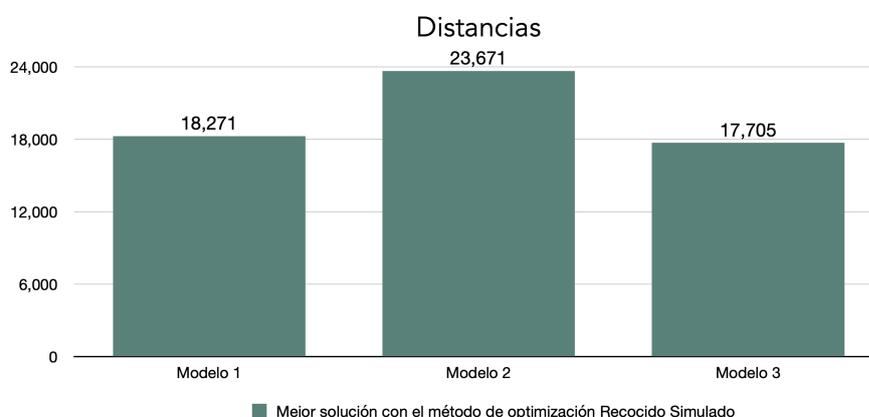


Figura 5.2: Distancias de las mejores soluciones de E_{20} en la Tesis de Estudillo.

Como se puede ver en la tabla 5.30 la distancia recorrida por las soluciones de este trabajo tiene una menor variación en comparación a la recorrida por las soluciones del trabajo de Estudillo.

Trabajo	\bar{x}	σ	CV
Becerril	21,409.25 km	928.96 km	4.34
Estudillo	19,882.33 km	2,688.94 km	13.54

Tabla 5.30: Variación de las distancias de las soluciones de E_{20} .

La mejor solución para este ejemplar en este trabajo viaja 19,979 km, la cual fue construida utilizando tanto el método CHM y heurística 2-opt como el método CHM y listas de candidatos de longitud $n/4 = 5$. Esta solución es 8.54% más larga que el modelo 1, 18.48% más corta que el modelo 2 y 11.38% más larga que el modelo 3.

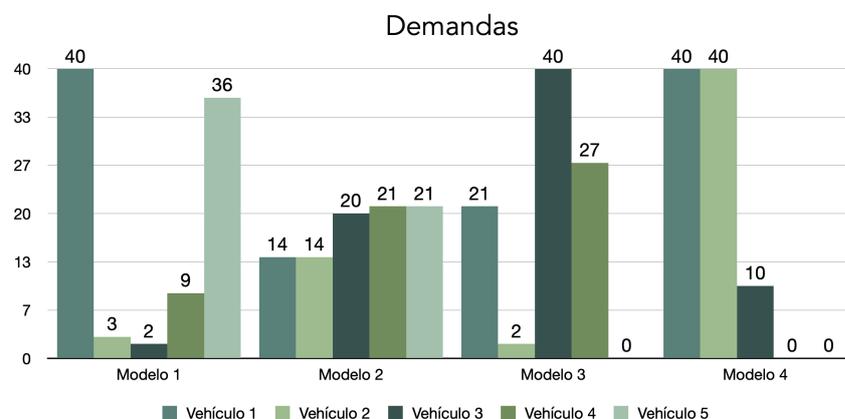


Figura 5.3: Demandas de las mejores soluciones de E_{20} en ambos trabajos.

En la figura 5.3 se muestran las demandas atendidas por cada vehículo de los modelos de las mejores soluciones. El modelo 4 representa a la mejor solución encontrada en este

5. EXPERIMENTOS Y RESULTADOS

trabajo. Como se muestra en la tabla 5.31 la demanda atendida por cada vehículo del modelo 4 tiene una menor variación en comparación a la atendida por los vehículos de los modelos 1 y 3, por lo que el balanceo de cargas del modelo 4 es mejor que el balanceo de los modelos 1 y 3 debido a que 2 vehículos del modelo 4 utilizan toda su capacidad de carga de tal forma que sólo se necesitan 3 vehículos para satisfacer la demanda de todos los clientes. Mientras que la demanda atendida por los vehículos del modelo 4 tiene una mayor variación en comparación a la atendida por cada vehículo del modelo 2, por lo cual el balanceo de cargas del modelo 4 es peor que el del modelo 2, ya que uno de los vehículos del modelo 4 sólo hace uso del 25% de su capacidad de carga. Podemos concluir que a pesar de que el modelo 4 recorre una distancia mayor que los modelos 1 y 3 necesita un menor número de vehículos para atender las demandas de todos los clientes. Por lo tanto, el modelo 4 es una buena solución para el ejemplar de 20 ciudades.

Modelo	\bar{x}	σ	CV
1	18	16.55	91.94
2	18	3.29	18.27
3	22.5	13.68	60.8
4	30	14.14	47.13

Tabla 5.31: Variación de las demandas de las soluciones de E_{20} .

5.5.2. Ejemplar de 30 ciudades

En la figura 5.4 se presentan las distancias totales recorridas por cada una de las soluciones generadas con los métodos de optimización (Colonia de hormiga única y múltiples) sin utilizar y utilizando las estrategias de mejora (listas de candidatos y heurística 2-opt). Estas soluciones sólo hacen uso de los vehículos necesarios para satisfacer la demanda total del ejemplar.

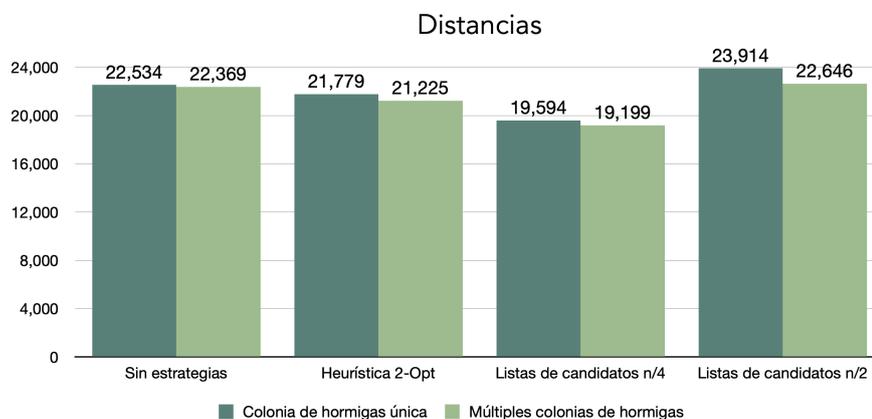


Figura 5.4: Distancias de las mejores soluciones de E_{30} en este trabajo.

En la figura 5.5 se muestran las distancias en kilómetros de las mejores soluciones halladas en la subsección *Ejemplar de 30 ciudades* de la sección *Resultados CVRP* de [20]. Los modelos tienen las siguientes variaciones:

- **Modelo 1:** La función de costo toma sólo la menor distancia recorrida y no permite que ningún vehículo supere su capacidad de carga.
- **Modelo 2:** La función de costo ajusta la carga a cada camión cercana al promedio y agrega un castigo al vehículo que no tenga $\pm 20\%$ de la carga promedio, en este caso es de 31 unidades por vehículo.
- **Modelo 3:** La función de costo toma sólo la menor distancia recorrida y permite anular algunas rutas. Además, permite que no se utilicen todos los camiones.

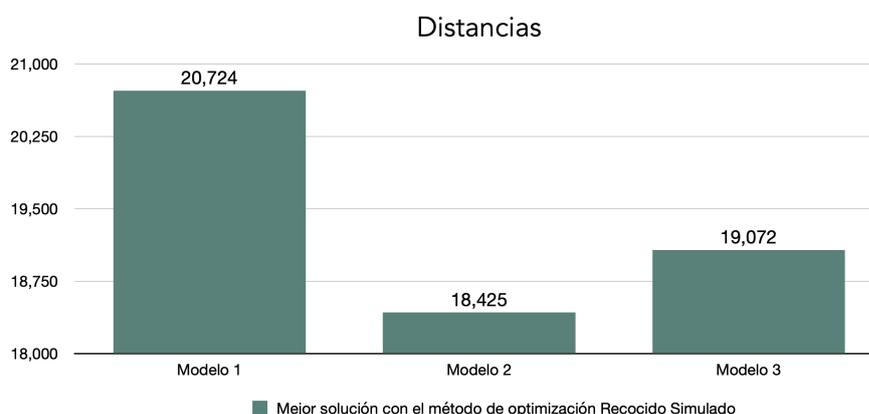


Figura 5.5: Distancias de las mejores soluciones de E_{30} en la Tesis de J. Estudillo.

Las soluciones de este trabajo recorren una distancia promedio de 21,657.50 km con una desviación estándar de 1,493.63 km, mientras que las del trabajo de Estudillo viajan una distancia promedio de 19,407 km con una desviación estándar de 967.99 km. Como se puede ver en la tabla 5.32 la distancia recorrida por las soluciones de este trabajo tiene una mayor variación en comparación a la recorrida por las soluciones del trabajo de Estudillo.

Trabajo	\bar{x}	σ	CV
Becerril	21,657.50 km	1,493.63 km	6.89
Estudillo	19,407 km	967.99 km	4.98

Tabla 5.32: Variación de las distancias de las soluciones de E_{30} .

La mejor solución para este ejemplar en este trabajo recorre 19,199 km, la cual fue construida utilizando tanto el método CHM y listas de candidatos de longitud $n/4 = 8$. Esta solución es 7.94 % más corta que el modelo 1, 4.03 % más larga que el modelo 2 y 0.66 % más larga que el modelo 3.

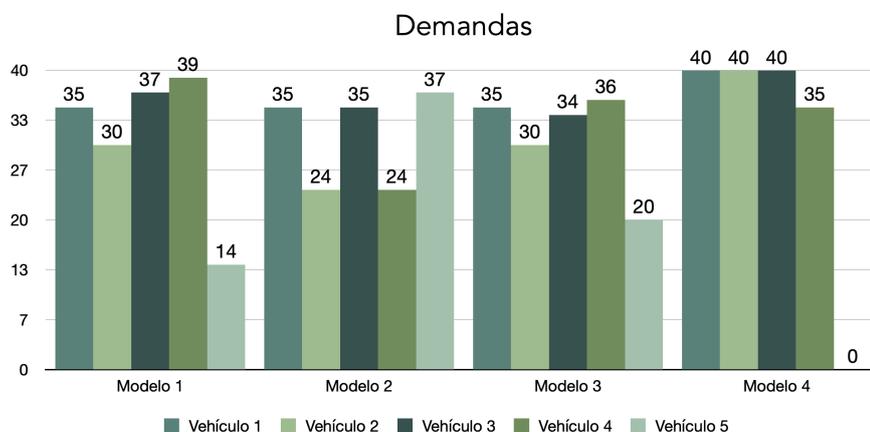


Figura 5.6: Demandas de las mejores soluciones de E_{30} en ambos trabajos.

En la figura 5.6 se presentan las demandas atendidas por cada vehículo de los modelos de las mejores soluciones. El modelo 4 representa a la mejor solución encontrada en este trabajo. Como se muestra en la tabla 5.33 la demanda atendida por cada vehículo del modelo 4 tiene una menor variación en comparación a la atendida por los vehículos de los modelos restantes, por lo que el balanceo de cargas del modelo 4 es mejor que el balanceo de los otros modelos debido a que uno de los vehículos del modelo 4 sólo hace uso del 87.5% de su capacidad de carga mientras que los 3 vehículos restantes del modelo 4 utilizan toda su capacidad de carga de tal forma que sólo se necesitan 4 vehículos para satisfacer la demanda de todos los clientes. Podemos concluir que a pesar de que el modelo 4 recorre una distancia mayor que los modelos 2 y 3 necesita un menor número de vehículos para satisfacer las demandas de todos los clientes. Por lo tanto, el modelo 4 es una buena solución para el ejemplar de 30 ciudades.

Modelo	\bar{x}	σ	CV
1	31	9.01	29.06
2	31	5.76	18.58
3	31	5.87	18.93
4	38.75	2.86	7.38

Tabla 5.33: Variación de las demandas de las soluciones de E_{30} .

Conclusiones

Todavía existen un gran número de problemas de optimización combinatoria que requieren el uso de un enfoque heurístico para la búsqueda de soluciones, sobre todo para ejemplares muy grandes del problema, ya que estamos limitados por los recursos o por el tiempo requeridos para solucionar el problema. El Problema de Rutas de Vehículos es uno de estos problemas y se utilizó la metaheurística de Optimización por Colonias de Hormigas para atacar el problema en este trabajo. Los resultados obtenidos nos lleva a los siguientes comentarios generales:

1. El método de optimización Colonias de Hormigas Múltiples genera mejores soluciones que el método de optimización Colonia de Hormigas Única, ya que utilizar diferentes rastros de feromonas permiten separar las mejores rutas para cada vehículo, favoreciendo la construcción de múltiples rutas de alta calidad simultáneamente.
2. La heurística 2-Opt genera buenas soluciones, pero está limitada a las mejores soluciones encontradas por los métodos de optimización sin estrategias de mejora. Además, su tiempo de ejecución va incrementando conforme el tamaño de la ruta que se quiere mejorar aumenta, por lo que no es eficiente para rutas de gran tamaño.
3. La longitud de la lista de candidatos afecta en la calidad de solución. Las listas de candidatos de tamaño $n/4$ generaron mejores soluciones que las listas de candidatos de tamaño $n/2$, por lo que al tener menos candidatos se puede seleccionar al cliente que más favorezca a la ruta sin desperdiciar recursos.
4. Las mejores soluciones para los ejemplares E_{20} y E_{30} en este trabajo recorren una mayor distancia total que dos de los modelos del trabajo del licenciado Estudillo, respectivamente. Sin embargo, se utilizan una menor cantidad de vehículos para atender las demandas de todos los clientes, a diferencia de los modelos del trabajo del licenciado Estudillo que utilizaba vehículos para atender a uno o dos clientes. Podemos concluir que a pesar de que se viaje una mayor distancia, se pueden reducir los costos asociados a los vehículos, por lo que las soluciones encontradas en este trabajo son competitivas con las soluciones encontradas en el trabajo del licenciado Estudillo.

5. EXPERIMENTOS Y RESULTADOS

5. Algunos trabajos a futuro que pueden desarrollarse para mejorar los resultados obtenidos son:

- Agregar un castigo a los vehículos al elegir un cliente que se encuentra a una distancia muy alejada, de tal forma que se evita desperdiciar recursos en clientes que no favorezcan la ruta.
- Mejorar el balanceo de cargas de manera que se utilice el menor número de vehículos y que cada uno de ellos atienda una demanda total equitativa.
- Variar el parámetro q_0 durante la ejecución del programa para mejorar la capacidad del programa no sólo para seleccionar una variedad de rutas, sino también para concentrarse en las rutas con mayores valores de feromona.

Apéndice A

Teoría de Gráficas

En este apéndice se describen las definiciones básicas sobre Teoría de Gráficas, el material está basado en los libros *Graphs* de Berge [21], *Discrete Mathematics with Graph Theory and Combinatorics* de Veerarajan [22] y *Discrete and Combinatorial Mathematics: An Applied Introduction* de Grimaldi [23].

A.1. Conceptos básicos

Definición A.1 Una **gráfica no dirigida** G es una tupla $G = (V, E)$ donde V es un conjunto no vacío de **vértices** y E es un conjunto de pares no ordenados de vértices llamados **aristas**, tales como $(u, v) \in E$. El número de vértices es el **orden** de G y el número de aristas es el **tamaño** de G . En este trabajo sólo se consideran conjuntos finitos tanto de vértices como de aristas.

Definición A.2 Sea $G = (V, E)$ una gráfica no dirigida. Decimos que una gráfica no dirigida $G' = (V', E')$ es una **subgráfica** de G si y sólo si $V' \subseteq V$ y $E' \subseteq E$.

Definición A.3 Sea $G = (V, E)$ una gráfica no dirigida y S un subconjunto no vacío de V . La **gráfica inducida** por S , denotada por $G[S]$, es la subgráfica de G que tiene a S como conjunto de vértices y para cualesquiera $u, v \in S$, (u, v) es una arista de $G[S]$ si y sólo si (u, v) es una arista de G .

Definición A.4 Sea $G = (V, E)$ una gráfica no dirigida. Decimos que G tiene un **lazo** si existe un vértice $v \in V$ tal que $(v, v) \in E$, es decir, si existe una arista en E que une a un vértice consigo mismo.

Definición A.5 Sea $G = (V, E)$ una gráfica no dirigida. Decimos que G tiene una **multiarista** si existen dos vértices $u, v \in V$ tales que están unidos por un número finito de aristas.

Definición A.6 Sea $G = (V, E)$ una gráfica no dirigida. Decimos que G es una **gráfica simple** si y sólo si G no tiene multiaristas ni lazos.

Definición A.7 Sea $G = (V, E)$ una gráfica. Si (u, v) es una arista de E , decimos que u y v son **vértices adyacentes** y que e **incide** en u y en v .

Definición A.8 Sea $G = (V, E)$ una gráfica. Decimos que dos vértices $u, v \in V$ son **vecinos** si y sólo si u y v son vértices adyacentes.

Definición A.9 Sean $G = (V, E)$ una gráfica y v un vértice de V . La **vecindad** de v en G es el conjunto $\{u \in V \mid u \text{ es vecino de } v\}$ y se denota $N_G(v)$.

Definición A.10 Sean $G = (V, E)$ una gráfica y v un vértice de V . El **grado** de v en G se denota $deg_G(v)$ y es el número de vecinos que tiene:

$$deg_G(v) = |N_G(v)|.$$

Definición A.11 Sea $G = (V, E)$ una gráfica. El **grado mínimo** de G se denota $\delta(G)$ y se define como:

$$\delta(G) = \min\{deg_G(v) \mid v \in V\}.$$

Definición A.12 Sea $G = (V, E)$ una gráfica. El **grado máximo** de G se denota $\Delta(G)$ y se define como:

$$\Delta(G) = \max\{deg_G(v) \mid v \in V\}.$$

Definición A.13 Sea $G = (V, E)$ una gráfica. Un **camino** en G es una sucesión alterna de vértices y aristas $C = (v_0, e_{01}, v_1, \dots, e_{(n-1)n}, v_n)$, en donde $e_{ij} = (v_i, v_j)$ para cada $i, j = 0, 1, 2, \dots, n$. Si $v_0 = v_n$ se dice que C es un **camino cerrado**.

Definición A.14 Sea $G = (V, E)$ una gráfica. Una **trayectoria** en G es un camino que no repite vértices, es decir, una sucesión alterna de vértices y aristas $T = (v_0, e_{01}, v_1, \dots, e_{(n-1)n}, v_n)$, en donde $v_i \neq v_j$ para cada $i, j = 0, 1, 2, \dots, n$. Si T también es un camino cerrado y no se repite ningún otro vértice de la sucesión, entonces se dice que T es un **ciclo**.

Definición A.15 Sea $G = (V, E)$ una gráfica. Un **paseo** en G es un camino que no repite aristas, es decir, una sucesión alterna de vértices y aristas $P = (v_0, e_0, v_1, e_1, \dots, e_{n-1}, v_n)$, en donde $e_i \neq e_j$ para cada $i, j = 0, 1, 2, \dots, n$. Si P también es un camino cerrado, se dice que P es un **circuito**.

Definición A.16 Sean $G = (V, E)$ una gráfica y u, v dos vértices de V . Un **uv -camino** es un camino en G de u a v .

Definición A.17 Sean $G = (V, E)$ una gráfica y C un camino en G . La **longitud** de C en G es igual al número de aristas que posee tal camino y se denota $l_G(C)$.

Definición A.18 Sea $G = (V, E)$ una gráfica. Decimos que G es **conexa** si y sólo si para cualquier par de vértices $u, v \in V$, existe un uv -camino en G .

Definición A.19 Sea $G = (V, E, W)$ una gráfica de orden n , tamaño m y $W = \{w_1, w_2, \dots, w_m\}$ es el conjunto de pesos asociados a cada arista, donde $w : E \rightarrow \mathbb{R}$ es una función de asignación de pesos. Decimos que G es una **gráfica ponderada** si y sólo si para cualquier arista $e_{ij} \in E$, su peso $w(e_{ij}) \in W$.

Definición A.20 Sea $G = (V, E)$ una gráfica de orden n . Decimos que G es una **gráfica completa** si y sólo si para todo par de vértices $u, v \in V$ con $u \neq v$ se tiene que $(u, v) \in E$. Se denota K_n y su tamaño es $\binom{n}{2} = \frac{n(n-1)}{2}$.

A.2. Notación para gráficas

A continuación, se presenta una tabla con la notación que es utilizada durante el desarrollo de este trabajo.

Notación	Significado
V	Conjunto de vértices de G
E	Conjunto de aristas de G
$G[S]$	Subgráfica de G inducida por S
$N_G(v)$	Vecindad del vértice v en G
$deg_G(v)$	Grado del vértice v en G
$\delta(G)$	Grado mínimo de G
$\Delta(G)$	Grado máximo de G
C	Camino en G
C_c	Camino cerrado en G
T	Trayectoria en G
T_c	Ciclo en G
P	Paseo en G
P_c	Circuito en G
$l_G(C)$	Longitud de C en G
$w(e)$	Peso de la arista e en G
K_n	Gráfica completa de orden n

Tabla A.1: Notación para gráficas

Apéndice B

Complejidad Computacional

En este apéndice se describen las definiciones básicas sobre Complejidad Computacional, el material está basado en el libro *Computers and Intractability: A Guide to the Theory of NP-Completeness* de Garey y Johnson [24].

B.1. Conceptos básicos

Definición B.1 Un **problema** es una pregunta general a responder y está descrito por los siguientes componentes:

- Una descripción de cada uno de sus **parámetros** y las restricciones de estos.
- Un enunciado que especifique qué propiedades debe satisfacer la **respuesta** o **solución**.

Definición B.2 Sea Π un problema. Un **ejemplar** de Π se obtiene especificando valores particulares para todos los parámetros de Π .

Definición B.3 Sea Π un problema. El **conjunto** de **todos** los **ejemplares** de Π se denota D_{Π} .

Definición B.4 Sea Π un problema. Decimos que Π es un **problema de decisión** si y sólo si el conjunto de respuestas de Π es $\{YES, NO\}$.

Definición B.5 Sea Π un problema. El **conjunto** de **todos** los **ejemplares** de Π para los cuales la solución es *YES* se denota Y_{Π} .

Definición B.6 Un **algoritmo** es un conjunto de pasos para resolver un problema.

Definición B.7 Sea Π un problema. Decimos que un algoritmo es **determinista** si y sólo si ese algoritmo se puede aplicar a cualquier ejemplar E de Π y siempre produce la misma solución para E .

Definición B.8 Sea Π un problema de decisión. Un **algoritmo no determinista** para Π consiste de dos fases separadas, la primera se llama **fase adivinadora** y la

segunda **fase verificadora**. Dado un ejemplar E de Π , la fase adivinadora se encarga de “adivinar” un candidato a solución S para E . A continuación, tanto E como S son pasados como entradas a la fase verificadora, que procede a computar de manera determinista si S es o no solución de E ; es decir, determina si S satisface las condiciones del problema Π . En caso afirmativo, el algoritmo termina su ejecución y la salida es *YES*. En caso contrario, el algoritmo termina su ejecución y la salida es *NO*, o bien, el algoritmo nunca termina su ejecución.

Definición B.9 Sea \mathcal{A} un algoritmo no determinista. \mathcal{A} es de **tiempo polinomial** si y sólo si tanto su fase adivinadora como su fase verificadora están acotadas por un polinomio con respecto al tamaño de la entrada $E \in D_\Pi$.

Definición B.10 Sean Π_1 y Π_2 dos problemas de decisión. Una **transformación polinomial** del problema Π_1 al problema Π_2 es una función $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$ que satisface las siguientes dos condiciones:

1. f es computable por un algoritmo de tiempo polinomial.
2. Para todo ejemplar $E \in D_{\Pi_1}$, $E \in Y_{\Pi_1}$ si y sólo si $f(E) \in Y_{\Pi_2}$.

Si existe una transformación polinomial de Π_1 a Π_2 , escribimos $\Pi_1 \propto_p \Pi_2$ y decimos que Π_1 es polinomialmente reducible a Π_2 .

Lema B.1 Sean Π_1 , Π_2 y Π_3 tres problemas de decisión. Si $\Pi_1 \propto_p \Pi_2$ y $\Pi_2 \propto_p \Pi_3$, entonces $\Pi_1 \propto_p \Pi_3$.

Demostración.

Sean Π_1 , Π_2 y Π_3 tres problemas de decisión.

Supongamos que $\Pi_1 \propto_p \Pi_2$ y $\Pi_2 \propto_p \Pi_3$

Por demostrar: $\Pi_1 \propto_p \Pi_3$; existe una transformación polinomial de Π_1 a Π_3 .

Como $\Pi_1 \propto_p \Pi_2$, existe una transformación polinomial f_1 de Π_1 a Π_2 . Análogamente, existe una transformación polinomial f_2 de Π_2 a Π_3 , dado que $\Pi_2 \propto_p \Pi_3$. Notemos que $f_2 \circ f_1$ es una transformación polinomial de Π_1 a Π_3 . Así, $\Pi_1 \propto_p \Pi_3$.

Ergo, si $\Pi_1 \propto_p \Pi_2$ y $\Pi_2 \propto_p \Pi_3$, entonces $\Pi_1 \propto_p \Pi_3$, para cualesquiera problemas de decisión Π_1 , Π_2 y Π_3 .

Q. E. D.

Definición B.11 Sea Π un problema de decisión. Π pertenece a la clase de complejidad **P** si y sólo si existe un algoritmo determinista de tiempo polinomial que resuelve Π .

Definición B.12 Sea Π un problema de decisión. Π es **NP** si y sólo si existe un algoritmo no determinista de tiempo polinomial para Π .

Definición B.13 Sea Π un problema de decisión. Decimos que Π es **NP-Completo** si y sólo si $\Pi \in \mathbf{NP}$; y para todo $\Pi' \in \mathbf{NP}$, $\Pi \propto_p \Pi'$.

Lema B.2 Sea Π_1 un problema de decisión. Si $\Pi \in \mathbf{NP}$ y existe un problema de decisión Π_2 **NP-Completo** tal que $\Pi_2 \propto_p \Pi_1$, entonces Π_1 es **NP-Completo**.

Demostración.

Sea Π_1 un problema de decisión.

Supongamos que $\Pi_1 \in \mathbf{NP}$ y que existe un problema de decisión Π_2 **NP-Completo** tal que $\Pi_2 \propto_p \Pi_1$.

Por demostrar: Π_1 es **NP-Completo**.

$\Pi \in \mathbf{NP}$ por hipótesis.

Como Π_2 es **NP-Completo**, se tiene que $\Pi' \propto_p \Pi_2$, para todo $\Pi' \in \mathbf{NP}$. Lo que implica, para todo $\Pi' \in \mathbf{NP}$, $\Pi' \propto_p \Pi_2$ y $\Pi_2 \propto_p \Pi_1$. Entonces, para todo $\Pi' \in \mathbf{NP}$, $\Pi' \propto_p \Pi_1$; por Lema B.1.

Por lo tanto, Π_1 es **NP-Completo**. Ergo, para todo problema de decisión Π_1 , si ocurre que $\Pi_1 \in \mathbf{NP}$ y existe un problema de decisión Π_2 **NP-Completo** tal que $\Pi_2 \propto_p \Pi_1$, entonces Π_1 es **NP-Completo**.

Q. E. D.

Definición B.14 Un problema de optimización combinatoria Π es un problema de **minimización** o un problema de **maximización** y consta de las siguientes tres partes:

1. Un conjunto D_Π de ejemplares.
2. Para cada ejemplar $E \in D_\Pi$, un conjunto finito $S_\Pi(E)$ de soluciones candidatas para E .
3. Una función objetivo f_Π que asigna a cada ejemplar $E \in D_\Pi$ y a cada solución candidata $\sigma \in S_\Pi(E)$ un número racional positivo $f_\Pi(E, \sigma)$ llamado el valor de solución para σ .

Definición B.15 Sea Π un problema de optimización combinatoria. Si Π es un problema de **minimización**, entonces una **solución óptima** para un ejemplar $E \in D_\Pi$ es una solución candidata $\sigma^* \in S_\Pi(E)$ tal que, para todo $\sigma \in S_\Pi(E)$, $f_\Pi(E, \sigma^*) \leq f_\Pi(E, \sigma)$. Se denota $OPT(E)$ el valor $f_\Pi(E, \sigma^*)$ de una solución óptima para E .

Definición B.16 Sea Π un problema de optimización combinatoria. Si Π es un problema de **maximización**, entonces una **solución óptima** para un ejemplar $E \in D_\Pi$ es una solución candidata $\sigma^* \in S_\Pi(E)$ tal que, para todo $\sigma \in S_\Pi(E)$, $f_\Pi(E, \sigma^*) \geq f_\Pi(E, \sigma)$.

Definición B.17 Sea Π un problema de optimización combinatoria. Un algoritmo \mathcal{A} es un **algoritmo de aproximación** para Π si, dado cualquier ejemplar $E \in D_\Pi$ encuentra una solución candidata $\sigma \in S_\Pi(E)$ para E .

Definición B.18 Una **heurística** es una técnica para encontrar soluciones aproximadas a problemas de forma rápida cuando los métodos clásicos son demasiados lentos o no logran encontrar una solución.

Definición B.19 Sea Π un problema de optimización combinatoria. Un **algoritmo de búsqueda local** \mathcal{A} para Π si, dado cualquier ejemplar $E \in D_\Pi$ encuentra un

conjunto finito $S_{\Pi}(\sigma_0)$ de soluciones candidatas a partir de una solución candidata inicial $\sigma_0 \in S_{\Pi}(E)$ para E con las siguientes características:

- $k \in \mathbb{N}$ y $\sigma_{i+1} \in S(\sigma_i) \forall i, i = 0, 1, \dots, k - 1$.
- **Problema de minimización:**
 - $f_{\Pi}(E, \sigma_{i+1}) < f_{\Pi}(E, \sigma_i), \forall i, i = 0, 1, \dots, k - 1$.
 - σ_k es un óptimo local: $f_{\Pi}(E, \sigma_k) < f_{\Pi}(E, \sigma), \forall \sigma \in S(\sigma_k)$.
- **Problema de maximización:**
 - $f_{\Pi}(E, \sigma_{i+1}) > f_{\Pi}(E, \sigma_i), \forall i, i = 0, 1, \dots, k - 1$.
 - σ_k es un óptimo local: $f_{\Pi}(E, \sigma_k) > f_{\Pi}(E, \sigma), \forall \sigma \in S(\sigma_k)$.

Finalmente, selecciona la mejor solución candidata $\sigma \in S_{\Pi}(\sigma_i)$.

Apéndice C

Heurística 2-opt

La heurística 2-opt es un algoritmo de búsqueda local que parte de una solución inicial y busca iterativamente oportunidades de mejora en la vecindad de dicha solución. Fue propuesta por Croes en 1958 [25] y fue implementado por Lin para resolver el Problema del Agente Viajero en 1965 [26].

Un intercambio 2-opt, consiste en eliminar dos aristas del ciclo y reconectar los dos caminos resultantes para generar un nuevo ciclo, reemplazando las aristas eliminadas por otras dos aristas. Sólo hay una forma de volver a conectar los caminos para obtener un ciclo con menor peso [27].

A continuación se describe brevemente el proceso de la heurística 2-opt:

1. Sea T_c el ciclo inicial.
2. Se seleccionan dos aristas del ciclo T_c , sean (i, j) y (k, l) .
3. Se realiza un intercambio 2-opt:
 - 2.1 Se eliminan las aristas (i, j) y (k, l) del ciclo T_c .
 - 2.2 Se reemplazan las aristas eliminadas por otras dos aristas (i, k) y (j, l) , o (k, i) y (l, j) . Hay que tener en cuenta que sólo uno de estos reemplazos genera un nuevo ciclo, mientras que el otro divide el ciclo T_c en dos ciclos.
 - 2.3 Se verifica que el reemplazo de aristas sea un intercambio 2-opt válido. El reemplazo va a ser un intercambio 2-opt válido si genera un nuevo ciclo T'_c .
4. Se compara el peso de los ciclos T_c y T'_c :
 - 4.1 Si el ciclo T'_c tiene un peso mayor o igual al del ciclo T_c , entonces se deshace el intercambio 2-opt y se repite el paso 2.
 - 4.2 Si el ciclo T'_c tiene un peso menor al del ciclo T_c , entonces se mantiene el intercambio 2-opt, se actualiza T_c con T'_c y se repite el paso 2.
5. Si no se puede encontrar ningún intercambio 2-opt, entonces se detiene. El ciclo resultante se denomina 2-óptimo.

Un ciclo es 2-óptimo si no se puede generar un ciclo de menor peso realizando un intercambio 2-opt. Verificar si existe un intercambio 2-opt requiere un tiempo $O(n^2)$, ya que tiene que examinar cada par de aristas del ciclo [27], donde n es el número de aristas. Esta técnica puede aplicarse al Problema del Agente Viajero y al VRP, así como a muchos problemas relacionados, para mejorar las soluciones encontradas.

Ejemplo

En la figura C.1a se muestra el ciclo inicial T_c , para empezar con el proceso de la heurística 2-opt se seleccionan las aristas (a, g) y (b, c) del ciclo T_c y posteriormente se eliminan del ciclo T_c como se ilustra en la figura C.1b.

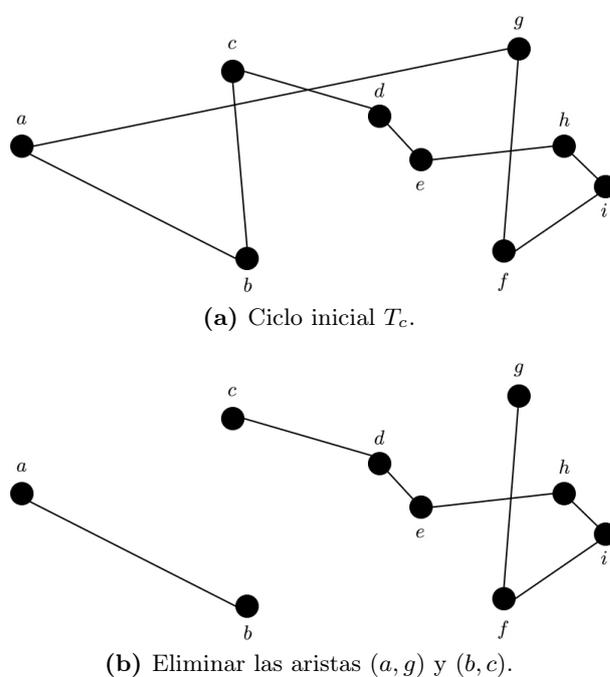
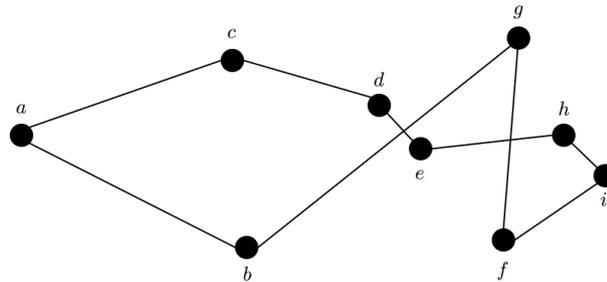
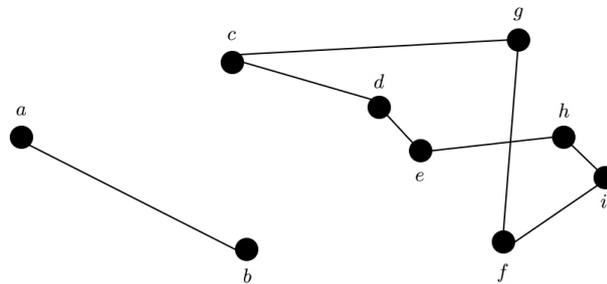


Figura C.1: Inicio del proceso de la heurística 2-opt.

A continuación se reemplazan las aristas eliminadas por otras dos aristas (a, b) y (g, c) , o (a, c) y (b, g) . El primer reemplazo genera un nuevo ciclo T'_c como se muestra en la figura C.2a, mientras que el segundo reemplazo divide a T_c en dos ciclos T_{c1} y T_{c2} como se ilustra en la figura C.2b.



(a) Intercambiar las aristas (a, g) y (b, c) por las aristas (a, c) y (b, g) .



(b) Intercambiar las aristas (a, g) y (b, c) por las aristas (a, b) y (g, c) .

Figura C.2: Posibles intercambios de las aristas (a, g) y (b, c) .

El reemplazo que se muestra en la figura C.2a es un intercambio 2-opt válido, se compara el peso de los ciclos y como el ciclo T'_c tiene un peso menor que el ciclo T_c el intercambio se mantiene. Posteriormente, se actualiza el ciclo T_c con el ciclo T'_c y se procede a seleccionar otro par de aristas, (b, g) y (d, e) . Estas aristas se eliminan del ciclo T_c como se muestra en la figura C.3.

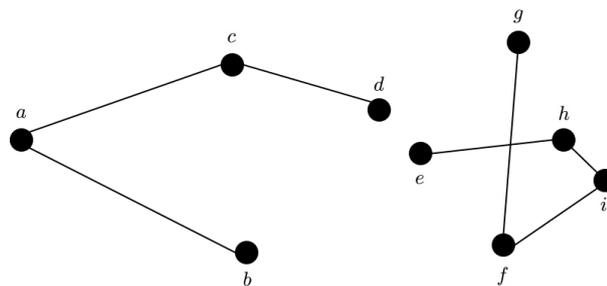


Figura C.3: Eliminar las aristas (b, g) y (d, e) .

A continuación se reemplazan las aristas eliminadas por otras dos aristas (b, d) y (g, e) , o (b, e) y (g, d) . Sólo la segunda sustitución genera un nuevo ciclo T'_c , por lo que es un intercambio 2-opt válido y como el ciclo T'_c tiene un peso menor que el ciclo T_c el intercambio se mantiene como se ilustra en la figura C.4.

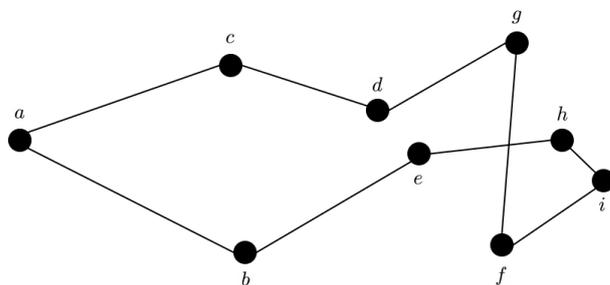


Figura C.4: Intercambiar las aristas (b, g) y (d, e) por las aristas (b, e) y (g, d) .

El ciclo T_c se actualiza con el ciclo T'_c y se procede a seleccionar el siguiente par de aristas, (e, h) y (g, f) . A continuación, se eliminan dichas aristas del ciclo T_c como se muestra en la figura C.5.

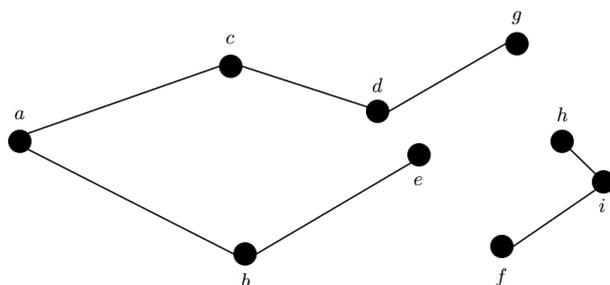


Figura C.5: Eliminar las aristas (e, h) y (g, f) .

A continuación se reemplazan las aristas eliminadas por otras dos aristas (e, f) y (h, g) , o (e, g) y (h, f) . Sólo el primer reemplazo genera un nuevo ciclo T'_c , por lo que es un intercambio 2-opt válido y como el ciclo T'_c tiene un peso menor que el ciclo T_c el intercambio se mantiene como se ilustra en la figura C.6, se actualiza el ciclo T_c con el ciclo T'_c y el proceso se detiene porque el ciclo T_c ya no puede mejorarse con intercambios 2-opt. Por lo que podemos decir que el ciclo T_c es un ciclo 2-óptimo.

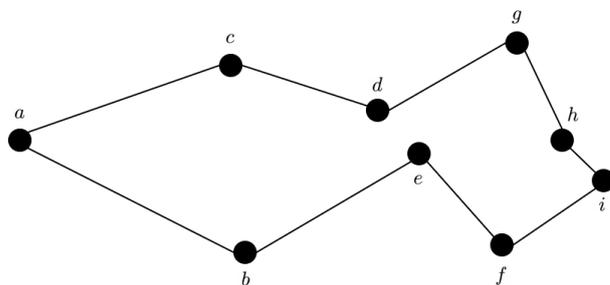


Figura C.6: Intercambiar las aristas (e, h) y (g, f) por las aristas (e, f) y (h, g) .

Apéndice D

Recursos de ejecución

En este apéndice se muestra el promedio de los recursos utilizados por cada método para los ejemplares E_{20} y E_{30} . Los detalles de la ejecución del programa de Estudillo no están disponibles. El criterio de término en ambos trabajos fue el número máximo de iteraciones que puede ejecutar el programa.

- **Recursos utilizados por cada método para E_{20}**

Método	No. de ejecuciones	Tiempo de ejecución	Criterio de término
CHU sin estrategias	54	467 s	1,500
CHU y heurística 2-opt	54	543 s	1,500
CHU y LC $n/4$	44	381 s	1,000
CHU y LC $n/2$	46	442 s	1,000
CHM sin estrategias	52	638 s	1,500
CHM y heurística 2-opt	52	713 s	1,500
CHM y LC $n/4$	46	572 s	1,000
CHM y LC $n/2$	48	651 s	1,000
Modelo 1	-	290 s	2,000
Modelo 2	-	607 s	2,000
Modelo 3	-	176 s	2,000

Tabla D.1: Promedio de los recursos utilizados por cada método para E_{20} .

D. RECURSOS DE EJECUCIÓN

- Recursos utilizados por cada método para E_{30}

Método	No. de ejecuciones	Tiempo de ejecución	Criterio de término
CHU sin estrategias	56	1,036 s	1,500
CHU y heurística 2-opt	56	1,259 s	1,500
CHU y LC $n/4$	44	998 s	1,000
CHU y LC $n/2$	48	1,147 s	1,000
CHM sin estrategias	51	1,306 s	1,500
CHM y heurística 2-opt	51	1,518 s	1,500
CHM y LC $n/4$	42	1,273 s	1,000
CHM y LC $n/2$	46	1,480 s	1,000
Modelo 1	-	979 s	2,000
Modelo 2	-	1,623 s	2,000
Modelo 3	-	936 s	2,000

Tabla D.2: Promedio de los recursos utilizados por cada método para E_{30} .

El número de ejecuciones promedio varían, ya que los métodos que utilizan las listas de candidatos seleccionan primero los clientes que se encuentran más cerca de su ubicación actual y al final a los clientes que faltan por visitar, por lo que las soluciones que generan recorren distancias más pequeñas que las generadas por los otros métodos.

Apéndice E

Detalles de las ciudades

En este apéndice se muestra la lista de ciudades que se utilizaron en los experimentos y la tabla de distancias en kilómetros entre cada par de ciudades.

E.1. Lista de Ciudades

En esta sección se muestra la lista de ciudades que se utilizaron en los experimentos:

1. Acapulco, Guerrero
2. Aguascalientes, Aguascalientes
3. Campeche, Campeche
4. Cancún, Quintana Roo
5. Chetumal, Quintana Roo
6. Chihuahua, Chihuahua
7. Chilpancingo de los Bravo, Guerrero
8. Ciudad Cuauhtémoc, Chiapas
9. Ciudad de México, CDMX
10. Ciudad Juárez, Chihuahua
11. Ciudad Victoria, Tamaulipas
12. Colima, Colima
13. Cuernavaca, Morelos
14. Culiacán Rosales, Sinaloa
15. Ensenada, Baja California

E. DETALLES DE LAS CIUDADES

16. Guadalajara, Jalisco
17. Guanajuato, Guanajuato
18. Hermosillo, Sonora
19. Heroica Matamoros, Tamaulipas
20. La Paz, Baja California Sur
21. León, Guanajuato
22. Manzanillo, Colima
23. Mazatlán, Sinaloa
24. Mexicali, Baja California
25. Mérida, Yucatán
26. Monterrey, Nuevo León
27. Morelia, Michoacán
28. Nogales, Sonora
29. Nuevo Laredo, Tamaulipas
30. Oaxaca de Juárez, Oaxaca
31. Pachuca, Hidalgo
32. Piedras Negras, Coahuila de Zaragoza
33. Puebla de Zaragoza, Puebla
34. Reynosa, Tamaulipas
35. Salina Cruz, Oaxaca
36. Saltillo, Coahuila de Zaragoza
37. San Luis Potosí, San Luis Potosí
38. Santiago de Querétaro, Querétaro
39. Tampico, Tamaulipas
40. Tapachula, Chiapas
41. Tepic, Nayarit
42. Tijuana, Baja California
43. Tlaxcala, Tlaxcala

44. Toluca de Lerdo, Estado de México
45. Torreón, Coahuila de Zaragoza
46. Tuxtla Gutiérrez, Chiapas
47. Veracruz, Veracruz
48. Victoria de Durango, Durango
49. Villahermosa, Tabasco
50. Xalapa-Enríquez, Veracruz
51. Zacatecas, Zacatecas

E.2. Tabla de distancias

En esta sección se muestra la tabla de distancias en kilómetros entre cada par de ciudades. El número en negrita indica el número de la ciudad dado en la lista de ciudades de la sección [E.1.](#)

1										
2	868									
3	1397	1613								
4	1873	2089	477							
5	1589	1804	418	384						
6	1827	944	2578	3054	2770					
7	105	766	1302	1778	1494	1710				
8	1196	1537	659	1129	785	2498	1218			
9	379	492	1140	1616	1332	1442	275	1067		
10	2194	1311	2944	3420	3136	360	2090	2872	1815	
11	1064	503	1523	1999	1715	1088	960	1451	684	1456
12	670	427	1864	2340	2056	1375	752	1791	735	1742
13	291	578	1178	1653	1370	1521	187	1105	87	1906
14	1584	862	2357	2833	2549	1080	1480	2284	1228	1440
15	3196	2474	3969	4445	4161	1419	3092	3896	2840	1237
16	776	221	1674	2150	1866	1182	797	1601	536	1549
17	732	180	1483	1959	1675	1141	628	1410	353	1408
18	2252	1530	3025	3501	3217	891	2148	2952	1896	747
19	1313	821	1772	2248	1964	1119	1208	1699	932	1331
20	4269	3547	5042	5518	5234	2563	4165	4969	3919	2418
21	752	127	1503	1979	1695	1088	648	1430	373	1455
22	687	532	1965	2441	2157	1476	1085	1892	836	1844
23	1253	656	2151	2627	2343	875	1274	2078	1022	1234
24	2951	2229	3724	4200	3916	1182	2847	3651	2595	1001
25	1574	1790	179	303	383	2750	1470	835	1309	3119
26	1283	569	1794	2270	1986	811	1179	1721	929	1163
27	597	315	1437	1913	1630	1276	561	1365	300	1643
28	2534	1642	3307	3751	3499	694	2421	3203	2151	549
29	1491	776	2018	2494	2210	1018	1387	1945	1198	946
	1	2	3	4	5	6	7	8	9	10

30	658	941	1071	1666	1334	1902	553	761	460	2270
31	445	477	1145	1620	1337	1421	350	1072	66	1806
32	1630	910	2192	2688	2573	810	1526	2119	1258	772
33	384	614	1020	1496	1212	1558	280	947	133	1943
34	1319	784	1779	2254	1972	1026	1215	1706	939	1288
35	640	1219	862	1337	1054	2179	899	551	738	2547
36	1216	481	1872	2351	2158	731	1111	1802	842	1098
37	790	165	1541	2016	1733	1020	686	1468	412	1387
38	591	287	1341	1817	1533	1248	486	1268	212	1615
39	819	602	1282	1758	1474	1312	718	1209	442	1680
40	1084	1629	948	1418	1135	2590	1230	165	1148	2958
41	977	426	1875	2351	2067	1088	998	1825	752	1448
42	3125	2403	3898	4374	4090	1383	3021	3823	2770	1201
43	418	592	1050	1526	1242	1535	314	977	118	1920
44	388	474	1201	1677	1394	1419	281	1129	64	1803
45	1359	501	2135	2598	2314	479	1255	2049	1006	838
46	967	1309	627	1103	753	2269	989	226	829	2638
47	667	852	850	1326	1042	1750	556	777	395	2118
48	1258	400	2034	2510	2226	630	1153	1961	905	989
49	1014	1229	384	859	577	2190	909	404	748	2558
50	559	744	956	1432	1148	1688	455	883	283	2073
51	975	118	1725	2227	1944	836	871	1652	623	1204
	1	2	3	4	5	6	7	8	9	10

11										
12	858									
13	774	798								
14	1249	851	1292							
15	2834	2463	2904	1634						
16	665	196	609	690	2293					
17	504	469	440	961	2564	275				
18	1931	1519	1960	690	931	1766	1623			
19	320	1180	1120	1387	2561	986	821	2021		
20	3948	3536	3977	2707	1373	3372	3641	2013	3677	
	11	12	13	14	15	16	17	18	19	20

E. DETALLES DE LAS CIUDADES

21	513	416	460	908	2511	222	57	1575	835	3593
22	959	103	900	953	2556	293	567	1619	1279	3638
23	1043	645	1086	218	1820	482	750	885	1156	2903
24	2594	2218	2659	1389	240	2055	2322	695	2324	1356
25	1694	2034	1347	2530	4133	1844	1652	3196	1943	5212
26	284	998	991	1047	2391	804	680	1707	311	3515
27	650	449	373	974	2578	289	176	1641	969	3659
28	1780	1801	2233	972	867	1638	1833	278	1823	1949
29	508	1206	1199	1254	2174	1012	888	1941	342	3288
30	1022	1185	498	1683	3284	995	803	2348	1271	4366
31	637	721	162	1219	2820	531	339	1884	886	3902
32	682	1339	1338	1314	1999	1146	1027	1519	516	3114
33	689	857	147	1356	2957	668	476	2020	938	4039
34	326	1186	1027	1262	2516	992	827	1929	89	3630
35	1171	1463	775	1958	3562	1273	1080	2625	1420	4643
36	365	910	923	967	2156	717	612	1627	408	3270
37	335	524	498	938	2445	330	187	1604	655	3623
38	531	539	298	1035	2638	349	148	1702	851	3720
39	237	960	530	1368	3081	767	602	2035	490	4131
40	1534	1873	1186	2369	3972	1683	1491	3036	1783	5054
41	866	369	810	484	2087	206	474	1151	1186	3169
42	2798	2392	2833	1563	105	2228	2497	869	2524	1472
43	647	836	190	1334	2934	646	453	1998	896	4016
44	710	662	94	1160	2761	472	336	1824	996	3842
45	612	931	1067	697	1894	737	693	1364	655	3009
46	1213	1533	866	2051	3652	1363	1171	2715	1462	4734
47	675	1096	433	1544	3195	906	714	2259	924	4277
48	801	830	991	463	2066	636	592	1129	908	3148
49	1134	1473	786	1971	3572	1283	1091	2636	1383	4654
50	708	988	332	1486	3087	796	606	2151	953	4169
51	518	513	683	754	2357	321	309	1420	787	3459
	11	12	13	14	15	16	17	18	19	20

21											
22	514										
23	697	744									
24	2270	2321	1583								
25	1675	2137	2326	3898							
26	695	1065	842	2062	1971						
27	199	582	770	2343	1614	830					
28	1781	1904	1166	584	3484	1501	1921				
29	903	1309	1050	1938	2195	219	1035	1509			
30	826	1288	1477	3049	1367	1293	757	2630	1538		
31	362	820	1013	2585	1321	893	288	2163	1101	464	
32	1042	1442	1025	1763	2558	393	1174	1334	175	1701	
33	499	961	1150	2722	1197	1030	430	2282	1205	340	
34	842	1289	1058	2280	1955	218	974	1774	258	1276	
35	1104	1322	1754	3326	1038	1442	1035	2905	1666	279	
36	620	1013	763	1924	2143	87	759	1478	295	1286	
37	180	623	734	2306	1717	515	334	1766	724	860	
38	171	638	831	2404	1518	711	189	1984	920	661	
39	617	1063	1164	2736	1458	508	687	2020	757	779	
40	1514	1976	2165	3737	1124	1805	1445	3318	2029	677	
41	421	468	280	1852	2052	1008	489	1433	1217	1195	
42	2444	2495	1758	176	4075	2352	2512	782	2135	3218	
43	476	935	1127	2699	1227	918	407	2260	1163	370	
44	359	761	953	2525	1378	890	234	2106	1099	521	
45	640	1034	493	1662	2287	334	829	1218	542	1429	
46	1194	1656	1844	3417	804	1484	1125	2977	1709	561	
47	737	1199	1388	2960	1026	946	668	2541	1191	444	
48	539	932	258	1814	1831	587	728	1411	795	1328	
49	1114	1576	1765	3337	560	1405	1045	2918	1650	797	
50	629	1091	1280	2852	1133	975	560	2413	1220	434	
51	257	650	550	2029	1928	465	446	1543	674	1045	
	21	22	23	24	25	26	27	28	29	30	

E. DETALLES DE LAS CIUDADES

31											
32	1238										
33	137	1377									
34	892	431	930								
35	768	1839	617	1425							
36	823	434	955	315	1530						
37	397	863	530	661	1139	447					
38	198	1053	340	856	939	643	216				
39	395	902	434	496	936	589	435	592			
40	1153	2216	1027	1789	466	1885	1550	1350	1291		
41	732	1315	864	1192	1474	928	530	549	964	1893	
42	2755	3143	2887	2481	3497	2111	2480	2572	2854	3916	
43	115	1355	40	902	649	939	512	312	405	1068	
44	128	1238	191	1002	800	822	395	196	504	1219	
45	992	603	1098	562	1708	253	578	801	836	2140	
46	832	1885	707	1468	350	1565	1229	1033	971	369	
47	375	1343	274	930	505	1027	773	573	432	797	
48	865	777	997	815	1607	506	477	680	899	2052	
49	778	1802	627	1388	479	1486	1150	950	891	565	
50	268	1314	173	959	611	1092	665	465	461	974	
51	582	808	714	693	1324	385	194	396	616	1743	
	31	32	33	34	35	36	37	38	39	40	

41											
42	2022										
43	846	2875									
44	672	2701	179								
45	701	1897	1079	964							
46	1564	3592	740	890	1817						
47	1111	3135	283	458	1278	547					
48	467	2006	978	863	243	1705	1238				
49	1488	3512	660	811	1731	244	467	1629			
50	999	3027	176	347	1246	653	107	1144	574		
51	554	2297	695	580	393	1422	955	291	1343	847	
	41	42	43	44	45	46	47	48	49	50	

Bibliografía

- [1] M. Dorigo. *Optimization, learning and natural algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992. [3](#)
- [2] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. In M. Dorigo D. Corne and F. Glover, editors, *New Ideas in Optimization*, pages 11–32. McGraw-Hill, 1999. [4](#)
- [3] M. Dorigo, G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999. [4](#)
- [4] P. Tarasewich and P. R. McMullen. Swarm intelligence: power in numbers. *Communications of the ACM*, 45(8):62–67, 2002. [4](#), [19](#)
- [5] J. L. Vázquez Lázaro. Coloreando gráficas con tres metaheurísticas basadas en colonia de hormigas, 2021. [6](#)
- [6] M. Dorigo and T. Stützle. *Ant Colony Optimization*. The MIT Press, 2004. [7](#)
- [7] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. [9](#)
- [8] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Oper Res*, 12(4):568–581, 1964. [9](#)
- [9] J.E. Bell and P. R. McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18:41–48, 2004. [10](#)
- [10] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014. [10](#)
- [11] D. Michel Tavera. Complejidad del problema de rutas de vehículos, 2017. [10](#)
- [12] I. Yusuf, M. S. Babal, and N. Iksan. Applied genetic algorithm for solving rich vrp. *Oper Res*, 28(10):957–991, 2014. [10](#)
- [13] M. A. Mohammed, M. S. Ahmad, and S. A. Mostafa. Using genetic algorithm in implementing capacitated vehicle routing problem. In *2012 International Conference on Computer Information Science (ICCIS)*, volume 1, pages 257–262, 2012. [10](#)

BIBLIOGRAFÍA

- [14] M. Gendreau, A. Hertz, and G. Laporte. A tabu search heuristic for the vehicle routing problem. *Management Science*, 40(10):1276–1290, 1994. [10](#)
- [15] L. H. Osman. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann Oper Res*, 41:421–451, 1993. [10](#)
- [16] N. De Jaegere, M. Defraeyea, and I. Van Nieuwenhuysea. The vehicle routing problem: State of the art classification and review. 2014. [10](#)
- [17] M. Dorigo and L. M. Gambardella. Ant colonies for the traveling salesman problem. *BioSystem*, 43:73–81, 1997. [23](#), [24](#)
- [18] B. Bullnheimer, R. F. Hartl, and C. Strauss. Applying the ant system to the vehicle routing problem. In *2nd International Conference on Metaheuristics - MIC97*, 1997. [26](#)
- [19] B. Bullnheimer, R. F. Hartl, and C. Strauss. An improved ant system algorithm for the vehicle routing problem. *Ann Oper Res*, 89:319–328, 1999. [26](#)
- [20] J. A. Estudillo Carranza. Recocido simulado aplicado a v r p, 2022. [36](#), [52](#), [55](#)
- [21] C. Berge. *Graphs*. North Holland, 1983. [59](#)
- [22] T. Veerarajan. *Discrete Mathematics with Graph Theory and Combinatorics*. McGraw Hill Education, 2006. [59](#)
- [23] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. Pearson Prentice Hall, 1985. [59](#)
- [24] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979. [63](#)
- [25] G. A. Croes. A method for solving traveling salesman problems. *Oper Res*, 6(6):791–812, 1958. [67](#)
- [26] S. Lin. Computer solutions of the travelling salesman problem. *Bell System Technical Journal*, 44(10):2245–2269, 1965. [67](#)
- [27] G.Reinelt. *The Traveling Salesman, Computational Solutions for TSP Applications*. Springer-Verlag, 1994. [67](#), [68](#)

Índice alfabético

- ACO, [3](#), [6](#), [19](#), [28](#)
- Actualización global, [24](#)
- Actualización local, [24](#)
- Algoritmo de búsqueda local, [65](#), [67](#)

- Ciclo, [60](#), [67](#)
- Ciclo 2-óptimo, [68](#)
- Colonia de hormiga única, [27](#), [35](#)
- Colonia de hormigas, [3](#), [28](#)
- Colonias de hormiga múltiples, [35](#)
- Colonias de hormigas múltiples, [25](#)
- CVRP, [10](#), [19](#), [27](#), [35](#)

- Estigmergia, [4](#)
- Estrategias de mejora, [10](#), [25](#), [28](#), [35](#)
- Evaporación, [4](#), [6](#)

- Feromonas, [3](#), [4](#)

- Gráfica completa, [14](#), [61](#)
- Gráfica no dirigida, [14](#), [59](#)
- Gráfica ponderada, [14](#), [60](#)

- Heurística, [3](#), [6](#), [65](#)
- Heurística 2-opt, [10](#), [25](#), [27](#), [29](#), [35](#), [67](#)

- Inteligencia de Enjambres, [3](#)
- Intercambio 2-opt, [25](#), [67](#)
- Intercambios 2-opt, [29](#)

- Listas de Candidatos, [35](#), [40](#)
- Listas de candidatos, [10](#), [26](#), [28](#), [29](#)

- Metaheurística, [3](#), [6](#)

- NP-Difícil, [10](#)

- Optimización por Colonias de Hormigas, [3](#), [10](#)

- Peso, [60](#), [67](#)
- Problema de optimización combinatoria, [10](#), [65](#)
- Problema de Rutas de Vehículos, [9](#)
- Problema del Agente Viajero, [3](#), [9](#), [67](#)

- Rastro de feromonas, [4](#), [6](#)
- Rastros de feromonas, [8](#), [24](#)

- Sistema de Hormigas, [3](#)
- Solución del VRP, [15](#)

- Variantes del VRP, [10](#)
- VRP, [9](#), [68](#)
- VRP asimétrico, [9](#)
- VRP Clásico, [10](#)
- VRP con Flota Heterogénea, [11](#)
- VRP con Flota Mixta, [11](#)
- VRP con Múltiples Depósitos, [12](#)
- VRP con Recolección y Entrega, [12](#)
- VRP con Ventanas de Tiempo, [13](#)
- VRP simétrico, [9](#)