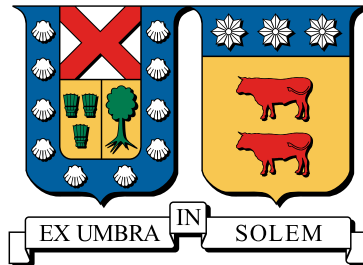


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE INFORMÁTICA  
VALPARAÍSO - CHILE



## **Nuevos óptimos para el Problema de Rutas de Camiones y Remolques con Restricciones de Capacidad usando un algoritmo colaborativo**

Andrés Felipe Navarro Galleguillos

Tesis presentada para la obtención del grado de  
Magíster en Ciencias de la Ingeniería Informática

Tutor: Dr. María Cristina Riff  
Correferente Interno: Dr. Elizabeth Montero  
Evaluador Externo: Dr. Ignacio Araya

13 de agosto de 2026



## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

### 1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

**Tipo de monografía (marcar una opción):** ☐ Memoria o trabajo de título; ☒ Tesis de Postgrado;

**Título del trabajo:** Nuevos óptimos para el Problema de Rutas de Camiones y Remolques con Restricciones de Capacidad usando un algoritmo colaborativo

**Nombre del candidato(a):** Andrés Felipe Navarro Galleguillos

**Carrera / Grado:** Magíster en Ciencias de la Ingeniería Informática

**Campus: ; Departamento:** Departamento de Informática

### 2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, María Cristina Riff Rojas, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

### 3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

### 4.- FIRMAS

**Profesor(a) guía o director(a) de memoria o tesis:**

**Firma:** María Cristina Riff Rojas

**Fecha:**

06/08/2026 ;

**Estudiante o Candidato(a):**

**Fecha:** 06/08/2026 ; **Firma:**

**Andrés Felipe Navarro Galleguillos**

*Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por*

TÍTULO DE LA TESIS:

**Nuevos óptimos para el Problema de Rutas de Camiones y Remolques con Restricciones de Capacidad usando un algoritmo colaborativo**

AUTOR:

**Andrés Felipe Navarro Galleguillos**

Tesis presentada para la obtención del grado de Magíster en Ciencias de la Ingeniería Informática de la Universidad Técnica Federico Santa María.

Dr. María Cristina Riff

---

Tutor

Dr. Elizabeth Montero

---

Correferente Interno

Dr. Ignacio Araya

---

Evaluador Externo

Valparaíso, Chile.  
13 de agosto de 2026.

Si he logrado llegar hasta aquí, es porque me he apoyado sobre los hombros de gigantes.  
Este trabajo está dedicado a mi pareja Bárbara Uribe Cataldo, a mi madre, Janet Galleguillos y a mi profesora guía, María Cristina Riff.

# Agradecimientos

El término de esta etapa no habría sido posible sin las personas que me apoyaron y acompañaron durante este proceso.

Quiero agradecer a la profesora María Cristina Riff, quien me acompañó una vez más en otra etapa de mi desarrollo profesional, apoyándome y trabajando conmigo para conseguir el mejor trabajo posible.

Quiero agradecer a mi madre, Janet Galleguillos, por estar siempre para mí.

A mis amigos, Matías Santander, Sebastian Muñoz, Sonny Muñoz, Kevin Lagos, Javier Moreno, Sebastián Vera, Sebastián Bórquez, Felipe Dumont y a todas las personas con las que pude compartir durante esta etapa.

También mencionar a mis amistades del liceo que aún conservo, David Morales, Camilo Vásquez y Héctor Carvajal, a quienes me terminé por unir al elegir el camino de la docencia.

Quiero agradecer también a las profesoras que me dieron la oportunidad de impartir clases y confiaron en mí, gracias profesora Andrea Vásquez y Elizabeth Montero.

Menciono además a todos los estudiantes que he tenido durante este tiempo y con quienes me reencuentro cada cierto tiempo. Tanto en la sala de clases como fuera de ella me confirman que quiero continuar siendo profesor.

También quiero agradecer a BaityBait, por acompañarme de fondo durante todas mis horas de trabajo.

Un agradecimiento especial a mi pareja, Bárbara Uribe Cataldo, pilar fundamental y compañera de vida con la que he compartido todo este proceso. También a Lucas y Wallie, nuestras mascotas.

Gracias a todas las personas con las que compartí.

Por último, esta tesis fue financiada por la Agencia Nacional de Investigación y Desarrollo (ANID) mediante el proyecto FONDECYT Regular N.º 1241112. Asimismo, agradezco el apoyo financiero otorgado por ANID-Subdirección de Capital Humano/Magíster Nacional/2022 - 22220786, beca que hizo posible cursar este programa de magíster.

# Resumen

El Problema de rutas de camiones y remolques con restricciones de capacidad es un problema de optimización combinatoria que se enfoca en determinar la manera de satisfacer un conjunto de clientes usando una flota de camiones y remolques. El objetivo del problema es identificar, para cada camión, la mejor secuencia de clientes a visitar para así minimizar el costo total, satisfaciendo la demanda del cliente y un conjunto de restricciones.

En esta tesis se presenta CORAL (*Construct then Repair Algorithm*), un enfoque colaborativo que incluye una fase de construcción y una fase de reparación, basadas en técnicas GRASP y *Hill-climbing* respectivamente. Se propone un conjunto de procedimientos simples basados en *Hill-climbing*, cada uno con un objetivo diferente, permitiendo al algoritmo encontrar vecindarios de buena calidad en los que buscar una solución.

Se usó un método de sintonización automático tanto para definir el orden de aplicación de cada procedimiento *Hill-climbing* como para determinar los mejores parámetros para el algoritmo.

Los resultados muestran que CORAL es capaz de obtener buenas soluciones para un conjunto de instancias de la literatura y obtiene 13 nuevas cotas para instancias de dicho conjunto.

**Palabras clave:** Problema de rutas de camiones y remolques, Búsqueda local, Heurística colaborativa, Problemas del mundo real.

# Abstract

The Truck and Trailer Routing Problem with capacity constraints is a combinatorial optimization problem focused on determining how to satisfy a set of customers using a fleet of trucks and trailers. The objective of the problem is to identify, for each truck, the best sequence of customers to visit in order to minimize the total cost, while satisfying customer demand and a set of constraints.

This thesis presents CORAL (Construct then Repair Algorithm), a collaborative approach comprising a construction phase and a repair phase, based on GRASP and Hill-climbing techniques, respectively. A set of simple Hill-climbing-based procedures is proposed, each with a different objective, allowing the algorithm to find good-quality neighborhoods in which to search for a solution.

An automatic tuning method was used to determine both the application order of each Hill-climbing procedure and the best parameters for the algorithm.

The results show that CORAL is able to obtain good solutions for a set of instances from the literature and achieves 13 new bounds for instances in that set.

**Keywords:** Truck and Trailer Routing Problem, Local search, Collaborative heuristic, Real-world problems.

# Índice general

|                                                                             |             |
|-----------------------------------------------------------------------------|-------------|
| <b>Resumen</b>                                                              | <b>III</b>  |
| <b>Abstract</b>                                                             | <b>IV</b>   |
| <b>Índice general</b>                                                       | <b>VI</b>   |
| <b>Índice de Figuras</b>                                                    | <b>VIII</b> |
| <b>Índice de Tablas</b>                                                     | <b>X</b>    |
| <b>Índice de Algoritmos</b>                                                 | <b>XI</b>   |
| <b>1. Introducción</b>                                                      | <b>1</b>    |
| 1.1. Objetivo General . . . . .                                             | 2           |
| 1.2. Objetivos Específicos . . . . .                                        | 3           |
| <b>2. Definición del problema</b>                                           | <b>4</b>    |
| 2.1. Modelo matemático . . . . .                                            | 7           |
| 2.1.1. Parámetros . . . . .                                                 | 7           |
| 2.1.2. Variables . . . . .                                                  | 8           |
| 2.1.3. Función objetivo . . . . .                                           | 8           |
| 2.1.4. Restricciones . . . . .                                              | 8           |
| 2.2. Variantes relacionadas . . . . .                                       | 11          |
| 2.3. Resumen del Capítulo . . . . .                                         | 11          |
| <b>3. Búsqueda Local Estocástica</b>                                        | <b>13</b>   |
| 3.1. Búsqueda constructiva . . . . .                                        | 14          |
| 3.2. Búsqueda reparadora . . . . .                                          | 16          |
| 3.3. Metaheurísticas poblacionales . . . . .                                | 18          |
| 3.4. Resumen del Capítulo . . . . .                                         | 20          |
| <b>4. Estado del arte del Capacitated Truck and Trailer Routing Problem</b> | <b>21</b>   |
| 4.1. Capacitated Truck and Trailer Routing Problem . . . . .                | 21          |
| 4.1.1. Métodos de resolución . . . . .                                      | 23          |
| 4.1.2. Variantes del problema . . . . .                                     | 32          |



|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| 4.2. Resumen del Capítulo . . . . .                                 | 35        |
| <b>5. Propuesta de solución</b>                                     | <b>37</b> |
| 5.1. Representación . . . . .                                       | 37        |
| 5.2. Función de evaluación . . . . .                                | 38        |
| 5.3. Fase de construcción . . . . .                                 | 38        |
| 5.3.1. Procedimiento A (Enfoque Round Robin) . . . . .              | 39        |
| 5.3.2. Procedimiento B (Construcción secuencial de rutas) . . . . . | 40        |
| 5.3.3. Procedimiento C (Enfoque de clustering) . . . . .            | 42        |
| 5.3.4. Procedimiento General . . . . .                              | 44        |
| 5.4. Fase de reparación . . . . .                                   | 44        |
| 5.4.1. Movimientos . . . . .                                        | 44        |
| 5.4.2. Estructura general . . . . .                                 | 48        |
| 5.4.3. Parámetros . . . . .                                         | 49        |
| 5.5. Conclusiones del Capítulo . . . . .                            | 50        |
| <b>6. Resultados</b>                                                | <b>51</b> |
| 6.1. Entorno experimental . . . . .                                 | 51        |
| 6.2. Proceso de sintonización . . . . .                             | 52        |
| 6.3. Estudio de componentes . . . . .                               | 53        |
| 6.4. Análisis de soluciones iniciales . . . . .                     | 53        |
| 6.5. Análisis de resultados . . . . .                               | 58        |
| 6.5.1. Efecto de soluciones iniciales . . . . .                     | 62        |
| 6.5.2. Test de Wilcoxon . . . . .                                   | 65        |
| 6.5.3. Boxplots . . . . .                                           | 66        |
| 6.6. Conclusiones del Capítulo . . . . .                            | 69        |
| <b>7. Conclusiones</b>                                              | <b>70</b> |
| 7.1. Trabajo futuro . . . . .                                       | 71        |
| <b>Referencias</b>                                                  | <b>72</b> |

# Índice de figuras

|                                                                                                                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1. Ejemplificación de distintas configuraciones con las que puede contar una flota de camiones. . . . .                                                                                                      | 5  |
| 2.2. Instancia de ejemplo con 25 clientes. Esta presenta 19 clientes de vehículo y 6 clientes de camión junto al depósito. . . . .                                                                             | 6  |
| 2.3. Tipos de ruta en el TTRP. Clientes de camión son presentados como cuadrados y clientes de vehículo como círculos. . . . .                                                                                 | 6  |
| 3.1. Taxonomía general de los algoritmos de búsqueda local estocástica . . . .                                                                                                                                 | 14 |
| 3.2. Las dos formas en que un algoritmo de búsqueda local puede definir su espacio de búsqueda: sobre soluciones parciales o sobre soluciones completas. . . . .                                               | 14 |
| 3.3. Estancamiento de una búsqueda reparadora monótona en un óptimo local $\hat{x}$ , y mecanismo de escape hacia regiones de mejor calidad. . . . .                                                           | 17 |
| 3.4. Ejemplo del operador de cruce de orden (OX) . . . . .                                                                                                                                                     | 20 |
| 4.1. Esquema de heurísticas de construcción <i>T-Cluster</i> y <i>T-Sweep</i> . . . . .                                                                                                                        | 24 |
| 4.2. Comparación de la distribución espacial de clientes entre las primeras instancias de 25 clientes generadas a partir del conjunto de instancias CHAO (izquierda) y TAI (derecha) . . . . .                 | 27 |
| 5.1. Representación de una ruta de vehículo con dos sub-tours. La ruta principal se muestra en la parte de arriba, seguida por los puntos iniciales de cada sub-tour y sus correspondientes sub-tours. . . . . | 38 |
| 5.2. Una solución inicial de tipo A para 25 clientes. Esta consiste de tres rutas principales, una ruta de camión pura y dos rutas de vehículo completas. Cada una con un sub-tour. . . . .                    | 40 |
| 5.3. Una solución inicial de tipo B para una instancia con 25 clientes, consiste en dos rutas principales, cada una con un sub-tour. . . . .                                                                   | 41 |
| 5.4. Una solución inicial de tipo C para una instancia de 25 clientes. Esta consiste de 3 rutas, una ruta de camión pura y dos rutas de vehículo completas con tres y un sub-tour, respectivamente. . . . .    | 43 |

|                                                                                                                                                                                                                                                                                                                                                                                |    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.5. Ejemplo del movimiento 2-opt. Primero se seleccionan dos arcos, que en este caso, deliberadamente se encuentran cruzados. Luego estos se eliminan, separando la ruta en tres segmentos. Por último, los cuatro clientes se reconectan intercambiando los clientes a los que están conectados, invirtiendo el orden del segmento intermedio, resultando en un menor costo. | 45 |
| 5.6. Ejemplo del movimiento de refinamiento de raíces de sub-tours. Los arcos que conectan el sub-tour a su origen en la ruta principal son removidos y el sub-tour es reconectado a un cliente distinto por medio de unir este cliente a dos clientes consecutivos que se encuentran en el sub-tour.                                                                          | 46 |
| 6.1. Distribución de n° de sub-tours según número de clientes.                                                                                                                                                                                                                                                                                                                 | 57 |
| 6.2. Distribución de largo de rutas según número de clientes.                                                                                                                                                                                                                                                                                                                  | 57 |
| 6.3. Distribución del gap porcentual respecto al mejor resultado conocido en la literatura, agrupado por conjunto de instancias y procedimiento de solución inicial.                                                                                                                                                                                                           | 67 |
| 6.4. Distribución del gap porcentual respecto al mejor resultado previamente conocido, para las 13 instancias en las que CORAL obtuvo una nueva cota.                                                                                                                                                                                                                          | 68 |
| 6.5. Distribución de la calidad de las soluciones obtenidas para la instancia Chao40 A-2, comparada con el mejor resultado conocido en la literatura.                                                                                                                                                                                                                          | 68 |

# Índice de tablas

|                                                                                                                                                                                                      |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 4.1. Características de las 21 instancias TTRP propuestas por Chao, derivadas de los problemas CMT del VRP . . . . .                                                                                 | 22 |
| 4.2. Características de las instancias del conjunto <i>Chao30</i> (30 clientes), tipos A y B . . . . .                                                                                               | 27 |
| 4.3. Características de las instancias del conjunto <i>Chao30</i> , tipo C (un camión y un remolque) . . . . .                                                                                       | 28 |
| 4.4. Características de las instancias del conjunto <i>Tai30</i> (30 clientes), tipos A y B . . . . .                                                                                                | 28 |
| 4.5. Características de las instancias del conjunto <i>Tai30</i> , tipo C (un camión y un remolque) . . . . .                                                                                        | 29 |
| 4.6. Características de las instancias del conjunto <i>Chao40</i> (40 clientes), tipos A y B . . . . .                                                                                               | 30 |
| 4.7. Características de las instancias del conjunto <i>Chao40</i> , tipo C (un camión y un remolque) . . . . .                                                                                       | 30 |
| 4.8. Características de las instancias del conjunto <i>Chao25</i> (25 clientes) . . . . .                                                                                                            | 31 |
| 4.9. Características de las instancias del conjunto <i>Tai25</i> (25 clientes) . . . . .                                                                                                             | 31 |
| 4.10. Contribuciones recientes por variante del problema en el TTRP. Los conjuntos de instancias marcados con <sup>A</sup> indican versiones adaptadas de los conjuntos de datos originales. . . . . | 36 |
| 6.1. Calidad promedio y desviación estándar para instancias de 25 clientes . . . . .                                                                                                                 | 54 |
| 6.2. Calidad promedio y desviación estándar para instancias de 30 clientes . . . . .                                                                                                                 | 55 |
| 6.3. Calidad promedio y desviación estándar para instancias de 40 clientes . . . . .                                                                                                                 | 56 |
| 6.4. Resultados del análisis de varianza (ANOVA) de un factor sobre métricas de número de sub-tours y largo de rutas . . . . .                                                                       | 58 |
| 6.5. Comparaciones post-hoc de Tukey HSD para el número de sub-tours . . . . .                                                                                                                       | 58 |
| 6.6. Resultados para las instancias del conjunto <i>Chao25</i> . . . . .                                                                                                                             | 58 |
| 6.7. Resultados para las instancias del conjunto <i>Chao30</i> . . . . .                                                                                                                             | 59 |
| 6.8. Resultados para las instancias del conjunto <i>Chao40</i> . . . . .                                                                                                                             | 59 |
| 6.9. Resultados para las instancias del conjunto <i>Tai25</i> . . . . .                                                                                                                              | 59 |
| 6.10. Resultados para las instancias del conjunto <i>Tai30</i> . . . . .                                                                                                                             | 59 |
| 6.11. Estadísticas resumen de todas las ejecuciones para las instancias del conjunto <i>Chao25</i> . . . . .                                                                                         | 60 |
| 6.12. Estadísticas resumen de todas las ejecuciones para las instancias del conjunto <i>Chao30</i> . . . . .                                                                                         | 61 |

## ÍNDICE DE TABLAS

---

|                                                                                                                                                                                        |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 6.13. Estadísticas resumen de todas las ejecuciones para las instancias del conjunto <i>Chao40</i> . . . . .                                                                           | 61 |
| 6.14. Estadísticas resumen de todas las ejecuciones para las instancias del conjunto <i>Tai25</i> . . . . .                                                                            | 61 |
| 6.15. Estadísticas resumen de todas las ejecuciones para las instancias del conjunto <i>Tai30</i> . . . . .                                                                            | 62 |
| 6.16. Distribución de las instancias del conjunto BART según los procedimientos de construcción cuyas ejecuciones alcanzan la mejor solución encontrada para dicha instancia . . . . . | 62 |
| 6.17. Comparación de la calidad obtenida por cada procedimiento para las instancias de 25 clientes . . . . .                                                                           | 63 |
| 6.18. Comparación de la calidad obtenida por cada procedimiento para las instancias de 30 clientes . . . . .                                                                           | 63 |
| 6.19. Comparación de la calidad obtenida por cada procedimiento para las instancias de 40 clientes ( <i>Chao40</i> ) . . . . .                                                         | 64 |
| 6.20. Rangos de la prueba de Wilcoxon entre los resultados de la literatura y el algoritmo propuesto para las 38 instancias sin óptimo probado . . . . .                               | 66 |
| 6.21. Estadísticos de la prueba de Wilcoxon . . . . .                                                                                                                                  | 66 |

# Índice de algoritmos

|                                     |    |
|-------------------------------------|----|
| 1. Esquema general de GRASP         | 15 |
| 2. Esquema general de ILS           | 19 |
| 3. Fase de construcción             | 44 |
| 4. Bloque de movimientos de cliente | 47 |
| 5. Fase de reparación               | 48 |
| 6. Algoritmo de reparación          | 49 |

# Capítulo 1

## Introducción

Los problemas de optimización combinatoria de tipo *NP-hard* pueden definirse como problemas que no pueden ser resueltos de manera óptima dentro de un tiempo razonable mediante un método de resolución determinista (técnica completa) (Talbi, 2009). Este tipo de problemas surge, en general, de una formulación matemática que modela una situación basada en la realidad y pueden clasificarse en distintas categorías dependiendo de si son continuos o discretos, restringidos o no restringidos, mono o multi-objetivo, estáticos o dinámicos.

Esta tesis está enfocada en resolver el Capacitated Truck and Trailer Routing Problem (CTTRP), un problema de optimización combinatoria que se presenta en diversas industrias de transporte y distribución de bienes. El CTTRP es una variante del clásico Vehicle Routing Problem (VRP) (Dantzig y Ramser, 1959), cuyo objetivo es satisfacer la demanda de un conjunto de clientes utilizando una flota de camiones que parte y regresa a un depósito central, minimizando el costo de distribución asociado. A diferencia del VRP clásico, el CTTRP incorpora la posibilidad de que cada camión posea un remolque, el cual puede ser desenganchado en ciertos puntos de la ruta, permitiendo que el camión realice subtours por sí solo para atender a un subconjunto de clientes. Esta característica es relevante debido a que existen clientes ubicados en zonas de difícil acceso, como caminos angostos o terrenos montañosos, que solo pueden ser atendidos por el camión sin el remolque acoplado, acercando así el problema a escenarios reales y aumentando su complejidad de resolución.

El problema fue introducido originalmente por Chao en el año 2002 (Chao, 2002), derivando sus instancias a partir de un conjunto de instancias conocidas del VRP (Christofides, Nicos, 1976). A partir de este trabajo se han propuesto numerosas variantes y extensiones en la literatura, incorporando elementos adicionales como ventanas de tiempo de entrega (S.-W. Lin, Yu, y Lu, 2011), demanda estocástica (Mirmohammadsadeghi y Ahmed, 2015), incertidumbre en los tiempos de viaje (Mirmohammadsadeghi y Kabir, 2024), así como escenarios con un único camión y remolque disponibles (Accorsi y Vigo, 2020). Un resumen de estas variantes y sus principales contribuciones se detalla en el Capítulo 4.

Dado que el CTTRP es un problema NP-hard, los métodos exactos, como los algoritmos de branch-and-cut propuestos por Bartolini y Schneider (Bartolini y Schneider, 2020), logran resolver de forma óptima solo una fracción de las instancias de mayor tamaño den-

tro de límites de tiempo razonables. Por esta razón, las metaheurísticas y los métodos de búsqueda local han sido ampliamente utilizados para abordar instancias de gran escala, ya que permiten obtener soluciones de buena calidad en tiempos computacionales considerablemente menores.

En este contexto, la calidad y la estructura de las soluciones iniciales juegan un rol fundamental en el desempeño de los algoritmos reparadores basados en búsqueda local, y contar con una diversidad de soluciones iniciales facilita que estos métodos exploren distintas regiones del espacio de búsqueda (Sarhani, Benadada, Erray, y Gendreau, 2022). Sin embargo, en la literatura relacionada con el CTTRP, la mayoría de los enfoques utilizan un único procedimiento de construcción de soluciones iniciales, dejando abierta la pregunta de si una estrategia colaborativa que combine múltiples procedimientos de construcción puede favorecer la búsqueda de mejores soluciones.

La forma de resolver el CTTRP en este trabajo es a través de CORAL (*Construct then Repair Algorithm*), un enfoque colaborativo de dos fases, que combina una fase de construcción basada en procedimientos tipo GRASP (*Greedy Randomized Adaptive Search Procedure*) con una fase de reparación basada en *Hill-climbing*. La fase de construcción incorpora distintos procedimientos, cada uno guiado por una función miope diferente, con el objetivo de generar un conjunto diverso de soluciones factibles y prometedoras. Estos procedimientos corresponden a un subconjunto de los propuestos y evaluados preliminarmente en un trabajo previo (Navarro, Riff, González, Barros, y Barriga, 2025), cuyos resultados se retoman y extienden en esta tesis al incorporarlos a un esquema de reparación. La fase de reparación, por su parte, aplica un conjunto de movimientos de búsqueda local, a nivel de ruta, de sub-tour y de cliente individual, con el objetivo de mejorar la calidad de dichas soluciones, permitiendo además que la búsqueda transite temporalmente por el espacio infactible mediante una estrategia de penalización adaptativa.

Como hipótesis: un enfoque colaborativo de construcción y reparación, que combine procedimientos de construcción diversos con una fase de búsqueda local basada en *Hill-climbing*, es capaz de resolver el Capacitated Truck and Trailer Routing Problem obteniendo resultados competitivos con respecto a los publicados en la literatura, mejorando incluso las mejores cotas conocidas para un subconjunto de instancias, dentro de tiempos computacionales razonables.

En base a lo mencionado anteriormente, el presente trabajo posee los siguientes objetivos.

### 1.1. Objetivo General

Diseñar e implementar un enfoque colaborativo de dos fases, de construcción y reparación, capaz de resolver el Capacitated Truck and Trailer Routing Problem, generando soluciones iniciales diversas y mejorándolas mediante búsqueda local, con el objetivo de obtener resultados competitivos o superiores en calidad y tiempo respecto a los reportados en el estado del arte.



## 1.2. Objetivos Específicos

- Analizar el estado del arte del Capacitated Truck and Trailer Routing Problem.
- Diseñar un conjunto de procedimientos de construcción, basados en GRASP, capaces de generar soluciones iniciales factibles y diversas para el CTTRP.
- Diseñar un conjunto de movimientos de búsqueda local y una fase de reparación basada en *Hill-climbing*, capaz de mejorar la calidad de las soluciones iniciales generadas.
- Evaluar el efecto de los distintos procedimientos de construcción de soluciones iniciales sobre la calidad de las soluciones finales obtenidas por el enfoque propuesto.

El presente trabajo se divide en los siguientes capítulos: en el Capítulo 2 se detalla la definición formal del Capacitated Truck and Trailer Routing Problem, junto con su modelo matemático. En el Capítulo 3 se presentan los conceptos fundamentales de la búsqueda local estocástica, distinguiendo entre búsqueda constructiva y búsqueda reparadora, como marco teórico para el resto del trabajo. En el Capítulo 4 se presenta el estado del arte, tanto de los métodos de resolución propuestos para el CTTRP como de la literatura relacionada con la diversidad de soluciones iniciales en metaheurísticas. En el Capítulo 5 se presenta CORAL, el enfoque propuesto, detallando su representación de soluciones, función de evaluación, fase de construcción y fase de reparación. En el Capítulo 6 se presentan y analizan los resultados experimentales obtenidos, incluyendo el proceso de sintonización de parámetros, el análisis de las soluciones iniciales generadas y la comparación con los resultados del estado del arte. Finalmente, en el Capítulo 7 se presentan las conclusiones de este trabajo junto con las líneas de investigación futuras.

# Capítulo 2

## Definición del problema

En este capítulo se presentará la definición formal del problema de rutas de camiones y remolques con restricciones de capacidad.

Actualmente uno de los servicios más importantes y más usados corresponde al de entrega de bienes a clientes (delivery). Uno de los problemas más estudiados en la literatura es el *Vehicle Routing Problem* o problema de ruteo de vehículos que consiste en servir a una serie de clientes, contando con una flota de vehículos que inician su viaje desde un depósito central, minimizando el costo o tiempo u otra medida de los viajes.

El origen de este problema se presenta en el trabajo de Dantzig y Ramser (Dantzig y Ramser, 1959), quienes formularon el problema bajo el nombre de *Truck Dispatching Problem*.

A diferencia de otros problemas, como el problema del vendedor viajero (Dantzig, Fulkerson, y Johnson, 1954), este considera una flota de vehículos con capacidad limitada.

Debido a que el modelo original incorpora restricciones de capacidad, en la literatura actual se le conoce como *Capacitated Vehicle Routing Problem* (CVRP), denominación utilizada para distinguirlo de las numerosas variantes del VRP propuestas posteriormente. En esta tesis se trabajará específicamente con el Problema de Rutas de Camiones y Remolques con Restricciones de Capacidad (CTTRP: en inglés *Capacitated Truck and Trailer Routing Problem*).

El CTTRP es una variante del clásico problema de enrutamiento de vehículos que incorpora elementos adicionales con el objetivo de representar situaciones más cercanas al mundo real. Además de contar con la capacidad del camión como tal, existe la opción de acoplar un remolque, incrementando así la capacidad total del vehículo para atender a un mayor número de clientes. El remolque puede ser desenganchado en medio de la ruta, permitiendo que el camión realice un sub-tour para atender a un subconjunto de clientes. En la Figura 2.1 se pueden observar distintas configuraciones que podría tener una flota de camiones. En este caso se trabajará con una flota con camiones que dispondrán de la misma capacidad y un conjunto de remolques que también tendrán la misma capacidad entre ellos.

Además, los clientes se clasifican en dos tipos según su accesibilidad:

- *Los clientes de vehículo* son fácilmente accesibles y pueden ser atendidos por el camión con el remolque acoplado (que se denominará vehículo compuesto).



Figura 2.1: Ejemplificación de distintas configuraciones con las que puede contar una flota de camiones.

- *Los clientes de camión* se ubican en zonas de difícil acceso, como lo podrían ser terrenos montañosos o calles estrechas y solo pueden ser atendidos por el camión sin remolque.

En la Figura 2.2 se presenta una instancia de la literatura con 40 clientes. En este caso la cantidad de clientes de camión es considerablemente mayor a la de clientes de vehículo, lo que incita a que existan rutas de camiones individuales, o rutas en las que, contando con un remolque, este se desacople en alguno de los clientes de vehículo para poder hacer un sub-tour y atender a los clientes de difícil acceso.

Bajo esta configuración, se definen tres tipos de rutas:

- Ruta de camión pura: Los clientes son atendidos únicamente por el camión.
- Ruta de vehículo pura: Los clientes son atendidos por el camión y el remolque sin desenganchar este en ningún momento.
- Ruta de vehículo completa: El camión, al menos una vez, desengancha el remolque en la ubicación de algún cliente de vehículo, atiende a un subconjunto de clientes y luego regresa a recoger el remolque para continuar la ruta.

Un ejemplo de las distintas rutas que se pueden formar se puede evidenciar en la Figura 2.3. En esta se muestra que la ruta de vehículo pura sólo puede atender a clientes de vehículo, mientras que la ruta de camión pura puede atender a ambos tipos de cliente. Por otro lado, la ruta de vehículo completa puede atender a clientes de vehículo solamente en la ruta principal, con el remolque acoplado, mientras que puede atender tanto a clientes de camión como de vehículo en los sub-tours que realiza, siempre que tenga la capacidad disponible para atenderlos solamente con el compartimento principal.

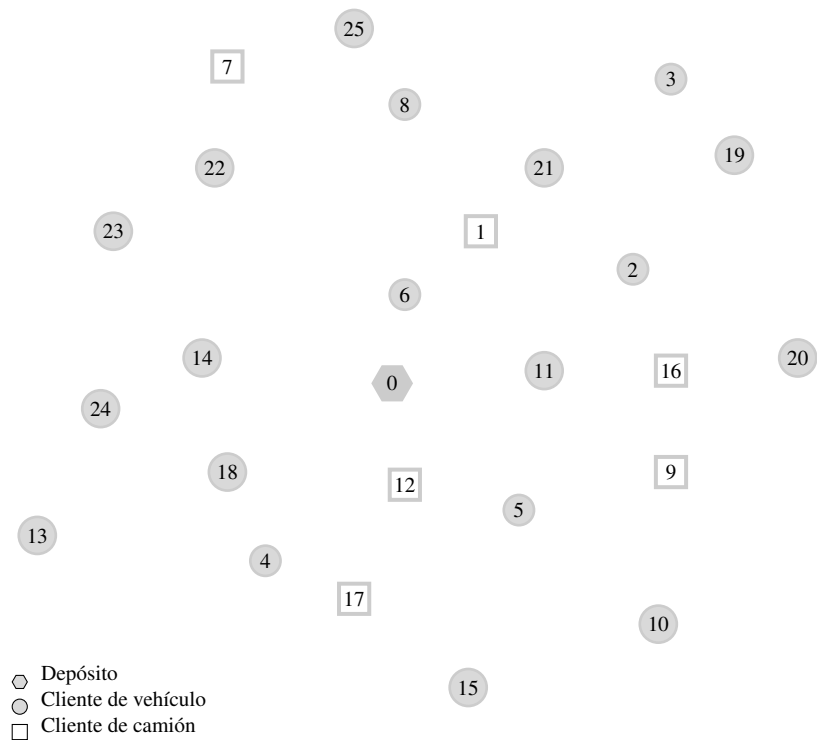


Figura 2.2: Instancia de ejemplo con 25 clientes. Esta presenta 19 clientes de vehículo y 6 clientes de camión junto al depósito.

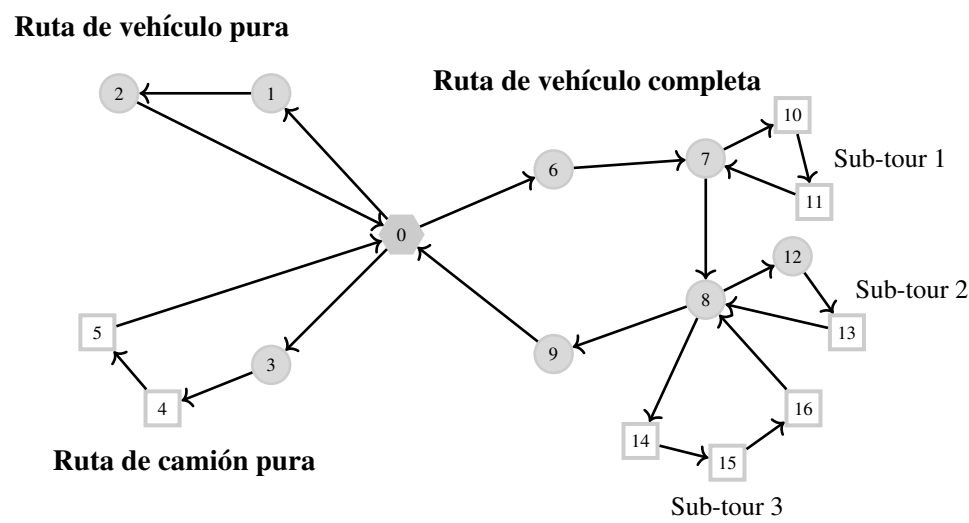


Figura 2.3: Tipos de ruta en el TTRP. Clientes de camión son presentados como cuadrados y clientes de vehículo como círculos.

Adicional a esto, se permite la transferencia de carga desde el remolque al camión, pudiendo realizar un sub-tour, en la que se considera solo la capacidad del camión, para luego continuar con la ruta principal y nuevamente mover carga al camión, permitiendo así realizar otro sub-tour más adelante.

El objetivo del problema es minimizar la distancia recorrida por los camiones, sin exceder la capacidad de los mismos. Con esto, se considera una flota de camiones homogénea, un conjunto de remolques homogéneo menor o igual a la flota de camiones, un grafo de clientes con una demanda definida que debe ser atendido solo una vez, y un depósito del que salen y al que vuelven todos los camiones al terminar su ruta. Finalmente, los sub-tours no pueden sobrepasar la capacidad del camión y deben volver al último cliente servido con remolque antes de continuar la ruta.

## 2.1. Modelo matemático

A continuación se presenta un modelo matemático propuesto en (Bartolini y Schneider, 2020). Este modelo corresponde a una formulación de flujo de dos mercancías (*two-commodity flow formulation*), en donde el flujo de bienes que son llevados por un vehículo se modela usando dos variables de flujo, una representando la carga del vehículo y otra representando la capacidad disponible al recorrer cada arco.

Se presenta un grafo completo no dirigido  $G = (V_0, E)$ , con un conjunto de vértices  $V_0 = \{0, \dots, n\}$  representando el depósito ( $V_0$ ) y los clientes. A partir de éste se define el conjunto de clientes  $V = V_0 \setminus \{0\}$  que diferencia entre  $V^c$  y  $V^t$ , representando los clientes de vehículo y camión respectivamente. Adicionalmente, definen los siguientes sub-conjuntos,  $V_0^c = V^c \cup \{0\}$  y  $V_0^t = V^t \cup \{0\}$ .

Con esto, se procede a definir los parámetros, variables, función objetivo y restricciones del modelo.

### 2.1.1. Parámetros

$q_i$  Demanda del cliente  $i$

$m_t$  Número de camiones

$Q_t$  Capacidad de camiones

$m_c$  Número de remolques

$Q_c$  Capacidad de remolques

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$  Un grafo extendido dónde:

- $\mathcal{V}$ : Corresponde a un conjunto de vértices extendido compuesto de  $V_0, 0'$ , que es una copia del depósito, usado como el final de todas las rutas, y una copia de cada  $i \in V^c, i'$ , representando el final de cada sub-tour.
- $\mathcal{E} = E \cup \{\{i, j'\} : i \in V, j \in V_0^c, i \neq j\}$ : El conjunto de arcos de  $\mathcal{G}$ .

- $\mathcal{E}_c = \{\{i, j\} \in E : i \in V_0^c, j \in V^c\} \cup \{\{i, 0'\} : i \in V_0^c\}$ : El subconjunto de arcos que son incidentes entre dos vértices de  $V_0^c$ .
- $\delta(S)$ : Notación usada para denotar el subconjunto de arcos en  $\mathcal{E}$  que cruza el subconjunto  $S \subseteq \mathcal{V}$ . Si  $S$  es un conjunto unitario  $\{i\}$ , se escribe como  $\delta(i)$ .
- $\mathcal{E}(S, T)$ : Notación usada para denotar el subconjunto de aristas que tiene un extremo en  $S$  y otro en  $T$ , para cualquier par de subconjuntos  $S, T \subseteq \mathcal{V}$ .

$c_{ij}$  Costo del arco  $\{i, j\} \in \mathcal{E}$ , para estos, el costo de viajar desde o hacia las copias es el mismo que para el vértice original.

### 2.1.2. Variables

- $x_{ij}, \forall \{i, j\} \in \mathcal{E}_c$ : 1 si una ruta de vehículo recorre el arco  $\{i, j\} \in \mathcal{E}_c$ , 0 en otro caso.
- $z_{ij}, \forall \{i, j\} \in \mathcal{E}$ : 1 si un sub-tour de camión recorre el arco  $\{i, j\} \in \mathcal{E}$ , 0 en otro caso.
- $f_{ij}$  y  $f_{ji}, \forall \{i, j\} \in \mathcal{E}_c$ : Representan la carga total y la capacidad disponible, respectivamente, en un vehículo compuesto recorriendo el arco  $\{i, j\} \in \mathcal{E}_c$ .
- $g_{ij}$  y  $g_{ji}, \forall \{i, j\} \in \mathcal{E}$ : Representan la carga total y la capacidad disponible, respectivamente, en un camión recorriendo el arco  $\{i, j\} \in \mathcal{E}$ .
- $y_i, \forall i \in V^c$ : 1 si el cliente  $i$  es atendido por un camión, 0 e.o.c.
- $h_i$ , Número de sub-tours que salen desde  $i \in V_0^c$ . Estas variables son redundantes pero son incluidas para facilitar la lectura del modelo.

### 2.1.3. Función objetivo

La función objetivo minimiza el costo total de viaje de las rutas.

$$\min \sum_{\{i,j\} \in \mathcal{E}_c} c_{ij} x_{ij} + \sum_{\{i,j\} \in \mathcal{E}} c_{ij} z_{ij}$$

### 2.1.4. Restricciones

- En cada cliente  $i$ , la diferencia entre el flujo de entrada y salida debe ser igual al doble de la demanda de  $i$ . En caso de tener sub-tours, la diferencia entre las variables de flujo  $f$  debe ser igual a  $2q_i + 2q(S)$  y la diferencia de las variables de flujo  $g$  igual a  $Q_t h_i - 2q(S)$ , debido a que  $i$  se comporta como un depósito para los sub-tours, dando como resultado que sea igual a  $2q_i + Q_t h_i$

$$\sum_{\{i,j\} \in \delta(i) \cap \mathcal{E}_c} (f_{ji} - f_{ij}) + \sum_{\{i,j\} \in \delta(i)} (g_{ji} - g_{ij}) = 2q_i + Q_t h_i, \quad \forall i \in V^c. \quad (2.1)$$

- El flujo de salida total del depósito debe ser igual a la demanda total.

$$\sum_{j \in V^c} f_{0j} + \sum_{j \in V} g_{0j} = q(V) \quad (2.2)$$

- El flujo de entrada total en el depósito debe ser igual a la capacidad sin usar de la flota de vehículos.

$$\sum_{j \in V_{0'}^c} f_{j0} + \sum_{j \in V_{0'}} g_{j0} = m_c Q_c + m_t Q_t - q(V) \quad (2.3)$$

- El flujo de salida total de la copia del depósito ( $0'$ ) debe ser igual a la capacidad total de la flota de vehículos.

$$\sum_{j \in V_0^c} f_{0'j} + \sum_{j \in V_0} g_{0'j} = m_c Q_c + m_t Q_t \quad (2.4)$$

- La suma del flujo de entrada y salida en cada arco debe ser igual a la capacidad del vehículo compuesto, si es atendido por uno.

$$f_{ij} + f_{ji} = (Q_t + Q_c) x_{ij}, \quad \forall \{i, j\} \in \mathcal{E}_c \setminus \{0, 0'\} \quad (2.5)$$

- Por cada cliente de camión  $i$ , la diferencia entre el flujo de entrada y el flujo de salida debe ser igual al doble de la demanda de  $i$ .

$$\sum_{\{i, j\} \in \delta(i)} (g_{ji} - g_{ij}) = 2q_i, \quad \forall i \in V^t \quad (2.6)$$

- La suma del flujo de entrada y flujo de salida en cada arco debe ser igual a la capacidad del camión, si es atendido por uno.

$$g_{ij} + g_{ji} = Q_t z_{ij}, \quad \forall \{i, j\} \in \mathcal{E} \setminus \{0, 0'\} \quad (2.7)$$

- El flujo de salida de las variables  $g$  para cada cliente de vehículo  $i$  desde el vértice  $i'$  debe ser igual a la capacidad total del sub-tour que tiene como origen dicho cliente.

$$\sum_{j \in V} g_{i'j} = h_i Q_t, \quad \forall i \in V_0^c \quad (2.8)$$

- El grado de cada cliente de vehículo debe ser par, a menos que sea el origen de un sub-tour, añadiendo  $h_i$  para considerar las conexiones adicionales.

$$\sum_{\{i, j\} \in \delta(i) \cap \mathcal{E}_c} x_{ij} + \sum_{\{i, j\} \in \delta(i)} z_{ij} = 2 + h_i, \quad \forall i \in V^c \quad (2.9)$$

- El grado de cada cliente de vehículo que es parte de un sub-tour debe ser par.

$$\sum_{\{i,j\} \in \delta(i)} z_{ij} = 2y_i + h_i, \quad \forall i \in V^c \quad (2.10)$$

- Los sub-tours no pueden iniciar desde un cliente de vehículo atendido por un camión.

$$z_{ij'} + y_j \leq 1, \quad \forall i \in V, \quad \forall j \in V^c \quad (2.11)$$

- El grado de cada cliente de camión debe ser par.

$$\sum_{\{i,j\} \in \delta(i)} z_{ij} = 2, \quad \forall i \in V^t \quad (2.12)$$

- El número de rutas no puede exceder el número de camiones.

$$\sum_{j \in V_{0'}^c} x_{0j} + h_0 \leq m_t \quad (2.13)$$

- El número de rutas de vehículo no puede exceder el número de remolques.

$$\sum_{j \in V_{0'}^c} x_{0j} \leq m_c \quad (2.14)$$

- Las rutas de camión deben finalizar en la copia del depósito. (0').

$$\sum_{j \in V} z_{0j} = h_0 \quad (2.15)$$

- Las rutas de vehículo deben finalizar en la copia del depósito.

$$\sum_{j \in V^c} x_{0j} = \sum_{j \in V^c} x_{j0'} \quad (2.16)$$

- Conectividad de sub-tour (asegura que cada sub-tour que inicia en el cliente  $i$ , termine en su copia).

$$\sum_{\{i,j\} \in \mathcal{E}(S_k, V_{k'} \setminus S_k)} z_{ij} - h_k \geq 0, \quad \forall S \subseteq V, \forall k \in V^c \quad (2.17)$$



- Se fijan los valores para las variables redundantes  $h_i$ .

$$\sum_{\{j,i'\} \in \delta(i') \setminus \{0,i'\}} z_{ji'} = h_i, \quad \forall i \in V_0^c \quad (2.18)$$

- Dominio de las variables:

$$x_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in \mathcal{E}_c \quad (2.19)$$

$$f_{ij} \geq 0, f_{ji} \geq 0, \quad \forall \{i, j\} \in \mathcal{E}_c \quad (2.20)$$

$$z_{ij} \in \{0, 1\}, \quad \forall \{i, j\} \in \mathcal{E} \quad (2.21)$$

$$g_{ij} \geq 0, g_{ji} \geq 0, \quad \forall \{i, j\} \in \mathcal{E} \quad (2.22)$$

$$y_i \in \{0, 1\}, \quad \forall i \in V^c \quad (2.23)$$

## 2.2. Variantes relacionadas

Respecto a variantes de este problema, se puede mencionar el *Truck and Trailer Routing Problem with Time Windows* (S.-W. Lin y cols., 2011) que añade otro elemento de la vida real, correspondiente a ventanas de tiempo para atender a los clientes. A partir de esta se han incorporado otros elementos como ventanas de tiempo blandas y duras junto con entregas fraccionadas (Batsyn y Ponomarenko, 2014) o demanda estocástica (Mirmohammadsadeghi y Ahmed, 2015).

Otras variantes de este problema introducen incertidumbre en los tiempos de viaje y servicio, denominándose *Stochastic TTRP* (Mirmohammadsadeghi y Kabir, 2024). Por otro lado, otros trabajos se enfocan en resolver este problema limitando la flota de vehículos a solamente un camión junto con un remolque (Accorsi y Vigo, 2020) o múltiples compartimentos dentro del vehículo (Davila-Pena, R. Penas, y Casas-Méndez, 2023).

Por otro lado se pueden mencionar aplicaciones específicas a un contexto de una empresa logística (Cavagnini, Schneider, y Theiß, 2023) o situaciones que se comportan de manera similar a un camión con un remolque acoplado (Nguyen y cols., 2018) en el que un repartidor recorre la ciudad con su vehículo de entregas y debe dejarlo estacionado para atender a un conjunto de clientes a pie, siendo similar a realizar un sub-tour.

## 2.3. Resumen del Capítulo

En este capítulo se presentó y detalló el problema de la presente Tesis, el Problema de Rutas de Camiones y Remolques con Restricciones de Capacidad. Se describieron sus componentes distintivos, correspondientes al uso de un conjunto de camiones y remolques que tienen la posibilidad de ser desacoplados para realizar sub-tours, además de la existencia de clientes con diferentes características de accesibilidad, limitando el tipo de vehículo por el que pueden ser atendidos.

Además, se estableció su formulación matemática mediante un modelo de flujo de dos mercancías, definiendo parámetros, variables, función objetivo y restricciones, con el fin de formalizar el problema y su complejidad.

## CAPÍTULO 2. DEFINICIÓN DEL PROBLEMA

---

En el siguiente capítulo se presentan los conceptos fundamentales de la búsqueda local estocástica, distinguiendo entre búsqueda constructiva y reparadora, que sustentan el enfoque propuesto en esta tesis.

# Capítulo 3

## Búsqueda Local Estocástica

En esta Tesis se propone un algoritmo de búsqueda local para resolver el CTTRP, por esto es necesario presentar una revisión de las principales técnicas que realizan búsqueda local estocástica. En primer lugar, se presentarán los conceptos comunes a todas esas técnicas, para luego profundizar en sus diferencias de diseño.

Un algoritmo de búsqueda local es un método de búsqueda que comienza en una posición inicial en el espacio de búsqueda y se desplaza iterativamente, basándose en información local, desde la posición actual hacia posiciones vecinas en el espacio de búsqueda.

En cada paso del proceso, las decisiones de hacia dónde moverse se toman evaluando las propiedades del vecindario inmediato (por ejemplo, mediante una función de evaluación que mide la calidad de las soluciones vecinas). Esta definición es aplicable tanto a métodos reparadores como constructivos. Si bien tradicionalmente el término *búsqueda local* suele asociarse a métodos reparadores (que modifican soluciones completas) su definición también puede extenderse a métodos constructivos:

- Búsqueda Reparadora: Modifica iterativamente una solución completa mediante pequeñas alteraciones o perturbaciones (como reemplazar aristas en un tour del TSP).
- Búsqueda Constructiva: Opera en el espacio de soluciones parciales. Una solución parcial representa una posición en el espacio de búsqueda, y sus vecinos se obtienen extendiendo la solución con uno o más componentes.

Antes de profundizar en cada técnica, la Figura 3.1 sitúa a la búsqueda local estocástica (*Stochastic Local Search*, SLS) dentro de una clasificación más amplia, distinguiendo métodos simples, híbridos y basados en poblaciones (H. H. Hoos y Stützle, 2004).

Esta distinción entre búsqueda constructiva y reparadora estructura el resto del capítulo, y anticipa la organización de la propia propuesta de esta tesis (Capítulo 5), que combina una fase de construcción con una fase de reparación. La Figura 3.2 resume esta segunda forma de clasificar la búsqueda local, según el tipo de espacio sobre el que opera.

Estas nociones pueden formalizarse considerando un problema de optimización representado por el par  $(S, f)$ , en donde  $S$  es el espacio de soluciones (parciales o completas, dependiendo del tipo de búsqueda) y  $f$  es la función de evaluación. Una estructura de vecindarios sobre  $S$  es una función  $E : S \rightarrow 2^S$  que asocia a cada posición  $x \in S$  un

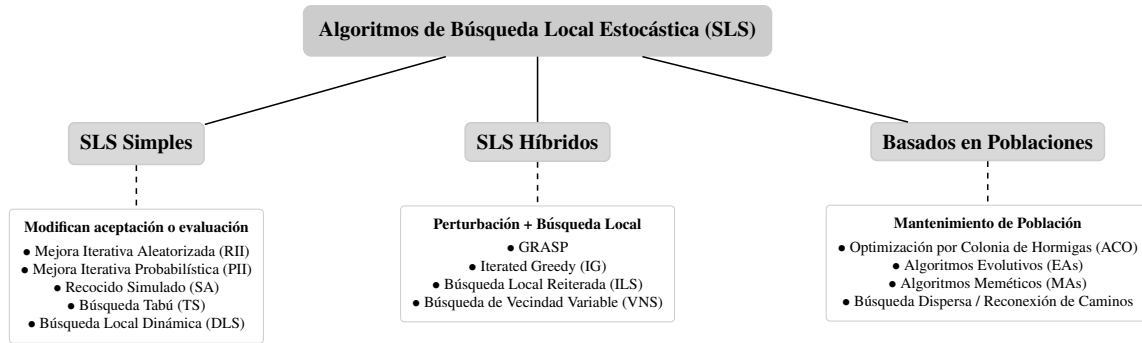


Figura 3.1: Taxonomía general de los algoritmos de búsqueda local estocástica (H. H. Hoos y Stützle, 2004).

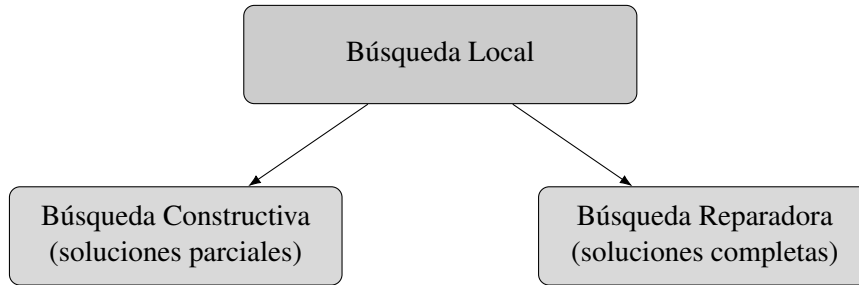


Figura 3.2: Las dos formas en que un algoritmo de búsqueda local puede definir su espacio de búsqueda: sobre soluciones parciales o sobre soluciones completas.

conjunto  $E(x) \subseteq S$  de posiciones vecinas, alcanzables desde  $x$  mediante un movimiento (Melian, Moreno-Pérez, y Moreno-Vega, 2003). En la búsqueda reparadora,  $x$  es una solución completa y sus vecinos se obtienen mediante pequeñas alteraciones sobre ella, mientras que en la búsqueda constructiva  $x$  es una solución parcial y sus vecinos se obtienen extendiéndola con nuevos componentes hasta completarla.

Una característica que comparten la gran mayoría de estas técnicas corresponde a la incorporación de un componente estocástico en alguna etapa del proceso. Este puede ser incluido en la selección del siguiente movimiento, en la construcción de la solución inicial, o en el criterio de aceptación de nuevas soluciones. Esta aleatoriedad controlada es lo que permite que un mismo procedimiento, aplicado repetidamente sobre una misma instancia, explore distintas regiones del espacio de búsqueda en cada ejecución, en lugar de obtener siempre la misma solución.

### 3.1. Búsqueda constructiva

La búsqueda constructiva opera sobre el espacio de soluciones parciales. Este tipo de búsqueda parte de una solución vacía e incorpora, en cada paso, un nuevo componente hasta completarla. La estrategia más simple de este tipo es la voraz (en inglés, *greedy*), que en cada paso selecciona el componente que ofrece el mejor resultado inmediato según

algún criterio, sin considerar el efecto de dicha elección sobre las decisiones futuras. Esta miopía hace que las soluciones voraces tiendan a quedar atrapadas en zonas de baja calidad del espacio de búsqueda, además de que, al ser un procedimiento determinista, siempre entrega la misma solución para una misma instancia, careciendo de un componente estocástico. Este tipo de procedimientos ha sido ampliamente usado para tener un punto de partida en problemas de optimización (Glover, Gutin, Yeo, y Zverovich, 2001). Dentro de las heurísticas constructivas es posible distinguir entre heurísticas **estáticas** y **adaptativas**, según el momento en que se calculan los valores heurísticos asociados a cada componente candidato. En las heurísticas estáticas estos valores se precálculan una única vez, antes de iniciar la construcción, y permanecen fijos durante todo el proceso. En las heurísticas adaptativas, en cambio, los valores se recalculan en cada paso de construcción, considerando el estado de la solución parcial actual. Si bien esta segunda alternativa es computacionalmente más costosa, generalmente produce soluciones de mejor calidad, ya que la evaluación de cada candidato refleja con mayor precisión su conveniencia real dado el contexto de la solución que se está construyendo (H. Hoos y Stützle, 2015). Por otro lado, la metaheurística GRASP (en inglés, *Greedy Randomized Adaptive Search Procedure*) incorpora aleatoriedad a la estrategia voraz para superar esta limitación, combinando una fase constructiva aleatorizada con una fase de mejora (Melian y cols., 2003). En cada iteración de GRASP se construye una solución completa y luego se mejora mediante búsqueda local. Este procedimiento se repite un número determinado de iteraciones, conservando la mejor solución encontrada. El Algoritmo 1 resume este esquema general.

---

**Algoritmo 1** Esquema general de GRASP

---

```

1: procedure GRASP( $maxIter$ )
2:    $mejorSol \leftarrow \emptyset$ 
3:   for  $iter = 1$  hasta  $maxIter$  do
4:      $sol \leftarrow$  CONSTRUCCIONVORAZALEATORIZADA
5:      $sol \leftarrow$  BUSQUEDALOCAL( $sol$ )
6:     if  $f(sol) < f(mejorSol)$  then
7:        $mejorSol \leftarrow sol$ 
8:     end if
9:   end for
10:  return  $mejorSol$ 
11: end procedure

```

---

La fase constructiva de GRASP se apoya en una **función miope** (también conocida como función de transición),  $g(c)$ , que estima la conveniencia inmediata de incorporar cada candidato  $c$  a la solución parcial, sin anticipar el efecto de esta decisión sobre el resto de la construcción. A diferencia de un procedimiento puramente voraz, que seleccionaría siempre al candidato con el mejor valor de  $g$ , GRASP construye una **Lista Restringida de Candidatos** (LRC) compuesta por los mejores candidatos según  $g$ , ya sea tomando los  $k$  mejores o aquellos cuyo valor cae dentro de un rango de calidad definido por un parámetro  $\alpha$ , seleccionando uno de ellos al azar. Este componente aleatorio y adaptativo permite

que distintas ejecuciones de GRASP generen soluciones iniciales diversas, sin sacrificar calidad, para una misma instancia.

## 3.2. Búsqueda reparadora

La búsqueda reparadora opera sobre el espacio de soluciones completas. Esta parte de una solución ya construida (pudiendo ser esta generada, por ejemplo, por una técnica constructiva) y la modifica iterativamente mediante pequeñas alteraciones o movimientos, en busca de una de mejor calidad.

Una *búsqueda local* es un proceso que, dada la solución actual, selecciona iterativamente una solución de su vecindario  $E(x)$  que la reemplaza si representa una mejora, deteniéndose cuando no es posible mejorar dentro del vecindario actual. En este proceso existen dos reglas de selección fundamentales:

- **Estrategia mejor mejora** (*Best improvement*): Se recorre todo el vecindario  $E(x)$  y selecciona la mejor solución dentro de todos los vecinos.
- **Estrategia alguna mejora** (*First improvement*): Recorre el vecindario, un vecino a la vez y selecciona la primera solución que mejore a la actual, sin revisar las demás candidatas a solución.

Un aspecto práctico determinante en el rendimiento de una búsqueda reparadora es el costo de evaluar cada solución vecina. En vez de recalcular la función objetivo completa para cada vecino generado, es posible aprovechar que un movimiento modifica solo una parte reducida de la solución, y calcular únicamente la diferencia (o *delta*) que dicho movimiento produce sobre el valor de la solución actual.

Esta evaluación incremental reduce significativamente el costo computacional de recorrer el vecindario, siendo una de las razones principales por las que la búsqueda local resulta viable en instancias de gran tamaño (H. Hoos y Stützle, 2015).

Una solución  $\hat{x}$  es un óptimo local respecto a una estructura de vecindarios  $E$  si  $f(\hat{x}) \leq f(y)$  para toda  $y \in E(\hat{x})$ . El principal inconveniente de una búsqueda reparadora monótona es que queda atrapada de forma inevitable al alcanzar dicho óptimo local, sin importar que tan alejado esté de este óptimo global (Figura 3.3).

La búsqueda local descrita anteriormente constituye el esquema básico del algoritmo *Hill-climbing*. Este algoritmo parte de una solución inicial y busca iterativamente en su vecindario una solución de mejor calidad, reemplazando la solución actual cuando se encuentra una mejora, pudiendo ser implementada mediante una estrategia de alguna mejora o mejor mejora.

Por esto, el propósito inicial de las primeras metaheurísticas era extender una búsqueda local para continuarla más allá de los óptimos locales. Estas técnicas reparadoras incorporan alguno de los siguientes tres mecanismos básicos (Melian y cols., 2003):

- Reiniciar la búsqueda desde otra solución.
- Modificar la estructura de vecindarios utilizada.

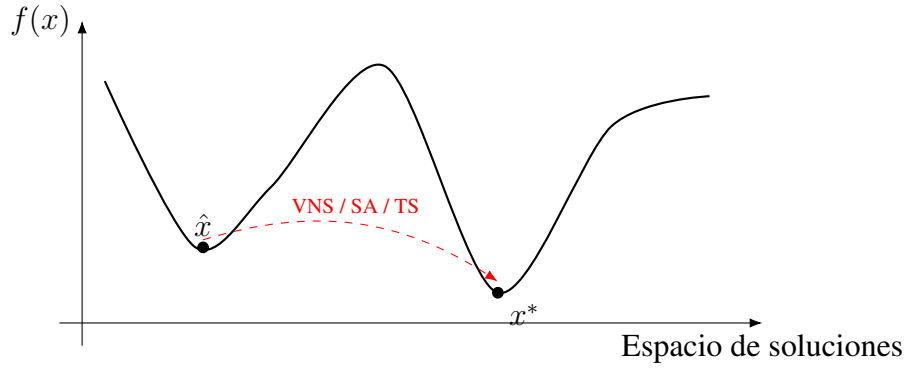


Figura 3.3: Estancamiento de una búsqueda reparadora monótona en un óptimo local  $\hat{x}$ , y mecanismo de escape hacia regiones de mejor calidad.

- Permitir movimientos que empeoren momentáneamente la solución actual.

El primer mecanismo da lugar a estrategias de *múltiples soluciones iniciales* (*Multi-Start*), que ejecutan varias búsquedas locales partiendo de distintas soluciones iniciales.

El segundo mecanismo corresponde a la *búsqueda por vecindarios variables* (*Variable Neighborhood Search*, VNS), que cambia sistemáticamente la estructura de vecindarios empleada, cada vez que la búsqueda se estanca, en lugar de utilizar una única estructura fija. Este cambio de vecindarios viene dado por el cambio de movimiento a emplear a la solución actual, permitiendo explorar distintas regiones del espacio de soluciones debido a la naturaleza de los movimientos empleados. Una variante determinista de esta idea es la *búsqueda descendente por vecindarios variables* (*Variable Neighborhood Descent*, VND), que utiliza una secuencia ordenada de estructuras de vecindarios  $E_1, E_2, \dots, E_k$ . VND comienza aplicando búsqueda reparadora sobre  $E_1$  hasta alcanzar un óptimo local; si no se logra ninguna mejora en la vecindad  $E_i$ , cambia a la siguiente estructura  $E_{i+1}$ , mientras que cada vez que se produce una mejora regresa a  $E_1$ . La búsqueda finaliza al alcanzar un óptimo local respecto a todas las estructuras consideradas. VNS extiende esta idea incorporando además una fase de *agitación* (*shaking*), que perturba aleatoriamente la solución dentro de la estructura de vecindario actual antes de aplicar la búsqueda reparadora, lo que le permite escapar de óptimos locales en lugar de solo alternar entre estructuras (H. Hoos y Stützle, 2015).

El tercer mecanismo, aceptar movimientos que no mejoran la solución, es el fundamento de dos de las técnicas de búsqueda reparadora más utilizadas en la literatura:

- Recocido Simulado (*Simulated Annealing*, SA): Esta técnica puede aceptar una solución vecina de peor calidad dependiendo de una probabilidad que decrece exponencialmente en función del empeoramiento producido y de un parámetro de temperatura  $T$ , que se reduce progresivamente durante la búsqueda. La función estándar aplicada en estos procedimientos es la siguiente:

$$P(\Delta f, T) = \exp\left(-\frac{\Delta f}{T}\right). \quad (3.1)$$

Cuando la temperatura es alta, la búsqueda tiene una mayor probabilidad de aceptar soluciones peores, pudiendo comportarse de forma casi aleatoria, permitiendo así favorecer la exploración en las primeras iteraciones. A medida que  $T$  se acerca a cero se aproxima a una búsqueda reparadora pura, favoreciendo la explotación y llegando finalmente a un óptimo local.

- **Búsqueda Tabú (*Tabu Search*)**: Este método emplea una lista tabú, en la que almacena el recorrido de la búsqueda, para prohibir temporalmente revertir los últimos movimientos aplicados y volver a una solución anterior. El objetivo es evitar ciclos y la reiteración en una misma zona del espacio de búsqueda. Esta restricción puede anularse mediante un criterio de aspiración cuando el movimiento tabú conduce a una solución mejor que la mejor conocida hasta el momento.

Además, estas técnicas necesitan almacenar la mejor solución obtenida hasta el momento, ya que debido a que pueden aceptar soluciones de peor calidad, podrían llegar a perderla al terminar su ejecución en un óptimo local de baja calidad.

Un procedimiento relacionado corresponde a la *Búsqueda Local Dinámica (Dynamic Local Search, DLS)*. Esta, en vez de aceptar movimientos que empeoran la solución actual, modifica la función de evaluación cada vez que la búsqueda alcanza un óptimo local, agregando penalizaciones a los componentes de la solución involucrados en dicho óptimo.

Otra estrategia para continuar la búsqueda más allá de un óptimo local es alternar entre fases de **perturbación** y de búsqueda reparadora, dando lugar a la **Búsqueda Local Reiterada (*Iterated Local Search, ILS*)**. Un algoritmo ILS se compone de cuatro elementos: un procedimiento de generación de la solución inicial, una búsqueda reparadora, un procedimiento de perturbación que modifica la solución actual para escapar del óptimo local alcanzado, y un criterio de aceptación que decide si continuar la búsqueda desde la solución perturbada y reparada o desde la solución anterior. El Algoritmo 2 resume su funcionamiento general.

Un aspecto relevante del diseño de ILS es que la solución generada por la perturbación debe ser suficientemente distinta de la solución actual para poder escapar del valle que llevaría a la búsqueda reparadora a obtener nuevamente el mismo óptimo local.

### 3.3. Metaheurísticas poblacionales

Hasta el momento se han descrito técnicas que recorren el espacio de búsqueda a través de una única trayectoria de soluciones. Un paradigma distinto es el de las *metaheurísticas evolutivas o poblacionales* (Talbi, 2009), que sustituyen la trayectoria única por un conjunto o *población* de soluciones que evolucionan conjuntamente. Los algoritmos genéticos, representativos de esta familia, codifican cada solución como un individuo sobre el que



**Algoritmo 2** Esquema general de ILS

---

```

1: procedure ILS( $maxIter$ )
2:    $sol \leftarrow$  CONSTRUCCIONINICIAL
3:    $sol \leftarrow$  BUSQUEDALOCAL( $sol$ )
4:   for  $iter = 1$  hasta  $maxIter$  do
5:      $sol' \leftarrow$  PERTURBAR( $sol$ )
6:      $sol' \leftarrow$  BUSQUEDALOCAL( $sol'$ )
7:      $sol \leftarrow$  CRITERIODEACEPTACION( $sol, sol'$ )
8:   end for
9:   return  $sol$ 
10: end procedure

```

---

se aplican operaciones de mutación, cruce y selección para producir nuevas soluciones a partir de las existentes. Cada uno de estos tiene un propósito definido:

- Operadores de mutación: Estos trabajan a nivel de individuo introduciendo una modificación de la solución, permitiendo así explorar el espacio de búsqueda. Generalmente se les asigna una probabilidad baja de ocurrir.
- Operadores de cruce: Estos trabajan con dos o mas individuos para generar uno o más descendientes que heredan características de ambos al combinar su material genético. Tiene el objetivo de combinar componentes prometedores de distintas soluciones en una nueva. A diferencia de la mutación, suele aplicarse con una probabilidad alta y su diseño depende fuertemente de la representación utilizada. Por ejemplo, en soluciones que corresponden a una secuencia o permutación de elementos, como ocurre en problemas de enrutamiento, un cruce ingenuo puede generar descendientes infactibles al repetir o eliminar elementos, por lo que se recurre a operadores especializados que preservan la estructura de la solución. Un ejemplo de operador aplicado a secuencias se puede ver en la Figura 3.4.
- Proceso de selección: Una vez aplicados los operadores, se aplica un proceso de selección sobre la población intermedia generada para definir la nueva población.

Además, estas técnicas pueden incluir como característica el *elitismo*, que consiste en conservar soluciones de buena calidad de la población actual para no perder su material genético.

Los *algoritmos meméticos* combinan esta estructura evolutiva poblacional con una búsqueda reparadora aplicada a cada descendiente, con el objetivo de intensificar la búsqueda alrededor de las soluciones generadas por los operadores evolutivos, siendo un híbrido de los dos paradigmas expuestos.

Finalmente, cabe destacar que el diseño de un algoritmo de búsqueda local estocástica no termina en la elección de sus componentes. Cada uno de los mecanismos revisados en este capítulo introduce parámetros cuyo ajuste tiene un impacto determinante en el desempeño del algoritmo resultante.

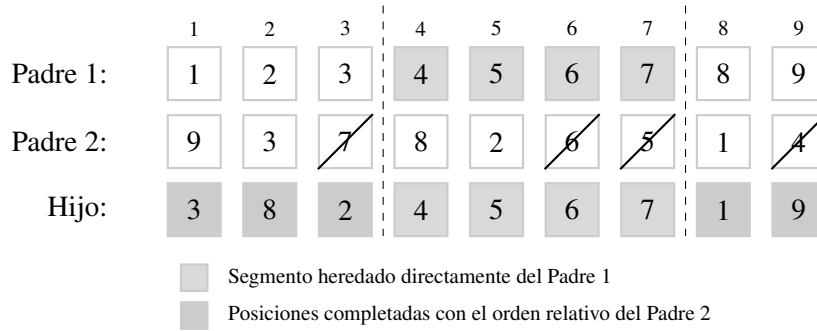


Figura 3.4: Ejemplo del operador de cruce de orden (*Order Crossover*, OX). El segmento entre las dos posiciones de corte se copia directamente del Padre 1 al Hijo, conservando su posición. Las posiciones restantes se completan recorriendo el Padre 2 en orden, comenzando justo después del segundo punto de corte y de forma circular, omitiendo los elementos que ya fueron heredados del Padre 1 (tachados en el Padre 2). Esto garantiza que el descendiente sea una permutación válida, sin elementos repetidos ni faltantes.

### 3.4. Resumen del Capítulo

En este capítulo se presentaron los conceptos fundamentales de la búsqueda local estocástica, distinguiendo entre búsqueda constructiva y búsqueda reparadora. Ambas comparten una misma base formal, correspondiente a un espacio de soluciones  $S$ , una función de evaluación  $f$  y una estructura de vecindarios  $E$ . Además, en su gran mayoría incorporan algún componente estocástico que les permite explorar distintas regiones del espacio de búsqueda en ejecuciones distintas, resolviendo una misma instancia.

Dentro de la búsqueda constructiva se destacó GRASP, que introduce aleatoriedad controlada en la estrategia voraz mediante una función miope y una LRC, generando así soluciones iniciales diversas y de buena calidad. Dentro de la búsqueda reparadora se revisó el esquema básico de *Hill-climbing* y los mecanismos que le permiten escapar de óptimos locales de baja calidad, además de las metaheurísticas más conocidas.

Cabe destacar que estas técnicas no son mutuamente excluyentes y pueden combinarse, dando lugar a esquemas híbridos o colaborativos. Esta es la idea de la propuesta de esta tesis presentada en el Capítulo 5, combinando una fase constructiva y una fase reparadora. En el siguiente capítulo se revisan las técnicas que han sido aplicadas para resolver el CTTRP y variantes de este.

## Capítulo 4

# Estado del arte del Capacitated Truck and Trailer Routing Problem

En este capítulo se hará una revisión de los principales algoritmos que existen en la literatura para resolver el CTTRP, así como de versiones de problemas relacionados. Cabe destacar que la literatura original no distingue explícitamente las restricciones de capacidad en su nomenclatura, por lo que los trabajos revisados en este capítulo se refieren al problema simplemente como TTRP. Sin embargo, en esta tesis se utiliza la denominación CTTRP para enfatizar dichas restricciones, tratándose del mismo problema.

### 4.1. Capacitated Truck and Trailer Routing Problem

El TTRP fue presentado por primera vez en (Chao, 2002). En ese trabajo, se proponen instancias del problema derivadas a partir de instancias conocidas del VRP propuestas por (Christofides, Nicos, 1976), variando el número de clientes entre 50 y 199, completando un total de 21 instancias. En ese trabajo se propone un algoritmo de dos fases, que comienza con un método de construcción de la solución inicial, que consiste en resolver un problema de asignación generalizada relajado para asignar los clientes a uno de los tres tipos de rutas, seguido de la construcción de rutas y una mejora por descenso. Posterior a esto, se intenta cambiar a los clientes a una nueva ruta, con el objetivo de convertir una solución que generalmente es infactible en una factible. Para esto se considera una penalización  $\theta$  que pondere la infactibilidad.

Estas soluciones se utilizan luego en una heurística de mejora basada en búsqueda tabú, junto con el concepto de desviación proveniente del recocido determinista (*deterministic annealing*) incluyendo los siguientes movimientos:

- *One-point descent movement*: La finalidad es mover un cliente de una ruta a otra. Si el movimiento candidato decrementa la penalización con o sin un aumento de distancia, o bien disminuye la distancia sin aumentar la penalización, entonces se ejecuta inmediatamente el cambio. Hay dos movimientos que deben ser excluidos: El primero es mover un cliente de camión a un tour principal perteneciente a una

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

ruta de vehículo completa, o bien a una ruta de vehículo pura. El segundo es mover un cliente de vehículo, que es el estacionamiento de un sub-tour (cliente raíz).

- *Two-point descent exchange*: En este movimiento se pueden intercambiar dos clientes de rutas diferentes. Se consideran las mismas restricciones que en el movimiento anterior.
- *Sub-tour root-refining step*: Este movimiento se incluye para dar la posibilidad de cambiar los orígenes de los sub-tours, lo que podría llevar a una mejor solución. En este se intenta intercambiar los nodos de origen de un sub-tour y hacer una re-secuenciación de los sub-tours, eliminando el primer y último arco que lo conectan con la ruta principal, probando a conectar otro par de clientes consecutivos.

En la Tabla 4.1 se muestran las características de las instancias generadas, que conforman el benchmark más conocido del problema.

Tabla 4.1: Características de las 21 instancias TTRP propuestas por Chao, derivadas de los problemas CMT del VRP

| Instancia | CMT original | Cientes veh. | Camión | Camiones Núm. | Cap. | Remolques Núm. | Cap. | Razón dem./cap. |
|-----------|--------------|--------------|--------|---------------|------|----------------|------|-----------------|
| 1         |              | 38           | 12     |               |      |                |      |                 |
| 2         |              | 25           | 25     |               |      |                |      |                 |
| 3         | CMT1         | 13           | 37     | 5             | 100  | 3              | 100  | 0.971           |
| 4         |              | 57           | 18     |               |      |                |      |                 |
| 5         |              | 38           | 37     |               |      |                |      |                 |
| 6         | CMT2         | 19           | 56     | 9             | 100  | 5              | 100  | 0.974           |
| 7         |              | 75           | 25     |               |      |                |      |                 |
| 8         |              | 50           | 50     |               |      |                |      |                 |
| 9         | CMT3         | 25           | 75     | 8             | 150  | 4              | 100  | 0.911           |
| 10        |              | 113          | 37     |               |      |                |      |                 |
| 11        |              | 75           | 75     |               |      |                |      |                 |
| 12        | CMT4         | 38           | 112    | 12            | 150  | 6              | 100  | 0.931           |
| 13        |              | 150          | 49     |               |      |                |      |                 |
| 14        |              | 100          | 99     |               |      |                |      |                 |
| 15        | CMT5         | 50           | 149    | 17            | 150  | 9              | 100  | 0.923           |
| 16        |              | 90           | 30     |               |      |                |      |                 |
| 17        |              | 60           | 60     |               |      |                |      |                 |
| 18        | CMT11        | 30           | 90     | 7             | 150  | 4              | 100  | 0.948           |

Tabla 4.1: Características de las 21 instancias TTRP propuestas por Chao, derivadas de los problemas CMT del VRP (continuación)

| Núm.<br>problema | CMT<br>original | Cientes<br>v.c. | Cientes<br>t.c. | Camiones<br>Núm. | Camiones<br>Cap. | Remolques<br>Núm. | Remolques<br>Cap. | Razón<br>dem./cap. |
|------------------|-----------------|-----------------|-----------------|------------------|------------------|-------------------|-------------------|--------------------|
| 19               |                 | 75              | 25              |                  |                  |                   |                   |                    |
| 20               |                 | 50              | 50              |                  |                  |                   |                   |                    |
| 21               | CMT12           | 25              | 75              | 10               | 150              | 5                 | 100               | 0.903              |

#### 4.1.1. Métodos de resolución

En (Scheuerer, 2006) se presentan dos heurísticas de construcción: una basada en un procedimiento de inserción secuencial por clústeres, y otra basada en el algoritmo de barrido (*sweep algorithm*). El primer procedimiento usa una heurística denominada *T-Cluster* en la que se emplea un procedimiento de inserción secuencial basado en grupos, donde las rutas se construyen cliente a cliente hasta la utilización total de la capacidad del vehículo. Luego, se inicializa una nueva ruta con el cliente sin asignar más alejado del depósito y un vehículo no utilizado con mayor capacidad. En caso de que sea un vehículo completo, si el cliente inicial es de vehículo, entonces se inserta en el recorrido principal, mientras que si es un cliente de camión, se inserta en un nuevo sub-tour. El siguiente cliente para ser insertado en la ruta se elige minimizando la siguiente ecuación:

$$e(k) = c_{ku} + c_{kf} - \pi c_{0k} \quad (4.1)$$

Donde  $k$  especifica el cliente a insertar,  $f$  es el cliente más cercano añadido a la ruta y 0 corresponde al depósito. El término  $c_{ku}$  mide la distancia entre el cliente más cercano que forma parte de la ruta  $u$  y el cliente  $k$ . Esta distancia debe ser la mínima posible para garantizar que la ruta sea compacta. El segundo término,  $c_{kf}$  asegura que el cliente esté cerca del borde de la ruta actual. Por último, el término  $\pi c_{0k}$  es denominado término de diversificación, en donde la variación del parámetro  $\pi$  resultará en la selección de una nueva estrategia, pudiendo elegir clientes que estén más lejanos del depósito.

La segunda heurística, *T-Sweep*, se basa en el algoritmo clásico *Sweep* (Gillett y Miller, 1974). Esta técnica de barrido es utilizada para construir soluciones factibles al girar un rayo centrado en el depósito e incluyendo gradualmente a los clientes en una ruta de vehículos. En este se inicializa una nueva ruta con el vehículo no utilizado de mayor capacidad total, y en caso de que no haya más vehículos disponibles para agregar un cliente, se permite que la última ruta sea infactible.

En la Figura 4.1 se puede ver una representación de cómo actúan estos procedimientos para construir una solución.

La etapa de reparación se realiza mediante una heurística de búsqueda tabú que incluye una función de evaluación que permite explorar soluciones infactibles e implementa un reinicio de la búsqueda, con un límite de tres reinicios en caso de que se obtenga el mismo óptimo local. Con esto, se logran mejores soluciones para el conjunto de instancias CHAO.

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

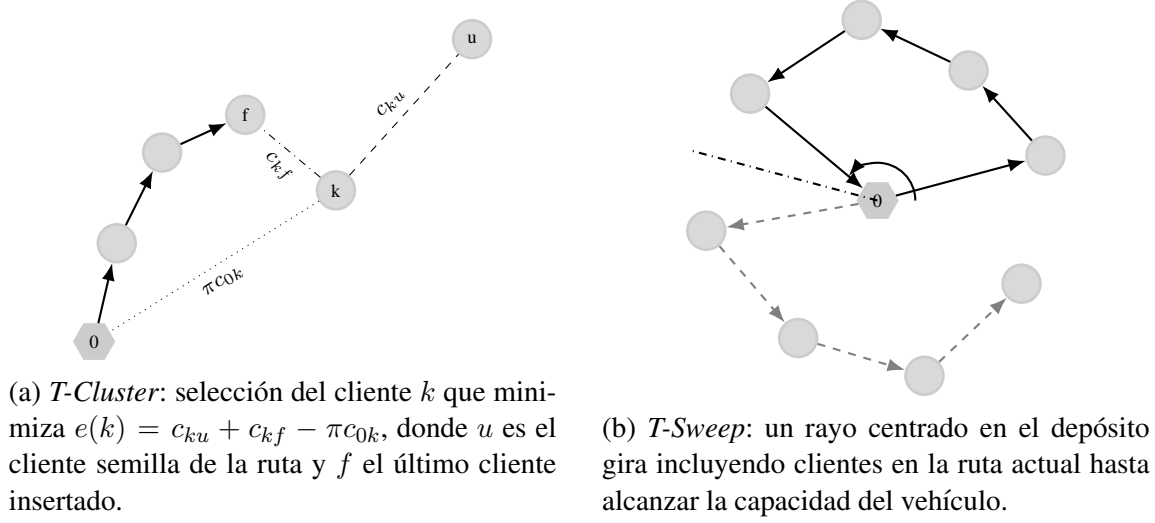


Figura 4.1: Esquema de heurísticas de construcción *T-Cluster* y *T-Sweep*

En (S.-W. Lin, Yu, y Chou, 2009) se propone la primera heurística basada en recocido simulado (*simulated annealing*) para el TTRP, con la propuesta de una representación ad-hoc al problema y tres movimientos e incluye además el movimiento de 2-Opt (S. Lin, 1965). Los movimientos usados son los siguientes:

- Un movimiento de inserción que selecciona un cliente y se inserta justo antes de otro cliente en la solución.
- Un movimiento de intercambio en el que se seleccionan dos clientes y se intercambian sus posiciones.
- Un movimiento específico para los clientes de vehículo. Dado que los clientes de vehículo pueden ser atendidos tanto por un vehículo completo como por un camión, el algoritmo cambia su tipo de vehículo asignado, pasando a ser asignado a sub-tours o rutas principales según corresponda.

Además, se intenta mejorar la búsqueda realizando  $N$  intentos, seleccionando el que produce la mejor solución. El algoritmo logra 11 nuevas mejores soluciones conocidas para el conjunto de instancias CHAO. Posteriormente, en (S.-W. Lin, Yu, y Chou, 2010) se introduce una variante, el TTRP Relajado, que elimina la restricción sobre el número de vehículos disponibles por instancia, con el fin de encontrar rutas de menor costo y mejores combinaciones de camiones y remolques. En éste se propone un nuevo algoritmo de recocido simulado basado en su trabajo previo obteniendo 18 mejores soluciones y se genera un nuevo conjunto de instancias basado en instancias conocidas del VRP, propuestas por Taillard ((Rochat y Taillard, 1995)) con 75, 100 y 150 clientes, denominado conjunto de instancias TAI en referencia a su origen.

En (Villegas, Prins, Prodhon, Medaglia, y Velasco, 2011) se propone un procedimiento de búsqueda voraz aleatorizada adaptativa (GRASP), mejorado con una búsqueda de vecindarios variables (VNS) y *path-relinking*. La adición de *path-relinking* tiene como objetivo

mitigar una característica del procedimiento GRASP tradicional, correspondiente a no tener memoria. Para esto, el algoritmo mantiene un conjunto de soluciones elite que permite almacenar y reutilizar información de soluciones prometedoras, para guiar la explotación del espacio de búsqueda.

En este trabajo, a diferencia de otros enfoques en los que se agrupa primero a los clientes, se construye una gran ruta que visita todos los clientes, ignorando las capacidades de los vehículos y las restricciones de acceso. Luego, en la fase de reparación, VNS cumple el rol de buscar mejores soluciones y reparar soluciones infactibles. En éste se emplean los siguientes cinco movimientos:

- Or-opt modificado: Similar al movimiento Or-opt, en el que se extrae una cadena de clientes y se trata de insertar en otra posición. La diferencia radica en que se evalúa simultáneamente la inclusión de la cadena en sentido contrario.
- Intercambio de dos nodos: Se intercambia la posición de dos nodos verificando las restricciones de capacidad de ambas rutas y las restricciones de acceso de cada cliente. Además, considera la posibilidad de intercambiar clientes en una misma ruta o sub-tour o entre rutas distintas.
- 2-opt: Se extrae una cadena de clientes eliminando los arcos que lo conectan a la ruta principal e invierte su orden. Además del enfoque tradicional para una sola ruta, se permite aplicar sobre dos arcos de rutas diferentes. Si los arcos pertenecen a un par de sub-tours, se asigna además el mejor origen posible a los sub-tours originales en sus rutas, considerando que no se exceda la capacidad de los vehículos.
- Reubicación de nodos: Este movimiento selecciona un nodo y lo reubica en otra posición, ya sea de la misma ruta u otra, considerando restricciones de capacidad y tipo de cliente.
- Refinamiento de raíces: Se aplica el *Sub-tour root-refining step* descrito en (Chao, 2002).

En este trabajo se considera además un umbral de penalización dinámico que permite que se exceda la capacidad de los vehículos o bien el número de vehículos disponibles, lo que ayuda a escapar de óptimos locales. Con su aproximación lograron obtener 12 mejores soluciones del conjunto de instancias CHAO.

Posteriormente, en (Villegas, Prins, Prodhon, Medaglia, y Velasco, 2013), basándose en su trabajo previo, los autores proponen una matheurística de dos fases que utiliza las rutas obtenidas mediante una heurística híbrida GRASP/ILS como columnas en una formulación de partición de conjuntos (*set-partitioning*) para el problema. En este trabajo la matheurística actúa como un generador de columnas que genera soluciones que son almacenadas en caso de tener buena calidad, incluyendo soluciones infactibles. Posterior a esto, la fase de optimización se encarga de seleccionar y combinar estas rutas para obtener una solución global óptima respetando las restricciones de capacidad de la flota de vehículos. Sus experimentos incluyen el conjunto de instancias TAI como un segundo banco de

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

pruebas en el que se obtuvo un nuevo óptimo para las 36 instancias del conjunto, mientras que se obtuvieron 15 nuevas cotas para el conjunto de instancias CHAO.

En (Derigs, Pullmann, y Vogel, 2013) se presenta una revisión de la literatura relacionada con el TTRP hasta ese momento. La revisión considera variantes del problema que permiten o no la transferencia de carga entre camiones y remolques. Los autores proponen un enfoque tradicional de búsqueda local y una búsqueda de vecindarios amplios (*large neighborhood search*), ambos partiendo de soluciones factibles, y utilizan el conjunto de instancias CHAO para evaluarlos, obteniendo 4 nuevas mejores soluciones.

En (Bartolini y Schneider, 2020) se introduce una formulación exacta para el problema basada en una formulación de flujo de dos productos (*two-commodity flow*) presentada en la Sección 2.1, en la que se utilizan dos conjuntos de variables de flujo para modelar el flujo de carga de los vehículos y los remolques. A partir de esta formulación se presenta un algoritmo de branch-and-cut, y se genera un nuevo conjunto de instancias, denominado conjunto de instancias BART, compuesto por 132 instancias con entre 25 y 40 clientes. Estas instancias incluyen diversas configuraciones de flota de camiones inspiradas en las propuestas por (Chao, 2002) y (S.-W. Lin y cols., 2010).

Para generar estas instancias se adaptaron las instancias de CHAO y TAI. Las primeras son instancias en las que los clientes están distribuidos uniformemente en el espacio, mientras que las segundas se ubican en clusters. Esta diferencia se puede observar en la Figura 4.2. Con éstas, debido a que se propone un método exacto, solo se consideran sets de 30 clientes, a partir de las instancias originales. Las instancias usadas fueron generadas de la siguiente manera:

- Se toman los primeros  $n_t = \lceil 0,25 \cdot 30 \rceil$  clientes de camión y los primeros  $30 - n_t$  clientes de vehículo de la primera instancia de cada conjunto.
- De la misma manera, para la segunda instancia se toman los primeros  $n_t = \lceil 0,5 \cdot 30 \rceil$  clientes de camión.
- Para la tercera instancia de cada conjunto, la proporción de clientes de camión tomados sube a 75 %.

Esto se repite con cada una de las instancias de cada conjunto.

Una vez seleccionado el conjunto de clientes, para cada instancia se asignó la flota de camiones de distinta manera, dividiéndose en tres tipos:

- Instancias de tipo A: El número de vehículos disponible así como su capacidad mantiene una razón similar a la demanda y capacidad total que en las instancias originales.
- Instancias de tipo B: Derivadas de las instancias de tipo A, pero se aumenta la capacidad de los vehículos.
- Instancias de tipo C: Solo se considera un único camión con un remolque.



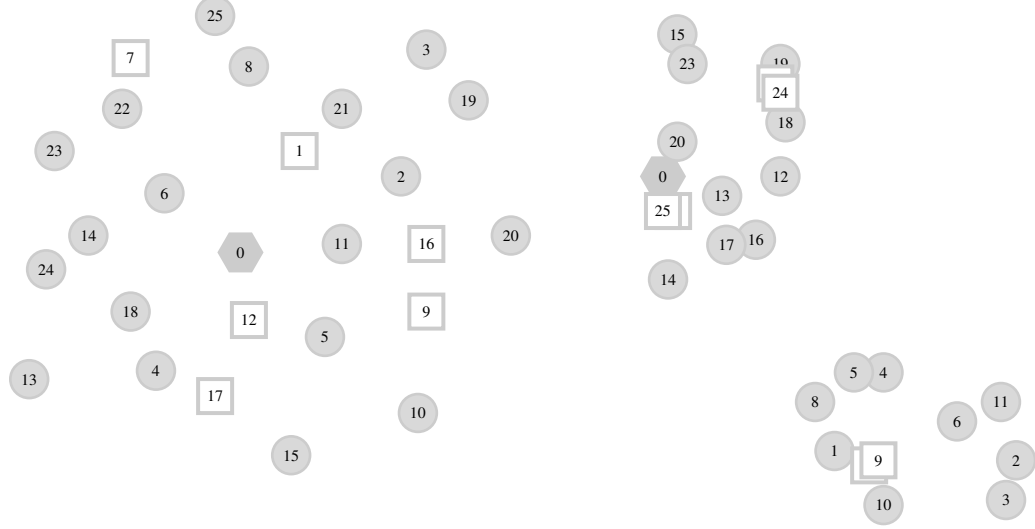


Figura 4.2: Comparación de la distribución espacial de clientes entre las primeras instancias de 25 clientes generadas a partir del conjunto de instancias CHAO (izquierda) y TAI (derecha)

En las Tablas 4.2, 4.3, 4.4 y 4.5 se presentan las características de las instancias. Se indica el número de clientes de vehículo ( $n_c$ ), número de remolques ( $m_c$ ) y su capacidad ( $Q_c$ ), número de camiones ( $m_t$ ) y su capacidad ( $Q_t$ ), mientras que  $\%q$  es la razón porcentual entre la demanda de los clientes y la capacidad total de la flota disponible. Para las instancias de tipo C no se reporta  $m_c$ ,  $m_t$  y  $Q_c$ , mientras que la columna  $\%q$  es reemplazada por la columna  $q_t$ , reportando la demanda total de los clientes de camión.

Tabla 4.2: Características de las instancias del conjunto *Chao30* (30 clientes), tipos A y B

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | $\%q$ |
|-----------|-------|-------|-------|-------|-------|-------|
| A-1       | 23    | 2     | 100   | 3     | 100   | 97.00 |
| A-2       | 15    | 2     | 100   | 3     | 100   | 96.60 |
| A-3       | 7     | 2     | 100   | 3     | 100   | 96.20 |
| A-4       | 23    | 2     | 100   | 4     | 100   | 95.67 |
| A-5       | 15    | 2     | 100   | 4     | 100   | 98.50 |
| A-6       | 8     | 2     | 100   | 4     | 100   | 99.33 |
| A-7       | 22    | 2     | 100   | 2     | 150   | 86.40 |
| A-8       | 15    | 1     | 150   | 2     | 150   | 92.67 |
| A-9       | 8     | 2     | 100   | 2     | 150   | 87.00 |
| A-10      | 22    | 2     | 100   | 3     | 100   | 81.00 |
| A-11      | 15    | 2     | 100   | 3     | 100   | 82.20 |
| A-12      | 8     | 2     | 100   | 3     | 100   | 80.00 |
| B-1       | 23    | 2     | 150   | 2     | 150   | 80.83 |
| B-2       | 15    | 2     | 150   | 2     | 150   | 80.50 |

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

Tabla 4.2: Características de las instancias del conjunto *Chao30* (30 clientes), tipos A y B (continuación)

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| B-3       | 7     | 2     | 150   | 2     | 150   | 80.17 |
| B-4       | 23    | 2     | 150   | 2     | 200   | 82.00 |
| B-5       | 15    | 2     | 150   | 2     | 200   | 84.43 |
| B-6       | 8     | 2     | 150   | 2     | 200   | 85.14 |
| B-7       | 22    | 2     | 150   | 2     | 200   | 61.71 |
| B-8       | 15    | 2     | 150   | 2     | 200   | 59.57 |
| B-9       | 8     | 2     | 150   | 2     | 200   | 62.14 |
| B-10      | 22    | 2     | 150   | 2     | 200   | 57.86 |
| B-11      | 15    | 2     | 150   | 2     | 200   | 58.71 |
| B-12      | 8     | 2     | 150   | 2     | 200   | 57.14 |

Tabla 4.3: Características de las instancias del conjunto *Chao30*, tipo C (un camión y un remolque)

| Instancia | $n_c$ | $Q_t$ | $q_t$ |
|-----------|-------|-------|-------|
| C-1       | 23    | 150   | 93    |
| C-2       | 15    | 150   | 224   |
| C-3       | 7     | 150   | 341   |
| C-4       | 23    | 150   | 113   |
| C-5       | 15    | 150   | 276   |
| C-6       | 8     | 150   | 422   |
| C-7       | 22    | 200   | 128   |
| C-8       | 15    | 100   | 217   |
| C-9       | 8     | 200   | 326   |
| C-10      | 22    | 100   | 93    |
| C-11      | 15    | 100   | 187   |
| C-12      | 8     | 100   | 317   |

Tabla 4.4: Características de las instancias del conjunto *Tai30* (30 clientes), tipos A y B

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| A-1       | 23    | 3     | 750   | 6     | 750   | 90.16 |
| A-2       | 15    | 3     | 750   | 5     | 750   | 87.30 |
| A-3       | 8     | 3     | 750   | 5     | 750   | 87.65 |
| A-4       | 23    | 4     | 850   | 5     | 850   | 92.82 |
| A-5       | 15    | 4     | 850   | 6     | 850   | 91.86 |

CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER  
ROUTING PROBLEM

Tabla 4.4: Características de las instancias del conjunto *Tai30* (30 clientes), tipos A y B (continuación)

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| A-6       | 8     | 3     | 850   | 8     | 850   | 93.93 |
| A-7       | 23    | 3     | 600   | 5     | 600   | 89.58 |
| A-8       | 15    | 3     | 600   | 5     | 600   | 89.58 |
| A-9       | 7     | 3     | 600   | 7     | 600   | 71.67 |
| A-10      | 23    | 3     | 850   | 5     | 850   | 91.66 |
| A-11      | 15    | 2     | 850   | 5     | 850   | 92.32 |
| A-12      | 7     | 3     | 850   | 5     | 850   | 91.32 |
| B-1       | 23    | 3     | 1000  | 6     | 1000  | 67.62 |
| B-2       | 15    | 3     | 1000  | 5     | 1000  | 65.48 |
| B-3       | 8     | 3     | 1000  | 5     | 1000  | 65.74 |
| B-4       | 23    | 4     | 1000  | 5     | 1000  | 78.90 |
| B-5       | 15    | 4     | 1000  | 6     | 1000  | 78.08 |
| B-6       | 8     | 3     | 1000  | 8     | 1000  | 79.84 |
| B-7       | 23    | 3     | 900   | 5     | 900   | 59.72 |
| B-8       | 15    | 3     | 900   | 5     | 900   | 59.72 |
| B-9       | 7     | 3     | 900   | 7     | 900   | 47.78 |
| B-10      | 23    | 3     | 1000  | 5     | 1000  | 77.91 |
| B-11      | 15    | 2     | 1000  | 5     | 1000  | 78.47 |
| B-12      | 7     | 3     | 1000  | 5     | 1000  | 77.63 |

Tabla 4.5: Características de las instancias del conjunto *Tai30*, tipo C (un camión y un remolque)

| Instancia | $n_c$ | $Q_t$ | $q_t$ |
|-----------|-------|-------|-------|
| C-1       | 23    | 750   | 1549  |
| C-2       | 15    | 750   | 2288  |
| C-3       | 8     | 750   | 3036  |
| C-4       | 23    | 850   | 1867  |
| C-5       | 15    | 850   | 2478  |
| C-6       | 8     | 850   | 4779  |
| C-7       | 23    | 600   | 262   |
| C-8       | 15    | 600   | 916   |
| C-9       | 7     | 600   | 1358  |
| C-10      | 23    | 850   | 1369  |
| C-11      | 15    | 850   | 2986  |
| C-12      | 7     | 850   | 4953  |

Se realiza un proceso similar para generar instancias con 40 clientes a partir del mismo

#### CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

conjunto de instancias y sus características reportadas en el trabajo se presentan en las Tablas 4.6 y 4.7. Por último, se crearon dos conjuntos de instancias pequeños extrayendo los primeros 25 clientes de las instancias de tipo A de 30 clientes generadas. En las Tablas 4.8 y 4.9 se presentan sus características.

Tabla 4.6: Características de las instancias del conjunto *Chao40* (40 clientes), tipos A y B

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| A-1       | 30    | 3     | 100   | 4     | 100   | 90.86 |
| A-2       | 20    | 3     | 100   | 4     | 100   | 92.14 |
| A-3       | 10    | 3     | 100   | 4     | 100   | 89.29 |
| A-4       | 30    | 4     | 100   | 4     | 100   | 96.38 |
| A-5       | 20    | 4     | 100   | 5     | 100   | 88.67 |
| A-6       | 10    | 3     | 100   | 6     | 100   | 88.67 |
| A-7       | 30    | 2     | 100   | 3     | 150   | 92.31 |
| A-8       | 20    | 1     | 150   | 3     | 150   | 89.00 |
| A-9       | 10    | 2     | 100   | 3     | 150   | 87.54 |
| A-10      | 30    | 3     | 100   | 3     | 100   | 93.83 |
| A-11      | 20    | 2     | 100   | 4     | 100   | 89.50 |
| A-12      | 10    | 3     | 100   | 3     | 100   | 89.17 |
| B-1       | 30    | 3     | 100   | 3     | 150   | 84.80 |
| B-2       | 20    | 2     | 150   | 3     | 150   | 86.00 |
| B-3       | 10    | 3     | 150   | 3     | 150   | 69.44 |
| B-4       | 30    | 2     | 100   | 3     | 200   | 96.38 |
| B-5       | 20    | 2     | 150   | 3     | 200   | 88.67 |
| B-6       | 10    | 2     | 150   | 3     | 200   | 88.67 |
| B-7       | 30    | 2     | 150   | 2     | 200   | 85.71 |
| B-8       | 20    | 2     | 150   | 2     | 200   | 76.29 |
| B-9       | 10    | 2     | 150   | 2     | 200   | 81.29 |
| B-10      | 30    | 2     | 150   | 2     | 200   | 80.43 |
| B-11      | 20    | 2     | 150   | 2     | 200   | 76.71 |
| B-12      | 10    | 2     | 150   | 2     | 200   | 76.43 |

Tabla 4.7: Características de las instancias del conjunto *Chao40*, tipo C (un camión y un remolque)

| Instancia | $n_c$ | $Q_t$ | $q_t$ |
|-----------|-------|-------|-------|
| C-1       | 30    | 150   | 134   |
| C-2       | 20    | 150   | 309   |
| C-3       | 10    | 150   | 449   |
| C-4       | 30    | 150   | 157   |

CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER  
ROUTING PROBLEM

Tabla 4.7: Características de las instancias del conjunto *Chao40*, tipo C (continuación)

| Instancia | $n_c$ | $Q_t$ | $q_t$ |
|-----------|-------|-------|-------|
| C-5       | 20    | 150   | 375   |
| C-6       | 10    | 150   | 577   |
| C-7       | 30    | 200   | 162   |
| C-8       | 20    | 100   | 262   |
| C-9       | 10    | 200   | 434   |
| C-10      | 30    | 100   | 117   |
| C-11      | 20    | 100   | 253   |
| C-12      | 10    | 100   | 440   |

Tabla 4.8: Características de las instancias del conjunto *Chao25* (25 clientes)

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| 1         | 19    | 2     | 100   | 3     | 100   | 84.40 |
| 2         | 13    | 2     | 100   | 3     | 100   | 84.80 |
| 3         | 6     | 2     | 100   | 3     | 100   | 84.20 |
| 4         | 19    | 2     | 100   | 4     | 100   | 82.17 |
| 5         | 12    | 2     | 100   | 4     | 100   | 86.17 |
| 6         | 6     | 2     | 100   | 4     | 100   | 80.83 |
| 7         | 18    | 2     | 100   | 2     | 150   | 68.20 |
| 8         | 12    | 2     | 150   | 1     | 150   | 76.67 |
| 9         | 7     | 2     | 100   | 2     | 150   | 69.00 |
| 10        | 20    | 2     | 100   | 3     | 100   | 66.40 |
| 11        | 12    | 2     | 100   | 3     | 100   | 66.40 |
| 12        | 6     | 2     | 100   | 2     | 100   | 66.40 |

Tabla 4.9: Características de las instancias del conjunto *Tai25* (25 clientes)

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| 1         | 19    | 3     | 750   | 6     | 750   | 64.28 |
| 2         | 13    | 3     | 750   | 5     | 750   | 81.25 |
| 3         | 6     | 3     | 750   | 5     | 750   | 65.08 |
| 4         | 19    | 4     | 850   | 5     | 850   | 84.77 |
| 5         | 13    | 4     | 850   | 6     | 850   | 88.91 |
| 6         | 6     | 3     | 850   | 7     | 850   | 94.24 |
| 7         | 19    | 2     | 600   | 5     | 600   | 63.45 |
| 8         | 13    | 2     | 600   | 5     | 600   | 64.05 |
| 9         | 5     | 2     | 600   | 5     | 600   | 63.98 |

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

Tabla 4.9: Características de las instancias del conjunto *Tai25* (continuación)

| Instancia | $n_c$ | $m_c$ | $Q_c$ | $m_t$ | $Q_t$ | %q    |
|-----------|-------|-------|-------|-------|-------|-------|
| 10        | 19    | 3     | 850   | 5     | 850   | 75.34 |
| 11        | 13    | 2     | 850   | 5     | 850   | 80.00 |
| 12        | 5     | 3     | 850   | 5     | 850   | 61.32 |

El algoritmo se ejecuta con una ventana de dos horas para cada instancia y logra encontrar soluciones óptimas para un gran porcentaje de estas instancias, siendo un resultado importante en la literatura al constituir el primer trabajo en el que se propone un método exacto eficiente para el problema.

Por otro lado, se prueba la versatilidad de su enfoque de dos maneras. Primero, provee buenos resultados tanto para instancias que consideran un solo vehículo o una configuración con múltiples camiones y remolques, además de considerar la prohibición de transferencia de carga del remolque al compartimento principal del camión, para servir a un sub-conjunto de clientes en un sub-tour. Segundo, es puesto a prueba con un conjunto de instancias de hasta 50 clientes para instancias del problema con un solo vehículo y depósitos satélite (*Single Truck and Trailer Routing Problem with Satellite Depots*), cuya característica principal es que los clientes de vehículo son reemplazados por depósitos satélite con demanda cero. En este caso la ruta principal consiste en visitar una serie de depósitos satélite, a partir de los cuales se puede iniciar un sub-tour para atender a una serie de clientes de camión, con la restricción de tener que volver al mismo depósito satélite, desde el que iniciaron el sub-tour.

Finalmente, en (Pérez-Rodríguez y Urbán-Rivero, 2025) se presenta un algoritmo evolutivo que incorpora la idea de ausencia de recombinación y cooperación, sin comunicación entre individuos, donde cada individuo representa rutas parciales para resolver el problema. En (Pérez-Rodríguez y Urbán-Rivero, 2024) se propone una mejora del algoritmo mediante la introducción de objetivos auxiliares para abordar limitaciones en la diversidad y la explotación. Esta versión mejorada agrega una segunda población y utiliza NSGA-II (Deb, Pratap, Agarwal, y Meyarivan, 2002) para seleccionar individuos desde un frente de Pareto. Ambos algoritmos se comparan con otros métodos, utilizando el conjunto de instancias CHAO, obteniendo resultados competitivos.

### 4.1.2. Variantes del problema

Además, se ha desarrollado una serie de trabajos con el objetivo de acercar el problema a aplicaciones del mundo real. Siguiendo esta motivación, se introduce el TTRP Generalizado (GTTRP) (Drexler y Sebastian, 2007), que permite la transferencia de carga entre camiones y el intercambio de remolques entre vehículos, junto con otras extensiones. En (Drexler, 2011) se emplea un método de branch-and-price con una heurística de generación de columnas. Los experimentos se realizan utilizando datos reales de un problema de recolección de leche, así como el conjunto de instancias original. Sin embargo, solo las instancias de hasta 30 clientes se resuelven de forma óptima.

En (S.-W. Lin y cols., 2011) se introduce por primera vez el Truck and Trailer Routing Problem with Time Windows (TTRPTW), proponiendo una heurística basada en recocido simulado. Para evaluar su algoritmo, generan un conjunto de instancias, denominado conjunto de instancias LIN2011, considerando entre 50 y 200 clientes, derivadas de instancias VRPTW conocidas ((Solomon, 1987; Gehring y Homberger, 2008)) siguiendo un procedimiento similar al propuesto en (Chao, 2002). Adicionalmente, en (Derigs y cols., 2013) se realizan experimentos sobre esta variante utilizando el mismo conjunto de instancias.

En (Batsyn y Ponomarenko, 2014) se propone una heurística voraz de arranques múltiples (*multi-start*) junto con un modelo MILP para resolver el TTRPTW con ventanas de tiempo blandas y duras, y entregas fraccionadas (*split deliveries*). El algoritmo se prueba utilizando una instancia real con 292 clientes.

En (Torres, Cruz, y Verdegay, 2015) se propone un enfoque difuso (*fuzzy*) mediante un enfoque paramétrico general, relajando las restricciones del problema original, específicamente las restricciones de capacidad. En este contexto, para modelar un problema más cercano a una aplicación real, se considera la incertidumbre referente a las restricciones de capacidad de los vehículos, permitiendo que los vehículos puedan transportar una carga superior a su capacidad según un factor  $\alpha$  definido, admitiendo así el uso de soluciones infactibles. El modelo resultante se resuelve mediante un algoritmo de *hill-climbing* que utiliza los movimientos descritos con anterioridad en el trabajo de (S.-W. Lin y cols., 2009). Este fue aplicado al conjunto de instancias original, demostrando que el enfoque permite reducir los costos al optimizar la asignación de clientes a los vehículos.

En (Mirmohammadsadeghi y Ahmed, 2015) se propone un algoritmo memético para resolver una nueva variante del problema, el TTRP con demandas estocásticas y ventanas de tiempo (TTRPSDTW). El algoritmo se evalúa utilizando instancias generadas a partir del conjunto de instancias LIN2011.

En (Parragh y Cordeau, 2017) se propone un enfoque de branch-and-price para el TTRPTW y se adapta un algoritmo de búsqueda de vecindario amplio adaptativo (ALNS) para generar columnas iniciales de alta calidad. Se creó un conjunto de instancias basado en los conjuntos de referencia presentados en (Solomon, 1987), denominado conjunto de instancias SOLO, el cual, junto con el conjunto de instancias LIN2011, se utiliza para evaluar el desempeño del método propuesto.

En (Rothenbächer, Drexler, y Irnich, 2018) se aborda el TTRPTW y se consideran dos extensiones: una relacionada con el horizonte de planificación y otra que incorpora tiempos de transferencia de carga dependientes de la cantidad trasladada del camión al remolque. Se propone un algoritmo de branch-and-price-and-cut, el cual se prueba sobre instancias de (Derigs y cols., 2013) y (Parragh y Cordeau, 2017).

En (Mirmohammadsadeghi y Kabir, 2024) se introduce aleatoriedad en los tiempos de viaje, los tiempos de servicio y las ventanas de tiempo, dando lugar a una nueva variante denominada TTRP Estocástico, que se considera una extensión del TTRPSTT introducido en (Mirmohammadsadeghi, Masoumik, y Alavi, 2020). El problema se resuelve mediante un método de recocido simulado multipunto, y las instancias de prueba se adaptan a partir del conjunto de referencia VRPTW propuesto en (X. Li, Tian, y Leung, 2010).

Otro grupo de variantes del problema incluye el Single Truck and Trailer Routing Pro-

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

blem (STTRP), que considera escenarios con un único camión y un único remolque. En (Villegas, Prins, Prodhon, Medaglia, y Velasco, 2010) se introduce el STTRP con depósitos satélite (STTRPSD), en el que se designan ubicaciones específicas para estacionar los remolques. Se proponen dos metaheurísticas, una basada en GRASP y descenso de vecindarios variables (VND), junto con una búsqueda local evolutiva. Se generó un conjunto de 32 instancias, denominado conjunto de instancias VILL, con entre 25 y 200 clientes. Posteriormente, en (Belenguer y cols., 2015) se presenta una formulación de flujo de dos índices para el STTRPSD, reforzada con desigualdades válidas. Se implementa un algoritmo de branch-and-cut, evaluado sobre el mismo conjunto de instancias, resolviendo instancias de hasta 100 clientes.

En (Accorsi y Vigo, 2020) se introduce una variante denominada Extended Single Truck and Trailer Routing Problem (XSTTRP), que considera clientes con instalaciones de estacionamiento y otras ubicaciones específicas designadas para el estacionamiento de remolques. Este problema tiene como objetivo modelar problemas que anteriormente se trataban por separado. Para esto modela cuatro tipos de nodos:

- Clientes de camión: Se comportan de manera similar al problema original.
- Clientes de vehículo: Al igual que en el problema original, pueden ser atendidos por un camión con o sin remolque. Sin embargo, para estos se introducen dos subdivisiones.
  - Clientes que cuentan con instalaciones para dejar el remolque estacionado y por ende, ser el origen de un sub-tour.
  - Clientes que no cuentan con dichas instalaciones y sólo pueden ser atendidos.
- Depósitos satélite: Ubicaciones sin demanda destinadas a servir como puntos en los que iniciar un sub-tour.

En este trabajo se presenta una metaheurística híbrida que se compone de un proceso de asignación, en el que cada cliente es asignado a ubicaciones de estacionamiento o al depósito, seguido de un proceso de construcción para generar soluciones iniciales factibles. Luego, la solución se mejora por medio de una búsqueda local iterativa y por último, una fase de pulido en la que se almacenan las soluciones de mejor calidad y se resuelve un modelo de partición de conjuntos que selecciona la combinación óptima de las rutas para formar la solución final.

La propuesta se evaluó usando una amplia variedad de instancias, incluyendo los conjuntos CHAO, VILL y BART, destacando que obteniendo cinco nuevas mejores soluciones conocidas para este último, específicamente para instancias de 40 clientes de tipo C provenientes del conjunto de CHAO.

En (da Cruz y Salles da Cunha, 2023) los autores presentan el Profitable Single Truck and Trailer Routing Problem with Time Windows (PSTTRPTW), en el que se obtiene una ganancia al atender a los clientes dentro de sus ventanas de tiempo, y en el que acoplar el remolque implica una penalización en el tiempo de viaje y en el costo. Se propone un



algoritmo de branch-and-cut, evaluado sobre instancias generadas a partir del conjunto de instancias LIN2011.

También se han propuesto diversas variantes específicas de aplicación del problema. En (Davila-Pena y cols., 2023) se introduce el Multi-Compartment Truck and Trailer Routing Problem (MC-TTRP), en el cual deben entregarse múltiples tipos de productos sin que estos se mezclen. Se implementa un algoritmo heurístico de dos fases, compuesto por una heurística constructiva seguida de una metaheurística de búsqueda tabú iterada. Se utilizan instancias adaptadas del conjunto de instancias CHAO. En la industria láctea, estudios como (Hoff y Løkketangen, 2007) abordan restricciones tales como granjas pequeñas y de difícil acceso. En (Caramia y Guerriero, 2010) se considera además el transporte de múltiples tipos de leche, lo que requiere estanques multicompartimentados en los camiones. Se presentan y evalúan dos formulaciones matemáticas y un procedimiento de búsqueda local utilizando instancias reales de granjas italianas. Otra aplicación se presenta en (Cavagnini y cols., 2023), donde el Asymmetric Single Truck and Trailer Routing Problem with Satellite Depots (ASTTRPSD) se resuelve mediante un método de búsqueda local iterada para una empresa de logística (DHL). En (H. Li, Chen, Wang, y Bai, 2021) se presenta una revisión del TTRP y de problemas relacionados, específicamente el problema de ruteo de vehículos de dos escalones (*two-echelon vehicle routing problem*) y el problema de ruteo y localización de dos escalones (*two-echelon location routing problem*).

## 4.2. Resumen del Capítulo

En este capítulo se puso en evidencia cómo la literatura relacionada con el CTTRP ha evolucionado considerablemente desde su formulación original (Chao, 2002), dando lugar a una serie de variantes que buscan abordar distintos aspectos de aplicaciones del mundo real, tales como ventanas de tiempo, incertidumbre en la demanda o en los tiempos de viaje. Además de considerar instancias con un único camión, junto a un remolque y distintas características para los clientes, como la inclusión de depósitos satélite. Respecto a los métodos de resolución, se observa una amplia aplicación de heurísticas constructivas combinadas con algoritmos de búsqueda local, que hacen uso de movimientos simples para reparar las soluciones. Por otro lado, los métodos exactos revisados han sido evaluados resolviendo instancias de menor tamaño que los métodos incompletos. En los trabajos más recientes se han propuesto algoritmos evolutivos cooperativos.

En la Tabla 4.10 se presenta un resumen con las contribuciones más recientes para cada variante del problema.

A pesar de la diversidad de enfoques, se observa que la gran mayoría de los trabajos revisados utiliza un único procedimiento para construir la solución inicial sobre la que se aplican los mecanismos de mejora. Esto plantea la posibilidad de explorar si una estrategia colaborativa que combine múltiples procedimientos de construcción, cada uno guiado por un criterio distinto, podría favorecer la exploración del espacio de búsqueda y, con ello, la calidad de las soluciones obtenidas por los métodos reparadores. Un estudio preliminar en esta línea fue realizado en (Navarro y cols., 2025), donde se propusieron y evaluaron cuatro procedimientos de construcción basados en GRASP para el CTTRP, confirmando

## CAPÍTULO 4. ESTADO DEL ARTE DEL CAPACITATED TRUCK AND TRAILER ROUTING PROBLEM

Tabla 4.10: Contribuciones recientes por variante del problema en el TTRP. Los conjuntos de instancias marcados con <sup>A</sup> indican versiones adaptadas de los conjuntos de datos originales.

| Problema    | Variante                                | Última referencia                     | Conjunto de instancias                   | Clientes |
|-------------|-----------------------------------------|---------------------------------------|------------------------------------------|----------|
| TTRP        | Estándar                                | (Pérez-Rodríguez y Urbán-Rivero 2024) | CHAO                                     | 50-199   |
|             |                                         | (Bartolini y Schneider 2020)          | BART                                     | 25-40    |
|             | TTRP Generalizado                       | (Drexl 2011)                          | Datos reales<br>CHAO                     | 6-75     |
|             | TTRP Relajado                           | (S.-W. Lin y cols. 2010)              | CHAO                                     | 50-199   |
|             | TTRP con restricciones difusas          | (Torres y cols. 2015)                 | CHAO                                     | 50-199   |
| TTRPTW      | Estándar                                | (Parragh y Cordeau 2017)              | LIN2011<br>SOLO <sup>A</sup>             | 50-100   |
|             | TTRP Estocástico                        | (Mirmohammadsadeghi y Kabir 2024)     | TAI <sup>A</sup><br>LIN2011 <sup>A</sup> | 50-100   |
|             | TTRPTW con demanda estocástica          | (Mirmohammadsadeghi y Ahmed 2015)     | LIN2011 <sup>A</sup>                     | 50-200   |
|             | STTRP con depósitos satélite            | (Villegas y cols. 2010)               | VILL                                     | 25-200   |
| TTRP Simple | STTRP Rentable con ventanas de tiempo   | (da Cruz y Salles da Cunha 2023)      | LIN2011 <sup>A</sup>                     | 9-46     |
|             | STTRP Extendido                         | (Accorsi y Vigo 2020)                 | CHAO                                     | 25-200   |
|             |                                         |                                       | VILL<br>BART                             |          |
| MC-TTRP     | STTRP Asimétrico con depósitos satélite | (Cavagnini y cols. 2023)              | Datos reales                             | -        |
|             | Estándar                                | (Davila-Pena y cols. 2023)            | CHAO <sup>A</sup>                        | 50-199   |
|             | Problema de recolección de leche        | (Caramia y Guerriero 2010)            | Datos reales                             | -        |

que cada uno explora regiones distintas del espacio de soluciones sin que ninguno domine consistentemente sobre los demás. Esta observación, junto con la escasez de estudios que analicen específicamente el efecto de la diversidad de soluciones iniciales en el desempeño de metaheurísticas para el CTTRP, motiva el enfoque colaborativo de construcción y reparación propuesto en el siguiente capítulo, que retoma tres de estos procedimientos y los integra a una fase de reparación basada en *Hill-climbing*.

# Capítulo 5

## Propuesta de solución

En este capítulo se presenta CORAL (*Construct then Repair Algorithm*), una heurística de búsqueda local de dos fases, capaz de resolver instancias que modelan la entrega de bienes en el contexto de los problemas tipo *Capacitated Truck and Trailer Routing Problem*. El nombre del algoritmo hace referencia a sus dos fases principales, una fase de construcción de soluciones iniciales, seguida de una fase de reparación basada en búsqueda local.

CORAL trabaja tanto sobre el espacio factible como infactible de soluciones candidatas. Se considera una fase de construcción basada en un procedimiento GRASP (*Greedy Randomized Adaptive Search Procedure*) que incluye tres maneras distintas para generar una solución inicial. Posterior a esto, se inicia la fase de reparación basada en un algoritmo *Hill-climbing* el cual usa una serie de movimientos de búsqueda local para mejorar la calidad de la solución actual. Se incluye además un procedimiento de perturbación, para poder explorar otros sectores del espacio de búsqueda cercanos al óptimo local encontrado.

Cabe destacar que, si bien el esquema colaborativo de construcción más reparación mediante búsqueda local es compartido con otros enfoques de la literatura, la fase de reparación de CORAL difiere de la búsqueda de vecindario variable (VNS) utilizada en trabajos cercanos, como el de (Villegas y cols., 2011). Mientras que en VNS la estructura de vecindario activa cambia de forma sistemática cada vez que no se logra una mejora, antes de aplicar la búsqueda local, CORAL organiza sus movimientos en bloques de orden fijo, definido una única vez mediante el proceso de sintonización presentado en el Capítulo 6, en lugar de alternar entre vecindarios de forma reactiva durante la ejecución.

A continuación se detallan cada una de las componentes del algoritmo propuesto: la representación, función de evaluación, fase de construcción y fase de reparación:

### 5.1. Representación

Se define una representación de camión + remolque que se utiliza en ambas fases del algoritmo. Cada ruta se representa mediante tres vectores. El primer vector representa la ruta principal del camión. En caso de que la ruta tenga un remolque asignado, constituyendo una ruta de vehículo, se utiliza un segundo vector para almacenar los orígenes de cada sub-tour junto con un tercer vector (compuesto de vectores) que almacena los sub-tour

para cada origen. En la Figura 5.1 se presenta un ejemplo.

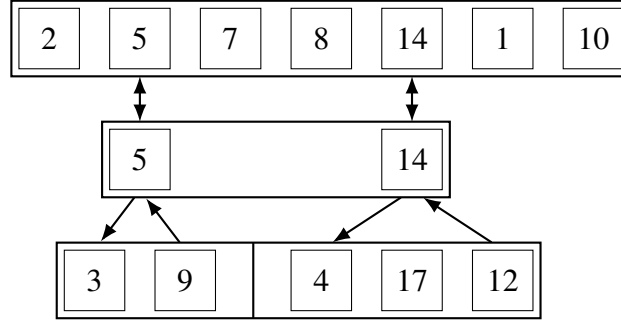


Figura 5.1: Representación de una ruta de vehículo con dos sub-tours. La ruta principal se muestra en la parte de arriba, seguida por los puntos iniciales de cada sub-tour y sus correspondientes sub-tours.

El diseño de esta representación busca proporcionar claridad estructural al separar explícitamente la ruta principal y cada sub-tour individual, al mismo tiempo que garantiza que cada cliente aparezca sólo una vez en la solución a través del manejo de los orígenes de los sub-tours en un vector separado, lo que permite el manejo directo de esas restricciones. Además, no es necesario incluir el depósito de la representación, lo que permite reducir el tamaño de la representación.

Asimismo, esta estructura permite el manejo eficiente de la solución, mediante operaciones vectoriales, facilitando el movimiento de clientes dentro de la misma ruta o entre rutas distintas. Por último, esta representación considera variables para mantener su costo y capacidad utilizada, lo que permite evaluaciones eficientes de movimientos, al calcular únicamente las diferencias producidas por los arcos afectados.

## 5.2. Función de evaluación

El objetivo del problema es minimizar el costo total de viaje. Sin embargo, el algoritmo permite la sobrecarga de rutas o sub-tours durante la búsqueda, ocasionando que el sub-tour transporte una carga mayor a la capacidad del camión o que la ruta atienda a clientes con una demanda total mayor a la capacidad del camión y el remolque (en caso de contar con uno). Por esto, para manejar la restricción del límite de carga, se aplica un factor de penalización sobre las soluciones infactibles.

$$f(s) = \sum_{i \in \text{routes}} (\text{travelCost}_i + \text{pen} \cdot \text{overload}_i) \quad \text{pen} \in \mathbb{R}^+ \quad (5.1)$$

## 5.3. Fase de construcción

La fase de construcción considera tres procedimientos basados en la metaheurística GRASP, con el objetivo de tener un conjunto de soluciones iniciales diverso. Estos fueron introdu-

cidos originalmente en un trabajo previo (Navarro y cols., 2025), en el que se propusieron y evaluaron cuatro procedimientos de construcción, denominados en dicho trabajo B, C, R y G. Para esta tesis se adoptó un subconjunto de tres de ellos, presentados a continuación y renombrados como A, B y C (correspondientes a los procedimientos B, C y G del trabajo original), tras descartar el procedimiento R al verificar que no aportaba diversidad adicional relevante. El análisis experimental que sustenta esta decisión se retoma en la Sección 6.4 del Capítulo 6.

Todos los procedimientos, construyen una solución seleccionando el siguiente cliente a visitar y decidiendo cuándo desacoplar el remolque basado en el tipo del siguiente cliente. Los procedimientos se diferencian en la función miope (también conocida como función de transición) que usan, que se detallan a continuación:

### 5.3.1. Procedimiento A (Enfoque Round Robin)

El Procedimiento A busca equilibrar la relación entre intensificación y diversificación. Comienza creando tantas rutas como camiones disponibles existan. Una vez hecho esto, todos los remolques disponibles se asignan a las primeras rutas del vector de rutas. Esto define la estructura inicial de la flota a la que se asignarán los clientes.

La asignación de clientes sigue un enfoque round-robin en el que las rutas se turnan en orden cíclico para agregar el siguiente cliente. Este proceso continúa hasta que todos los clientes han sido asignados.

En este procedimiento se incorpora un componente probabilístico inspirado en GRASP para introducir aleatoriedad controlada. En cada turno de una ruta, el algoritmo construye una Lista Restringida de Candidatos (LRC) con los  $k$  mejores candidatos según una función miope, y luego selecciona uno al azar de esta lista.

La función miope se define como:

$$\text{Función miope} = d \cdot \frac{d_b}{c} \quad (5.2)$$

Donde:

- $d$ : Distancia al siguiente cliente.
- $d_b$ : Distancia de retorno (ya sea al depósito o al cliente con el remolque, dependiendo de si el camión está acoplado al remolque).
- $c$ : Capacidad disponible en el camión, incluyendo la capacidad del remolque cuando está acoplado.

Al intentar asignar un cliente de camión a una ruta de vehículo, se crea un nuevo sub-tour utilizando el último cliente añadido como origen del sub-tour. Los clientes posteriores asignados a la misma ruta se agregarán a este sub-tour, siempre que no se supere su capacidad. Si se supera la capacidad, el sub-tour finaliza y el cliente seleccionado se agrega a la ruta principal. En caso de que sea nuevamente un cliente de camión, se crea un nuevo sub-tour cuyo origen es el mismo que el del sub-tour anterior.

Si en algún momento un cliente no puede agregarse a ninguna ruta, sin generar sobrecarga, la solución se declara infactible y el proceso de construcción reinicia desde cero. Un ejemplo de solución inicial obtenida mediante este procedimiento se muestra en la Figura 5.2

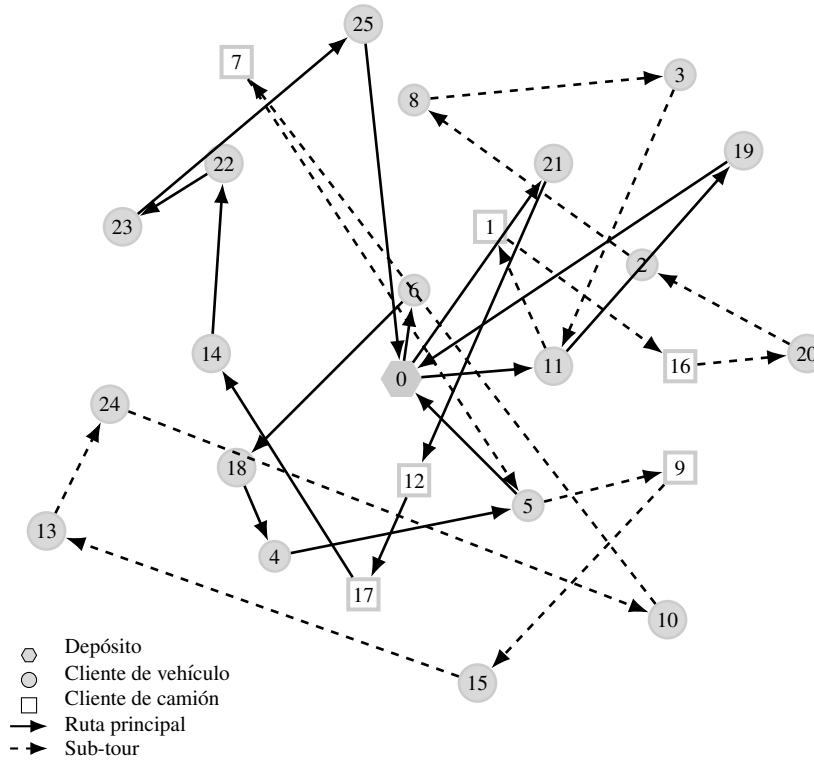


Figura 5.2: Una solución inicial de tipo A para 25 clientes. Esta consiste de tres rutas principales, una ruta de camión pura y dos rutas de vehículo completas. Cada una con un sub-tour.

### 5.3.2. Procedimiento B (Construcción secuencial de rutas)

El Procedimiento B busca construir las rutas manteniendo una distribución equitativa de la demanda entre ellas. A diferencia del Procedimiento A, construye las rutas de forma secuencial, buscando soluciones con menos rutas que el número de camiones disponibles, lo que puede llevar a soluciones globales más eficientes. El algoritmo se enfoca en una sola ruta a la vez y no pasa a la siguiente, hasta que la ruta actual esté llena o la *stop probability* (definida a continuación) indique que debe finalizar.

La lógica para insertar clientes y manejar sub-tours sigue las mismas reglas descritas para el Procedimiento A. El proceso de selección utiliza la misma función miope del Procedimiento A presentada en la ecuación 5.2, usando una LRC de los  $k$  mejores candidatos.

Este procedimiento se diferencia del primero al introducir una probabilidad basada en la capacidad disponible, que determina si la ruta actual debe dejar de agregar clientes e iniciar

una nueva ruta. Después de seleccionar un cliente con la función miope, la probabilidad se calcula en función de la demanda total de la ruta en comparación con la demanda promedio esperada por camión:

$$\text{stop\_probability} = \begin{cases} \left( \frac{rt\_d - avg\_d}{avg\_d} \right) & \text{si } rt\_d > avg\_d \\ 0 & \text{en otro caso} \end{cases} \quad (5.3)$$

Donde:

- $rt\_d$ : Demanda total de la ruta actual.
- $avg\_d$ : Demanda promedio esperada por camión, calculada dividiendo la demanda total de los clientes entre el número de camiones.

Esta probabilidad se calcula y aplica al seleccionar el siguiente cliente, y determina si agregarlo a la ruta actual o insertar un “cliente nulo”, lo que obliga a la ruta a regresar al depósito y proceder a construir la siguiente ruta. Este mecanismo garantiza un mejor equilibrio de carga entre las rutas y ayuda a evitar una concentración excesiva de demanda en una sola ruta. La Figura 5.3 ilustra una solución inicial generada por este procedimiento.

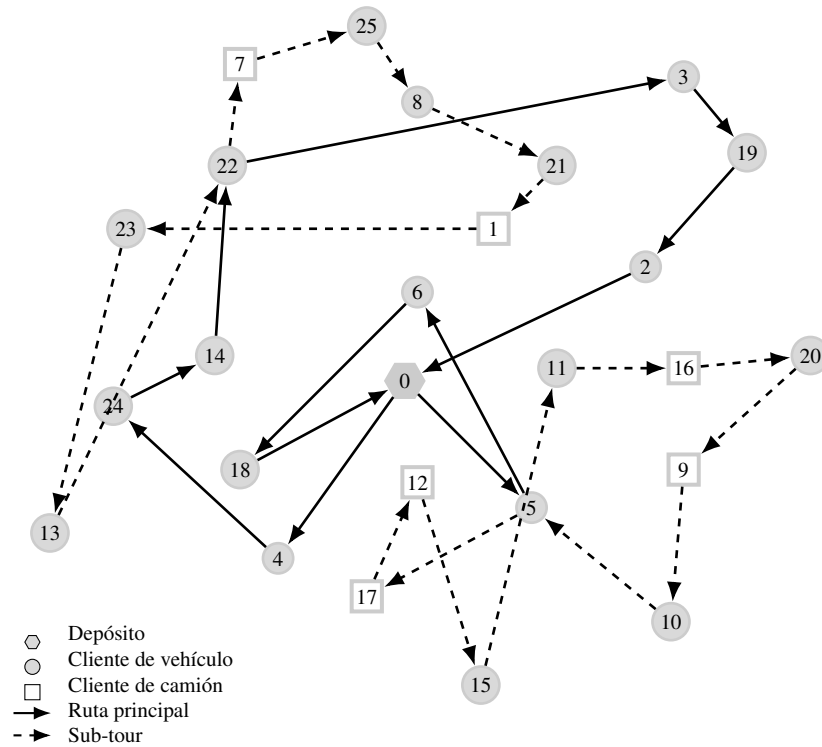


Figura 5.3: Una solución inicial de tipo B para una instancia con 25 clientes, consiste en dos rutas principales, cada una con un sub-tour.

### 5.3.3. Procedimiento C (Enfoque de clustering)

Este enfoque busca construir rutas agregando clientes en función de qué tan cercanos están de un cliente específico. Primero, el algoritmo crea tantas rutas como camiones disponibles existan. Luego, los remolques disponibles se asignan aleatoriamente a estas rutas.

A continuación, se asigna un cliente inicial a cada ruta, seleccionado de modo que esté lo más lejos posible de los ya elegidos y del depósito. Una vez que todas las rutas tienen su cliente inicial, los clientes restantes se dividen en dos listas según su tipo, clientes de vehículo o clientes de camión.

El procedimiento itera sobre las rutas de forma cíclica, intentando asignar un cliente de la lista de clientes de vehículo. En este caso, la función miope (Ecuación 5.4) depende del último cliente agregado a la ruta (o del cliente inicial si aún no se ha agregado ninguno) y del cliente inicial de la ruta. Se diseñó para promover el agrupamiento local alrededor del cliente inicial en cada ruta.

$$\text{Función miope} = d_{last} + d_{ret} \quad (5.4)$$

Donde:

- $d_{last}$  : Distancia al último cliente agregado a la ruta/sub-tour.
- $d_{ret}$  : Distancia de retorno al cliente inicial/origen del sub-tour.

Una vez calculada la función miope para todos los clientes candidatos de la ruta actual, la LRC se forma con los  $w$  clientes de vehículo con los valores más bajos de la función miope. El procedimiento selecciona entonces un cliente al azar de la LRC e intenta agregarlo a la ruta, sin superar la capacidad del vehículo (incluyendo un pequeño umbral). Si esto no es posible, intenta asignar el cliente a la siguiente ruta con capacidad disponible. Si no existe tal ruta, el cliente se asigna a la ruta con mayor capacidad disponible. Esto puede llevar a una solución inicial infactible por sobrecarga de la ruta, pero dichos casos se abordan en la fase de reparación posterior, ya que la prioridad en esta etapa es generar una solución inicial rápidamente. Este proceso continúa hasta que todos los clientes de vehículo han sido asignados.

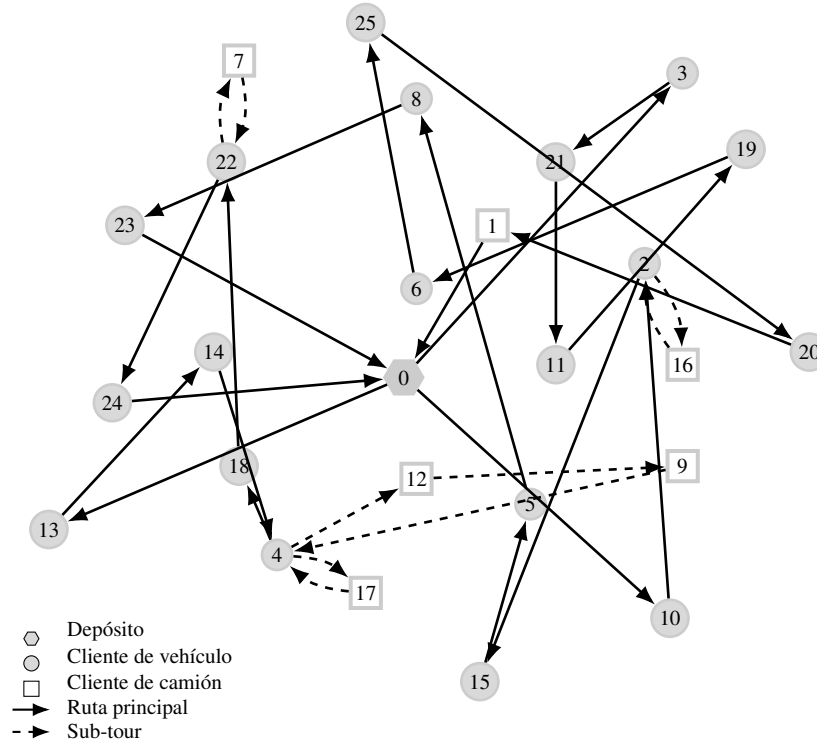
Luego, se aplica el mismo proceso a la lista de clientes de camión, utilizando la función miope con respecto al último cliente de vehículo agregado a la ruta. Sin embargo, en lugar de insertarlos directamente en la ruta, los clientes de camión se almacenan en una lista temporal de cada ruta, para que su integración en ella se decida posteriormente.

Una vez asignados todos los clientes de camión, se integran según las siguientes reglas:

- Si la ruta no tiene remolque o si la ruta no tiene clientes en el tour principal, el cliente se agrega al final de la ruta.
- Si la ruta tiene remolque, el algoritmo evalúa dos opciones de inserción basadas en la minimización de la distancia:
  - Creación de un nuevo sub-tour: el costo se calcula como el doble de la distancia entre el cliente y el cliente más cercano en el tour principal, que se convierte en el punto inicial del sub-tour.



- Integración en un sub-tour existente: el costo se calcula como la suma de la distancia desde el cliente hasta el último cliente del sub-tour y la distancia necesaria para regresar a su punto inicial. Solo se consideran los sub-tours con capacidad restante suficiente.



menor tamaño. Por otro lado, genera rutas peores al considerar la distancia tanto al cliente inicial como al último agregado, presentando un mayor número de arcos cruzados, tal como se muestra en la Figura 5.4.

### 5.3.4. Procedimiento General

En el Algoritmo 3 se presenta el esquema general usado para construir las soluciones iniciales. Este asigna clientes de manera iterativa usando la LRC basada en la función miope del procedimiento seleccionado. La selección round-robin solo es aplicada en el procedimiento A.

---

**Algoritmo 3** Fase de construcción

---

```
1: procedure CONSTRUCCION(clientes, camiones, tipoAlg)
2:   rutas  $\leftarrow$  InicializarRutasVacias(camiones)
3:   rutaAct  $\leftarrow$  PrimeraRuta(rutas)
4:   while  $\exists$  clientesSinAsignar do
5:     clientes  $\leftarrow$  ObtenerElegibles(clientes, rutaAct)
6:      $\triangleright$  Calcular valores de función miope
7:     LRC  $\leftarrow$  ObtenerTop(FuncionMiope(clientes))
8:     clienteSel  $\leftarrow$  SeleccionAleatoria(LRC)
9:     AsignarARuta(clienteSel, rutaAct)
10:    rutaAct  $\leftarrow$  ActualizarEstadoDeRuta(rutaAct, tipoAlg)
11:   end while
12:   return rutas
13: end procedure
```

---

## 5.4. Fase de reparación

La fase de reparación contempla tres procedimientos basados en *Hill-climbing*, en el que se ocupa la estrategia de Alguna Mejora (en inglés First Improvement) para aceptar una nueva solución. Estos procedimientos fueron diseñados para reparar la ruta principal, los sub-tours, o trabajar a nivel de cliente individual. Además, este conjunto de procedimientos usa un mecanismo de control de la penalización en la función de evaluación por exceso de capacidad, para promover la exploración y permitir que el algoritmo se mueva a través de soluciones infactibles y promoviendo al final la búsqueda de una solución factible.

A continuación se presentan los movimientos utilizados y la estructura de la fase de reparación.

### 5.4.1. Movimientos

Se incluyeron un total de siete movimientos, para reparar las soluciones. Dos movimientos trabajan a nivel de ruta para reparar una ruta de manera individual, mientras que los cinco

movimientos restantes tienen como objetivo mejorar la solución, reubicando clientes de manera individual en otra ubicación. Cabe destacar que cada movimiento es evaluado sólo calculando la diferencia en el costo de viaje resultante debido a la modificación de los arcos producida por el movimiento, considerando además la sobrecarga que se puede incurrir en el camión o remolque, antes de mover realmente al cliente a su nueva ubicación, lo que permite reducir el costo de evaluar cada movimiento, mejorando la eficiencia del algoritmo. Por último, si un movimiento mejora la solución, se aplica a esta y el algoritmo continúa a la siguiente iteración. A continuación se presentan los siete movimientos.

### 1. Movimientos de reparación de ruta

- a) El primer movimiento considerado corresponde al algoritmo 2-opt (S. Lin, 1965). Este operador explora el vecindario de una ruta considerando el intercambio de dos arcos. Primero selecciona un par de arcos y los elimina para reconectar los elementos resultantes, invirtiendo la secuencia de clientes entre los puntos de corte. En la Figura 5.5 se presenta un esquema de su funcionamiento. Este tiene como objetivo reparar las rutas principales eliminando arcos cruzados para crear caminos más cortos.

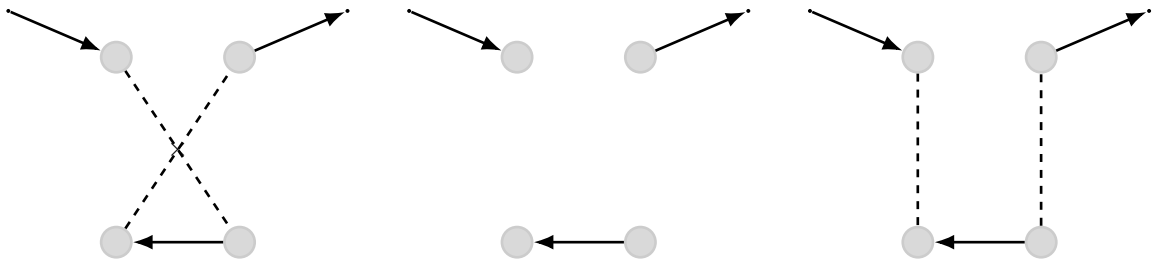


Figura 5.5: Ejemplo del movimiento 2-opt. Primero se seleccionan dos arcos, que en este caso, deliberadamente se encuentran cruzados. Luego estos se eliminan, separando la ruta en tres segmentos. Por último, los cuatro clientes se reconectan intercambiando los clientes a los que están conectados, invirtiendo el orden del segmento intermedio, resultando en un menor costo.

- b) El segundo movimiento corresponde al paso de refinamiento de raíces de sub-tours (*sub-tour root-refining step*) propuesto en (Chao, 2002). Este tiene como objetivo mejorar la manera en la que cada sub-tour está conectado a la ruta principal. Este movimiento modifica el cliente que actúa como origen del sub-tour y los clientes de inicio y final del sub-tour que son usados para conectarlo a la ruta principal.

Primero, los dos arcos que conectan el sub-tour a su origen actual se eliminan y el arco que une a los clientes que corresponden a los bordes del sub-tour se re-construye. Luego, se evalúan reconexiones alternativas, seleccionando pares consecutivos de clientes del sub-tour, removiendo el arco entre ellos y reconectando ambos clientes a un cliente de la ruta principal, convirtiéndolo en el nuevo origen del sub-tour. Si una reconexión resulta en una distancia menor,

se acepta el movimiento. Además, en caso de ser necesario, el sub-tour es re-secuenciado rotando su orden interno, de tal manera que los nuevos clientes de borde preserven su orden relativo en el sub-tour original. Un ejemplo es presentado en la Figura 5.6.

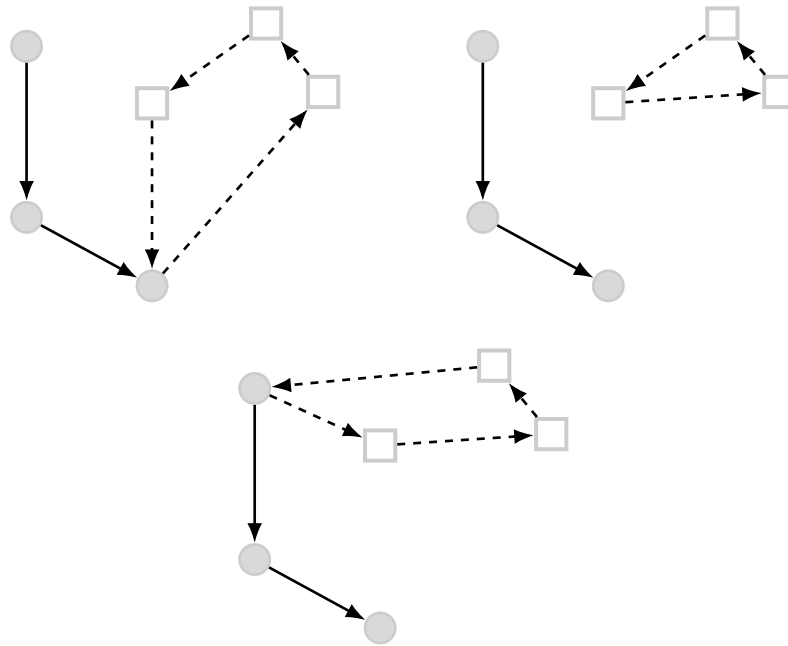


Figura 5.6: Ejemplo del movimiento de refinamiento de raíces de sub-tours. Los arcos que conectan el sub-tour a su origen en la ruta principal son removidos y el sub-tour es reconectado a un cliente distinto por medio de unir este cliente a dos clientes consecutivos que se encuentran en el sub-tour.

Cada uno de estos movimientos se maneja como bloque independiente dentro del proceso de reparación, lo que permite versatilidad en el orden en que se aplican.

## 2. Movimientos de cliente

El conjunto de cinco movimientos que trabajan a nivel de cliente es considerado como otro bloque independiente. Este bloque de movimientos tiene como objetivo mejorar el costo de viaje, a través de reubicar clientes individuales y asegurar su factibilidad:

- a) *movToTour*: Este movimiento selecciona un cliente y lo reubica en la ruta principal de cualquier ruta. En caso de que sea un cliente de camión, se restringe el movimiento solamente a rutas de camión pura, evitando las rutas de vehículo y asegurando la factibilidad de la solución.
- b) *movToSubtour*: Este movimiento selecciona un cliente y lo reubica en un sub-tour.

- c) *newSubtour*: Este movimiento selecciona un cliente y evalúa si generar un nuevo sub-tour sólo con este cliente incurre en un menor costo de viaje. Este busca un cliente en cualquiera de las rutas disponibles, de tal manera que mejore la solución.
- d) *swap*: Este selecciona un cliente y lo intercambia con otro de una ruta distinta. Este considera si el cambio es posible dependiendo de los tipos de los clientes y la ubicación a la que quieren ser movidos (ruta principal de una ruta de vehículo o un sub-tour/ruta de camión pura).
- e) *swapBest*: Este movimiento es similar al anterior, con la excepción de que, en vez de intercambiar los clientes en sus posiciones originales, trata de ubicarlos en la ruta buscando la posición que reduzca el costo de viaje. A diferencia de los movimientos anteriores, al buscar la mejor posición en la que insertar a los clientes en cada una de las rutas, este es más costoso computacionalmente y el que incurre en una mayor mejora de la solución al ser aplicado, ya que explora la mejor alternativa en la que se pueden ubicar los clientes.

Estos movimientos son aplicados recorriendo los clientes de cada ruta y explorando el espacio de búsqueda de cada movimiento con un cliente específico, antes de pasar al siguiente cliente y repetir el procedimiento. Este se muestra en el Algoritmo 4.

---

**Algoritmo 4** Bloque de movimientos de cliente

---

```
1: procedure MOVIMIENTOSCLIENTE(sol, pen)
2:   movimientos  $\leftarrow$  [MOVTOUR, MOVTOSUBTOUR, NEWSUBTOUR,
3:     SWAP, SWAPBEST]
4:   for each ruta in sol.rutas do
5:     for each cliente in ruta.rutaPrincipal do
6:       for each mov in movimientos do
7:         if mov(sol, ruta, cliente) then
8:           return true
9:         end if
10:      end for
11:    end for
12:    for each subtour in ruta.subtours do
13:      for each cliente in subtour do
14:        for each mov in movimientos do
15:          if mov(sol, ruta, cliente) then
16:            return true
17:          end if
18:        end for
19:      end for
20:    end for
21:    return false
22: end procedure
```

---

### 5.4.2. Estructura general

El proceso general de la fase de reparación se presenta en el Algoritmo 5. Este considera una solución inicial, una penalización inicial, una cantidad de reinicios y un límite de iteraciones.

El objetivo es reparar la solución candidata actual, aplicando el conjunto de procedimientos basados en *Hill-climbing* presentados. En el caso de que el óptimo local obtenido no sea factible, la penalización se incrementa significativamente, con el objetivo de forzar la búsqueda con movimientos que restauren la factibilidad a la solución. Para esta implementación, el valor de la penalización incrementada se fijó en 100, que es un valor elevado en relación con el costo total de viaje de las instancias trabajadas (que varía aproximadamente entre 300 y 1000). Esto garantiza que incluso, solo una unidad de exceso de capacidad resulte prohibitivamente costosa. Posteriormente, los procedimientos se ejecutan nuevamente utilizando la penalización actualizada en la función de evaluación.

Esta estrategia de penalización adaptativa permite tanto la exploración del espacio de búsqueda como la convergencia hacia soluciones factibles, logrando restaurar la factibilidad en la totalidad de las ejecuciones realizadas en los experimentos presentados en el Capítulo 6. Finalmente, la solución generada se perturba mediante el intercambio de la posición de clientes con una probabilidad del  $P_{pert}$ , tras lo cual se aplica nuevamente la fase de reparación.

Este procedimiento se repite según el número de reinicios, contribuyendo a buscar soluciones cercanas al último óptimo local obtenido, asegurando un balance entre exploración y explotación.

---

**Algoritmo 5** Fase de reparación

---

```
1: procedure REPARAR(solInicial, initPen, nReset, totalIt)
2:   sol  $\leftarrow$  solInicial
3:   mejorSol  $\leftarrow$  solInicial
4:   reset  $\leftarrow$  0
5:   while reset < nReset do
6:     pen  $\leftarrow$  initPen
7:     sol  $\leftarrow$  HC(sol, pen, totalIt)
8:     if sol no es factible then
9:       pen  $\leftarrow$  100
10:      sol  $\leftarrow$  HC(sol, pen, totalIt)
11:    end if
12:    if esMejor(sol, mejorSol) then
13:      mejorSol  $\leftarrow$  sol
14:    end if
15:    sol  $\leftarrow$  perturbar(sol)
16:    reset = reset + 1
17:  end while
18:  return mejorSol
19: end procedure
```

---

El Algoritmo 6 muestra la estructura del conjunto de procedimientos basados en *Hill-climbing*. En cada iteración, se intenta mejorar la solución aplicando el conjunto de movi-

mientos en un orden específico.

Cada bloque corresponde a un tipo de movimiento específico, ya sea uno de los dos movimientos de reparación de ruta o el conjunto de movimientos de cliente. Estos se presentan como *BloqueN* en el algoritmo, ya que su orden será definido posteriormente en el proceso de sintonización.

---

**Algoritmo 6** Algoritmo de reparación

---

```
1: procedure HC(sol, pen, totalIt)
2:   mejora  $\leftarrow$  true
3:   iter  $\leftarrow$  0
4:   while iter < totalIt and mejora do
5:     mejora  $\leftarrow$  false
6:     mejora  $\leftarrow$  Bloque1(sol, pen)
7:     if !mejora then
8:       mejora  $\leftarrow$  Bloque2(sol, pen)
9:     end if
10:    if !mejora then
11:      mejora  $\leftarrow$  Bloque3(sol, pen)
12:    end if
13:  end while
14:  return sol
15: end procedure
```

---

### 5.4.3. Parámetros

Este enfoque requiere definir los valores para los siguientes parámetros:

- Factor de penalización inicial (*initPen*): Determina la penalización por unidad por exceder la capacidad durante la primera ejecución de la fase de reparación.
- Iteraciones totales (*totalIt*): Corresponde al máximo número de iteraciones que serán ejecutadas en cada proceso de reparación.
- Número de reinicios (*nReset*): Determina el número de veces que una solución es perturbada.
- Orden de bloques: Define el orden en que los bloques independientes son ejecutados en el algoritmo.
- Orden de movimientos de cliente: Determina el orden de ejecución de los cinco movimientos a nivel de cliente.
- Versión de *Hill-climbing*: Determina cómo el algoritmo se comporta, este varía de las siguientes maneras:

1. Aplica el movimiento a cada cliente antes de probar el siguiente movimiento.

2. Aplica los 5 movimientos a cada cliente antes de probar con el siguiente cliente.
3. Aplica el movimiento a cada cliente, si mejora, aplica el siguiente movimiento usando la nueva solución.
4. Aplica los 5 movimientos a cada cliente, si mejora, intenta mejorar la nueva solución iniciando desde siguiente movimiento.
5. Remueve el movimiento *swapBest*, comportándose de manera similar a la segunda versión presentada.

Además, deben considerarse los parámetros  $k$  y  $w$  que definen el largo de la LRC para los procedimientos constructivos, junto con  $P_{pert}$ , mencionados anteriormente.

### 5.5. Conclusiones del Capítulo

El diseño metodológico de CORAL permite concluir que la combinación de una construcción diversa y una reparación intensiva constituye una estrategia coherente para abordar el CTTRP. La representación adoptada y la evaluación incremental de movimientos entregan una base racional para explorar vecindarios amplios.

Asimismo, la organización de la fase de reparación en bloques independientes, junto con una penalización adaptativa, busca favorecer un equilibrio entre exploración e intensificación, lo que es un aspecto clave para enfrentar la alta complejidad combinatoria del problema. En consecuencia, la propuesta queda suficientemente fundamentada para ser contrastada empíricamente.

Cabe hacer notar que cada estrategia de diseño tiene un objetivo asociado relacionado con el problema y que cada función miope, así como cada movimiento buscan suplir una funcionalidad diferente en el espacio de búsqueda de soluciones. En el siguiente capítulo se valida esta conclusión mediante el análisis experimental del algoritmo, considerando calidad de soluciones, robustez y comparación con el estado del arte.



# Capítulo 6

## Resultados

En este capítulo se presenta la evaluación de CORAL, usando varias instancias de la literatura.

CORAL se compara con los resultados presentados en el estado del arte por (Bartolini y Schneider, 2020), en el que se implementa un algoritmo exacto de branch-and-cut con un límite de 2 horas de ejecución y se introduce el conjunto de instancias utilizado. Además, se consideran los resultados adicionales reportados por (Accorsi y Vigo, 2020) para cinco instancias de tipo C con 40 clientes, obtenidos por medio de una metaheurística híbrida. En ambos casos los valores de la literatura se toman directamente de las tablas publicadas en los artículos originales, sin reimplementar dichos métodos.

Para los experimentos computacionales se usó un subconjunto de 132 instancias provenientes del conjunto de instancias BART.

Las primeras 84 instancias (referidas en las tablas como Chao25, Chao30 y Chao40) fueron creadas a partir del benchmark original propuesto en (Chao, 2002). Estas instancias fueron generadas designando diferentes proporciones de clientes de vehículo y de camión. Las instancias tienen 25, 30 o 40 clientes. Las últimas dos se dividen en tres clases:

- Tipo A: Se mantiene una proporción similar entre la demanda total de los clientes y la capacidad total de la flota.
- Tipo B: Se derivan de las de Tipo A, tanto camiones como remolques cuentan con una mayor capacidad.
- Tipo C: Cuentan únicamente con un camión y un remolque disponible.

Las características de estas instancias se encuentran en el Capítulo 4, en las Tablas 4.8, 4.9, 4.2, 4.3, 4.4, 4.5, 4.6 y 4.7.

### 6.1. Entorno experimental

Los experimentos realizados con la propuesta presentada fueron ejecutados en un servidor con Ubuntu 16.10 con 125 GB de RAM DDR4, utilizando dos procesadores Intel Xeon E5-2680 v3, con 12 núcleos y 24 hilos, cada uno operando a una frecuencia de 2.5 GHz.

## 6.2. Proceso de sintonización

Se utilizó Irace (López-Ibáñez, Dubois-Lacoste, Pérez Cáceres, Stützle, y Birattari, 2016) para sintonizar los parámetros mencionados en la Sección 5.4.3. Este es un método de *racing* iterado para la configuración automática de algoritmos. Este, en cada iteración, genera un conjunto de configuraciones candidatas y las evalúa ejecutando el algoritmo sobre un conjunto de instancias definidas, descartando estadísticamente aquellas configuraciones cuyo desempeño es significativamente peor que el resto. Este proceso se repite hasta agotar el presupuesto de ejecuciones definido, quedando al final un pequeño grupo de *configuraciones elite*, correspondientes a las que sobrevivieron a todas las rondas de descarte y que, por lo tanto, presentan el mejor desempeño estadísticamente respaldado.

Para sintonizar se utilizaron las 132 instancias evaluando la calidad de cada ejecución con la función de evaluación presentada en el Capítulo 5. El criterio de parada de cada ejecución individual queda determinado por los parámetros sintonizados. Por otro lado, Irace se ejecutó con un presupuesto de 150000 experimentos.

Para los parámetros descritos en la Sección 5.4.3 se consideraron los siguientes valores:

- Factor de penalización inicial: Valores que varían entre 0.2 y 1.4, con incrementos de 0.2.
- Iteraciones totales: 250, 500, 750 y 1000 iteraciones.
- Número de reinicios (`nReset`): Se probaron valores de 0 a 10 reinicios.
- Orden de bloques: Debido a que son tres bloques, se pueden generar 6 permutaciones diferentes para el orden de estos.
- Orden de movimientos de cliente: Debido a que son cinco movimientos, estos se pueden organizar de 120 formas distintas.
- Versión de *Hill-climbing*: Se selecciona una opción de las descritas previamente.

Con esto la configuración seleccionada es la siguiente:

- Factor de penalización inicial: 0.8
- Iteraciones totales: 500
- Número de reinicios (`nReset`): 4
- Orden de bloques: paso de refinamiento de raíces de sub-tours, movimientos de cliente, 2-opt.
- Orden de movimientos de cliente: `movToTour`, `swap`, `newSubtour`, `movToSubtour`, `swapBest`
- Versión de *Hill-climbing*: Aplica los 5 movimientos a cada cliente antes de probar con el siguiente cliente.

En cuanto al orden de bloques seleccionado, se observa que el paso de refinamiento de raíces de sub-tours se ubica primero. Esto es consistente con que corresponde a la corrección estructuralmente más compleja de la solución, ya que redefine la conexión de cada sub-tour a la ruta principal. Luego, se ubica el bloque de movimientos de clientes, que trabaja modificaciones más pequeñas una vez se reconectan los sub-tours, destacando que el movimiento *swapBest*, correspondiente al movimiento con mayor potencial de mejora dentro del bloque se ubica al final, seguido del bloque de 2-opt. Esto sugiere que la configuración sintonizada favorece corregir primero los aspectos estructurales más difíciles de la solución, dejando los movimientos de mayor impacto en la calidad para etapas finales del ciclo de reparación.

Como se mencionó en el Capítulo 5, este orden de bloques y de movimientos, fijado por el proceso de sintonización, se mantiene constante a lo largo de toda la ejecución de una instancia, a diferencia de VNS, donde la estructura de vecindario activa cambia de forma reactiva ante la falta de mejora. En particular, la versión de *Hill-climbing* seleccionada explora los cinco movimientos de cliente de forma secuencial y en el mismo orden para cada cliente, reforzando el carácter predeterminado de la exploración de vecindarios en CORAL.

### 6.3. Estudio de componentes

Como se mencionó anteriormente, a diferencia del resto de movimientos a nivel de cliente, *swapBest* presenta un mayor costo computacional y a la vez, es el que potencialmente, incurre en una mayor mejora de la solución al ser aplicado, en comparación a movimientos como *swap* o la generación de un nuevo sub-tour. Es por esto que se incluyó como alternativa la quinta versión de *Hill-climbing* a probar, en la que se remueve este movimiento, con el objetivo de evaluar si la contribución de aplicar un movimiento más complejo permite llegar a mejores soluciones o si, por el contrario, se reduce el número de movimientos que se aplican dentro del número de iteraciones definido y estanca el proceso de búsqueda.

Irace seleccionó la versión de *Hill-climbing* que conserva el movimiento *swapBest* por sobre la que aplica los movimientos a los clientes de la misma manera, pero sin incluirlo, esto para todas las configuraciones élite entregadas. Esto sugiere que la calidad adicional que aporta este movimiento, al buscar la mejor posición de inserción en una ruta, compensa la potencial reducción en el número de movimientos ejecutados dentro del límite de iteraciones definido, validando su inclusión como parte relevante del proceso de búsqueda realizado por el bloque de movimientos de clientes.

### 6.4. Análisis de soluciones iniciales

Este análisis fue presentado originalmente en (Navarro y cols., 2025), donde se evaluó la calidad y diversidad de las soluciones generadas por los cuatro procedimientos de construcción originales. Esta sección retoma dicho análisis para los tres procedimientos adoptados en esta tesis (A, B y C).

## CAPÍTULO 6. RESULTADOS

Tabla 6.1: Calidad promedio y desviación estándar para instancias de 25 clientes

|    | A       |           | B       |           | C       |           |
|----|---------|-----------|---------|-----------|---------|-----------|
|    | Calidad | Dev. Est. | Calidad | Dev. Est. | Calidad | Dev. Est. |
| 1  | 506.2   | 16.6      | 512.5   | 16.4      | 670.2   | 44.2      |
| 2  | 532.2   | 20.7      | 545.1   | 17.1      | 726.1   | 51.0      |
| 3  | 551.2   | 20.0      | 573.1   | 20.2      | 746.8   | 57.8      |
| 4  | 522.4   | 18.6      | 528.9   | 18.3      | 701.2   | 41.8      |
| 5  | 565.6   | 19.3      | 564.6   | 20.6      | 774.6   | 50.7      |
| 6  | 630.5   | 21.0      | 611.8   | 21.9      | 795.1   | 58.4      |
| 7  | 472.1   | 18.1      | 495.9   | 18.5      | 682.1   | 50.5      |
| 8  | 539.0   | 19.0      | 531.4   | 18.4      | 713.8   | 47.6      |
| 9  | 589.7   | 20.9      | 591.5   | 19.5      | 781.1   | 54.2      |
| 10 | 543.7   | 20.4      | 564.6   | 20.2      | 790.2   | 52.1      |
| 11 | 556.6   | 20.0      | 597.1   | 22.7      | 795.0   | 55.3      |
| 12 | 601.8   | 21.6      | 620.1   | 22.5      | 799.4   | 58.2      |

Para analizar la diversidad de las soluciones iniciales generadas se realizaron 250 experimentos para cada procedimiento sobre el conjunto de instancias. En las Tablas 6.1, 6.2 y 6.3 se presenta el promedio y desviación estándar de las calidades obtenidas para cada procedimiento, categorizadas por el número de clientes. Los resultados son redondeados a una unidad decimal. Para estos el tamaño de la LRC para los procedimientos A y B fue de  $k = 3$ , mientras que para el procedimiento C fue de  $w = 5$ .

Se observa que las soluciones obtenidas por los procedimientos A y B poseen una mejor calidad en la mayoría de los casos. Por otro lado, el procedimiento C genera las soluciones de peor calidad y con una alta variabilidad, mientras que el resto presenta una menor desviación estándar, debido a que se eligen clientes según una función miope que toma como referencia un cliente estático para cada ruta. Este comportamiento se mantiene y las diferencias se acentúan a medida que aumenta el número de clientes. Esto es relevante, pues si bien es intuitivo pensar que generar soluciones cercanas al óptimo lleva a un mejor desempeño de algoritmos metaheurísticos, trabajos recientes (Q. Li, Liu, y Yang, 2020) demuestran que esta cercanía no asegura ni tiene correlación demostrable en mejoras de rendimiento global, por lo que es fundamental considerar métricas adicionales que consideren elementos como la estructura de las soluciones generadas, dando una mayor libertad de exploración del espacio de búsqueda.

En cuanto al número de sub-tours que se generan en las soluciones, se puede apreciar que los procedimientos A y B poseen un valor relativamente bajo, lo que sugiere configuraciones iniciales con sub-tours más extensas. En cambio, el procedimiento C, al ubicar los clientes de camión luego de los de vehículo, favorece la creación de un mayor número de sub-tours, lo que se mantiene al aumentar el número de clientes. Este patrón está alineado con la menor calidad observada en las soluciones generadas. La distribución de sub-tours se muestra en la Figura 6.1.

Por otro lado, respecto al largo de las rutas creadas, se presenta un comportamiento similar

Tabla 6.2: Calidad promedio y desviación estándar para instancias de 30 clientes

|      | A       |           | B       |           | C       |           |
|------|---------|-----------|---------|-----------|---------|-----------|
|      | Calidad | Dev. Est. | Calidad | Dev. Est. | Calidad | Dev. Est. |
| A-1  | 604.0   | 19.1      | 609.8   | 19.8      | 784.4   | 48.7      |
| A-2  | 608.9   | 21.6      | 616.2   | 20.5      | 795.7   | 47.9      |
| A-3  | 636.8   | 22.6      | 646.8   | 23.1      | 824.3   | 59.4      |
| A-4  | 625.2   | 21.6      | 629.6   | 22.9      | 831.5   | 52.0      |
| A-5  | 710.7   | 30.7      | 694.8   | 25.4      | 888.4   | 56.4      |
| A-6  | 839.8   | 41.1      | 783.4   | 36.2      | 939.0   | 67.4      |
| A-7  | 591.5   | 20.4      | 610.4   | 22.3      | 824.4   | 55.6      |
| A-8  | 609.7   | 23.1      | 593.5   | 20.2      | 850.6   | 53.7      |
| A-9  | 696.0   | 24.8      | 707.0   | 23.0      | 867.9   | 50.4      |
| A-10 | 638.0   | 21.0      | 686.7   | 23.5      | 878.4   | 59.4      |
| A-11 | 645.9   | 23.3      | 691.7   | 25.1      | 878.6   | 55.1      |
| A-12 | 742.5   | 24.3      | 767.5   | 24.4      | 955.3   | 66.9      |
| B-1  | 559.6   | 17.4      | 573.1   | 20.4      | 757.8   | 47.0      |
| B-2  | 571.8   | 18.5      | 575.1   | 20.3      | 732.9   | 38.3      |
| B-3  | 604.2   | 19.6      | 594.1   | 22.5      | 737.4   | 39.4      |
| B-4  | 541.1   | 19.0      | 579.0   | 20.3      | 809.3   | 53.2      |
| B-5  | 585.0   | 19.4      | 594.7   | 19.6      | 793.7   | 50.2      |
| B-6  | 657.1   | 20.0      | 631.4   | 23.8      | 796.3   | 45.8      |
| B-7  | 539.0   | 16.5      | 588.3   | 19.1      | 818.0   | 60.0      |
| B-8  | 566.7   | 18.4      | 612.3   | 21.9      | 824.0   | 59.1      |
| B-9  | 642.2   | 18.4      | 660.7   | 23.2      | 861.1   | 58.9      |
| B-10 | 572.0   | 18.0      | 614.7   | 25.5      | 897.4   | 78.9      |
| B-11 | 546.8   | 17.9      | 615.2   | 23.4      | 824.5   | 49.8      |
| B-12 | 623.8   | 19.6      | 668.4   | 26.3      | 894.2   | 78.8      |
| C-1  | 558.5   | 21.5      | 550.0   | 20.6      | 721.8   | 50.3      |
| C-2  | 569.2   | 21.3      | 553.2   | 17.4      | 680.1   | 37.6      |
| C-3  | 577.1   | 20.2      | 564.9   | 18.9      | 644.1   | 30.8      |
| C-4  | 574.6   | 22.4      | 561.5   | 20.4      | 755.7   | 50.6      |
| C-5  | 613.0   | 22.8      | 587.4   | 17.0      | 733.5   | 48.8      |
| C-6  | 664.9   | 28.0      | 633.7   | 22.2      | 712.2   | 40.2      |
| C-7  | 557.9   | 21.7      | 550.4   | 19.7      | 772.4   | 52.6      |
| C-8  | 676.2   | 28.4      | 647.6   | 23.2      | 762.7   | 48.2      |
| C-9  | 646.1   | 25.6      | 624.9   | 21.1      | 766.6   | 43.4      |
| C-10 | 660.0   | 27.6      | 643.4   | 22.4      | 811.9   | 51.3      |
| C-11 | 668.7   | 26.8      | 646.5   | 24.2      | 745.3   | 42.0      |
| C-12 | 734.4   | 28.5      | 708.7   | 22.7      | 783.0   | 37.2      |

Tabla 6.3: Calidad promedio y desviación estándar para instancias de 40 clientes

|      | A       |           | B       |           | C       |           |
|------|---------|-----------|---------|-----------|---------|-----------|
|      | Calidad | Dev. Est. | Calidad | Dev. Est. | Calidad | Dev. Est. |
| A-1  | 792.8   | 26.6      | 799.3   | 27.3      | 1005.2  | 65.1      |
| A-2  | 846.9   | 28.3      | 850.4   | 29.0      | 1063.2  | 63.8      |
| A-3  | 887.3   | 31.4      | 906.1   | 28.9      | 1116.9  | 77.3      |
| A-4  | 799.3   | 28.1      | 802.6   | 27.8      | 1047.0  | 60.1      |
| A-5  | 890.1   | 28.8      | 904.5   | 28.5      | 1141.3  | 63.7      |
| A-6  | 985.1   | 25.0      | 943.1   | 28.1      | 1231.5  | 90.2      |
| A-7  | 730.3   | 21.9      | 762.2   | 26.8      | 1027.0  | 62.6      |
| A-8  | 720.8   | 20.7      | 722.1   | 21.2      | 1031.8  | 62.1      |
| A-9  | 830.9   | 26.7      | 848.0   | 28.2      | 1079.5  | 63.7      |
| A-10 | 841.2   | 26.5      | 889.5   | 28.2      | 1072.2  | 63.1      |
| A-11 | 875.6   | 25.1      | 892.8   | 25.0      | 1165.5  | 72.3      |
| A-12 | 943.0   | 28.5      | 975.2   | 33.8      | 1176.4  | 66.6      |
| B-1  | 728.7   | 22.6      | 760.5   | 24.7      | 972.1   | 62.5      |
| B-2  | 765.0   | 26.2      | 777.0   | 23.0      | 997.4   | 65.8      |
| B-3  | 810.8   | 27.1      | 823.4   | 26.9      | 1008.2  | 74.3      |
| B-4  | 700.7   | 21.8      | 713.9   | 23.7      | 1018.6  | 64.3      |
| B-5  | 739.3   | 23.4      | 767.3   | 22.8      | 1033.8  | 58.9      |
| B-6  | 797.3   | 19.8      | 791.6   | 23.6      | 989.6   | 55.7      |
| B-7  | 705.9   | 23.2      | 727.4   | 20.8      | 1035.1  | 66.4      |
| B-8  | 712.5   | 19.8      | 724.4   | 23.1      | 974.8   | 57.4      |
| B-9  | 778.5   | 25.3      | 786.1   | 25.0      | 981.6   | 52.8      |
| B-10 | 745.2   | 24.4      | 784.0   | 27.0      | 1105.0  | 66.1      |
| B-11 | 779.0   | 24.5      | 797.6   | 23.4      | 1082.9  | 60.7      |
| B-12 | 834.6   | 30.2      | 823.6   | 24.2      | 1073.4  | 65.7      |
| C-1  | 719.5   | 28.0      | 697.0   | 25.3      | 902.7   | 58.8      |
| C-2  | 759.8   | 31.4      | 736.8   | 24.9      | 864.3   | 49.6      |
| C-3  | 783.1   | 26.4      | 753.1   | 23.7      | 817.0   | 42.9      |
| C-4  | 727.6   | 28.6      | 703.5   | 23.7      | 901.6   | 57.4      |
| C-5  | 804.4   | 29.9      | 766.5   | 20.8      | 901.7   | 55.0      |
| C-6  | 815.4   | 29.5      | 783.5   | 21.7      | 853.9   | 41.3      |
| C-7  | 706.8   | 23.7      | 690.8   | 22.2      | 968.0   | 59.2      |
| C-8  | 820.1   | 32.0      | 783.6   | 26.7      | 894.6   | 51.9      |
| C-9  | 746.4   | 23.9      | 730.6   | 21.9      | 854.0   | 47.4      |
| C-10 | 850.7   | 32.8      | 820.4   | 27.7      | 1057.8  | 63.2      |
| C-11 | 897.2   | 38.3      | 857.5   | 26.2      | 975.5   | 51.2      |
| C-12 | 969.1   | 35.9      | 925.9   | 26.4      | 952.0   | 48.8      |

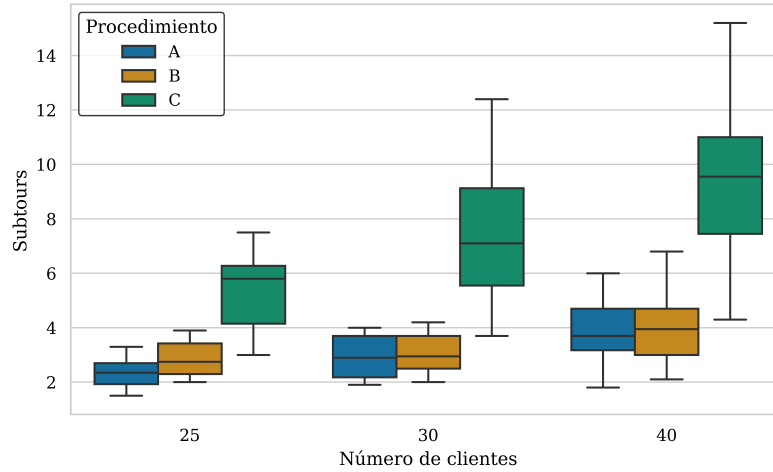


Figura 6.1: Distribución de n° de sub-tours según número de clientes.

para instancias de 25 clientes. Sin embargo, en instancias con 30 y 40 clientes, el procedimiento C tiende a generar rutas de mayor longitud, reflejando una menor calidad en la solución inicial construida, como se observa en la Figura 6.2.

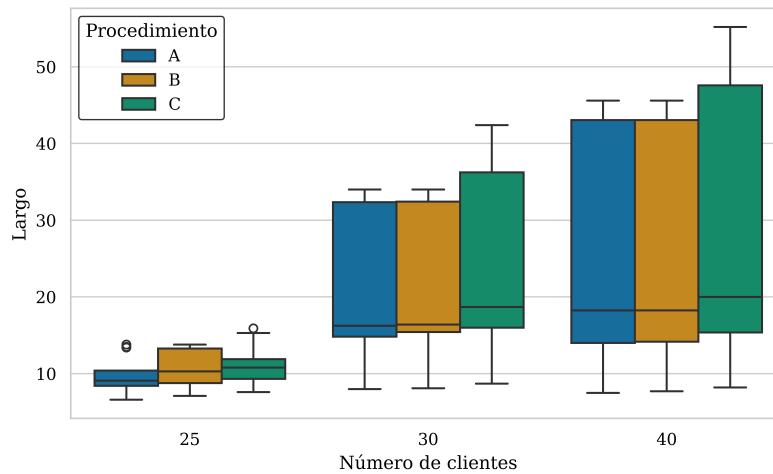


Figura 6.2: Distribución de largo de rutas según número de clientes.

Finalmente, se aplicó un análisis de varianza (ANOVA) de un factor sobre las métricas de número de sub-tours y largo total de rutas. Para el número de sub-tours, ANOVA mostró diferencias altamente significativas entre heurísticas ( $F=197,41$ ,  $p<0,001$ ), y la prueba post-hoc de Tukey HSD indicó que el procedimiento C difiere significativamente de A y B, mientras que no se encontró una diferencia significativa entre estos últimos. Para el largo de rutas no se observó diferencia significativa ( $F=1,32$ ,  $p=0,269$ ), y todas las comparaciones post-hoc resultaron no significativas. La Tabla 6.4 resume estos resultados y la Tabla 6.5 detalla las comparaciones post-hoc para el número de sub-tours. Esto indica que el procedimiento C produce un comportamiento distintivo en términos de sub-tours,

mientras que en el largo de rutas presentan desempeños comparables, a pesar de generar soluciones estructuralmente distintas.

Tabla 6.4: Resultados del análisis de varianza (ANOVA) de un factor sobre métricas de número de sub-tours y largo de rutas

| Métrica         | F      | Valor- $p$ | Significativo ( $\alpha = 0,05$ ) |
|-----------------|--------|------------|-----------------------------------|
| N° de sub-tours | 197.41 | < 0,001    | Sí                                |
| Largo de rutas  | 1.32   | 0.269      | No                                |

Tabla 6.5: Comparaciones post-hoc de Tukey HSD para el número de sub-tours

| Comparación | Diferencia de medias | Valor- $p$ ajustado | Significativo |
|-------------|----------------------|---------------------|---------------|
| A vs. B     | 0.18                 | 0.775               | No            |
| A vs. C     | 4.72                 | < 0,001             | Sí            |
| B vs. C     | 4.53                 | < 0,001             | Sí            |

## 6.5. Análisis de resultados

Los resultados y la comparación de todos los algoritmos, clasificados según el número de clientes, se presentan en las Tablas 6.6, 6.7, 6.8, 6.9 y 6.10. Las tablas se encuentran organizadas de acuerdo con el número de clientes. Los resultados se redondean a un decimal, mientras que los tiempos de ejecución se presentan en segundos, redondeados a la unidad más cercana. Para los resultados de la literatura se consideró un límite de tiempo de 7200 segundos.

Tabla 6.6: Resultados para las instancias del conjunto *Chao25*

| Inst. | Propuesta    |        | Literatura |        | Inst. | Propuesta    |        | Literatura |        | Inst. | Propuesta    |        | Literatura |        |
|-------|--------------|--------|------------|--------|-------|--------------|--------|------------|--------|-------|--------------|--------|------------|--------|
|       | Calidad      | Tiempo | Calidad    | Tiempo |       | Calidad      | Tiempo | Calidad    | Tiempo |       | Calidad      | Tiempo | Calidad    | Tiempo |
| 1     | <i>340.5</i> | 68     | 340.5      | 187    | 5     | <i>378.1</i> | 94     | 378.1      | 19     | 9     | <i>364.2</i> | 70     | 364.2      | 9      |
| 2     | <i>366.3</i> | 72     | 366.3      | 49     | 6     | <i>401.0</i> | 158    | 401.0      | 13     | 10    | <i>374.0</i> | 98     | 374.0      | 146    |
| 3     | <i>373.2</i> | 88     | 373.2      | 26     | 7     | <i>324.6</i> | 55     | 324.6      | 35     | 11    | <i>389.7</i> | 122    | 389.7      | 97     |
| 4     | <i>355.6</i> | 127    | 355.6      | 147    | 8     | <i>353.6</i> | 46     | 353.6      | 28     | 12    | <i>401.3</i> | 143    | 401.3      | 39     |

Los resultados reportados por (Bartolini y Schneider, 2020) alcanzan la solución óptima en 94 instancias. Las instancias para las cuales no fue posible obtener la solución óptima dentro del límite de tiempo establecido aparecen subrayadas, indicando que aún podría encontrarse una mejor cota superior. Los resultados que mejoran el estado del arte se destacan en negrita, mientras que aquellos que igualan el estado del arte se presentan en cursiva. Además, (Accorsi y Vigo, 2020) reportó nuevas cotas superiores para cinco instancias de tipo C con 40 clientes (específicamente, las instancias C-2, C-4, C-7, C-10 y C-12).



Tabla 6.7: Resultados para las instancias del conjunto *Chao30*

| Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        |
|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|
|       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |
| A-1   | 384.0     | 118    | 384.0      | 4554   | B-1   | 370.6     | 85     | 370.6      | 2732   | C-1   | 350.6     | 62     | 350.6      | 1830   |
| A-2   | 408.8     | 150    | 408.8      | 3990   | B-2   | 375.9     | 93     | 375.9      | 353    | C-2   | 371.6     | 66     | 371.6      | 3026   |
| A-3   | 421.8     | 166    | 421.8      | 255    | B-3   | 387.1     | 100    | 387.1      | 11     | C-3   | 383.6     | 71     | 383.6      | 61     |
| A-4   | 402.9     | 130    | 402.9      | 760    | B-4   | 356.8     | 80     | 356.8      | 163    | C-4   | 351.7     | 62     | 351.7      | 223    |
| A-5   | 458.8     | 178    | 458.8      | 7200   | B-5   | 395.0     | 97     | 395.0      | 43     | C-5   | 383.8     | 66     | 383.8      | 194    |
| A-6   | 504.7     | 211    | 512.1      | 7200   | B-6   | 417.0     | 99     | 417.0      | 12     | C-6   | 435.8     | 74     | 435.8      | 1064   |
| A-7   | 385.4     | 88     | 385.4      | 145    | B-7   | 372.5     | 93     | 372.5      | 216    | C-7   | 365.1     | 57     | 365.1      | 322    |
| A-8   | 389.6     | 111    | 389.6      | 28     | B-8   | 374.6     | 100    | 374.6      | 60     | C-8   | 419.5     | 81     | 419.5      | 137    |
| A-9   | 444.1     | 116    | 444.1      | 5      | B-9   | 387.2     | 124    | 387.2      | 7      | C-9   | 408.0     | 69     | 408.0      | 9      |
| A-10  | 431.8     | 134    | 431.8      | 3857   | B-10  | 374.6     | 103    | 374.6      | 57     | C-10  | 411.5     | 70     | 411.5      | 2012   |
| A-11  | 410.4     | 137    | 410.4      | 101    | B-11  | 350.7     | 98     | 350.7      | 8      | C-11  | 398.8     | 78     | 398.8      | 132    |
| A-12  | 451.2     | 260    | 451.2      | 403    | B-12  | 373.1     | 119    | 373.1      | 2      | C-12  | 471.8     | 91     | 471.8      | 601    |

Tabla 6.8: Resultados para las instancias del conjunto *Chao40*

| Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura         |        |
|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|-------|-----------|--------|--------------------|--------|
|       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad            | Tiempo |
| A-1   | 461.5     | 390    | 461.5      | 7200   | B-1   | 429.1     | 372    | 437.6      | 7200   | C-1   | 399.2     | 190    | 399.2              | 3844   |
| A-2   | 520.3     | 463    | 561.4      | 7200   | B-2   | 456.4     | 308    | 461.3      | 7200   | C-2   | 441.3     | 195    | 441.3 <sup>†</sup> | 7200   |
| A-3   | 530.2     | 552    | 530.2      | 7200   | B-3   | 478.8     | 446    | 477.5      | 7200   | C-3   | 472.3     | 218    | 472.3              | 7200   |
| A-4   | 506.6     | 379    | 507.7      | 7200   | B-4   | 440.7     | 259    | 440.7      | 7200   | C-4   | 414.5     | 185    | 414.5 <sup>†</sup> | 7200   |
| A-5   | 549.7     | 875    | 565.3      | 7200   | B-5   | 461.4     | 306    | 461.4      | 7200   | C-5   | 455.5     | 201    | 455.5              | 4725   |
| A-6   | 588.8     | 1082   | 593.0      | 7200   | B-6   | 485.0     | 297    | 485.5      | 7200   | C-6   | 501.0     | 233    | 501.0              | 7200   |
| A-7   | 454.7     | 265    | 465.0      | 7200   | B-7   | 435.4     | 189    | 434.4      | 3253   | C-7   | 422.2     | 165    | 422.2 <sup>†</sup> | 7200   |
| A-8   | 433.3     | 231    | 433.3      | 762    | B-8   | 413.6     | 269    | 413.6      | 314    | C-8   | 461.5     | 214    | 461.5              | 7200   |
| A-9   | 485.1     | 360    | 485.1      | 346    | B-9   | 464.9     | 241    | 464.9      | 13     | C-9   | 449.0     | 201    | 449.0              | 84     |
| A-10  | 522.6     | 341    | 526.2      | 7200   | B-10  | 468.5     | 210    | 468.0      | 4274   | C-10  | 485.5     | 243    | 485.5 <sup>†</sup> | 7200   |
| A-11  | 538.2     | 404    | 541.8      | 7200   | B-11  | 458.4     | 280    | 458.4      | 121    | C-11  | 500.7     | 250    | 500.7              | 2399   |
| A-12  | 597.6     | 450    | 597.6      | 3489   | B-12  | 473.2     | 293    | 473.2      | 33     | C-12  | 574.5     | 282    | 574.5 <sup>†</sup> | 7200   |

<sup>†</sup> Cota reportada en (Accorsi y Vigo, 2020), obtenida en un tiempo computacional de aproximadamente 10 segundos. El tiempo indicado en la tabla para estas instancias corresponde al límite de tiempo del método exacto de (Bartolini y Schneider, 2020), que no logró obtener un óptimo global en dicha ventana de tiempo.

Tabla 6.9: Resultados para las instancias del conjunto *Tai25*

| Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        |
|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|
|       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |
| 1     | 497.2     | 305    | 497.2      | 11     | 5     | 512.1     | 167    | 512.1      | 145    | 9     | 402.3     | 202    | 402.3      | 18     |
| 2     | 672.0     | 132    | 672.0      | 12     | 6     | 689.1     | 121    | 689.1      | 673    | 10    | 706.9     | 96     | 706.9      | 8      |
| 3     | 691.6     | 261    | 691.6      | 10     | 7     | 406.1     | 122    | 406.1      | 7200   | 11    | 738.9     | 131    | 738.9      | 182    |
| 4     | 444.4     | 150    | 444.4      | 225    | 8     | 413.4     | 170    | 413.4      | 241    | 12    | 819.8     | 411    | 820.3      | 7200   |

Tabla 6.10: Resultados para las instancias del conjunto *Tai30*

| Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        | Inst. | Propuesta |        | Literatura |        |
|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|-------|-----------|--------|------------|--------|
|       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |       | Calidad   | Tiempo | Calidad    | Tiempo |
| A-1   | 739.8     | 159    | 739.8      | 7200   | B-1   | 566.4     | 389    | 566.4      | 12     | C-1   | 508.9     | 66     | 508.9      | 2      |
| A-2   | 830.6     | 323    | 830.6      | 42     | B-2   | 638.9     | 351    | 638.9      | 10     | C-2   | 711.3     | 60     | 711.3      | 13     |
| A-3   | 875.2     | 189    | 875.2      | 61     | B-3   | 655.5     | 431    | 655.5      | 13     | C-3   | 757.4     | 77     | 757.4      | 6      |
| A-4   | 536.4     | 163    | 536.4      | 7200   | B-4   | 467.2     | 245    | 467.2      | 51     | C-4   | 310.6     | 65     | 310.4      | 4      |
| A-5   | 525.9     | 174    | 525.9      | 181    | B-5   | 511.2     | 447    | 511.2      | 533    | C-5   | 333.9     | 62     | 333.9      | 1      |
| A-6   | 806.7     | 260    | 806.7      | 7200   | B-6   | 676.5     | 592    | 676.5      | 7200   | C-6   | 359.1     | 81     | 359.1      | 1      |
| A-7   | 675.1     | 114    | 675.1      | 7200   | B-7   | 544.0     | 348    | 544.0      | 373    | C-7   | 491.1     | 54     | 491.1      | 5      |
| A-8   | 697.9     | 131    | 697.9      | 7200   | B-8   | 568.2     | 419    | 568.2      | 97     | C-8   | 538.5     | 63     | 538.5      | 4      |
| A-9   | 699.1     | 337    | 699.1      | 7200   | B-9   | 568.7     | 575    | 568.7      | 21     | C-9   | 585.4     | 69     | 585.4      | 11     |
| A-10  | 743.1     | 132    | 743.1      | 32     | B-10  | 729.2     | 200    | 729.2      | 727    | C-10  | 608.8     | 62     | 608.8      | 5      |
| A-11  | 788.3     | 141    | 788.3      | 7200   | B-11  | 761.6     | 259    | 761.0      | 7200   | C-11  | 667.6     | 62     | 667.6      | 21     |
| A-12  | 926.6     | 221    | 936.0      | 7200   | B-12  | 815.1     | 392    | 815.1      | 7200   | C-12  | 719.1     | 98     | 719.1      | 6      |

Inicialmente, se consideró utilizar únicamente el mejor de los tres procedimientos de generación de soluciones iniciales. Sin embargo, tras analizar los resultados, la evidencia sugiere que la combinación de los tres procedimientos constituye la estrategia más efectiva para abordar el problema. Para cada procedimiento se realizó un total de 1500 ejecuciones, partiendo de distintas soluciones iniciales, lo que dio lugar a 4500 experimentos en total. El tiempo total mostrado corresponde a la suma de los tiempos de ejecución de las 1500 ejecuciones realizadas para cada tipo de solución inicial.

En la totalidad de las 4500 ejecuciones realizadas por instancia, la solución final obtenida al finalizar la fase de reparación resultó factible, es decir, ninguna ruta ni sub-tour excedió la capacidad del camión o el remolque correspondiente. Esto confirma que el mecanismo de penalización adaptativa es suficiente para restaurar la factibilidad de la solución en la práctica, incluso partiendo de soluciones iniciales infactibles como las que puede generar el Procedimiento C.

De las 38 instancias en las que era posible mejorar el estado del arte, CORAL obtuvo una nueva cota en 13 instancias, igualó los mejores resultados conocidos en 23 y obtuvo resultados inferiores en 2. Para las 94 instancias restantes, se igualó el estado del arte en 91 casos y se obtuvieron resultados inferiores en los 3 restantes.

La desviación estándar reportada en las Tablas 6.11, 6.12, 6.13, 6.14 y 6.15 se mantiene relativamente baja en todos los conjuntos de instancias de Chao, lo que indica que la propuesta converge de manera consistente a soluciones de alta calidad. En la mayoría de las instancias, la desviación estándar se mantiene por debajo o cercana a 20. A medida que aumenta el tamaño de las instancias, ésta tiende a incrementarse ligeramente, existiendo algunos casos en los que supera el valor de 25. En los conjuntos de instancias de Tai se observa una mayor variabilidad. Este aumento en la dispersión se asocia a la mayor complejidad estructural de dichas instancias, principalmente por el mayor número de camiones y remolques, lo que genera un conjunto más diverso de óptimos locales. En conjunto, estos resultados indican que CORAL es robusto y presenta un comportamiento estable, incluso en instancias de mayor complejidad donde existe una mayor diversidad de soluciones.

Un aspecto destacable de CORAL es su eficiencia computacional en comparación con el método exacto propuesto por (Bartolini y Schneider, 2020). En la mayoría de las instancias más complejas, el método propuesto requiere un tiempo de ejecución considerablemente menor, lo que lo convierte en una alternativa más adecuada para aplicaciones de gran escala o con restricciones de tiempo.

Tabla 6.11: Estadísticas resumen de todas las ejecuciones para las instancias del conjunto *Chao25*

| Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. |
|-------|-------|---------|-----------|-------|-------|---------|-----------|-------|-------|---------|-----------|
| 1     | 363.6 | 361.9   | 14.6      | 5     | 404.0 | 403.0   | 13.1      | 9     | 409.3 | 404.4   | 27.9      |
| 2     | 391.8 | 391.5   | 15.6      | 6     | 432.6 | 433.2   | 17.9      | 10    | 396.9 | 394.9   | 14.6      |
| 3     | 400.1 | 398.2   | 20.7      | 7     | 349.6 | 347.7   | 16.1      | 11    | 420.1 | 418.6   | 16.0      |
| 4     | 376.0 | 374.9   | 14.0      | 8     | 371.4 | 369.2   | 11.3      | 12    | 438.5 | 438.0   | 22.3      |

Tabla 6.12: Estadísticas resumen de todas las ejecuciones para las instancias del conjunto *Chao30*

| Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. |
|-------|-------|---------|-----------|-------|-------|---------|-----------|-------|-------|---------|-----------|
| A-1   | 418.0 | 417.7   | 17.3      | B-1   | 396.6 | 394.9   | 12.7      | C-1   | 377.3 | 375.8   | 12.3      |
| A-2   | 444.5 | 444.1   | 16.5      | B-2   | 407.0 | 406.0   | 14.4      | C-2   | 397.6 | 397.2   | 14.0      |
| A-3   | 458.0 | 456.7   | 24.6      | B-3   | 421.2 | 420.3   | 19.9      | C-3   | 411.0 | 410.0   | 16.1      |
| A-4   | 436.6 | 436.1   | 16.3      | B-4   | 381.9 | 381.7   | 14.7      | C-4   | 373.8 | 371.9   | 15.3      |
| A-5   | 501.5 | 499.5   | 21.2      | B-5   | 420.9 | 419.9   | 14.3      | C-5   | 412.1 | 411.0   | 15.5      |
| A-6   | 579.8 | 578.7   | 33.2      | B-6   | 456.1 | 454.3   | 23.8      | C-6   | 463.3 | 460.1   | 17.6      |
| A-7   | 416.8 | 415.7   | 15.4      | B-7   | 395.4 | 394.0   | 14.9      | C-7   | 390.9 | 390.5   | 12.5      |
| A-8   | 412.6 | 411.8   | 13.0      | B-8   | 415.3 | 413.9   | 21.0      | C-8   | 443.7 | 443.0   | 14.3      |
| A-9   | 485.5 | 484.9   | 27.1      | B-9   | 446.5 | 445.3   | 29.2      | C-9   | 435.1 | 433.8   | 17.2      |
| A-10  | 459.3 | 457.5   | 12.9      | B-10  | 399.8 | 396.4   | 15.2      | C-10  | 434.4 | 433.0   | 11.0      |
| A-11  | 449.5 | 449.2   | 18.1      | B-11  | 382.5 | 380.6   | 16.3      | C-11  | 432.6 | 431.2   | 19.7      |
| A-12  | 515.9 | 514.9   | 20.3      | B-12  | 428.6 | 432.6   | 35.7      | C-12  | 504.4 | 502.9   | 16.5      |

Tabla 6.13: Estadísticas resumen de todas las ejecuciones para las instancias del conjunto *Chao40*

| Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. |
|-------|-------|---------|-----------|-------|-------|---------|-----------|-------|-------|---------|-----------|
| A-1   | 504.8 | 502.7   | 19.6      | B-1   | 478.4 | 477.8   | 18.7      | C-1   | 439.9 | 437.0   | 17.9      |
| A-2   | 577.9 | 577.0   | 20.3      | B-2   | 501.4 | 500.4   | 16.3      | C-2   | 485.5 | 484.0   | 19.8      |
| A-3   | 608.7 | 607.6   | 28.3      | B-3   | 532.5 | 531.1   | 21.7      | C-3   | 521.2 | 519.6   | 21.3      |
| A-4   | 540.5 | 538.9   | 17.4      | B-4   | 477.0 | 475.6   | 14.3      | C-4   | 445.2 | 442.2   | 17.1      |
| A-5   | 606.7 | 606.3   | 23.0      | B-5   | 497.1 | 496.6   | 15.7      | C-5   | 491.2 | 489.2   | 20.7      |
| A-6   | 649.9 | 648.0   | 23.3      | B-6   | 524.4 | 522.2   | 19.3      | C-6   | 535.3 | 532.5   | 20.0      |
| A-7   | 493.6 | 492.3   | 16.7      | B-7   | 473.4 | 472.7   | 15.0      | C-7   | 455.2 | 453.0   | 17.4      |
| A-8   | 461.1 | 460.5   | 13.2      | B-8   | 457.4 | 456.9   | 16.4      | C-8   | 499.6 | 498.0   | 17.7      |
| A-9   | 539.4 | 539.0   | 21.7      | B-9   | 516.1 | 515.5   | 24.2      | C-9   | 487.4 | 484.7   | 20.2      |
| A-10  | 574.5 | 574.1   | 18.3      | B-10  | 512.2 | 508.6   | 21.2      | C-10  | 518.3 | 515.7   | 15.2      |
| A-11  | 576.5 | 575.3   | 16.4      | B-11  | 502.0 | 501.5   | 16.1      | C-11  | 538.4 | 536.0   | 17.2      |
| A-12  | 655.2 | 650.9   | 32.7      | B-12  | 535.0 | 534.5   | 29.4      | C-12  | 609.1 | 606.7   | 17.7      |

Tabla 6.14: Estadísticas resumen de todas las ejecuciones para las instancias del conjunto *Tai25*

| Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. |
|-------|-------|---------|-----------|-------|-------|---------|-----------|-------|-------|---------|-----------|
| 1     | 558.9 | 548.2   | 42.2      | 5     | 595.6 | 569.7   | 74.0      | 9     | 414.1 | 410.8   | 10.3      |
| 2     | 773.7 | 752.9   | 64.7      | 6     | 748.2 | 728.0   | 58.4      | 10    | 780.5 | 767.3   | 63.5      |
| 3     | 742.6 | 732.8   | 42.1      | 7     | 423.9 | 421.3   | 13.5      | 11    | 854.8 | 863.2   | 70.8      |
| 4     | 560.5 | 601.7   | 100.7     | 8     | 426.9 | 422.4   | 11.9      | 12    | 916.6 | 881.9   | 76.9      |

Tabla 6.15: Estadísticas resumen de todas las ejecuciones para las instancias del conjunto *Tai30*

| Inst. | Media  | Mediana | Dev. Est. | Inst. | Media  | Mediana | Dev. Est. | Inst. | Media | Mediana | Dev. Est. |
|-------|--------|---------|-----------|-------|--------|---------|-----------|-------|-------|---------|-----------|
| A-1   | 828.5  | 808.4   | 58.0      | B-1   | 694.8  | 697.7   | 47.1      | C-1   | 537.4 | 529.4   | 28.6      |
| A-2   | 940.4  | 931.8   | 51.9      | B-2   | 728.6  | 720.0   | 61.6      | C-2   | 756.1 | 739.7   | 47.8      |
| A-3   | 969.6  | 962.4   | 50.0      | B-3   | 737.8  | 727.5   | 56.8      | C-3   | 802.3 | 784.2   | 40.7      |
| A-4   | 668.6  | 647.4   | 96.1      | B-4   | 526.5  | 489.7   | 65.7      | C-4   | 327.7 | 317.0   | 37.0      |
| A-5   | 617.9  | 585.2   | 79.8      | B-5   | 537.0  | 532.6   | 22.1      | C-5   | 339.1 | 333.9   | 19.8      |
| A-6   | 884.3  | 871.6   | 51.5      | B-6   | 734.0  | 702.7   | 59.3      | C-6   | 380.4 | 367.0   | 44.4      |
| A-7   | 700.1  | 697.8   | 15.3      | B-7   | 576.0  | 569.2   | 23.4      | C-7   | 506.7 | 501.0   | 15.3      |
| A-8   | 716.5  | 714.0   | 16.8      | B-8   | 608.2  | 605.1   | 33.0      | C-8   | 562.8 | 549.6   | 27.7      |
| A-9   | 714.7  | 714.1   | 14.2      | B-9   | 611.0  | 606.6   | 43.2      | C-9   | 619.5 | 614.2   | 30.6      |
| A-10  | 858.3  | 858.7   | 84.1      | B-10  | 779.5  | 776.7   | 41.0      | C-10  | 626.8 | 613.0   | 28.3      |
| A-11  | 941.5  | 928.3   | 78.7      | B-11  | 852.3  | 830.8   | 70.8      | C-11  | 715.8 | 676.8   | 66.7      |
| A-12  | 1103.6 | 1099.5  | 87.5      | B-12  | 1006.3 | 1016.1  | 91.9      | C-12  | 856.1 | 768.8   | 152.9     |

### 6.5.1. Efecto de soluciones iniciales

Utilizando soluciones iniciales generadas por el procedimiento A, el algoritmo obtuvo las mejores soluciones en 97 instancias, con el procedimiento B en 106 instancias y con el procedimiento C en 103 instancias. Naturalmente, en algunas instancias dos o más procedimientos contribuyeron a obtener las mejores soluciones, al igual que múltiples ejecuciones de cada uno de ellos. La Tabla 6.16 muestra que en 27 de las 132 instancias (20,5 %) la mejor solución obtenida por el algoritmo es alcanzada exclusivamente por uno de los tres procedimientos, mientras que en 36 instancias (27,3 %) coinciden exactamente dos. Por último en 69 instancias (52,3 %) los tres logran alcanzarlo. Esto confirma que ningún procedimiento predomina sobre otro y en caso de haber seleccionado un único procedimiento para construir las soluciones iniciales, no habría sido posible alcanzar la mejor solución encontrada para, al menos, una quinta parte de las instancias evaluadas.

Tabla 6.16: Distribución de las instancias del conjunto BART según los procedimientos de construcción cuyas ejecuciones alcanzan la mejor solución encontrada para dicha instancia

| Procedimiento(s) que alcanzan el óptimo | N° instancias |
|-----------------------------------------|---------------|
| Solo A                                  | 9             |
| Solo B                                  | 6             |
| Solo C                                  | 12            |
| A y B                                   | 14            |
| A y C                                   | 5             |
| B y C                                   | 17            |
| A, B y C                                | 69            |

Esta diversidad en el desempeño, junto con la relativamente baja variabilidad entre ejecuciones, sugiere que la integración de múltiples estrategias de generación de soluciones iniciales no solo está justificada, sino que resulta esencial, ya que permite al algoritmo aprovechar las fortalezas de cada procedimiento y explorar distintas regiones del espacio de búsqueda.

En las Tablas 6.17, 6.18 y 6.19 se presenta el óptimo obtenido al reparar las soluciones según el procedimiento de construcción aplicado. En negrita se destacan los procedimientos que alcanzan la mejor calidad encontrada. Se observa que en la mayoría de los casos dos o tres procedimientos coinciden en alcanzar el óptimo de la instancia, lo que es consistente con la Tabla 6.16. Sin embargo, existen instancias en las que solo uno de los procedimientos logra alcanzarlo, lo que refuerza la idea de que ningún procedimiento es predominante y justifica la estrategia colaborativa de construcción.

Tabla 6.17: Comparación de la calidad obtenida por cada procedimiento para las instancias de 25 clientes

| Instancia | Chao25       |              |              | Tai25        |              |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | A            | B            | C            | A            | B            | C            |
| 1         | <b>340.5</b> | <b>340.5</b> | <b>340.5</b> | <b>497.2</b> | <b>497.2</b> | <b>497.2</b> |
| 2         | <b>366.3</b> | <b>366.3</b> | <b>366.3</b> | <b>672.0</b> | <b>672.0</b> | <b>672.0</b> |
| 3         | <b>373.2</b> | <b>373.2</b> | <b>373.2</b> | 692.1        | <b>691.6</b> | <b>691.6</b> |
| 4         | <b>355.6</b> | <b>355.6</b> | <b>355.6</b> | 448.7        | 444.6        | <b>444.4</b> |
| 5         | <b>378.1</b> | <b>378.1</b> | <b>378.1</b> | <b>512.1</b> | <b>512.1</b> | <b>512.1</b> |
| 6         | 414.8        | <b>401.0</b> | <b>401.0</b> | <b>689.1</b> | 689.7        | 689.9        |
| 7         | 332.0        | <b>324.6</b> | <b>324.6</b> | <b>406.1</b> | <b>406.1</b> | <b>406.1</b> |
| 8         | <b>353.6</b> | <b>353.6</b> | <b>353.6</b> | <b>413.4</b> | <b>413.4</b> | 414.5        |
| 9         | 396.9        | <b>364.2</b> | <b>364.2</b> | <b>402.3</b> | <b>402.3</b> | <b>402.3</b> |
| 10        | <b>374.0</b> | 375.5        | <b>374.0</b> | 715.0        | <b>706.9</b> | <b>706.9</b> |
| 11        | 395.5        | <b>389.7</b> | <b>389.7</b> | <b>738.9</b> | 739.8        | <b>738.9</b> |
| 12        | 418.2        | <b>401.3</b> | <b>401.3</b> | <b>819.8</b> | <b>819.8</b> | <b>819.8</b> |

Tabla 6.18: Comparación de la calidad obtenida por cada procedimiento para las instancias de 30 clientes

| Instancia | Chao30       |              |              | Tai30        |              |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | A            | B            | C            | A            | B            | C            |
| A-1       | <b>384.0</b> | <b>384.0</b> | <b>384.0</b> | <b>739.8</b> | <b>739.8</b> | <b>739.8</b> |
| A-2       | <b>408.8</b> | <b>408.8</b> | <b>408.8</b> | <b>830.6</b> | 833.1        | <b>830.6</b> |
| A-3       | <b>421.8</b> | <b>421.8</b> | <b>421.8</b> | <b>875.2</b> | <b>875.2</b> | 877.0        |
| A-4       | <b>402.9</b> | <b>402.9</b> | <b>402.9</b> | <b>536.4</b> | <b>536.4</b> | 540.2        |
| A-5       | <b>458.8</b> | <b>458.8</b> | <b>458.8</b> | <b>525.9</b> | <b>525.9</b> | <b>525.9</b> |
| A-6       | <b>504.7</b> | 505.3        | 505.3        | <b>806.7</b> | <b>806.7</b> | <b>806.7</b> |
| A-7       | <b>385.4</b> | <b>385.4</b> | <b>385.4</b> | <b>675.1</b> | <b>675.1</b> | <b>675.1</b> |
| A-8       | <b>389.6</b> | <b>389.6</b> | <b>389.6</b> | <b>697.9</b> | <b>697.9</b> | <b>697.9</b> |
| A-9       | <b>444.1</b> | <b>444.1</b> | <b>444.1</b> | <b>699.1</b> | <b>699.1</b> | <b>699.1</b> |
| A-10      | <b>431.8</b> | <b>431.8</b> | 432.5        | <b>743.1</b> | <b>743.1</b> | <b>743.1</b> |

## CAPÍTULO 6. RESULTADOS

Tabla 6.18: Comparación de la calidad obtenida por cada procedimiento para las instancias de 30 clientes (continuación)

| Instancia | Chao30       |              |              | Tai30        |              |              |
|-----------|--------------|--------------|--------------|--------------|--------------|--------------|
|           | A            | B            | C            | A            | B            | C            |
| A-11      | <b>410.4</b> | <b>410.4</b> | <b>410.4</b> | 788.8        | 791.4        | <b>788.3</b> |
| A-12      | 472.1        | 456.3        | <b>451.2</b> | <b>926.6</b> | <b>926.6</b> | 929.5        |
| B-1       | 374.5        | <b>370.6</b> | 374.5        | <b>566.4</b> | <b>566.4</b> | 585.4        |
| B-2       | <b>375.9</b> | <b>375.9</b> | <b>375.9</b> | 639.0        | <b>638.9</b> | <b>638.9</b> |
| B-3       | <b>387.1</b> | <b>387.1</b> | <b>387.1</b> | 657.9        | <b>655.5</b> | 657.7        |
| B-4       | <b>356.8</b> | <b>356.8</b> | <b>356.8</b> | 474.6        | 475.7        | <b>467.2</b> |
| B-5       | <b>395.0</b> | <b>395.0</b> | <b>395.0</b> | 518.7        | 512.8        | <b>511.2</b> |
| B-6       | <b>417.0</b> | <b>417.0</b> | <b>417.0</b> | 679.8        | <b>676.5</b> | 687.2        |
| B-7       | 373.1        | 373.1        | <b>372.5</b> | <b>544.0</b> | <b>544.0</b> | <b>544.0</b> |
| B-8       | 397.9        | <b>374.6</b> | <b>374.6</b> | <b>568.2</b> | <b>568.2</b> | <b>568.2</b> |
| B-9       | 434.1        | <b>387.2</b> | <b>387.2</b> | <b>568.7</b> | <b>568.7</b> | <b>568.7</b> |
| B-10      | 381.7        | <b>374.6</b> | <b>374.6</b> | <b>729.2</b> | <b>729.2</b> | <b>729.2</b> |
| B-11      | 368.6        | <b>350.7</b> | <b>350.7</b> | 763.3        | 763.4        | <b>761.6</b> |
| B-12      | 422.0        | 394.0        | <b>373.1</b> | 834.1        | <b>815.1</b> | 816.8        |
| C-1       | <b>350.6</b> | <b>350.6</b> | 352.7        | <b>508.9</b> | <b>508.9</b> | <b>508.9</b> |
| C-2       | <b>371.6</b> | <b>371.6</b> | 371.8        | <b>711.3</b> | <b>711.3</b> | <b>711.3</b> |
| C-3       | <b>383.6</b> | <b>383.6</b> | <b>383.6</b> | <b>757.4</b> | <b>757.4</b> | <b>757.4</b> |
| C-4       | <b>351.7</b> | <b>351.7</b> | <b>351.7</b> | <b>310.6</b> | <b>310.6</b> | <b>310.6</b> |
| C-5       | <b>383.8</b> | <b>383.8</b> | <b>383.8</b> | <b>333.9</b> | <b>333.9</b> | <b>333.9</b> |
| C-6       | <b>435.8</b> | <b>435.8</b> | <b>435.8</b> | <b>359.1</b> | <b>359.1</b> | <b>359.1</b> |
| C-7       | <b>365.1</b> | <b>365.1</b> | <b>365.1</b> | <b>491.1</b> | <b>491.1</b> | <b>491.1</b> |
| C-8       | <b>419.5</b> | <b>419.5</b> | <b>419.5</b> | <b>538.5</b> | <b>538.5</b> | <b>538.5</b> |
| C-9       | <b>408.0</b> | <b>408.0</b> | <b>408.0</b> | <b>585.4</b> | <b>585.4</b> | <b>585.4</b> |
| C-10      | <b>411.5</b> | <b>411.5</b> | 412.3        | <b>608.8</b> | <b>608.8</b> | <b>608.8</b> |
| C-11      | <b>398.8</b> | <b>398.8</b> | <b>398.8</b> | <b>667.6</b> | <b>667.6</b> | <b>667.6</b> |
| C-12      | <b>471.8</b> | <b>471.8</b> | <b>471.8</b> | <b>719.1</b> | <b>719.1</b> | <b>719.1</b> |

Tabla 6.19: Comparación de la calidad obtenida por cada procedimiento para las instancias de 40 clientes (Chao40)

| Instancia | A            | B            | C            |
|-----------|--------------|--------------|--------------|
| A-1       | 467.5        | <b>461.5</b> | <b>461.5</b> |
| A-2       | <b>520.3</b> | 526.5        | 526.5        |
| A-3       | <b>530.2</b> | <b>530.2</b> | <b>530.2</b> |
| A-4       | <b>506.6</b> | <b>506.6</b> | <b>506.6</b> |
| A-5       | <b>549.7</b> | <b>549.7</b> | 552.0        |

Tabla 6.19: Comparación de la calidad obtenida por cada procedimiento para las instancias de 40 clientes

| Instancia | A            | B            | C            |
|-----------|--------------|--------------|--------------|
| A-6       | <b>588.8</b> | <b>588.8</b> | 593.4        |
| A-7       | 456.0        | <b>454.7</b> | <b>454.7</b> |
| A-8       | <b>433.3</b> | 433.4        | 434.2        |
| A-9       | <b>485.1</b> | <b>485.1</b> | <b>485.1</b> |
| A-10      | <b>522.6</b> | 523.6        | 530.8        |
| A-11      | <b>538.2</b> | 541.5        | 541.5        |
| A-12      | <b>597.6</b> | <b>597.6</b> | <b>597.6</b> |
| B-1       | 440.0        | 430.9        | <b>429.1</b> |
| B-2       | <b>456.4</b> | 458.3        | <b>456.4</b> |
| B-3       | 497.2        | <b>478.8</b> | 478.9        |
| B-4       | 444.2        | 448.8        | <b>440.7</b> |
| B-5       | <b>461.4</b> | <b>461.4</b> | <b>461.4</b> |
| B-6       | <b>485.0</b> | <b>485.0</b> | <b>485.0</b> |
| B-7       | <b>435.4</b> | 439.1        | <b>435.4</b> |
| B-8       | 429.5        | 429.5        | <b>413.6</b> |
| B-9       | <b>464.9</b> | <b>464.9</b> | <b>464.9</b> |
| B-10      | <b>468.5</b> | 468.6        | 476.9        |
| B-11      | 466.4        | 466.4        | <b>458.4</b> |
| B-12      | 499.1        | <b>473.2</b> | <b>473.2</b> |
| C-1       | 400.4        | <b>399.2</b> | 400.4        |
| C-2       | <b>441.3</b> | <b>441.3</b> | 441.7        |
| C-3       | <b>472.3</b> | <b>472.3</b> | 473.9        |
| C-4       | <b>414.5</b> | <b>414.5</b> | <b>414.5</b> |
| C-5       | <b>455.5</b> | <b>455.5</b> | <b>455.5</b> |
| C-6       | <b>501.0</b> | <b>501.0</b> | <b>501.0</b> |
| C-7       | <b>422.2</b> | <b>422.2</b> | <b>422.2</b> |
| C-8       | <b>461.5</b> | <b>461.5</b> | 464.3        |
| C-9       | <b>449.0</b> | <b>449.0</b> | <b>449.0</b> |
| C-10      | <b>485.5</b> | 488.6        | 487.9        |
| C-11      | <b>500.7</b> | 504.3        | 504.3        |
| C-12      | 576.7        | <b>574.5</b> | <b>574.5</b> |

### 6.5.2. Test de Wilcoxon

Se realizó una prueba de rangos con signo de Wilcoxon entre el conjunto de resultados reportados en la literatura y los mejores resultados obtenidos por el algoritmo propuesto en las 38 instancias de (Bartolini y Schneider, 2020) para las cuales no fue posible obtener la solución óptima. La Tabla 6.20 resume los rangos de las diferencias pareadas, dando como resultado que en 13 de las 38 instancias el método propuesto obtuvo un resultado de

menor costo que el reportado en la literatura, en 2 fue peor y en las 23 restantes se obtuvo el mismo resultado.

Tabla 6.20: Rangos de la prueba de Wilcoxon entre los resultados de la literatura y el algoritmo propuesto para las 38 instancias sin óptimo probado

|                  | N  | Rango promedio | Suma de rangos |
|------------------|----|----------------|----------------|
| Rangos negativos | 2  | 4.00           | 8.00           |
| Rangos positivos | 13 | 8.62           | 112.00         |
| Empates          | 23 |                |                |
| Total            | 38 |                |                |

Se utilizó como hipótesis alternativa,  $H_1$ , que las diferencias entre las muestras pareadas son significativamente distintas de cero. Los estadísticos de la prueba se presentan en la Tabla 6.21, obteniéndose  $Z = -2,926$  y un valor- $p$  (bilateral) de 0,003. Dado que  $0,003 < 0,05$ , se rechaza la hipótesis nula a favor de la hipótesis alternativa. Además, se observa que la suma de rangos positivos domina ampliamente por sobre la de rangos negativos, siendo consistente con la obtención de mejores resultados que los reportados en la literatura.

Tabla 6.21: Estadísticos de la prueba de Wilcoxon

|                         |        |
|-------------------------|--------|
| Z                       | -2,926 |
| Sig. asintót. (2 colas) | 0,003  |

### 6.5.3. Boxplots

La Figura 6.3 muestra que, en general, gran parte de las soluciones obtenidas se concentra cerca del óptimo reportado en la literatura. La mediana del gap se mantiene entre 6 % y 9 % para la mayoría de instancias. Por otro lado el rango intercuartil de las instancias Chao se mantiene por debajo del 15 % en su mayoría, sin embargo, las instancias Tai fueron más difíciles de resolver, presentando una dispersión considerablemente mayor, presentando una cantidad de valores atípicos no despreciable que supera un gap de 50 %, lo que sugiere que, para estos casos, algunas ejecuciones quedan estancadas en óptimos locales de baja calidad en lugar de converger a soluciones competitivas. Este comportamiento es consistente con la variabilidad reportada en las Tablas 6.14 y 6.15. Esto se explica por la mayor cantidad de camiones y remolques disponibles en estas instancias, que amplía el espacio de búsqueda.

Por otro lado, la Figura 6.4 presenta la distribución de las soluciones obtenidas para las trece instancias en las que se pudo encontrar una nueva cota. Si bien, en todas ellas se pudo encontrar una mejor solución por lo menos una vez, el porcentaje de ejecuciones que lo logra varía considerablemente entre instancias. Esto se puede evidenciar en la instancia de



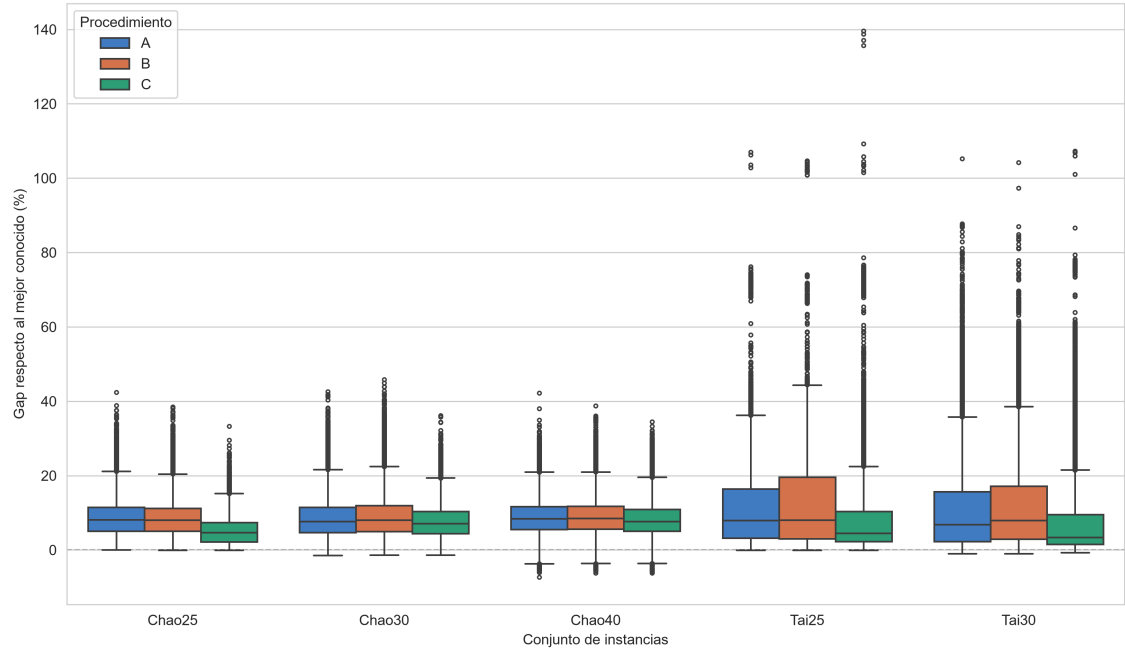


Figura 6.3: Distribución del gap porcentual respecto al mejor resultado conocido en la literatura, agrupado por conjunto de instancias y procedimiento de solución inicial.

CHAO con 40 clientes, A-2, en la que un 21.6 % de las ejecuciones obtienen una solución mejor que la cota previa, lo que indica que en algunas instancias la mejora es un evento poco frecuente, mientras que en otras el enfoque propuesto encuentra soluciones mejores que el estado del arte consistentemente.

Finalmente, la Figura 6.5 detalla este último caso para la instancia de CHAO con 40 clientes. En esta se observa que, si bien la mediana de las tres cajas se ubica por sobre la línea del mejor conocido, una fracción considerable de las ejecuciones logra igualar o superar dicha cota.

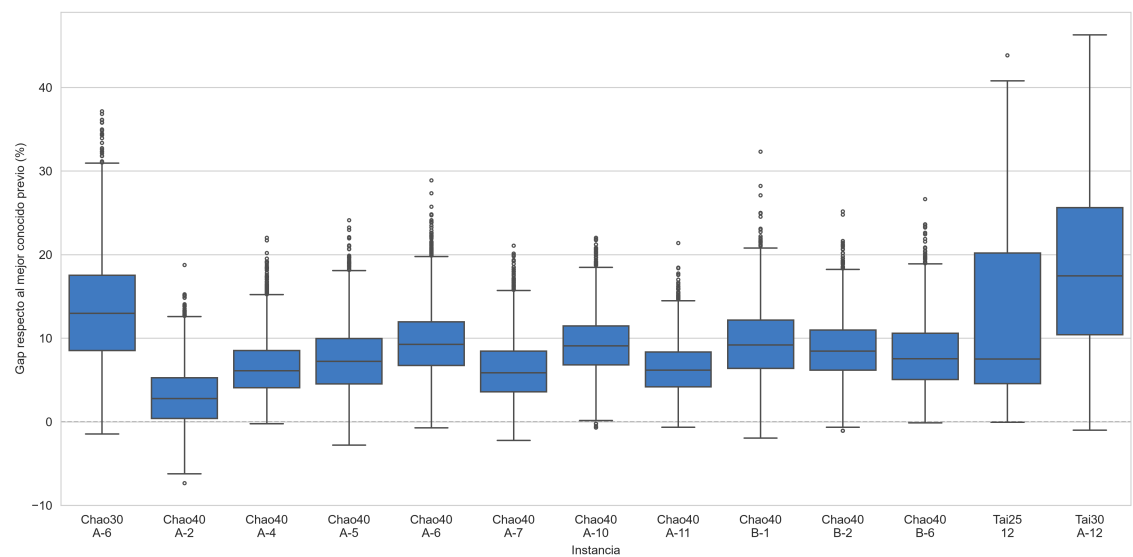


Figura 6.4: Distribución del gap porcentual respecto al mejor resultado previamente conocido, para las 13 instancias en las que CORAL obtuvo una nueva cota.

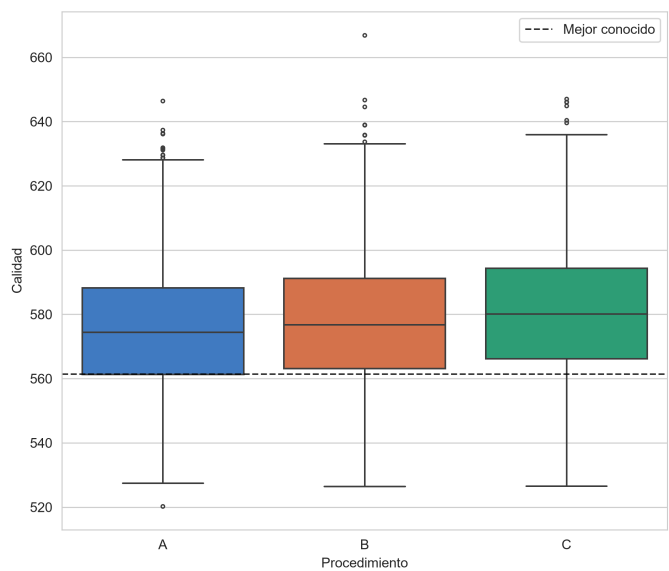


Figura 6.5: Distribución de la calidad de las soluciones obtenidas para la instancia Chao40 A-2, comparada con el mejor resultado conocido en la literatura.

## **6.6. Conclusiones del Capítulo**

Los resultados obtenidos permiten concluir que CORAL no solo es competitivo, sino también robusto para resolver instancias del conjunto BART. La capacidad de igualar una alta proporción de los mejores resultados conocidos y de mejorar cotas en instancias abiertas respalda la efectividad práctica del método en escenarios de alta dificultad.

Además, la evidencia experimental confirma que la diversidad en la fase de construcción no es un elemento accesorio, sino un componente determinante del desempeño global: la combinación de los tres procedimientos aporta complementariedad en la exploración y aumenta la probabilidad de alcanzar soluciones de mayor calidad.

En conjunto, estos resultados sostienen la hipótesis de la tesis y justifican la propuesta como una alternativa válida frente a enfoques previos. Sobre esta base, el capítulo siguiente presenta las conclusiones generales y las proyecciones de trabajo futuro.

# Capítulo 7

## Conclusiones

Se presentó CORAL, un algoritmo de búsqueda local de dos fases, centrado en resolver el CTTRP. Este consideró una fase de construcción basada en procedimientos de tipo GRASP y una fase de reparación basada en procedimientos *Hill-climbing*.

La representación elegida permitió que el algoritmo pudiese manejar de manera eficiente las soluciones candidatas obtenidas, evitando tener que recalcular la totalidad de la solución al aplicar un movimiento. Esto fue logrado, al solo considerar las diferencias incurridas en la distancia total de viaje al modificar arcos y modificar la capacidad disponible de la ruta al remover o incluir un cliente a ésta.

La fase de construcción fue capaz de generar soluciones factibles e infactibles fácilmente mejorables, cuyas características eran distintas y variadas. Esta diversidad, tanto en calidad como en estructura de las soluciones, es útil como punto de partida para algoritmos reparadores, poblacionales o estrategias de búsqueda local (Sarhani y cols., 2022). En particular, los procedimientos A y B mostraron una mejor calidad promedio, mientras que el procedimiento C generó soluciones con mayor número de sub-tours y mayor variabilidad, confirmando la influencia de la estrategia de construcción en la estructura de las soluciones. Además, se observa que el tamaño de la instancia influye en las características de la solución, por lo que distintos métodos pueden ser preferibles dependiendo del problema a resolver.

Los tres procedimientos de construcción de soluciones iniciales propuestos se guían por una función miope distinta y exploran regiones diferentes del espacio de búsqueda. El análisis del efecto de estas soluciones permite concluir que ningún procedimiento domina de manera absoluta sobre los demás, lo que es altamente deseable en metaheurísticas.

Por otro lado, en la fase de reparación, la secuencia de aplicación de procedimientos *Hill-climbing* fue estructurada en bloques de movimientos independientes, dos a nivel de ruta (2-opt y el refinamiento de raíces de sub-tours) y uno a nivel de cliente, compuesto por cinco movimientos simples. Para esta fase, el uso de Irace no se limitó a la sintonización de parámetros numéricos, como la penalización inicial o el número de iteraciones, sino que permitió abordar decisiones estructurales del algoritmo, como el orden de los bloques de movimientos o la inclusión del movimiento *swapBest*, tratándolos como parámetros categóricos sintonizables. De esta forma, el proceso de sintonización no solo permitió sistematizar la calibración de parámetros, sino también ser utilizado como herramienta de

apoyo al diseño estructural del algoritmo, definiendo configuraciones que obtienen mejores resultados.

Además, se introdujo una estrategia de penalización adaptativa que permite que la búsqueda considere soluciones infactibles momentáneamente, aumentando significativamente la penalización para forzar la convergencia a una solución factible. Esto, seguido de un procedimiento de perturbación que permite explorar otros óptimos locales cercanos al último encontrado.

Este acercamiento fue probado en un conjunto de instancias BART, comparando los resultados con los presentados en (Bartolini y Schneider, 2020) y (Accorsi y Vigo, 2020). Para cada instancia el algoritmo fue ejecutado con un tiempo límite de dos horas, siendo ejecutado 1500 veces por cada tipo de solución inicial, empezando de soluciones iniciales diferentes.

De las 132 instancias, en 114 se obtuvo un gap del 0 % respecto a los óptimos reportados en la literatura. En cuanto a las 38 instancias en las que no se conocía el óptimo global, CORAL obtuvo una nueva cota superior en 13 de ellas, igualando el mejor resultado en otras 23. En 5 instancias se obtuvieron resultados de peor calidad, siendo dos de estos en instancias en las que no se conoce el óptimo global y tres en instancias en las que el método completo sí pudo obtener el óptimo global.

Los experimentos presentados muestran que CORAL es considerablemente más eficiente en términos computacionales, lo que se evidencia a medida que el tamaño y la complejidad de las instancias crece, situaciones en las que una técnica completa puede no tener garantía de encontrar una solución óptima dentro de un límite de tiempo. Además, el algoritmo es capaz de explorar regiones del espacio de búsqueda y encontrar soluciones de buena calidad en un gran número de instancias.

## 7.1. Trabajo futuro

Como trabajo futuro, el método propuesto podría ser evaluado con otros conjuntos de instancias como los mencionados en el Estado del Arte, analizando su escalabilidad y calidad de soluciones obtenidas respecto a otros enfoques en los que un método exacto no puede entregar una solución en un tiempo acotado. De ser necesario, se podrían incluir otros movimientos más específicos al conjunto de instancias que se quiere resolver para mejorar la calidad de la solución o incluir elementos específicos del problema que se quiere resolver, considerando, por ejemplo, ventanas de tiempo o depósitos satélite.

Por otro lado, los experimentos mostraron que tanto el método exacto como CORAL fueron capaces de encontrar soluciones de buena calidad, por lo que un enfoque colaborativo que incluya ambos métodos podría ser una estrategia prometedora para abordar distintos tipos de instancias.

# Referencias

- Accorsi, L., y Vigo, D. (2020, 05). A hybrid metaheuristic for single truck and trailer routing problems. *Transportation Science*, 54, 1351-1371. doi: 10.1287/trsc.2019.0943
- Bartolini, E., y Schneider, M. (2020). A two-commodity flow formulation for the capacitated truck-and-trailer routing problem. *Discrete Applied Mathematics*, 275, 3 – 18. Descargado de <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85053353614&doi=10.1016%2fj.dam.2018.07.033&partnerID=40&md5=32629e4d5f55f5d276a71f8398e2007f> doi: 10.1016/j.dam.2018.07.033
- Batsyn, M., y Ponomarenko, A. (2014). Heuristic for a real-life truck and trailer routing problem. *Procedia Computer Science*, 31, 778-792. Descargado de <https://www.sciencedirect.com/science/article/pii/S1877050914005055> (2nd International Conference on Information Technology and Quantitative Management, ITQM 2014) doi: <https://doi.org/10.1016/j.procs.2014.05.328>
- Belenguer, J. M., Benavent, E., Martínez, A., Prins, C., Prodhon, C., y Villegas, J. G. (2015). A branch-and-cut algorithm for the single truck and trailer routing problem with satellite depots. *Transportation Science*, 50(2), 735–749. doi: 10.1287/trsc.2014.0593
- Caramia, M., y Guerriero, F. (2010, 04). A milk collection problem with incompatibility constraints. *Interfaces*, 40, 130-143. doi: 10.1287/inte.1090.0475
- Cavagnini, R., Schneider, M., y Theiß, A. (2023, 08). A granular iterated local search for the asymmetric single truck and trailer routing problem with satellite depots at dhl group. *Networks*, 83, 3-29. doi: 10.1002/net.22178
- Chao, I.-M. (2002). A tabu search method for the truck and trailer routing problem. *Computers & Operations Research*, 29(1), 33-51. Descargado de <https://www.sciencedirect.com/science/article/pii/S0305054800000563> doi: [https://doi.org/10.1016/S0305-0548\(00\)00056-3](https://doi.org/10.1016/S0305-0548(00)00056-3)
- Christofides, Nicos. (1976). The vehicle routing problem. *R.A.I.R.O. Recherche opérationnelle*, 10, 55-70. Descargado de <https://doi.org/10.1051/ro/197610V100551> doi: 10.1051/ro/197610V100551
- da Cruz, H. F. A., y Salles da Cunha, A. (2023). The profitable single truck and trailer routing problem with time windows: Formulation, valid inequalities and branch-and-cut algorithms. *Computers and Industrial Engineering*, 180. Descargado

- de <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85153276939&doi=10.1016%2fj.cie.2023.109238&partnerID=40&md5=2053b895b74380ca9e612ee5207e21b6> doi: 10.1016/j.cie.2023.109238
- Dantzig, G. B., Fulkerson, D. R., y Johnson, S. M. (1954). Solution of a large-scale traveling-salesman problem. *Operations Research*, 2(4), 393–410. doi: 10.1287/opre.2.4.393
- Dantzig, G. B., y Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1), 80–91. doi: 10.1287/mnsc.6.1.80
- Davila-Pena, L., R. Penas, D., y Casas-Méndez, B. (2023). A new two-phase heuristic for a problem of food distribution with compartmentalized trucks and trailers. *International Transactions in Operational Research*, 30(2), 1031 – 1064. Descargado de <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85117487396&doi=10.1111%2fitor.13071&partnerID=40&md5=a56ecd8a81e0356e641754861c39d71f> doi: 10.1111/itor.13071
- Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. doi: 10.1109/4235.996017
- Derigs, U., Pullmann, M., y Vogel, U. (2013). Truck and trailer routing—problems, heuristics and computational experience. *Computers & Operations Research*, 40(2), 536–546. Descargado de <https://www.sciencedirect.com/science/article/pii/S0305054812001724> doi: <https://doi.org/10.1016/j.cor.2012.08.007>
- Drexl, M. (2011, 12). Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, 12, 5–38.
- Drexl, M., y Sebastian, H.-J. (2007). On some generalized routing problems.. Descargado de <https://api.semanticscholar.org/CorpusID:33582346>
- Gehring, H., y Homberger, J. (2008, abril). *Extended vrptw benchmark instances*. <https://www.sintef.no/projectweb/top/vrptw/>. (Accessed: 2025-06-24)
- Gillett, B. E., y Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2), 340–349. Descargado 2026-08-06, de <http://www.jstor.org/stable/169591>
- Glover, F., Gutin, G., Yeo, A., y Zverovich, A. (2001). Construction heuristics for the asymmetric tsp. *European Journal of Operational Research*, 129(3), 555–568. Descargado de <https://www.sciencedirect.com/science/article/pii/S0377221799004683> doi: [https://doi.org/10.1016/S0377-2217\(99\)00468-3](https://doi.org/10.1016/S0377-2217(99)00468-3)
- Hoff, A., y Løkketangen, A. (2007). A tabu search approach for milk collection in western norway using trucks and trailers. En *6th triennial symposium on transportation analysis tristan vi*.
- Hoos, H., y Stützle, T. (2015, 01). Stochastic local search algorithms: An overview. ,

- 1085-1105. doi: 10.1007/978-3-662-43505-2\_54
- Hoos, H. H., y Stützle, T. (2004). Stochastic local search: Foundations & applications.. Descargado de <https://api.semanticscholar.org/CorpusID:42920353>
- Li, H., Chen, J., Wang, F., y Bai, M. (2021). Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. *European Journal of Operational Research*, 294(3), 1078-1095. Descargado de <https://www.sciencedirect.com/science/article/pii/S0377221721001156> doi: <https://doi.org/10.1016/j.ejor.2021.02.022>
- Li, Q., Liu, S.-Y., y Yang, X.-S. (2020). Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing*, 91, 106193. Descargado de <https://www.sciencedirect.com/science/article/pii/S1568494620301332> doi: <https://doi.org/10.1016/j.asoc.2020.106193>
- Li, X., Tian, P., y Leung, S. C. (2010). Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, 125(1), 137-145. Descargado de <https://www.sciencedirect.com/science/article/pii/S092552731000023X> doi: <https://doi.org/10.1016/j.ijpe.2010.01.013>
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 44(10), 2245-2269. doi: 10.1002/j.1538-7305.1965.tb04146.x
- Lin, S.-W., Yu, V., y Chou, S.-Y. (2010, 01). A note on the truck and trailer routing problem. *Expert Systems with Applications*, 37, 899-903. doi: 10.1016/j.eswa.2009.06.077
- Lin, S.-W., Yu, V. F., y Chou, S.-Y. (2009). Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research*, 36(5), 1683-1692. Descargado de <https://www.sciencedirect.com/science/article/pii/S0305054808000762> (Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X)) doi: <https://doi.org/10.1016/j.cor.2008.04.005>
- Lin, S.-W., Yu, V. F., y Lu, C.-C. (2011). A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, 38(12), 15244-15252. Descargado de <https://www.sciencedirect.com/science/article/pii/S0957417411008591> doi: <https://doi.org/10.1016/j.eswa.2011.05.075>
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Stützle, T., y Birattari, M. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43–58. doi: 10.1016/j.orp.2016.09.002
- Melian, B., Moreno-Pérez, J., y Moreno-Vega, J. (2003, 01). Metaheurísticas: Una visión global. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, ISSN 1137-3601, null 7, N.º. 19, 2003, pags. 7-28, 7.
- Mirmohammadsadeghi, S., y Ahmed, S. (2015, December). Memetic heuristic approach for solving truck and trailer routing problems with stochastic demands and time windows. *Networks and Spatial Economics*, 15(4), 1093-



1115. Descargado de <https://ideas.repec.org/a/kap/netspa/v15y2015i4p1093-1115.html> doi: 10.1007/s11067-014-9282-2
- Mirmohammadsadeghi, S., y Kabir, G. (2024). Formulation and solution of the stochastic truck and trailer routing problem. *Engineering Proceedings*, 76(1). Descargado de <https://www.mdpi.com/2673-4591/76/1/17> doi: 10.3390/engproc2024076017
- Mirmohammadsadeghi, S., Masoumik, S. M., y Alavi, S. (2020). Multi-point simulated annealing algorithm for solving truck and trailer routing problem with stochastic travel and service time. *Journal of Soft Computing & Decision Support Systems*, 7(6), 14.
- Navarro, A., Riff, M. C., González, B., Barros, T., y Barriga, R. (2025). Generation of diverse feasible solutions for the capacitated truck and trailer routing problem. En *2025 44th international conference of the chilean computer science society (sccc)* (p. 1-8). doi: 10.1109/SCCC67219.2025.11420346
- Nguyen, T., Bektas, T., Cherrett, T., Mcleod, F., Allen, J., Bates, O., ... Wise, S. (2018, 05). Optimising parcel deliveries in london using dual-mode routing. *Journal of the Operational Research Society*, 70. doi: 10.1080/01605682.2018.1480906
- Parragh, S. N., y Cordeau, J.-F. (2017). Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers & Operations Research*, 83, 28-44. Descargado de <https://www.sciencedirect.com/science/article/pii/S0305054817300266> doi: <https://doi.org/10.1016/j.cor.2017.01.020>
- Pérez-Rodríguez, R., y Urbán-Rivero, L. E. (2024, September). The cooperative algorithm with auxiliary objectives for the truck and trailer routing problem. *IAES International Journal of Artificial Intelligence*, 13(3), 2683–2693. doi: 10.11591/ijai.v13.i3.pp2683-2693
- Pérez-Rodríguez, R., y Urbán-Rivero, L. E. (2025). A cooperative algorithm to tackle the truck and trailer routing problem. *Computación y Sistemas*, 29(1), 365–375.
- Rochat, Y., y Taillard, E. (1995, 09). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1, 147-167. doi: 10.1007/BF02430370
- Rothenbächer, A.-K., Drexler, M., y Irnich, S. (2018, 05). Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science*, 52, 1174-1190. doi: 10.1287/trsc.2017.0765
- Sarhani, A., Benadada, Y., Erray, A., y Gendreau, M. (2022). Initialization of metaheuristics: comprehensive review, critical analysis, and practical guidelines. *International Transactions in Operational Research*, 29(2), 647–700. doi: 10.1111/itor.12939
- Scheuerer, S. (2006). A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research*, 33(4), 894-909. Descargado de <https://www.sciencedirect.com/science/article/pii/S0305054804001923> (Part Special Issue: Optimization Days 2003) doi: <https://doi.org/10.1016/j.cor.2004.08.002>
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems

- with time window constraints. *Oper. Res.*, 35, 254-265. Descargado de <https://api.semanticscholar.org/CorpusID:15346313>
- Talbi, E.-G. (2009). *Metaheuristics: From design to implementation* (Vol. 74). doi: 10.1002/9780470496916
- Torres, I., Cruz, C., y Verdegay, J. (2015, 05). Solving the truck and trailer routing problem with fuzzy constraints. *International Journal of Computational Intelligence Systems*, 8, 713. doi: 10.1080/18756891.2015.1046330
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., y Velasco, N. (2010). Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5), 780-794. Descargado de <https://www.sciencedirect.com/science/article/pii/S095219761000031X> (Advances in metaheuristics for hard optimization: new trends and case studies) doi: <https://doi.org/10.1016/j.engappai.2010.01.013>
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., y Velasco, N. (2011). A grasp with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9). Descargado de <https://doi.org/10.1016/j.cor.2010.11.011> doi: 10.1016/j.cor.2010.11.011
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., y Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2), 231-244. Descargado de <https://www.sciencedirect.com/science/article/pii/S037722171300324X> doi: <https://doi.org/10.1016/j.ejor.2013.04.026>