



Universidade de Vigo

Trabajo Fin de Máster

Modelos y algoritmos en una clase de problemas de rutas de vehículos

Laura Davila Pena

Máster en Técnicas Estadísticas

Curso 2018-2019

Propuesta de Trabajo Fin de Máster

<p>Título en galego: Modelos e algoritmos nunha clase de problemas de rutas de vehículos</p>
<p>Título en español: Modelos y algoritmos en una clase de problemas de rutas de vehículos</p>
<p>English title: Models and algorithms in a class of vehicle routing problems</p>
<p>Modalidad: Modalidad A</p>
<p>Autora: Laura Davila Pena, Universidade de Santiago de Compostela</p>
<p>Directores: Balbina Virginia Casas Méndez, Universidade de Santiago de Compostela; David Rodríguez Penas, Universidade de Santiago de Compostela</p>
<p>Breve resumen del trabajo:</p> <p>En el presente trabajo se realizará una revisión bibliográfica de algunos de los modelos, algoritmos y aplicaciones más relevantes relacionados con el <i>truck and trailer routing problem</i> (TTRP). Se complementará con la implementación en AMPL de alguna de las soluciones estudiadas y la programación en R de una heurística que se ha diseñado para resolver ciertos TTRPs.</p>
<p>Otras observaciones:</p> <p>Este TFM ha sido propuesto por Laura Davila Pena con el consentimiento de Balbina V. Casas Méndez y David Rodríguez Penas.</p>

Doña Balbina Virginia Casas Méndez, Profesora Titular del Departamento de Estadística, Análisis Matemático y Optimización de la Universidade de Santiago de Compostela y don David Rodríguez Penas, Contratado Post-Doctoral del Instituto de Matemáticas (IMAT) de la Universidade de Santiago de Compostela, informan que el Trabajo Fin de Máster titulado

Modelos y algoritmos en una clase de problemas de rutas de vehículos

fue realizado bajo su dirección por doña Laura Davila Pena para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En Santiago de Compostela, a 4 de julio de 2019.

La directora:

El director:

Doña Balbina Virginia Casas Méndez

Don David Rodríguez Penas

La autora:

Doña Laura Davila Pena

Índice general

Resumen	IX
Prefacio	XI
1. Una panorámica del TTRP	1
1.1. Evolución del TTRP y modelos representativos	3
1.1.1. Modelo VRPT de Gerdessen	4
1.1.2. Modelo TTRP de Chao	10
1.2. Formulación del TTRP con flota heterogénea	14
2. Resolución del TTRP	19
2.1. Métodos de resolución	19
2.1.1. Método de resolución de Chao	23
2.1.2. Métodos de resolución exactos	31
2.1.3. Instancias de referencia para el TTRP	32
2.2. Resolución del TTRP en AMPL	32
2.2.1. TTRP de Chao en AMPL	33
2.2.2. TTRP con flota heterogénea en AMPL	34
2.2.3. Resolución de un ejemplo real en AMPL	35
3. Heurística para el TTRP	39
3.1. Algoritmo de Clarke & Wright para el CVRP	40
3.2. Heurística basada en los ahorros para el TTRP	41
3.3. Resultados computacionales de la heurística CW_TTRP	51
4. TTRP con compartimentos	53
4.1. Revisión de problemas con compartimentos	54
4.2. Formulación del MC-TTRP	56
4.3. Resolución del MC-TTRP en AMPL	63

4.3.1. Resolución de un ejemplo en AMPL	63
4.3.2. Resolución de un problema real en AMPL	66
5. Conclusiones y trabajo futuro	73
A. VRP	75
A.1. Elementos principales del VRP	75
A.2. Formulaciones del CVRP	78
A.3. Métodos de resolución	80
A.3.1. Heurísticas clásicas	81
A.3.2. Metaheurísticas	83
B. Código de AMPL	85
C. Algoritmo de ahorros para un CVRP	111
Lista de acrónimos	119
Bibliografía	121

Resumen

Resumen en español

Los problemas de rutas de vehículos (VRPs) admiten diversas variantes, según las necesidades de los clientes. Una de ellas es el truck and trailer routing problem (TTRP), donde una flota de camiones y tráileres atienden a un conjunto de clientes tal que cuando el tráiler no puede acceder al cliente, este es atendido solo por el camión.

En este trabajo se revisan algunos de los modelos con tráileres más representativos de la literatura, así como sus métodos de resolución. Además, se formulan estos modelos como problemas de programación lineal y entera binaria. Como complemento al marco teórico, se resuelven algunos ejemplos de forma exacta, implementando su correspondiente formulación en AMPL. Posteriormente, se introduce e implementa en \mathbb{R} un nuevo algoritmo heurístico para el TTRP, con el que resolveremos los ejemplos anteriores.

Finalmente, se presenta un nuevo modelo, derivado del TTRP, que incorpora vehículos divididos en compartimentos: el MC-TTRP. Se incluye un problema real, relativo al reparto de piensos por parte de una cooperativa agrícola gallega y se resuelve con Gurobi haciendo uso de la formulación propuesta.

English abstract

Vehicle routing problems (VRPs) admit different variants, depending on the clients' needs. One of them is the truck and trailer routing problem (TTRP), in which a fleet of trucks and trailers serve a set of customers in such a way that when the trailer is not able to reach a customer, they are attended only by the truck.

In this project we review some of the most representative models with trailers in the literature, as well as their solution methods. Furthermore, these models are formulated as

mixed integer linear programming problems. As a complement to the theoretical framework, some examples are solved exactly, by implementing their corresponding formulation in AMPL. Afterwards, a new heuristic algorithm for the TTRP is introduced and implemented in R, with which we will solve the aforementioned examples.

Finally, a new model is presented, derived from the TTRP, which incorporates vehicles divided into compartments: the MC-TTRP. A real-world problem is included, related to the delivery of animal feed by a Galician agricultural cooperative and it is solved with Gurobi using the proposed formulation.

Prefacio

Los temas de logística y de rutas llevan siendo de vital trascendencia una gran cantidad de tiempo, pero cobran ahora mayor importancia ante una economía cada vez más globalizada. Nos encontramos ante una sociedad en la que los gobiernos, fábricas y particulares pueden requerir bienes de otra parte del mundo y obtenerlos en un tiempo razonable. El aumento de la demanda produce rutas logísticas más complejas, haciendo que los problemas de ruta óptima resultantes sean cada vez más grandes y difíciles de resolver. Así, es de vital importancia proponer nuevos modelos y algoritmos de optimización en este campo para reducir el impacto económico asociado a los procesos logísticos.

El problema de rutas de vehículos (VRP) trata de determinar el conjunto óptimo de rutas que debe realizar una flota de vehículos para servir a un cierto conjunto de clientes, y es uno de los problemas combinatorios de optimización más importantes y estudiados.

Este problema fue introducido por [Dantzig y Ramser \(1959\)](#), y surge como una generalización del problema del viajante ([Dantzig et al. 1954](#)). Dichos autores describían una aplicación real relacionada con el abastecimiento de gasolina desde una terminal a diversas estaciones de servicio, y propusieron tanto la primera formulación matemática del problema como una resolución algorítmica. Unos años más tarde, [Clarke y Wright \(1964\)](#) desarrollaron una heurística voraz efectiva, que mejoraba la aproximación de Dantzig–Ramser. Teniendo estos trabajos seminales como punto de partida, numerosos fueron los modelos y algoritmos de resolución propuestos en los años posteriores para obtener soluciones óptimas o aproximadas de los problemas de rutas.

El VRP clásico concierne la entrega de mercancía a unos clientes distribuidos geográficamente y cuyas demandas son determinísticas, conocidas con antelación y que no se pueden dividir. Existe un único depósito, del cual se debe partir y al cual han de volver todos los vehículos, que son idénticos. El objetivo es minimizar la distancia total necesaria para servir a todos los clientes. En la [Figura 1](#) podemos ver el esquema básico de un VRP.

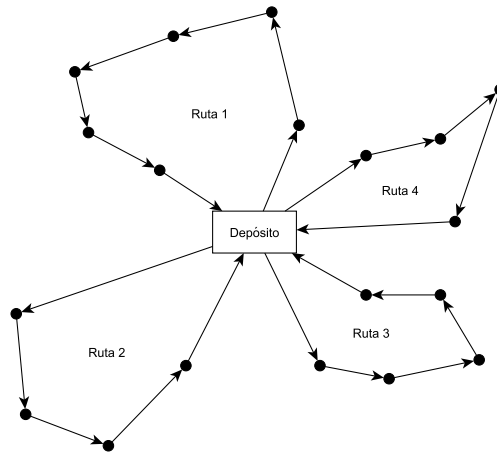


Figura 1: Esquema básico de un VRP.

Naturalmente, el VRP presenta múltiples variantes, que surgen como respuesta a situaciones reales. Algunas de estas modificaciones son:

- VRP con vehículos capacitados (CVRP), donde cada vehículo tiene asociada una capacidad máxima a transportar.
- VRP con ventanas de tiempo (VRPTW), donde cada cliente tiene asociado un intervalo de tiempo en el que debe ser atendido.
- VRP con recolección y entrega (VRPPD), en el cual los vehículos deben recoger y entregar una cierta cantidad de bienes a cada uno de los clientes.
- VRP con demandas estocásticas (VRPSD), donde las demandas de los clientes son variables aleatorias.
- VRP con *split-delivery* (SDVRP), tal que un mismo cliente puede ser atendido por varios vehículos.
- VRP con depósitos múltiples (MDVRP), en el cual existen diversos depósitos desde los cuales se pueden abastecer a los clientes.
- VRP con multi-compartimentos (MC-VRP), donde los clientes demandan diferentes tipos de productos que no se pueden mezclar, de modo que los vehículos deben estar divididos en compartimentos para poder transportarlos.

En la Figura 2 presentamos una taxonomía con una parte representativa de variantes del VRP. Cualquier combinación de ellas es también una extensión válida del problema.

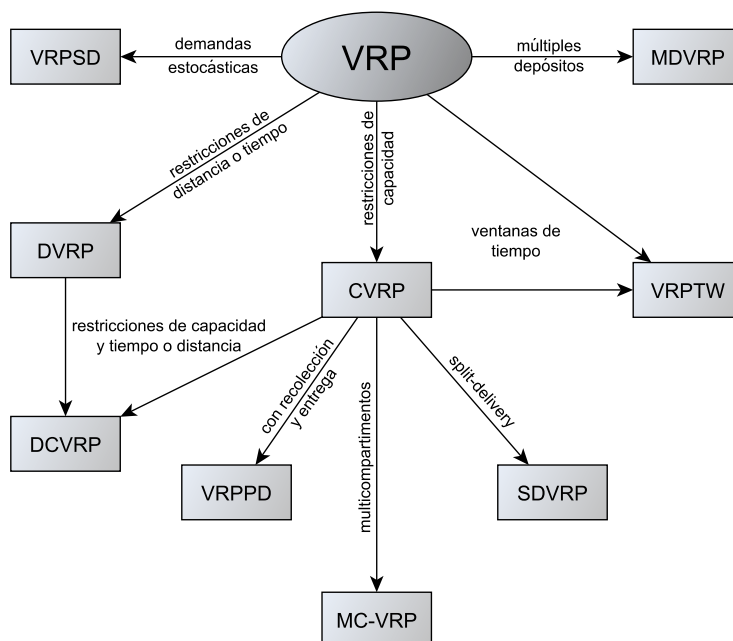


Figura 2: Diagrama con algunas variantes del VRP.

En la actualidad, hay disponibles docenas de paquetes comerciales para obtener soluciones de VRP reales. El interés por estudiar este problema se debe tanto a su relevancia práctica como a su considerable dificultad: determinar la solución óptima del VRP es un problema NP-duro (Lenstra y Rinnooy Kan 1981). Así, las instancias del VRP más grandes que han sido resueltas de forma consistente hasta este momento mediante los algoritmos exactos más efectivos, no contienen más de 50 clientes; mientras que los problemas más grandes solo pueden ser resueltos de forma óptima en algunos casos particulares.

En este trabajo nos centraremos en estudiar otra de las variantes del VRP, el *truck and trailer routing problem* (TTRP), introducida a principios de siglo. En esta ocasión, la flota de vehículos está formada por camiones y tráileres (que funcionan como remolques), y existen restricciones de acceso a los clientes: algunos no pueden ser atendidos con la combinación del camión y el tráiler, de modo que se debe desacoplar este último mientras los clientes inaccesibles con él son servidos solamente por el camión. Este problema se presenta, mayoritariamente, en situaciones de distribución de pienso o recolección de leche en granjas, ya que se requiere el transporte de elevadas cantidades de mercancía y, debido a las características de las carreteras y caminos, no siempre es viable el uso de vehículos de grandes dimensiones.

En el Capítulo 1 de este trabajo haremos una revisión exhaustiva del TTRP, mostrando algunos de los modelos más representativos existentes en la literatura, y formularemos también una versión del problema con flota heterogénea.

A continuación, en el Capítulo 2, se comienza presentando una panorámica de los métodos de resolución utilizados para el TTRP y algunas de sus variantes, explicando de forma detallada el empleado por Chao (2002). También se implementa el modelo en AMPL y se resuelven algunos ejemplos reales.

En el Capítulo 3 se propone y explica una heurística constructiva para resolver un TTRP, basada en el algoritmo de los ahorros de Clarke y Wright (1964).

Posteriormente, en el Capítulo 4, se introduce, formula y resuelve un nuevo modelo, derivado del TTRP, que incorpora vehículos compartimentados.

Para finalizar esta memoria, se incluye un capítulo de conclusiones y trabajo futuro, pues se pretenden continuar los estudios aquí presentes como parte de una tesis doctoral.

Capítulo 1

Una panorámica del TTRP

El problema de rutas de vehículos (VRP) presenta múltiples variantes, como la existencia de ventanas de tiempo, vehículos con capacidad limitada, divididos en compartimentos, o demandas que se pueden fraccionar al ser servidas o recogidas. Una de estas variantes, relacionada con la restricción de accesibilidad, es el denominado *truck and trailer routing problem* (TTRP). En este problema, los vehículos están formados por un camión y un tráiler, tal y como puede verse en la Figura 1.1. En lo que sigue, llamaremos vehículo al medio de transporte completo, combinación de ambos elementos.

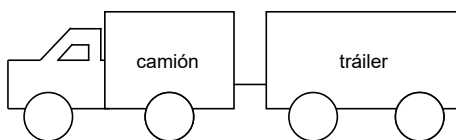


Figura 1.1: Vehículo, compuesto por un camión y un tráiler.

Los problemas de transporte con tráileres surgen con el fin de adaptarse a situaciones reales. El uso del vehículo puede hacerse inviable según qué ruta estemos considerando: en diversas ocasiones, nos podemos encontrar con caminos o fincas que requieren una gran capacidad de maniobra, imposible de llevar a cabo con el tráiler. El hecho de servir a determinados clientes solamente con el camión, puede presentar ventajas como el ahorro de combustible y la disminución del tiempo de trayecto.

Dentro de este contexto, el papel de las restricciones de accesibilidad es prohibir a unos determinados clientes, llamados clientes de camión (*truck customers*, t.c.), ser abastecidos por el vehículo completo, debido a la imposibilidad de llegar a ellos con el tráiler. Los clientes que sí son alcanzables por el vehículo, se denominan clientes de vehículo (*vehicle*

customers, v.c.), y podrán ser atendidos tanto por la combinación de camión y tráiler, como únicamente por el camión. A la hora de resolver este problema, distinguiremos tres tipos de rutas, detalladas a continuación e ilustradas en la Figura 1.2:

- Ruta de camión pura (PTR), recorrida únicamente por camiones, y que incluye tanto a clientes v.c. como a t.c.
- Ruta de vehículo pura (PVR), recorrida íntegramente por el vehículo, y en la cual solo se puede servir a clientes de tipo v.c.
- Ruta de vehículo completa (CVR), consistente en una ruta principal (MT) que recorre el vehículo y una o más sub-rutas a las que solo puede acceder el camión.

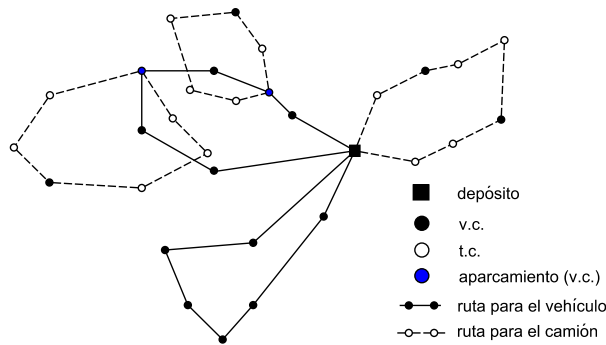


Figura 1.2: Rutas posibles en una solución a un TTRP.

Tal y como podemos observar en la Figura 1.2, las rutas de vehículo y camión puras se representan como circuitos clásicos, en los que el medio de transporte considerado en cada caso abandona el depósito, sirve a los clientes que le corresponden y regresa al depósito. No obstante, la ruta de vehículo completa presenta una particularidad: el vehículo parte del depósito, sirve a algunos v.c. y desacopla el tráiler, dejándolo aparcado en uno de esos clientes. En ese momento, el camión atiende, en lo que se conoce como sub-ruta, a ciertos clientes (de ambos tipos) y regresa al aparcamiento para montar de nuevo el tráiler, dirigiéndose al siguiente v.c. de la ruta principal. El proceso se repite hasta que todos los clientes de la ruta estén servidos, pudiendo presentarse más de una sub-ruta, o bien en distintos aparcamientos o incluso en el mismo. Finalmente, el vehículo regresa al depósito.

En la Sección 1.1 vamos a hacer una revisión de la evolución del TTRP y mencionar algunos de sus antecedentes o variantes más representativas, ahondando en los modelos de [Gerdessen \(1993\)](#) y [Chao \(2002\)](#). En la Sección 1.2, formularemos una extensión de este último modelo, en la cual se contempla el uso de flota heterogénea.

1.1. Evolución del TTRP y modelos representativos

El término “Truck and Trailer Routing Problem” fue introducido por primera vez por [Chao \(2002\)](#). Sin embargo, existen trabajos anteriores en los que se hace referencia al uso de los tráileres para resolver problemas reales de características similares al TTRP.

El primero de ellos se debe a [Semet y Taillard \(1993\)](#): motivados por el reparto de comida a supermercados en dos cantones de Suiza, estos autores estudiaron un VRP que incluía el uso de tráileres bajo restricciones de accesibilidad. Una diferencia con respecto al TTRP clásico, es que los tráileres estaban cargados con bienes únicamente para los clientes v.c. Además, se incluían ventanas de tiempo y costes distintos según el tipo de vehículo a utilizar.

Dos años más tarde, [Semet \(1995\)](#) propone una nueva variante del VRP, el PACVRP (*Partial Accesibility Constrained Vehicle Routing Problem*) y lo formula como un problema de programación lineal entera. A pesar de tratarse de un modelo muy similar al TTRP de Chao, presenta ciertas disimilitudes, como por ejemplo:

- Solo puede haber una sub-ruta para cada aparcamiento.
- Se usan todos los camiones disponibles.
- Hay que determinar el número de tráileres.

El hecho de utilizar toda la flota de camiones y tener que hallar el número de tráileres necesario, incrementa el coste total asociado, pues implica la contratación de más conductores (cada camión necesita un conductor).

Otra autora que modificó el VRP y consideró una flota heterogénea de vehículos, con camiones y tráileres, fue [Gerdessen \(1996\)](#), cuyo modelo se conoce como VRPT (*Vehicle Routing Problem with Trailers*). En la Subsección 1.1.1 detallaremos en qué consiste esta variante, presentando también su formulación como un problema de programación lineal entera mixta.

Unos años más tarde, [Chao et al. \(1998\)](#) estudian el SVRP (*Site-Dependent Vehicle Routing Problem*), en el que cada cliente tenía asignado un tipo de vehículo. Este dependerá de varios factores relativos a los clientes, como por ejemplo: su demanda, que puede hacer que dicho cliente tenga que ser atendido por un vehículo de gran capacidad; su localización, como en el caso del TTRP, ya que algunas zonas solo son accesibles con vehículos

pequeños, etc.

El modelo que propone [Chao \(2002\)](#), como ya se ha mencionado, es el que se corresponde con el TTRP clásico, y ahondaremos en él en la Subsección [1.1.2](#). A partir de ese modelo, se han ido construyendo otros con el objetivo de suplir algunas de las necesidades presentes en la vida real. Un ejemplo de ello es el TTRP con múltiples depósitos (MDTTRP), introducido por [Scheuerer \(2004\)](#); o el TTRP con ventanas de tiempo (TTRPTW), de [Lin et al. \(2011\)](#). En este último, cada cliente tiene asociados, a mayores de su tipo y su demanda, tres tiempos: el tiempo más temprano y último del día en el que puede ser atendido y el tiempo de servicio requerido.

1.1.1. Modelo VRPT de Gerdessen

El VRPT es una de las primeras variantes del VRP en la que se alude al uso de camiones con tráileres para realizar determinadas rutas. A diferencia del TTRP, aquí no existen clientes de distintos tipos, sino que se asume que todos son alcanzables por cualquier vehículo. No obstante, se establecen diferencias de tiempo, para un mismo trayecto, según se utilice el camión o el vehículo completo. Esta diferencia de tiempo es lo que se conoce como tiempo de maniobra, y será uno de los factores que este modelo tratará de minimizar.

Antes de nada, vamos a exponer cuáles son las tareas a realizar diariamente por el vehículo en cuestión:

- Partir del depósito con el camión y el tráiler.
- Servir a aquellos clientes fácilmente alcanzables para el vehículo.
- Estacionar el tráiler en algún aparcamiento.
- Servir, solamente con el camión, a los clientes difíciles de atender con el vehículo.
- Volver al aparcamiento y montar el tráiler.
- De nuevo, atender a los clientes de fácil acceso con el vehículo.
- Regresar al depósito.

En la [Figura 1.3](#), mostramos un ejemplo de solución para el VRPT. Podemos ver que difiere de la vista en la [Figura 1.2](#) en un aspecto fundamental: no existen diferentes tipos de rutas, sino que todas son rutas completas de vehículo con una sub-ruta, recorrida por el camión, y cuyo aparcamiento puede ser cualquiera de los clientes.

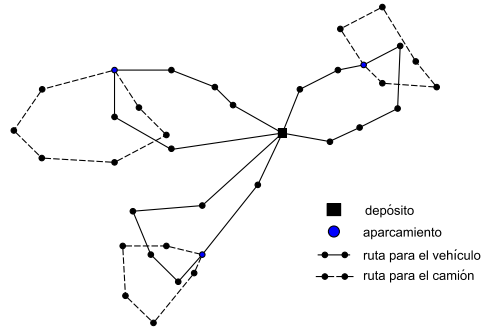


Figura 1.3: Solución al VRPT.

Como cabría esperar de lo mencionado anteriormente, el modelo que propone [Gerdessen \(1996\)](#) es una simplificación de lo que ocurre en la mayor parte de las situaciones reales, ya que se asumen los siguientes supuestos:

- El depósito y los clientes son vértices de un espacio euclídeo.
- El tiempo que un camión o vehículo tardan en realizar un trayecto es proporcional a la distancia.
- No hay ventanas de tiempo.
- Un cliente es atendido exactamente por un vehículo (o camión).
- Cada cliente es válido como aparcamiento del tráiler.
- Cada tráiler se aparca una sola vez.
- Todos los clientes tienen demanda única.
- La capacidad para cada vehículo es la misma.

En [Gerdessen \(1993\)](#) se formula el VRPT como un problema de programación lineal entera mixta, algo que no se había hecho hasta el momento con ninguno de los modelos con tráileres. La función objetivo a minimizar será la suma del tiempo que tardan tanto los vehículos como los camiones en realizar sus rutas y el tiempo total de maniobra. Con el fin de facilitar la comprensión de dicha formulación, presentamos en la Tabla 1.1 las constantes, conjuntos, índices y variables utilizadas.

Constantes y conjuntos	
n	Número de clientes
N	Conjunto de índices de los vértices (0 para el depósito; $1, \dots, n$ para los clientes)
K	Número de vehículos (combinaciones de un camión y un tráiler)
dv_{ij}	Tiempo que un vehículo necesita para viajar del nodo i al j
dt_{ij}	Tiempo que un camión necesita para viajar del nodo i al j
m_i	Tiempo extra de maniobra necesario si el cliente i es atendido por el vehículo en lugar de por el camión
q_i	Demanda del cliente i
$capt$	Capacidad del camión
$capv$	Capacidad del vehículo
Índices	
i, j	Índice para un vértice/cliente, $0 \leq i \leq n, 0 \leq j \leq n$
k	Índice para el vehículo/camión, $1 \leq k \leq K$
Variables	
xv_{ijk}	$= \begin{cases} 1 & \text{si el vehículo } k \text{ pasa por el arco } (i, j) \\ 0 & \text{si el vehículo } k \text{ no pasa por el arco } (i, j) \end{cases}$
xt_{ijk}	$= \begin{cases} 1 & \text{si el camión } k \text{ pasa por el arco } (i, j) \\ 0 & \text{si el camión } k \text{ no pasa por el arco } (i, j) \end{cases}$
yp_{ik}	$= \begin{cases} 1 & \text{si el vehículo } k \text{ aparca el tráiler en el vértice } i \\ 0 & \text{si el vehículo } k \text{ no aparca el tráiler en el vértice } i \end{cases}$
yv_{ik}	$= \begin{cases} 1 & \text{si el vehículo } k \text{ sirve al cliente } i \\ 0 & \text{si el vehículo } k \text{ no sirve al cliente } i \end{cases}$
yt_{ik}	$= \begin{cases} 1 & \text{si el camión } k \text{ sirve al cliente } i \\ 0 & \text{si el camión } k \text{ no sirve al cliente } i \end{cases}$

Tabla 1.1: Constantes, conjuntos, índices y variables para la formulación del VRPT.

Una vez establecida esta notación, estamos en condiciones de formular el modelo:

$$\text{minimizar } \left\{ \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K dv_{ij} xv_{ijk} + \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K dt_{ij} xt_{ijk} + \sum_{i=1}^n \sum_{k=1}^K m_i yv_{ik} \right\} \quad (1.1)$$

sujeto a

$$\sum_{j=0}^n \sum_{k=1}^K (xv_{ijk} + xt_{ijk}) = 1 + \sum_{k=1}^K yp_{ik}, \quad i = 1, \dots, n; \quad (1.2)$$

$$\sum_{i=0}^n \sum_{k=1}^K (xv_{ijk} + xt_{ijk}) = 1 + \sum_{k=1}^K yp_{jk}, \quad j = 1, \dots, n; \quad (1.3)$$

$$\sum_{i=0}^n yp_{ik} = 1, \quad k = 1, \dots, K; \quad (1.4)$$

$$\sum_{j=0}^n xv_{jik} = \sum_{j=0}^n xv_{ijk}, \quad i = 0, \dots, n; \quad k = 1, \dots, K; \quad (1.5)$$

$$\sum_{j=0}^n xt_{jik} = \sum_{j=0}^n xt_{ijk}, \quad i = 0, \dots, n; \quad k = 1, \dots, K; \quad (1.6)$$

$$\sum_{j=0}^n xv_{ijk} = yv_{ik}, \quad i = 1, \dots, n; \quad k = 1, \dots, K; \quad (1.7)$$

$$\sum_{j=0}^n xt_{ijk} = yt_{ik}, \quad i = 1, \dots, n; \quad k = 1, \dots, K; \quad (1.8)$$

$$\sum_{k=1}^K yv_{ik} \leq 1, \quad i = 1, \dots, n; \quad (1.9)$$

$$\sum_{k=1}^K yt_{ik} \leq 1, \quad i = 1, \dots, n; \quad (1.10)$$

$$yv_{ik} + yt_{ik} \leq 1 + yp_{ik}, \quad i = 1, \dots, n; \quad k = 1, \dots, K; \quad (1.11)$$

$$\sum_{j=0}^n xv_{0jk} = 1, \quad k = 1, \dots, K; \quad (1.12)$$

$$\sum_{i=0}^n xv_{i0k} = 1, \quad k = 1, \dots, K; \quad (1.13)$$

$$yv_{ik} \geq yp_{ik}, \quad i = 1, \dots, n; \quad k = 1, \dots, K; \quad (1.14)$$

$$yt_{ik} \geq yp_{ik}, \quad i = 1, \dots, n; \quad k = 1, \dots, K; \quad (1.15)$$

$$\sum_{i=1}^n q_i (yv_{ik} + yt_{ik} - yp_{ik}) \leq capv, \quad k = 1, \dots, K; \quad (1.16)$$

$$\sum_{i=1}^n q_i(yt_{ik} - yp_{ik}) \leq capt, \quad k = 1, \dots, K; \quad (1.17)$$

$$\sum_{i \in B} \sum_{j \in B} \sum_{k=1}^K xv_{ijk} \leq |B| - 1, \quad \forall B \subseteq N \setminus \{0\}; \quad (1.18)$$

$$\sum_{i \in B} \sum_{j \in B} \sum_{k=1}^K xt_{ijk} \leq |B| - 1 + \sum_{i \in B} \sum_{k=1}^K yp_{ik}, \quad \forall B \subseteq N \setminus \{0\}; \quad (1.19)$$

Resumimos a continuación a qué hace referencia cada una de las restricciones anteriores:

(1.1): la función objetivo es el tiempo total empleado para visitar a los clientes (tiempo de viaje y tiempo de maniobra).

(1.2): el número de arcos salientes de i es 2 si i se usa como aparcamiento, y 1 en caso contrario.

(1.3): el número de arcos entrantes en j es 2 si j se usa como aparcamiento, y 1 en caso contrario.

(1.4): el vehículo k usa exactamente un único vértice como aparcamiento.

(1.5): si el vehículo k entra en el vértice i , también tiene que salir de él.

(1.6): si el camión k entra en el vértice i , también tiene que salir de él.

(1.7): el vehículo k visita al cliente i , si dicho vehículo viaja a i .

(1.8): el camión k visita al cliente i , si dicho camión viaja a i .

(1.9): el cliente i no puede ser visitado por más de un vehículo.

(1.10): el cliente i no puede ser visitado por más de un camión.

(1.11): al cliente i solo lo pueden visitar un camión y un vehículo si ejerce como aparcamiento para el tráiler de dicho vehículo.

(1.12): todos los vehículos parten del depósito.

(1.13): cada vehículo regresa al depósito.

(1.14): el cliente i no puede ejercer como aparcamiento para el tráiler k si el vehículo k no lo visita.

(1.15): el cliente i no puede ejercer como aparcamiento para el tráiler k si el camión k no lo visita.

(1.16): restricción de capacidad para un vehículo. El aparcamiento es un cliente servido por el vehículo.

(1.17): restricción de capacidad para un camión.

(1.18): restricción de eliminación de subtour para las rutas de los vehículos.

(1.19): restricción de eliminación de subtour para las rutas de camiones.

El trabajo de [Gerdessen \(1996\)](#), como ya hemos mencionado, supone el estudio detallado de un campo hasta el momento desconocido: los problemas de rutas con tráileres. Dos de las grandes simplificaciones que presenta su modelo, con respecto al que veremos a continuación, son (i) que todos los clientes tienen demanda única y (ii) que cada tráiler se aparca exactamente una vez.

La asunción (i) hace más sencillo el problema ya que podemos conocer de antemano el número mínimo de vehículos a utilizar. No es difícil obtener una solución factible con un número de vehículos mayor o igual que el mínimo. Esto no ocurre cuando las demandas son diferentes: en tal caso, el cálculo del número mínimo de automóviles, así como la obtención de una solución factible donde se utilice esa cantidad mínima de vehículos, se complica.

Por otro lado, la asunción (ii) establece que cada tráiler se estaciona una única vez. En la práctica, esto podría ser útil si el número de veces que se puede aparcar el tráiler estuviese determinado por el modelo. Entonces, el coste derivado de acoplar y desacoplar el tráiler debería verse reflejado en la función objetivo (por ejemplo, en unidades de tiempo). La autora propone desarrollar un modelo donde el número de veces que se puede aparcar el tráiler sea libre.

Además, en relación con los aparcamientos, cabe destacar que en el VRPT cualquier cliente es candidato a servir como *parking* del tráiler. No todas las ubicaciones pueden ser igualmente elegibles, bien sea por cuestiones de seguridad o de espacio. Gerdessen también propone restringir el número de posibles aparcamientos a un subconjunto de los clientes tal que no suponga un inconveniente.

El modelo que veremos a continuación mejora algunas de las carencias del VRPT, siendo de esta forma una adaptación más fiel a lo que ocurre en ciertas situaciones reales.

1.1.2. Modelo TTRP de Chao

En esta subsección vamos a mostrar una formulación del TTRP de Chao, presentada en el trabajo de Scheuerer (2004), y que se trata de una adaptación de la propuesta por Semet (1995) para el PACVRP. Cabe destacar que Chao (2002), si bien considera que los camiones tardan el mismo tiempo en realizar un trayecto independientemente de que tengan o no un tráiler acoplado, sí hace hincapié en el tiempo máximo de utilización de un camión/vehículo. Es importante valorar esta restricción, que no se había visto antes, ya que puede reflejar, entre otras cosas, las limitaciones de combustible o la jornada laboral del conductor en cuestión. Además, también se tendrá en cuenta un nuevo parámetro, s_0 , que hace referencia al tiempo empleado en cargar/descargar (según entreguemos o recojamos producto, respectivamente) el vehículo en el depósito. Asimismo, Chao permite que más de una sub-ruta comience en el mismo aparcamiento (habiendo como máximo tantas como número de clientes en el problema). En la Tabla 1.2, de forma análoga a lo que hemos hecho para el modelo de Gerdessen, presentamos las constantes, conjuntos, índices y variables involucradas en dicha formulación. Nótese que estamos considerando una flota de vehículos con m_k camiones y m_l tráileres ($m_l \leq m_k$), en donde cada tráiler está asociado únicamente a un camión. De tal modo, tendremos un total de m_l vehículos completos, y asumiremos que los tráileres $1, \dots, m_l$ van acompañados de los camiones $1, \dots, m_l$, respectivamente; mientras que los restantes camiones, $m_l + 1, \dots, m_k$, viajan por sí solos (es decir, son los únicos que pueden realizar rutas de camión puras).

Constantes y conjuntos	
n	Número de clientes
n_1	Número de clientes v.c.
$N = N_1 \cup N_2$	Conjunto de índices de los clientes ($N = \{1, \dots, n\}$)
N_1	Conjunto de índices de los clientes v.c. ($N_1 = \{1, \dots, n_1\}$)
N_2	Conjunto de índices de los clientes t.c. ($N_2 = \{n_1 + 1, \dots, n\}$)
m_l	Número de tráileres
m_k	Número de camiones
$K = K_1 \cup K_2$	Conjunto de los vehículos
K_1	Conjunto de los vehículos completos (camión más tráiler, $K_1 = \{1, \dots, m_l\}$)
K_2	Conjunto de los camiones sin tráiler acoplado ($K_2 = \{m_l + 1, \dots, m_k\}$)
c_{ij}	Distancia que un camión/vehículo necesita para viajar del nodo i al j
$d_i \geq 0$	Demanda del cliente $i \in N$
$Q_{truck} \geq 0$	Capacidad del camión

$Q_{trailer} \geq 0$	Capacidad del tráiler
$D \geq 0$	Tiempo máximo permitido de uso de un camión/vehículo
$vm \geq 0$	Velocidad media de los camiones/vehículos
$s_i \geq 0$	Tiempo de servicio requerido para el cliente o depósito, $i \in N \cup \{0\}$
Índices	
i, j	Índice para un vértice/cliente, $0 \leq i \leq n, 0 \leq j \leq n$
k	Índice para el vehículo/camión, $k = 1, \dots, m_l, \dots, m_k$
l	Índice para el aparcamiento, $l = 0, \dots, n_1$
v	Índice para la v -ésima sub-ruta, $v = 1, \dots, n$
Variables	
$x_{ij}^k = \begin{cases} 1 & \text{si el vehículo } k \in K_1 \text{ pasa por el arco } (i, j) \text{ con } i, j \in N_1 \cup \{0\} \\ 0 & \text{en otro caso} \end{cases}$ $y_{ij}^{klv} = \begin{cases} 1 & \text{si el camión } k \in K \text{ pasa por el arco } (i, j) \text{ con } i, j \in N \cup \{0\} \\ & \text{en la } v\text{-ésima } (v = 1, \dots, n) \text{ ruta/sub-ruta con aparcamiento} \\ & l \in N_1 \cup \{0\} \\ 0 & \text{en otro caso} \end{cases}$	

Tabla 1.2: Constantes, conjuntos, índices y variables para la formulación del TTRP de Chao.

Tras presentar la notación anterior, ya estamos en condiciones de formular el modelo:

$$\text{minimizar} \quad \left\{ \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} \sum_{k=1}^{m_l} c_{ij} x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n c_{ij} y_{ij}^{klv} \right\} \quad (1.20)$$

sujeto a

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} x_{ij}^k + \sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{\substack{l=0 \\ l \neq j}}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = 1, \dots, n_1; \quad (1.21)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = n_1 + 1, \dots, n; \quad (1.22)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=1}^{n_1} \sum_{v=1}^n y_{i0}^{klv} = 0, \quad (1.23)$$

$$\sum_{j=0}^n y_{lj}^{klv} \leq 1, \quad \begin{array}{l} k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \\ v = 1, \dots, n; \end{array} \quad (1.24)$$

$$y_{lj}^{klv} \leq \sum_{i=0}^{n_1} x_{il}^k, \quad \begin{array}{l} j = 1, \dots, n; \quad k = 1, \dots, m_l; \\ l = 1, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.25)$$

$$y_{ij}^{klv} \leq \sum_{p=0}^n y_{lp}^{klv}, \quad \begin{array}{l} i, j = 0, \dots, n; \quad k = 1, \dots, m_l; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.26)$$

$$y_{ij}^{klv} = 0, \quad \begin{array}{l} i, j = 0, \dots, n; \quad k = m_l + 1, \dots, m_k; \\ l = 1, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.27)$$

$$\sum_{v=1}^n \sum_{j=1}^n y_{0j}^{k0v} \leq 1, \quad k = m_l + 1, \dots, m_k; \quad (1.28)$$

$$\sum_{j=1}^{n_1} x_{0j}^k \leq 1, \quad k = 1, \dots, m_l; \quad (1.29)$$

$$\sum_{i=0}^n \sum_{\substack{j=1 \\ j \neq l}}^n d_j y_{ij}^{klv} \leq Q_{truck}, \quad \begin{array}{l} k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \\ v = 1, \dots, n; \end{array} \quad (1.30)$$

$$\sum_{i=0}^{n_1} \sum_{j=1}^{n_1} d_j x_{ij}^k + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{i=0}^n \sum_{\substack{j=1 \\ j \neq l}}^n d_j y_{ij}^{klv} \leq Q_{truck} + Q_{trailer}, \quad k = 1, \dots, m_l; \quad (1.31)$$

$$\begin{aligned} \sum_{i=0}^{n_1} \sum_{j=1}^{n_1} s_j x_{ij}^k + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{i=0}^n \sum_{\substack{j=1 \\ j \neq l}}^n s_j y_{ij}^{klv} + \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} (c_{ij}/vm) x_{ij}^k + \\ + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (c_{ij}/vm) y_{ij}^{klv} \leq D - s_0, \end{aligned} \quad k = 1, \dots, m_l; \quad (1.32)$$

$$\sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (s_j + c_{ij}/vm) y_{ij}^{k0v} \leq D, \quad k = m_l + 1, \dots, m_k; \quad (1.33)$$

$$\sum_{i=0}^n y_{ij}^{klv} = \sum_{p=0}^n y_{jp}^{klv}, \quad \begin{array}{l} j = 0, \dots, n, \quad k = 1, \dots, m_k; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.34)$$

$$\sum_{i=0}^{n_1} x_{ij}^k = \sum_{p=0}^{n_1} x_{jp}^k, \quad j = 0, \dots, n_1; \quad k = 1, \dots, m_l; \quad (1.35)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ij}^k \leq |B| - 1, \quad k = 1, \dots, m_l; \quad \forall B \subseteq N_1 : |B| \geq 2; \quad (1.36)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{klv} - \sum_{i \in B \cap N_1} \sum_{j \in N_1 \setminus B} x_{ij}^k \leq |B| - 1, \quad k = 1, \dots, m_l; \quad l = 0, \dots, n_1; \\ v = 1, \dots, n; \quad \forall B \subseteq N : |B| \geq 2; \quad (1.37)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{k0v} \leq |B| - 1, \quad k = m_l + 1, \dots, m_k; \\ v = 1, \dots, n, \quad \forall B \subseteq N : |B| \geq 2; \quad (1.38)$$

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} x_{ii}^k + \sum_{j=0}^n \sum_{r=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{jj}^{rlv} = 0, \quad (1.39)$$

$$x_{ij}^k \in \{0, 1\}, \quad i, j = 0, \dots, n_1; \quad k = 1, \dots, m_l; \quad (1.40)$$

$$y_{ij}^{klv} \in \{0, 1\}, \quad i, j = 0, \dots, n, \quad k = 1, \dots, m_k; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \quad (1.41)$$

A continuación, vamos a explicar a qué hace referencia cada una de las restricciones anteriores, separándolas en bloques, con el fin de entender por completo el modelo:

- Función objetivo (1.20):

Minimiza el coste total del recorrido. El primer sumando hace referencia a las rutas con el vehículo completo (rutas de vehículo puras y rutas principales de una completa), mientras que el segundo sumando incluye las rutas puras de camión y las sub-rutas.

- Restricciones específicas de los clientes (1.21)-(1.26):

(1.21), cada cliente v.c. tiene que estar presente exactamente una vez o bien en una ruta principal, o bien en una sub-ruta de la que no es aparcamiento.

(1.22), los clientes t.c. deben ser atendidos una única vez o bien en una sub-ruta o en una ruta de camión pura (cuando $l = 0$).

(1.23), el depósito no puede estar presente en ninguna sub-ruta que comience en un cliente v.c.

(1.24), para cada sub-ruta o ruta de camión pura, el camión en cuestión pasa del aparcamiento o depósito (según corresponda) a un cliente no más de una vez.

(1.25), si no se llega a un cliente v.c. con el vehículo completo, entonces no puede ser candidato a aparcamiento para iniciar una sub-ruta.

(1.26), solo si un candidato a ser aparcamiento se utiliza realmente como aparcamiento, pueden otros clientes pertenecer a una sub-ruta que parte de él.

- Restricciones específicas de los vehículos (1.27)-(1.33):

(1.27), para los camiones que no tienen tráiler asociado no existe ruta principal.

(1.28), para los camiones sin tráiler puede haber como mucho una única ruta, es decir, una ruta de camión pura (no hay múltiple uso del vehículo).

(1.29), para los vehículos completos, el número de sub-rutas no está limitado, pero

sí el de rutas principales, que no puede ser más que uno.

(1.30), en una ruta/sub-ruta sin tráiler, las demandas de los clientes no pueden sobrepasar Q_{truck} .

(1.31), en una ruta completa, o bien en una ruta de vehículo pura, las demandas no pueden exceder $Q_{truck} + Q_{trailer}$.

(1.32), para los vehículos completos no se puede superar su tiempo máximo de utilización, incluido el tiempo de carga o descarga de un vehículo en el depósito, s_0 .

(1.33), para los camiones solos, tampoco se puede superar el tiempo máximo de uso.

- Restricciones específicas de formulación (1.34)-(1.41):

(1.34), conservación del flujo en las sub-rutas y rutas de camión puras.

(1.35), conservación del flujo en rutas de vehículo puras y rutas principales.

(1.36), eliminación de ciclos en las rutas principales de una ruta completa y en las rutas de vehículo puras.

(1.37), eliminación de ciclos en las sub-rutas.

(1.38), eliminación de ciclos en las rutas de camión puras.

(1.39), de un vértice no se puede pasar al mismo.

(1.40), la variable x_{ij}^k es binaria.

(1.41), la variable y_{ij}^{klv} es binaria.

El modelo que acabamos de ver, si bien generaliza la situación expuesta por [Gerdessen \(1993\)](#), contempla la utilización de una flota homogénea. Es cierto que se dispone de dos tipos de automóviles: camiones y tráileres; no obstante, dentro de cada grupo, sus capacidades son idénticas. Además, también se asume que cualquier cliente de tipo v.c. puede servir como aparcamiento del tráiler, algo que no tiene por qué ser cierto en todas las situaciones reales. A continuación, proponemos un modelo que trata de paliar la primera de las carencias que acabamos de mencionar.

1.2. Formulación del TTRP con flota heterogénea

El modelo de [Scheuerer \(2004\)](#) formulado en la Subsección 1.1.2 presenta la particularidad de que todos los camiones y todos los tráileres tienen la misma capacidad: Q_{truck} y $Q_{trailer}$, respectivamente. No obstante, es muy común que las empresas dispongan o se planteen utilizar una flota totalmente heterogénea, pudiendo escoger así los vehículos que mejor satisfagan sus demandas. De hecho, trabajos como el de [Caramia y Guerriero \(2010b\)](#) consideran camiones y tráileres con capacidades diferentes, pero sin proporcionar en ningún momento la formulación del modelo. Si bien es cierto que los costes de alquiler o gasto en combustible derivados de los vehículos con mayor capacidad pueden ser más

elevados, no tendremos esta consideración en cuenta a la hora de formular el problema, pues dependerá de factores externos al modelo.

Para poder considerar vehículos con distintas capacidades es necesario modificar las variables x 's anteriores, ya que ahora, en las rutas realizadas por el camión y el tráiler, no bastará con considerar un vehículo completo k (con $k \in \{1, \dots, m_l\}$), sino que será necesario distinguir sus dos componentes: el camión, que denotaremos por k ; y el tráiler, r . Ambos índices recorrerán el conjunto $K_1 = \{1, \dots, m_l\}$; mientras que los camiones que realicen las rutas de camión puras seguirán perteneciendo a $K_2 = \{m_l + 1, \dots, m_k\}$.

Por lo tanto, modificamos las variables x 's tal y como sigue:

$$x_{ij}^{kr} = \begin{cases} 1 & \text{si el camión } k \in K_1 \text{ con el tráiler acoplado } r \in K_1 \\ & \text{pasa por el arco } (i, j) \text{ con } i, j \in N_1 \cup \{0\} \\ 0 & \text{en otro caso} \end{cases}$$

Además, nótese que ahora Q_{truck} y $Q_{trailer}$ pasan a ser vectores, cuyas componentes dependerán del camión $k \in K$ o del tráiler $r \in K_1$ utilizado, respectivamente.

Ahora ya estamos en condiciones de formular el presente modelo:

$$\text{minimizar} \quad \left\{ \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} c_{ij} x_{ij}^{kr} + \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n c_{ij} y_{ij}^{klv} \right\} \quad (1.42)$$

sujeto a

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ij}^{kr} + \sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{\substack{l=0 \\ l \neq j}}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = 1, \dots, n_1; \quad (1.43)$$

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ij}^{kr} + \sum_{i=0}^n \sum_{k=1}^{m_k} y_{ij}^{k j v} \leq 2, \quad j = 1, \dots, n_1; \quad v = 1, \dots, n; \quad (1.44)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = n_1 + 1, \dots, n; \quad (1.45)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=1}^{n_1} \sum_{v=1}^n y_{i0}^{klv} = 0, \quad (1.46)$$

$$\sum_{j=0}^n y_{ij}^{klv} \leq 1, \quad k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \quad v = 1, \dots, n; \quad (1.47)$$

$$y_{lj}^{klv} \leq \sum_{i=0}^{n_1} \sum_{r=1}^{m_l} x_{il}^{kr}, \quad \begin{array}{l} j = 1, \dots, n; \quad k = 1, \dots, m_l; \\ l = 1, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.48)$$

$$y_{ij}^{klv} \leq \sum_{p=0}^n y_{lp}^{klv}, \quad \begin{array}{l} i, j = 0, \dots, n; \quad k = 1, \dots, m_k; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.49)$$

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^n y_{ij}^{klv} = 0, \quad \begin{array}{l} k = m_l + 1, \dots, m_k; \\ l = 1, \dots, n_1; \end{array} \quad (1.50)$$

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^n y_{ij}^{k0v} = 0, \quad k = 1, \dots, m_l; \quad (1.51)$$

$$\sum_{v=1}^n \sum_{j=1}^n y_{0j}^{k0v} \leq 1, \quad k = m_l + 1, \dots, m_k; \quad (1.52)$$

$$\sum_{j=1}^{n_1} \sum_{r=1}^{m_l} x_{0j}^{kr} \leq 1, \quad k = 1, \dots, m_l; \quad (1.53)$$

$$\sum_{j=1}^{n_1} \sum_{k=1}^{m_l} x_{0j}^{kr} \leq 1, \quad r = 1, \dots, m_l; \quad (1.54)$$

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq l}}^n d_j y_{ij}^{klv} \leq Q_{truckk}, \quad \begin{array}{l} k = 1, \dots, m_l; \quad l = 1, \dots, n_1; \\ v = 1, \dots, n; \end{array} \quad (1.55)$$

$$\sum_{i=0}^n \sum_{j=1}^n d_j y_{ij}^{k0v} \leq Q_{truckk}, \quad \begin{array}{l} k = m_l + 1, \dots, m_k; \\ v = 1, \dots, n; \end{array} \quad (1.56)$$

$$\sum_{i=0}^{n_1} \sum_{j=1}^{n_1} d_j x_{ij}^{kr} + \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq l}}^{n_1} \sum_{l=1}^n \sum_{v=1}^n d_j y_{ij}^{klv} \leq Q_{truckk} + Q_{trailer_r}, \quad \begin{array}{l} k = 1, \dots, m_l; \\ r = 1, \dots, m_l; \end{array} \quad (1.57)$$

$$\begin{aligned} & \sum_{i=1}^{n_1} \sum_{j=0}^{n_1} s_i x_{ij}^{kr} + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{\substack{i=1 \\ i \neq l}}^n \sum_{j=0}^n s_i y_{ij}^{klv} + \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} (c_{ij}/vm) x_{ij}^{kr} + \\ & + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (c_{ij}/vm) y_{ij}^{klv} \leq D - s_0, \end{aligned} \quad \begin{array}{l} k = 1, \dots, m_l; \\ r = 1, \dots, m_l; \end{array} \quad (1.58)$$

$$\sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (s_i + c_{ij}/vm) y_{ij}^{k0v} \leq D, \quad k = m_l + 1, \dots, m_k; \quad (1.59)$$

$$\sum_{i=0}^n y_{ij}^{klv} = \sum_{p=0}^n y_{jp}^{klv}, \quad \begin{array}{l} j = 0, \dots, n, \quad k = 1, \dots, m_k; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.60)$$

$$\sum_{i=0}^{n_1} x_{ij}^{kr} = \sum_{p=0}^{n_1} x_{jp}^{kr}, \quad \begin{array}{l} j = 0, \dots, n_1; \quad k = 1, \dots, m_l; \\ r = 1, \dots, m_l; \end{array} \quad (1.61)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ij}^{kr} \leq |B| - 1, \quad \begin{array}{l} k = 1, \dots, m_l; \quad r = 1, \dots, m_l; \\ \forall B \subseteq N_1 : |B| \geq 2; \end{array} \quad (1.62)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{klv} - \sum_{i \in B \cap N_1} \sum_{j \in N_1 \setminus B} \sum_{r=1}^{m_l} x_{ij}^{kr} \leq |B| - 1, \quad \begin{array}{l} k = 1, \dots, m_l; \quad l = 1, \dots, n_1; \\ v = 1, \dots, n; \quad \forall B \subseteq N : |B| \geq 2; \end{array} \quad (1.63)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{k0v} \leq |B| - 1, \quad \begin{array}{l} k = m_l + 1, \dots, m_k; \\ v = 1, \dots, n; \quad \forall B \subseteq N : |B| \geq 2; \end{array} \quad (1.64)$$

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ii}^{kr} + \sum_{j=0}^n \sum_{t=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{jj}^{tlv} = 0, \quad (1.65)$$

$$x_{ij}^{kr} \in \{0, 1\}, \quad i, j = 0, \dots, n_1; \quad k, r = 1, \dots, m_l; \quad (1.66)$$

$$y_{ij}^{klv} \in \{0, 1\}, \quad \begin{array}{l} i, j = 0, \dots, n; \quad k = 1, \dots, m_k; \\ l = 0, \dots, n_1; \quad v = 1, \dots, n; \end{array} \quad (1.67)$$

La mayor parte de las restricciones expuestas en este modelo son análogas a las del visto en la Subsección 1.1.2, por lo que no requieren ser explicadas. No obstante, ha sido necesaria la inclusión de cuatro restricciones adicionales: (1.44), (1.51), (1.53) y (1.54). La primera indica que, si un cliente v.c. estuviese ejerciendo como aparcamiento de un tráiler para iniciar una sub-ruta, entonces podrían llegar a él dos arcos: uno de la ruta principal y otro de dicha sub-ruta. La restricción (1.51) señala que los camiones que pueden llevar tráiler acoplado, no pueden realizar PTRs. Las restricciones (1.53) y (1.54), por su parte, establecen que, para cada camión o tráiler de $\{1, \dots, m_l\}$, respectivamente, puede haber como mucho una ruta principal.

Hasta ahora hemos tratado de proporcionar una visión general de los problemas de rutas con tráileres, mostrando su evolución y formulando algunos de los modelos representativos en la literatura. Además, hemos modelizado también una extensión natural del TTRP, en la cual se tienen en cuenta vehículos con distintas capacidades. Como ya se ha mencionado, la motivación principal de dicha variante es adaptarse a circunstancias concretas que pueda tener una determinada empresa.

A continuación, nos centraremos en cómo resolver estos problemas, comenzando por revisar los métodos utilizados desde la aparición del TTRP.

Capítulo 2

Resolución del TTRP

2.1. Métodos de resolución

El VRP es uno de los problemas de optimización más populares para el modelado de rutas, y para el que se han desarrollado una gran cantidad de métodos de resolución. En el Apéndice A explicamos algunos de los más comunes que, como veremos, han sido también adaptados y utilizados para obtener soluciones de los problemas que nos conciernen. En esta sección comenzaremos realizando una revisión de los algoritmos de resolución para el TTRP y algunas de sus variantes. Presentaremos de un modo más exhaustivo la heurística propuesta por [Chao \(2002\)](#), al tratarse su modelo del TTRP clásico. Además, también haremos mención de las aproximaciones exactas de este problema vistas hasta el momento.

Los modelos de rutas explicados en el Capítulo 1 presentan la particularidad de que, con un número pequeño de vehículos y destinos de reparto, implican problemas con un número muy elevado de variables y restricciones. En situaciones reales con decenas o cientos de vehículos y clientes, los problemas de optimización resultantes son extremadamente grandes y complejos de resolver. En estos casos tendremos que conformarnos con una buena solución, aunque no sea el óptimo global, debido al gran tiempo computacional que involucraría resolverlo de forma exacta. Esto motiva el uso de heurísticas y metaheurísticas como las que vamos a revisar a continuación.

[Gerdessen \(1996\)](#) propone heurísticas constructivas y de mejora para la resolución del VRPT expuesto en la Sección 1.1. El primer grupo trata de buscar una *buena* solución factible inicial, mientras que el segundo intenta mejorarla.

[Semet \(1995\)](#), en cambio, describe un método heurístico de dos fases para obtener so-

luciones del PACVRP, extendiendo el algoritmo de resolución de Fisher y Jaikumar (1981) para el VRP. Con el fin de simplificar el problema, asume que no existe transferencia de carga entre el camión y el tráiler, de modo que los tráileres estarán cargados solamente con bienes para los clientes de vehículo. La primera fase del algoritmo consiste en resolver el problema combinatorio que asigna tráileres a camiones y determina qué clientes están asignados a cada camión o vehículo. En la segunda fase, se construyen las rutas resolviendo o bien los TSPs o los ACTSPs (*Travelling Salesman Problem under Accessibility Constraints*) dependiendo del vehículo utilizado.

El algoritmo que propone Chao (2002), tal y como veremos en la Subsección 2.1.1, consiste en un método de construcción para la solución inicial y una heurística de mejora basada en la búsqueda tabú (Glover 1986) y en el concepto de desviación del recocido determinístico (brevemente introducido en el Apéndice A).

Siguiendo el mismo enfoque de Chao, Scheuerer (2006) presenta un método de resolución para el TTRP en dos fases. En la primera fase utiliza dos heurísticas de construcción para obtener una solución inicial, llamadas *T-Cluster* y *T-Sweep*. La primera de ellas consiste en un procedimiento de inserción secuencial basado en agrupamiento, en el cual se construyen las rutas de forma secuencial hasta completar la utilización del vehículo. Así, se prioriza la construcción de rutas de vehículo completo en lugar de las rutas de camión puras. La segunda heurística de construcción es una adaptación del algoritmo propuesto por Gillett y Miller (1974) para el VRP. Este método construye las rutas rotando un eje centrado en el depósito e insertando a los clientes de forma gradual según los encuentra. En ambas heurísticas, tras la introducción de un cliente en una de las rutas, se lleva a cabo un procedimiento de post-optimización mediante las heurísticas de mejora 2-opt (Lin 1965) y Or-opt (Or 1976). La segunda fase del algoritmo de Scheuerer consiste en mejorar la solución obtenida en la primera mediante una heurística basada en la búsqueda tabú. Se mejoraron todas las soluciones conocidas hasta el momento para las 21 instancias del TTRP de Chao (2002).

Posteriormente, Lin et al. (2009), proponen por primera vez un algoritmo basado en la metaheurística del recocido simulado para conseguir soluciones aproximadas del TTRP, partiendo de una solución generada de forma aleatoria. Dos años más tarde, lo extienden al TTRPTW, tal y como se puede ver en Lin et al. (2011). Los resultados obtenidos son competitivos con los de Scheuerer (2006), llegando a mejorar la solución obtenida en 11 de las 21 instancias de Chao (2002).

La propuesta de [Caramia y Guerriero \(2010a\)](#) utiliza programación matemática y búsqueda local para resolver el TTRP. Se resuelven dos sub-problemas de forma secuencial: el problema de asignación cliente-ruta (CAP), cuyo objetivo es minimizar la flota de vehículos requerida para servir a los clientes; y el problema de definición de rutas (RDP), que minimiza la distancia total de la ruta de cada vehículo una vez le son asignados los clientes. Dado que el primero de los sub-problemas puede dar lugar a soluciones no factibles, se introduce la búsqueda local para corregir la no factibilidad. Las formulaciones matemáticas y la búsqueda local trabajan de forma iterativa, integrando un mecanismo de múltiple reinicio capaz de diversificar las soluciones al (i) identificar restricciones adicionales a tener en cuenta en la formulación del CAP y (ii) utilizar una matriz tabú para evitar asignaciones de clientes a rutas ya realizadas en las iteraciones anteriores del algoritmo.

[Derigs et al. \(2013\)](#) analizan diferentes variantes del TTRP, tales como la posibilidad de las ventanas de tiempo para la entrega, o la opción de transferir o no carga entre el camión y el tráiler. Proponen heurísticas de dos fases para la resolución de los problemas, comenzando por construir una solución inicial para después pasar a una fase de mejora en la que se combinan técnicas de búsqueda en la vecindad (*local search* (LS) y *large neighborhood search* (LNS)) con dos metaheurísticas (*attribute based hill climber* (ABHC) y *record-to-record travel* (RRT)), evitando así terminar en un óptimo local. El objetivo de la heurística de construcción es generar una solución inicial factible de calidad aceptable en poco tiempo. El hecho de que la solución inicial sea factible es crucial, ya que los métodos de mejora utilizados no aceptan no factibilidades. Dependiendo de la variante del TTRP que se esté considerando, se aplicará una determinada heurística de construcción: para el TTRP clásico (con transferencia de carga), se adaptan las heurísticas de ahorros de [Clarke y Wright \(1964\)](#) y el método de barrido de [Gillett y Miller \(1974\)](#); para el TTRP sin transferencia de carga, la heurística de asignación generalizada de [Fisher y Jaikumar \(1981\)](#); y una heurística de inserción para el TTRPTW. En cuanto a los movimientos que pueden tener lugar en la fase de mejora, ilustramos en las Figuras 2.1 y 2.2 algunos ejemplos de operaciones que se pueden llevar a cabo al aplicar la búsqueda en la vecindad.

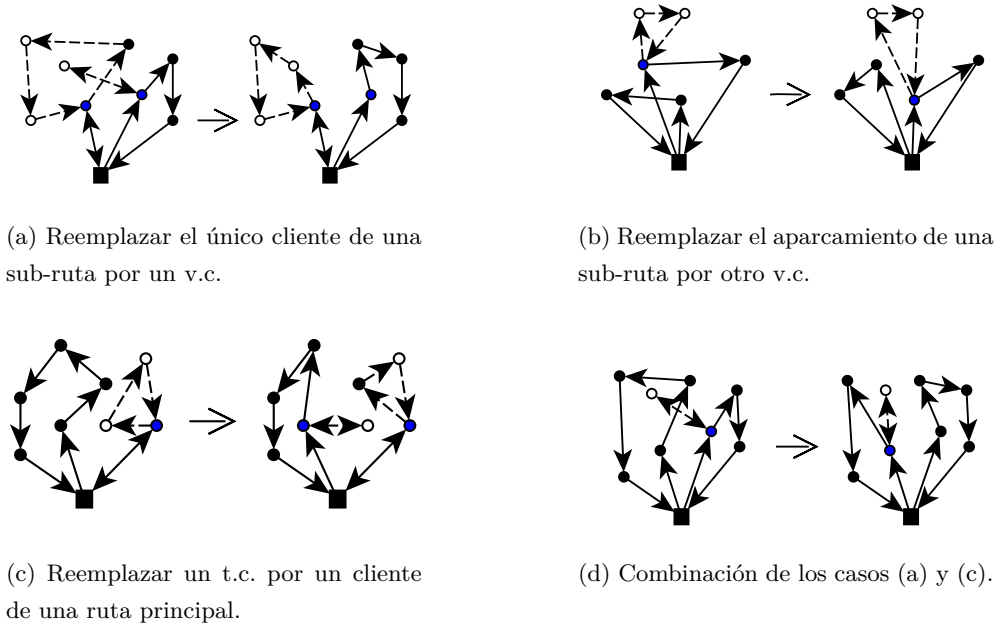


Figura 2.1: Movimientos especiales de intercambio en el TTRP (LS).

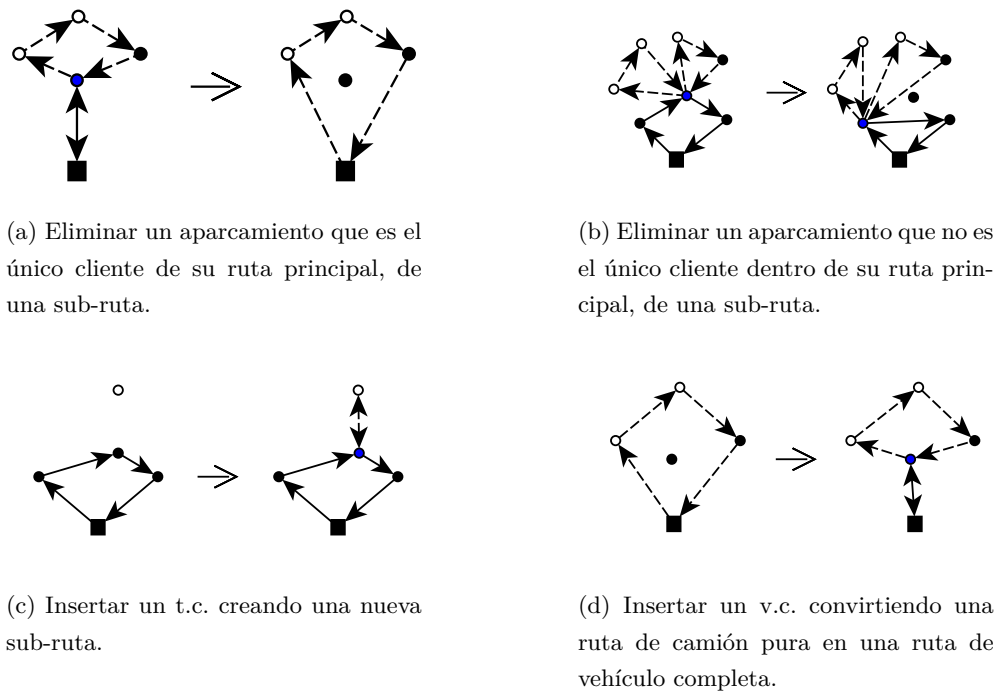


Figura 2.2: Operaciones especiales de eliminación e inserción en el TTRP (LNS).

2.1.1. Método de resolución de Chao

Como ya hemos mencionado, [Chao \(2002\)](#) propone un algoritmo de resolución del TTRP en dos fases: en la primera, utiliza un método de construcción para obtener una solución inicial; y en la segunda mejora dicha solución aplicando una modificación de la metaheurística de búsqueda tabú introducida por [Glover \(1986\)](#). Vamos a describir con detalle cada una de las etapas del algoritmo.

Construcción de la solución inicial

La construcción de la solución inicial para el TTRP consiste en (i) una asignación generalizada relajada, (ii) construcción de las rutas, y (iii) descenso de mejora.

(i) El problema de asignación generalizado relajado (RGAP). En este paso, se trata de asignar los clientes a las rutas resolviendo un problema de asignación generalizado relajado. Antes de nada, se escoge un punto semilla para cada una de las m_l rutas de vehículo puras o completas, y para las $m_k - m_l$ rutas de camión puras. Inicialmente, se selecciona la primera semilla; las restantes se van escogiendo como clientes que presentan mayores distancias entre las semillas ya elegidas y el depósito. Variando la elección de la primera semilla se genera un conjunto de semillas distinto, dando así lugar a una solución inicial diferente. De este modo, para elegir la solución inicial, se realiza el proceso anterior 10 veces y se escoge el mejor resultado. La semilla seleccionada para la primera solución inicial será aquel cliente con mayor distancia al depósito; para la segunda solución inicial, se elige como primera semilla el segundo cliente más alejado del depósito, y así sucesivamente. El coste de asignar el cliente i a la ruta de vehículo o camión j es $d_{ij} = c_{0i} + c_{is_j} - c_{0s_j}$, donde s_j es la semilla de la ruta j . Este problema de asignación se formula del siguiente modo:

$$\text{minimizar} \quad \sum_{i=1}^n \sum_{j=1}^{m_k} d_{ij} x_{ij} \quad (2.1)$$

$$\text{sujeto a} \quad \sum_{j=1}^{m_k} x_{ij} = 1, \quad i = 1, 2, \dots, n; \quad (2.2)$$

$$\sum_{i=1}^n q_i x_{ij} \leq Q_{trailer} + Q_{truck}, \quad j = 1, 2, \dots, m_l; \quad (2.3)$$

$$\sum_{i=1}^n q_i x_{ij} \leq Q_{truck}, \quad j = m_l + 1, \dots, m_k; \quad (2.4)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m_k. \quad (2.5)$$

La función objetivo (2.1) minimiza el coste total de asignación. Las restricciones (2.2) y (2.5) aseguran que se seleccione exactamente una ruta para cada cliente. La restricción (2.3) impone que las demandas de los clientes en las rutas de vehículo completas o puras, no superen la capacidad total del vehículo, $Q_{trailer} + Q_{truck}$; mientras que la restricción (2.4) fuerza a que las demandas de los clientes en una ruta de camión pura no sobrepase su capacidad, Q_{truck} . Para resolver el problema entero, relajamos la restricción (2.5) con:

$$0 \leq x_{ij}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m_k. \quad (2.6)$$

Las restricciones (2.2) y (2.6) imponen que los x_{ij} 's tengan valores en el intervalo $[0, 1]$. Tal y como se puede ver en Lasdon (1970) y Chao et al. (1995), la solución del problema relajado tiene como mucho $2m_k$ valores fraccionarios; por lo tanto habrá como mucho m_k clientes con valores x_{ij} 's no enteros. Sean n_1 ¹ los clientes con valores x_{ij} enteros. En una solución óptima de un problema lineal (LP) acotado y factible hay como mucho $n + m_k$ valores no nulos de x_{ij} 's. Esto implica que:

$$n - n_1 \leq m_k \quad (2.7)$$

y

$$n + m_k - n_1 \leq 2m_k. \quad (2.8)$$

El valor $(n + m_k) - n_1$ de la ecuación (2.8) es una cota del número de variables x_{ij} con valores fraccionarios, y $n - n_1$ en (2.7) es el número de clientes con variable x_{ij} fraccionaria. En la mayoría de los problemas TTRP, m_k es mucho menor que n , de modo que m_k o $2m_k$ no son muy grandes. Después de resolver el LP relajado, se redondea el mayor valor fraccionario de x_{ij} a 1 para el cliente i . La solución del LP relajado asigna exactamente una ruta (de vehículo o de camión) a cada cliente, pero el redondeo de los valores x_{ij} puede hacer que en algunos se exceda la capacidad total de la ruta, dando lugar a una solución no factible. En un primer momento, independientemente de la factibilidad, se construyen tres tipos de rutas, y luego se corrigen las no factibilidades usando una función de penalización en la etapa de mejora.

(ii) Construcción de rutas. Hemos visto que en la fase anterior se asigna cada uno de los clientes a un tipo de ruta. En este paso, se construyen las rutas. Nótese que las rutas de camión puras con clientes v.c. o t.c. y las rutas de vehículo puras con solo clientes v.c., se pueden tratar como problemas del viajante (TSPs) y su construcción se lleva a cabo

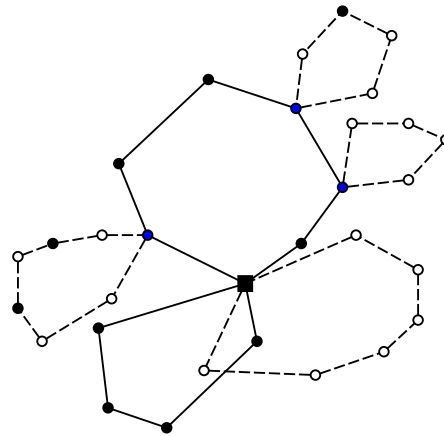
¹No confundir este valor con el número de clientes v.c. visto en la Subsección 1.1.2.

mediante una de las heurísticas de inserción más barata presente en Raff (1983).

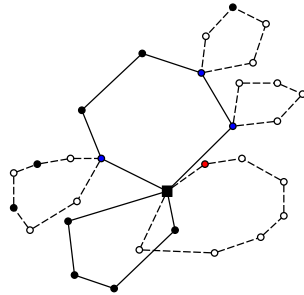
Por otro lado, una ruta de vehículo completa se calcula del siguiente modo: primero, se halla la ruta principal para los clientes de tipo v.c. usando una de las heurísticas de inserción más barata. Luego, se construyen las sub-rutas para los clientes t.c. La construcción de la primera sub-ruta se lleva a cabo conectando el primer cliente t.c. al cliente v.c. más cercano de la ruta principal. El resto de clientes t.c. se insertan o bien en las sub-rutas existentes siempre que no se exceda la capacidad del camión, o bien se conectan a clientes v.c. de la ruta principal, generando una nueva sub-ruta. A pesar de que un cliente v.c. puede ser abastecido en una sub-ruta, por cuestiones de simplicidad impondremos que en este paso solo puedan estar en la ruta principal. Estos clientes podrán ser movidos de la ruta principal a una sub-ruta cuando se consideren movimientos de mejora.

(iii) **Descenso de mejora.** El descenso de mejora presenta cuatro pasos, en los que tratamos de mover a los clientes entre las distintas rutas con el objetivo de convertir una solución no factible en una factible, quizás incrementando la distancia total recorrida o disminuyéndola sin aumentar la penalización. La penalización, θ , de una solución es una medida del grado de no factibilidad y se puede ver como la suma total de los excesos de demanda en todas las rutas. Dada una ruta R , calculamos su penalización como $\theta_R = \max(0, \eta_R - \gamma_R)$, donde η_R es la demanda total en R y γ_R la capacidad de dicha ruta. Nótese que para las rutas de vehículo puras y completas, $\gamma_R = Q_{truck} + Q_{trailer}$; mientras que para las rutas puras de camión y las sub-rutas, $\gamma_R = Q_{truck}$. Describimos a continuación cada uno de los cuatro pasos que tienen lugar:

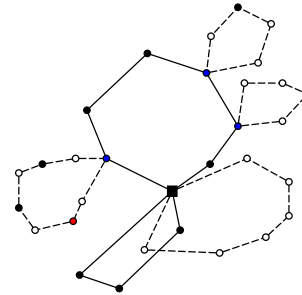
- **Movimiento de descenso de un punto (OPD).** Se examina un único cliente a la vez, moviéndolo de una ruta a otra. Si un movimiento disminuye la penalización (con o sin aumentar la distancia), o disminuye la distancia sin aumentar la penalización, entonces es ejecutado inmediatamente. Nótese que durante esta etapa de descenso se descartan dos tipos de movimientos: (1) mover un cliente t.c. a la ruta principal de una ruta de vehículo completa o a una ruta de vehículo pura y (2) mover un cliente v.c. que está siendo usado como aparcamiento de una sub-ruta. Los primeros clientes estudiados son los servidos en las rutas de camión y vehículo puras y en las rutas principales de una completa, seguidos por los clientes de las sub-rutas. Una vez examinado un candidato, se recalcula tanto la función objetivo como las penalizaciones en ambas rutas. En la Figura 2.3 mostramos un ejemplo de algunos de los movimientos que podrían tener lugar en esta fase. Con el fin de facilitar la comprensión del lector, pintamos de rojo el cliente estudiado en cada caso.



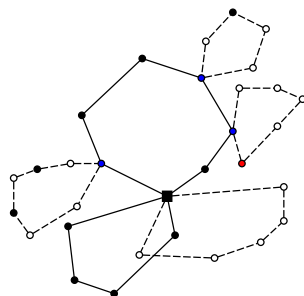
(a) Solución inicial.



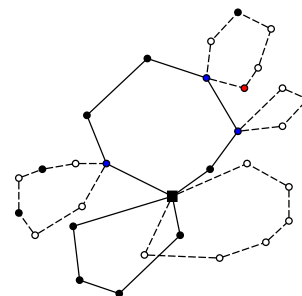
(b) Mover un v.c. de una ruta principal a una ruta de camión pura.



(c) Mover un v.c. de una ruta de vehículo pura a una sub-ruta.



(d) Mover un t.c. de una ruta de camión pura a una sub-ruta.

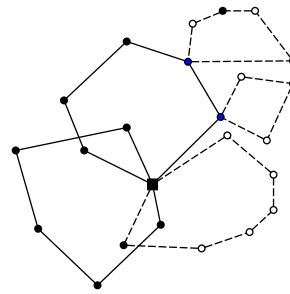


(e) Mover un t.c. de una sub-ruta a otra.

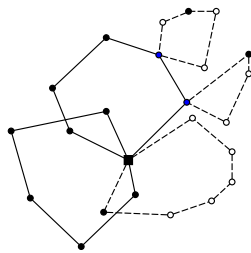
Figura 2.3: Ejemplos de movimientos (OPD).

- **Intercambio de descenso de dos puntos (TPD).** Se propone el intercambio de dos clientes pertenecientes a diferentes rutas. Para ello se tienen en cuenta las penalizaciones de las dos rutas y la distancia del mismo modo que en el OPD. Se prohíbe mover un cliente t.c. a la ruta principal de una ruta completa o a una ruta pura de

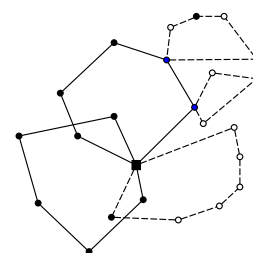
vehículo, o mover el aparcamiento de una sub-ruta, y es imprescindible mantener la factibilidad de las sub-rutas con respecto a la capacidad del camión. Algunos de los posibles intercambios que pueden tener lugar en esta etapa se muestran en la Figura 2.4.



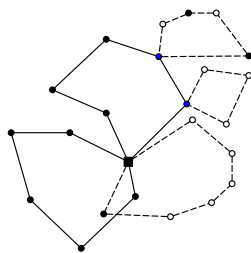
(a) Solución inicial.



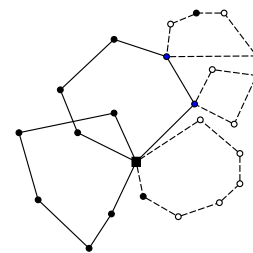
(b) Intercambiar un v.c. de una sub-ruta con un t.c. de otra.



(c) Intercambiar un t.c. de una sub-ruta con otro de una ruta pura de camión.



(d) Intercambiar un v.c. de una ruta pura de vehículo con otro de una ruta principal.



(e) Intercambiar un v.c. de una ruta pura de vehículo con otro de una ruta pura de camión.

Figura 2.4: Ejemplos de intercambios (TPD).

- Refinamiento del aparcamiento en una sub-ruta.** En los dos pasos anteriores, los aparcamientos no sufren cambios. Sin embargo, hay ocasiones en las que es posible mejorar la solución si se reemplaza dicho *parking*. En este paso, se intentan seleccionar otros aparcamientos o variar la secuencia de clientes de las sub-rutas. Para cada

sub-ruta, el proceso consiste en eliminar el primer y el segundo arco, esto es, los dos arcos que conecten la sub-ruta a la ruta principal, y substituirlos por otros dos que conecten clientes consecutivos en la sub-ruta a un cliente v.c. de la ruta principal. Si existe algún movimiento que disminuya la distancia, se ejecuta; en otro caso, la sub-ruta permanece como está. Después de reemplazar los arcos, se tiene un nuevo aparcamiento, y los clientes de la sub-ruta deben ser ordenados de nuevo para mejorar la solución. Ilustramos en la Figura 2.5 los pasos a seguir en esta fase.

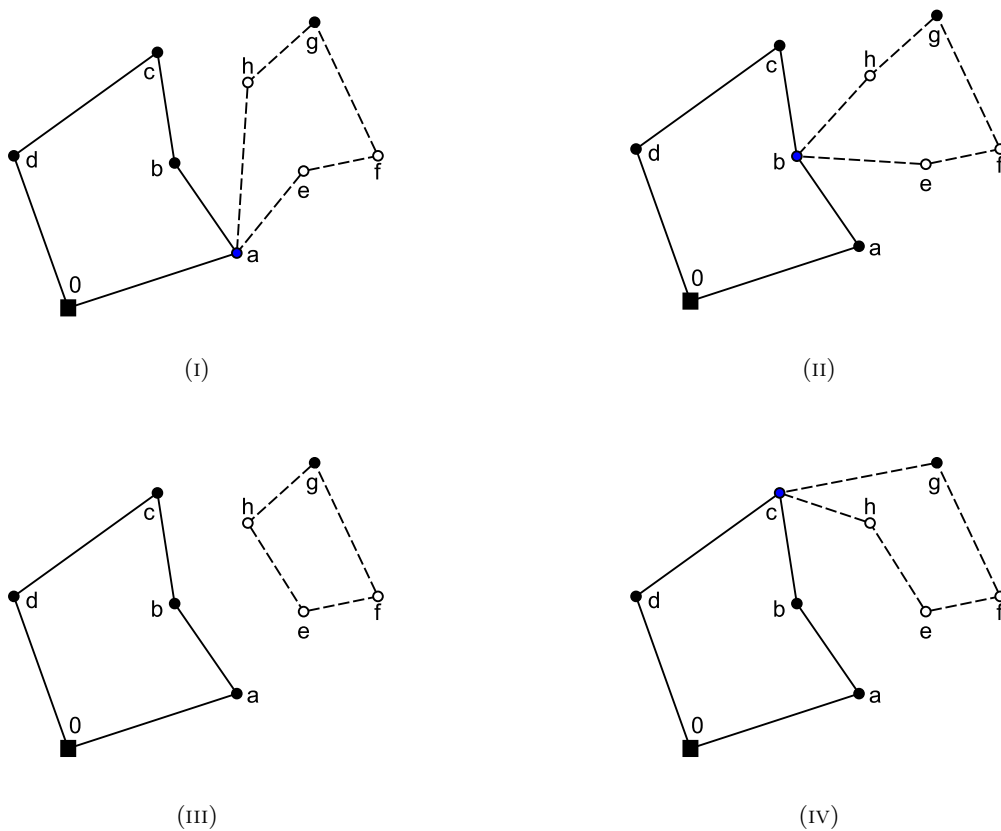


Figura 2.5: Refinamiento del aparcamiento.

Nótese que en la Figura 2.5i tenemos una ruta de vehículo completa, que consiste en una ruta principal $(0 - a - b - c - d - 0)$ y una sub-ruta $(a - e - f - g - h - a)$, donde el cliente a es utilizado como aparcamiento. Al pasar a la Figura 2.5ii, los arcos (a, e) y (h, a) son reemplazados por (b, e) y (h, b) para formar una nueva sub-ruta, de modo que el tráiler queda estacionado en el cliente b . Luego, tal y como vemos en la Figura 2.5iii, se eliminan los arcos (b, e) y (h, b) temporalmente, uniéndose los

nodos e y h con el objetivo de mejorar la sub-ruta. Finalmente, en la Figura 2.5IV se deshace la unión (g, h) y se conectan (c, h) y (g, c) para formar de nuevo otra sub-ruta con el cliente c como aparcamiento. De tal modo, podemos ver que se ha cambiado dos veces el cliente en el cual se desacopla el tráiler, y se han reordenado los clientes de la sub-ruta para perfeccionarla.

- **Limpieza 2-opt.** Una vez aplicado el algoritmo de construcción para obtener la solución inicial, se llevan a cabo los tres métodos anteriores para tratar de mejorarla, consecutivamente. El proceso termina cuando ya no se puede realizar ningún movimiento. Es en ese momento cuando se aplica un procedimiento basado en el 2-opt de Croes (1958) a cada una de las rutas para tratar de depurarlas y reducir la distancia total recorrida.

Búsqueda Tabú

Chao (2002) utiliza un método basado en la metaheurística de búsqueda tabú (TS), introducida por Glover (1986), y la combina con el concepto de desviación del recocido determinístico (DA) para mejorar la solución inicial generada en la etapa de construcción.

Los procedimientos basados en TS exploran el espacio de búsqueda al pasar de una solución a su mejor vecino (aquel que presenta un valor más bajo de la función objetivo entre todos los candidatos examinados), incluso si esto resulta en un aumento de dicho valor, con el fin de aumentar la probabilidad de escapar de un óptimo local. Para evitar un proceso cíclico en el transcurso de la búsqueda, se crea lo que se conoce como *lista tabú*, en la que se introducen algunos de los últimos movimientos realizados y se prohíbe ejecutarlos de nuevo durante un cierto número de iteraciones (*tiempo tabú*).

La heurística de mejora que propone Chao (2002) emplea una restricción tabú basada en frecuencia (FTB) que no permite realizar algunos movimientos hacia atrás. Además, como ya hemos mencionado, usa también el concepto de desviación del DA y desarrolla un nuevo tipo de restricción tabú, basada en la función objetivo (OTB). Dicha restricción se utiliza para implementar estrategias de intensificación y diversificación variando los valores de las desviaciones dentro de dos rangos distintos.

A continuación se detallan los dos tipos de restricciones tabú utilizados por este autor, un criterio de aspiración y tres reglas de parada. Todo esto se integrará posteriormente en los métodos OPD y TPD de búsqueda tabú. Finalmente, se integran todos estos pasos en una nueva heurística de mejora para resolver el TTRP.

- **Restricciones tabú.** Los métodos OPD y TPD, vistos en la fase de construcción de rutas, solo permiten realizar movimientos que disminuyen o bien la distancia o bien la penalización, sin aplicar ningún tipo de restricción tabú. Esto se debe a que el interés de esa etapa se centra en generar una solución inicial, y no en obtener una solución de gran calidad.

La primera restricción tabú es FTB, que prohíbe mover de nuevo un cliente a una ruta (de la cual acaba de ser eliminado) durante π iteraciones, siendo π un número aleatorio entre 5 y 10. La segunda restricción tabú, OTB, se corresponde con la llamada *desviación* utilizada en el recocido determinístico. Nótese que en el recocido determinístico se tienen en cuenta aquellas soluciones cuyo valor no exceda el mejor valor de la función objetivo más una cierta desviación. Así, se tomarán como soluciones prohibidas, o tabú, las que superen dicho umbral. El valor de la desviación se ajusta automáticamente dentro de dos rangos de valores: más pequeños para intensificación y mayores para diversificación. En la búsqueda de intensificación se establece la proporción inicial δ igual al 1%, siendo el de la desviación δ veces el valor de la mejor función objetivo encontrada hasta el momento. Si ninguna solución candidata pasa la restricción tabú OTB durante el transcurso de una iteración, se actualiza el valor de δ aumentándolo un 1%. Se ejecuta una nueva iteración con el δ restablecido, a menos que supere el 10%. En la búsqueda de diversificación, δ comienza fijándose como 10%, e incrementándose en cada iteración un 5%. La diversificación se detiene cuando se obtiene al menos una nueva solución (la búsqueda ha conducido a una nueva parte del espacio de soluciones y se reinicia la intensificación usando como punto de partida para la búsqueda la solución generada en la diversificación). El valor de δ vuelve a ser 1% y se va auto-ajustando de nuevo con un incremento del 1%.

- **Criterio de aspiración.** En la búsqueda tabú, se utiliza un criterio de aspiración (AC) para determinar cuándo pueden ser descartadas las restricciones tabú. En este caso, se utiliza un AC que ignora una restricción tabú una vez se obtiene una solución cuyo valor objetivo es mejor que el de la solución más buena hasta el momento. Se utilizan dos tipos de movimientos para generar la vecindad de una solución: un movimiento de búsqueda tabú en un punto (OPT), que trata de mover un cliente de una ruta a otra factible; y un intercambio de búsqueda tabú en dos puntos (TPT), que intercambia dos clientes entre dos rutas factibles. Ambos métodos, OPT y TPT, involucran dos tipos de restricciones tabú y un criterio de aspiración para convertir los pasos de descenso de la construcción de la solución en pasos de mejora para la TS.

- **Depuración local y criterios de parada.** Para mejorar una solución de forma local, se aplica un procedimiento 2-opt a cada ruta de la solución en la mejora de la búsqueda tabú. Al igual que en los pasos de descenso, no se permiten soluciones que empeoren el valor de la función objetivo (pues las restricciones tabú y el criterio de aspiración no se aplican en esta fase). Se lleva a cabo la depuración solamente a la mejor solución obtenida en la etapa de intensificación, para así mejorar la solución de forma local.

La fase de mejora tiene tres etapas: el descenso, la intensificación y la diversificación. La primera de ellas ya ha sido explicada con anterioridad. En la etapa de intensificación, se busca el mejor vecino usando valores pequeños de la desviación. Por otro lado, la diversificación permite explorar una nueva región de soluciones usando valores grandes de la desviación y aplicándole el procedimiento de búsqueda a una buena solución encontrada en las etapas de intensificación y descenso. Estas tres etapas se llevan a cabo de forma consecutiva y cada una finaliza de acuerdo a una regla de parada local diferente. Además, existe también una regla de parada global, que finaliza todo el proceso de búsqueda. Cuando al realizar 30 iteraciones seguidas de las fases de intensificación, descenso y diversificación, no se obtiene una nueva mejor solución durante 10 iteraciones consecutivas, el procedimiento finaliza y la mejor solución encontrada es la solución final.

2.1.2. Métodos de resolución exactos

[Drexler \(2007\)](#) introduce una variante del TTRP, el TTRP generalizado (GTTRP), en la cual se consideran ubicaciones adicionales como aparcamiento, ventanas de tiempo, así como una flota de vehículos heterogénea. Este autor propone por primera vez un método de resolución exacto para el GTTRP, desarrollando un algoritmo de ramificación y acotación para este problema. No obstante, solo ha sido probado en instancias relativamente pequeñas del TTRP, con no más de 10 clientes de cada tipo.

Posteriormente, trabajos como el de [Parragh y Cordeau \(2017\)](#) o el de [Rothenbächer et al. \(2018\)](#) resuelven el TTRPTW de forma exacta, empleando algoritmos de ramificación y acotación, y de ramificación, acotación y corte, respectivamente. Los resultados obtenidos en ambos casos son competitivos con las metaheurísticas creadas hasta el momento. No obstante, no han sido testados en instancias con más de 100 clientes.

2.1.3. Instancias de referencia para el TTRP

Hasta el trabajo de Chao (2002), no existían en la literatura de la Investigación Operativa *benchmarks* que permitieran testar los problemas del TTRP. Este autor, con el fin de probar su heurística, vista en la Subsección 2.1.1, seleccionó siete VRPs clásicos de las instancias de Christofides et al. (1979) (CMT), obteniendo 21 TTRPs. Estos, fueron generados del siguiente modo: para cada cliente i en un problema de CMT, se calcula la distancia con su vecino más próximo y se denota por A_i . Para cada uno de los problemas CMT, se crean tres TTRPs. En el primero, el 25 % de los clientes con los valores A_i más pequeños son identificados como clientes t.c.; en el segundo problema, el 50 %; y en el tercero, el 75 %. Además, se presentan también el número y capacidades de los camiones y tráileres, así como la relación entre la demanda de los clientes y la capacidad total.

Posteriormente, Lin et al. (2010), con el fin de aportar más información a la literatura del TTRP, generaron 36 nuevas instancias para este problema a partir de los 12 VRPs básicos de Roachat y Taillard (1995) de manera análoga a lo realizado por Chao (2002).

Tanto los *benchmarks* de Chao (2002) como los de Lin et al. (2010) para el TTRP se pueden descargar desde la página web: <http://web.ntust.edu.tw/~vincent/ttrp/>.

2.2. Resolución del TTRP en AMPL

Con el fin de testar los modelos de programación lineal entera binaria formulados en el Capítulo 1, los hemos escrito en el lenguaje *A Modeling Language for Mathematical Programming* (AMPL) (véase <https://ampl.com/>). Los modelos implementados en AMPL se pueden resolver por medio de un solver tanto en el servidor NEOS (véase <https://neos-server.org/neos/>) como haciendo uso de una licencia (académica o comercial). Aquí utilizaremos el solver Gurobi (véase <https://www.gurobi.com/>).

A continuación vamos a presentar un pequeño ejemplo que, salvo ligeras modificaciones en algunos parámetros, sirve para ambas versiones (TTRP de Chao y TTRP con flota heterogénea). Este, nos vale para comprobar que la formulación expuesta es la adecuada. Para facilitar la lectura de este trabajo, no incluimos en esta sección el código fuente, sino que los archivos de AMPL correspondientes se encuentran en el Apéndice B para su consulta.

2.2.1. TTRP de Chao en AMPL

Los archivos `TTRP_Chao.mod`, `TTRP_Chao.dat` y `TTRP_Chao.run`, presentes en el Apéndice B, ilustran la formulación del modelo visto en la Subsección 1.1.2.

Sea $\{0\}$ el vértice que hace referencia al depósito y $N = \{1, \dots, 7\}$ el conjunto de clientes. De ellos, los 4 primeros son clientes v.c., y los restantes son de tipo t.c. De este modo, $N_1 = \{1, 2, 3, 4\}$ y $N_2 = \{5, 6, 7\}$. Disponemos de una flota de $m_k = 3$ camiones y $m_l = 2$ tráileres, por lo que tendremos dos vehículos completos y un camión que viaja solo. Las capacidades (en toneladas) del camión y del tráiler son $Q_{truck} = 15$ y $Q_{trailer} = 30$, respectivamente (por lo tanto, la del vehículo completo será 45). Asumimos que el tiempo que se tarda en realizar un recorrido es proporcional a la distancia. Cada medio de transporte, que realiza una única ruta, no puede ser utilizado durante más de 10 horas, debido a la jornada laboral de los conductores. Se estima, además, que la velocidad media a la que viajan los automóviles es de 80 km/h. Las distancias, en kilómetros, entre los distintos nodos del problema las representamos en la siguiente matriz simétrica:

$$C = \begin{pmatrix} 0 & 130 & 125 & 150 & 190 & 360 & 250 & 245 \\ 130 & 0 & 125 & 280 & 335 & 460 & 320 & 260 \\ 125 & 125 & 0 & 210 & 315 & 440 & 360 & 345 \\ 150 & 280 & 210 & 0 & 165 & 280 & 310 & 370 \\ 190 & 335 & 315 & 165 & 0 & 100 & 205 & 310 \\ 360 & 460 & 440 & 280 & 100 & 0 & 260 & 390 \\ 250 & 320 & 360 & 310 & 205 & 260 & 0 & 145 \\ 245 & 260 & 345 & 370 & 310 & 390 & 145 & 0 \end{pmatrix}$$

El valor de s_0 , esto es, el tiempo de carga de mercancía en el depósito, es media hora; mientras que las demandas (en toneladas de producto) y los tiempos de servicio de cada uno de los clientes (en minutos), d_i y s_i , respectivamente, con $i \in N$, vienen dadas por los siguientes vectores:

$$d = \left(25 \ 20 \ 14 \ 16 \ 14 \ 6 \ 7 \right)^t \quad \text{y} \quad s = \left(10 \ 8 \ 5 \ 6 \ 5 \ 12 \ 9 \right)^t.$$

Hemos resuelto este problema usando una versión gratuita del solver Gurobi (a través del servidor NEOS) que utiliza una variación del algoritmo de ramificación y acotación. El tiempo de ejecución fue de 0.479964 segundos y se obtuvo una solución en la que estaban presentes los tres tipos de rutas posibles:

- Valor de la función objetivo: 1725 kilómetros.
- Variables de decisión para el vehículo 1 (combinación del camión 1 y el tráiler 1): $x_{04}^1 = x_{43}^1 = x_{30}^1 = 1$ e $y_{45}^{147} = y_{54}^{147} = 1$. Se trata de una ruta de vehículo completa con una sub-ruta cuyo aparcamiento es el cliente 4, y una longitud de 705 km.
- Variables de decisión para el vehículo 2 (combinación del camión 2 y el tráiler 2): $x_{01}^2 = x_{12}^2 = x_{20}^2 = 1$. Es una ruta de vehículo pura, cuya longitud es 380 km.
- Variables de decisión para el camión 3: $y_{07}^{307} = y_{76}^{307} = y_{60}^{307} = 1$. Ruta de camión pura con distancia total de 640 km.

Representamos este resultado en la Figura 2.6.

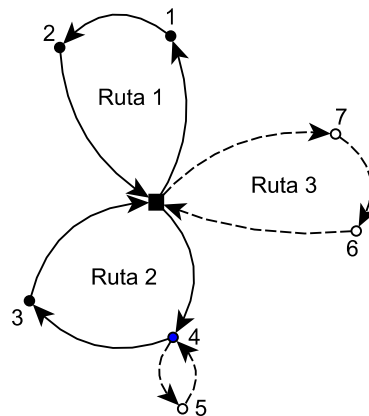


Figura 2.6: Solución obtenida por Gurobi al ejemplo propuesto del TTRP de Chao.

2.2.2. TTRP con flota heterogénea en AMPL

Como ya se ha comentado en la Sección 1.2, hay ocasiones en las que o bien las compañías disponen de camiones y tráileres con distintas capacidades; o bien tienen la posibilidad de alquilar una flota de vehículos con estas características. De ese modo, es necesario reflejar las cantidades máximas que puede cargar cada medio de transporte involucrado: los que antes eran parámetros, Q_{truck} y $Q_{trailer}$, pasan ahora a ser vectores. Así, consideramos Q_{truckk} y $Q_{trailerr}$ las capacidades del camión $k \in K$ y del tráiler $r \in K_1$, respectivamente. Manteniendo el resto de valores invariantes con respecto a los de la Subsección 2.2.1,

tomamos:

$$Q_{truck} = \begin{pmatrix} 18 \\ 15 \\ 15 \end{pmatrix} \quad y \quad Q_{trailer} = \begin{pmatrix} 30 \\ 35 \end{pmatrix}.$$

Los archivos relativos a la formulación de este modelo, `TTRP_het.mod` y `TTRP_het.dat` se encuentran en el Apéndice B. De nuevo, hemos resuelto el problema en NEOS con el solver Gurobi, con un tiempo de computación de 0.541792 segundos, dando lugar a los siguientes resultados:

- Valor de la función objetivo: 1725 kilómetros.
- Ruta 1: es una ruta de vehículo pura, recorrida por el camión 1 y el tráiler 2, y cuya longitud es 380 km. Las variables de decisión son: $x_{02}^{12} = x_{21}^{12} = x_{10}^{12} = 1$.
- Ruta 2: se trata de una ruta de vehículo completa con una longitud total de 705 km. La ruta principal está recorrida por el camión 2 con el tráiler 1 acoplado, el cual queda estacionado en el cliente 4 para iniciar una sub-ruta solo con el camión. Las variables de decisión son: $x_{04}^{21} = x_{43}^{21} = x_{30}^{21} = 1$ e $y_{45}^{247} = y_{54}^{247} = 1$.
- Ruta 3: es una ruta de camión pura, con una distancia total de 640 km. Las variables de decisión para el camión 3 son: $y_{07}^{305} = y_{76}^{305} = y_{60}^{305} = 1$.

Omitimos ilustrar las rutas aquí, por ser el resultado muy semejante al de la Figura 2.6, con la particularidad de que la ruta 1 se recorre en sentido inverso. Esto no tiene mayor trascendencia en la solución, pues la matriz de distancias es simétrica.

2.2.3. Resolución de un ejemplo real en AMPL

En los últimos años, la logística del transporte ha alcanzado un papel fundamental en la industria. Numerosas empresas, tanto públicas como privadas, están interesadas en desarrollar herramientas informáticas para diseñar sus rutas, con objetivos como minimizar los costes y/o maximizar la distribución de sus productos.

En esta subsección nos centraremos en un ejemplo concreto, y del cual tenemos datos: el de una cooperativa agrícola gallega que reparte pienso a diferentes granjas. La compañía dispone de una flota de camiones con diferentes capacidades. En particular, conocemos las distancias entre los distintos vértices (socios y depósito central), así como las demandas

de los clientes y las capacidades de los camiones.

Con el fin de resolver el problema haciendo uso de AMPL en un tiempo considerable, hemos seleccionado los 10 primeros clientes, que podrán ser atendidos por alguno de los tres camiones de la compañía. Dos de ellos tienen capacidad para transportar 15300 kilogramos, mientras que el tercero 15000. Los clientes tienen las siguientes demandas (en kilogramos):

$$d = \left(3300 \ 6041 \ 5959 \ 2951 \ 4885 \ 3003 \ 3016 \ 4478 \ 5413 \ 3490 \right)^t,$$

y la matriz de distancias (en kilómetros) entre los vértices viene dada por:

$$C = \begin{pmatrix} 0 & 21 & 20 & 17 & 65 & 63 & 60 & 19 & 22 & 24 & 60 \\ 21 & 0 & 4 & 6 & 60 & 58 & 55 & 15 & 18 & 20 & 55 \\ 20 & 4 & 0 & 4 & 59 & 56 & 53 & 13 & 8 & 12 & 53 \\ 17 & 6 & 4 & 0 & 57 & 54 & 52 & 11 & 13 & 16 & 52 \\ 65 & 60 & 59 & 57 & 0 & 3 & 7 & 66 & 69 & 71 & 6 \\ 63 & 58 & 56 & 54 & 3 & 0 & 4 & 64 & 66 & 69 & 3 \\ 60 & 55 & 53 & 52 & 7 & 4 & 0 & 61 & 64 & 66 & 2 \\ 19 & 15 & 13 & 11 & 66 & 64 & 61 & 0 & 3 & 5 & 61 \\ 22 & 18 & 8 & 13 & 69 & 66 & 64 & 3 & 0 & 7 & 64 \\ 24 & 20 & 12 & 16 & 71 & 69 & 66 & 5 & 7 & 0 & 66 \\ 60 & 55 & 53 & 52 & 6 & 3 & 2 & 61 & 64 & 66 & 0 \end{pmatrix}. \quad (2.9)$$

Sabemos que la jornada laboral de los conductores es de 8 horas y estimamos la velocidad media de los automóviles en 60 km/h. Además, dado que no disponemos de información acerca del tiempo de servicio de cada socio ni del empleado en cargar los camiones, lo suponemos despreciable. Hemos resuelto este problema usando Gurobi en NEOS, y, tras un tiempo de ejecución de 351.94 segundos, obtuvimos los siguientes resultados:

- Valor de la función objetivo: 232 kilómetros.
- El camión 1 les reparte mercancía a los socios 1, 2 y 3, recorriendo 46 kilómetros. Las variables de decisión son: $y_{03}^{1,0,10} = y_{32}^{1,0,10} = y_{21}^{1,0,10} = y_{10}^{1,0,10} = 1$.

- El camión 2 realiza la ruta 0–6–4–5–10–0, cuya longitud es de 133 kilómetros. Las variables de decisión son: $y_{06}^{206} = y_{64}^{206} = y_{45}^{206} = y_{5,10}^{206} = y_{10,0}^{206} = 1$.
- El camión 3 recorre 53 kilómetros y sirve a los clientes 7, 8 y 9. Las variables de decisión para el camión 3 son: $y_{07}^{304} = y_{78}^{304} = y_{89}^{304} = y_{90}^{304} = 1$.

En la Figura 2.7 ilustramos la solución anterior.

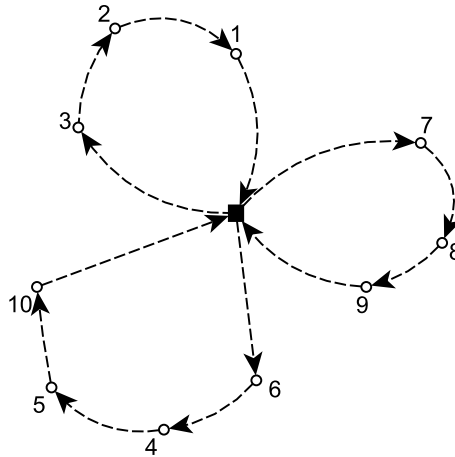


Figura 2.7: Solución obtenida por Gurobi al ejemplo real de los piensos (CVRP).

Tal y como podemos deducir, al utilizar solamente camiones, nos encontramos ante un CVRP. En el Apéndice B presentamos el código de los archivos `CVRP_piensos.mod` y `CVRP_piensos10.dat`. Nótese que el primero solo difiere del `TTRP_het.mod` mencionado con anterioridad en que ahora los conjuntos N_1 y K_1 (aka *Trailer*) son vacíos.

Ahora bien, debido a la gran cantidad de pienso que debe repartir la cooperativa diariamente, proponemos el uso de tráileres, para así dar cabida a la utilización del modelo TTRP con flota heterogénea presentado en este trabajo. Además, dada la existencia de restricciones de accesibilidad a algunas de las granjas, este modelo parece adecuado. Si bien es cierto que la compra o alquiler de estos vehículos adicionales puede suponer una inversión inicial grande, los beneficios suelen compensar a largo plazo (pues, entre otras cosas, no es necesario contratar a más conductores). De tal modo, hemos supuesto que la compañía dispone de dos tráileres, con capacidades de 25300 y 20500 kilogramos.

Los archivos de AMPL que estamos utilizando para resolver este problema, presentes en el Apéndice B, son `TTRP_het.mod`, y `TTRP_piensos10.dat`. Tras un tiempo de ejecución de 10967.5 segundos (3 horas), se obtuvo la siguiente solución:

- Valor de la función objetivo: 207 kilómetros.
- Hay una ruta de vehículo completa de 74 kilómetros, cuya ruta principal, 0–3–2–1–0 está recorrida por el camión 2 con el tráiler 2 acoplado. El cliente 2 sirve como aparcamiento del tráiler para realizar la sub-ruta 2–8–7–9–2 con el camión. Las variables de decisión son: $x_{03}^{22} = x_{32}^{22} = x_{21}^{22} = x_{10}^{22} = 1$ e $y_{28}^{227} = y_{87}^{227} = y_{79}^{227} = y_{92}^{227} = 1$.
- El resto de clientes están servidos por el camión 3 en una ruta de camión pura, 0–10–5–4–6–0. Las variables de decisión son: $y_{0,10}^{308} = y_{10,5}^{308} = y_{54}^{308} = y_{46}^{308} = y_{60}^{308} = 1$.

Tal y como podemos ver, si la compañía introduce tráileres se consigue una reducción no solo de la longitud total recorrida (que disminuye en 25 kilómetros), sino también en el número de conductores necesarios (que pasa de 3 a 2). En la Figura 2.8 ilustramos esta solución.

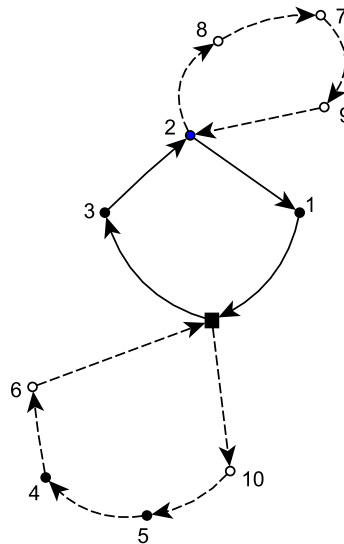


Figura 2.8: Solución obtenida por Gurobi al ejemplo real de los piensos (TTRP).

Capítulo 3

Heurística para el TTRP

En el Capítulo 1 presentamos la formulación del TTRP como un problema de programación lineal y entera binaria. La posterior implementación en AMPL nos permitía obtener la solución de los problemas planteados. El solver empleado, Gurobi, hace uso de algoritmos para la resolución de problemas de programación lineal y lineal entera mixta que, cuando estos son pequeños, nos proporciona la solución óptima en un tiempo razonablemente bajo. No obstante, tal y como ya hemos visto anteriormente, a medida que aumentamos el número de clientes de nuestro problema, el tiempo computacional se hace demasiado grande.

Ante la creciente necesidad de dar respuesta a situaciones reales con la mayor brevedad posible, tales como el reparto de piensos a granjas expuesto en la Subsección 2.2.3, nos planteamos utilizar otro tipo de métodos para resolver nuestros TTRPs.

Como se vio en la Sección 2.1, numerosos han sido los autores que, tras la modelización del TTRP por parte de Chao (2002), han propuesto heurísticas y metaheurísticas para obtener soluciones a distintas variantes de este problema. El propio Chao combinó una heurística de construcción con la metaheurística de la búsqueda tabú, tal y como vimos en la Subsección 2.1.1.

Las heurísticas constructivas, como su nombre indica, construyen la solución paso a paso, de manera que se le van añadiendo elementos a medida que se tienen en cuenta distintos criterios. La heurística de construcción más conocida y utilizada para los VRPs se debe a Clarke y Wright (1964) y está basada en la noción de los ahorros. En el Apéndice A, de hecho, se presentan de forma resumida los pasos de dicho algoritmo.

A continuación explicamos la heurística de Clarke & Wright para el CVRP en su versión paralela¹, y resolvemos el primer ejemplo de la Subsección 2.2.3.

3.1. Algoritmo de Clarke & Wright para el CVRP

Cuando dos rutas $(0, \dots, i, 0)$ y $(0, j, \dots, 0)$ se pueden unir de modo factible en una simple ruta $(0, \dots, i, j, \dots, 0)$, se genera un ahorro en la distancia total de $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. En la Figura 3.1 mostramos, mediante un diagrama de flujo, los pasos de este algoritmo.

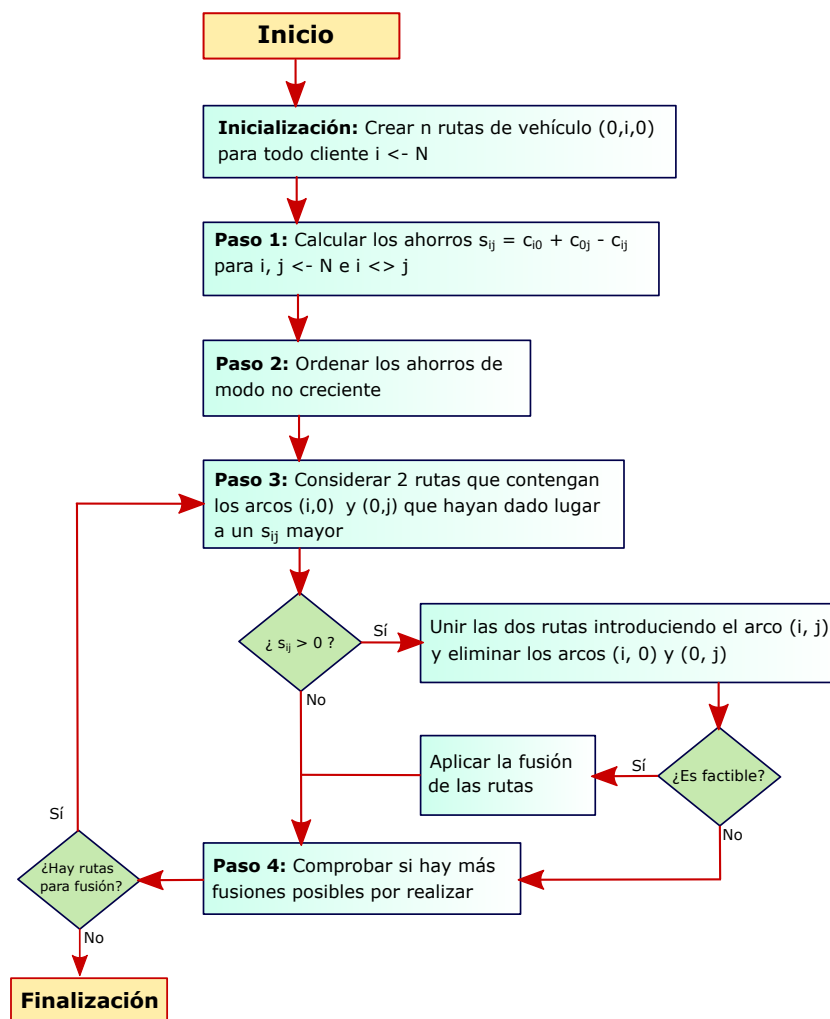
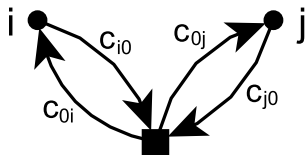
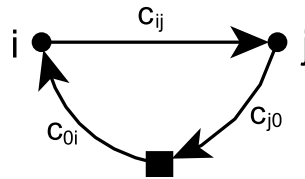


Figura 3.1: Pasos del algoritmo de Clarke & Wright.

¹El paso 2 de este algoritmo presenta dos versiones: una paralela y una secuencial. Ambas han sido explicadas en el Apéndice A. Para lo que nos concierne en este capítulo, nos interesa solamente la primera.

Gráficamente, podemos ver cómo funciona este algoritmo en las Figuras 3.2 y 3.3. En la primera de ellas, el coste de las rutas es $C_1 = c_{0i} + c_{i0} + c_{0j} + c_{j0}$; mientras que en la segunda sería $C_2 = c_{0i} + c_{ij} + c_{j0}$. Claramente, la diferencia de costes entre el primer escenario y el segundo, es decir, el ahorro, sería $s_{ij} = C_2 - C_1 = c_{i0} + c_{0j} - c_{ij}$. Cuando este ahorro es positivo y las demandas de los clientes i y j no exceden la capacidad del vehículo en cuestión, la ruta es factible. Podríamos por tanto implementar la fusión entre ambos clientes.

Figura 3.2: Rutas $(0, i, 0)$ y $(0, j, 0)$.Figura 3.3: Ruta $(0, i, j, 0)$.

Clarke & Wright es una sencilla heurística constructiva que puede proporcionar rápidamente buenos resultados en problemas realmente complejos, como el que se ha explicado en la Subsección 2.2.3, donde considerábamos un ejemplo real relativo al reparto de piensos por parte de una cooperativa agrícola gallega. En primer lugar, tomábamos el problema como un CVRP y lo resolvíamos con Gurobi utilizando la formulación implementada en AMPL. La solución obtenida puede verse en la Figura 2.7.

Ahora bien, para poner a prueba esta heurística, resolvemos el ejemplo anteriormente mencionado. Con el fin de facilitar la lectura de este trabajo, mostramos en el Apéndice C cada una de las iteraciones de este método. Tal y como podemos comprobar, al aplicar dicho algoritmo ya llegamos a la solución óptima (232 kilómetros) en la iteración 18, pero debemos continuar hasta que no nos queden ahorros positivos, lo cual se alcanza en la 83.

3.2. Heurística basada en los ahorros para el TTRP

La eficacia del algoritmo de Clarke & Wright a la hora de construir una solución de un CVRP y la necesidad de resolver TTRPs que involucren un número relativamente alto de clientes, ha servido de motivación para adaptar este método heurístico al caso del TTRP.

Hasta donde conocemos, solamente Derigs et al. (2013) mencionan la adaptación de este algoritmo para construir una solución inicial del TTRP, sin dar más detalles acerca

de dicha adaptación. A continuación, proponemos una modificación del método de Clarke & Wright para este problema, el cual hemos programado en R.

Hemos llamado a este método CW_TTRP (Clarke & Wright Truck and Trailer Routing Problem), siendo explicado su principal funcionamiento en el Algoritmo 1. Como argumentos de entrada, tenemos el número de clientes total y de tipo v.c., n y n_1 , respectivamente; la matriz de distancias, C ; las demandas de los clientes, d ; y las capacidades tanto del truck como del tráiler, Q_{truck} y $Q_{trailer}$, respectivamente.

A diferencia de lo que ocurre en el algoritmo de Clarke & Wright clásico, en CW_TTRP se tienen que contemplar dos matrices de rutas, R y \hat{R} . La primera la iremos modificando de modo que aparezcan en ella las rutas conectadas con el depósito (PTRs, PVRs y MT de las CVRs), mientras que \hat{R} la completaremos con las sub-rutas cuyo aparcamiento es un v.c.². Por lo tanto, tal y como indica la línea 3 del Algoritmo 1, se inicializan ambas matrices de la siguiente forma:

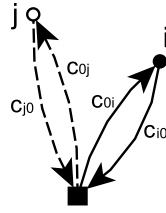
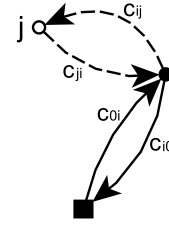
$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ \vdots & \vdots & \vdots \\ 0 & n & 0 \end{pmatrix} = \hat{R}$$

A continuación, el método propuesto también inicializa las matrices de ahorros S y \hat{S} , las cuales representan el cálculo de ahorros de tamaño $n \times n$. Las entradas de la matriz S se calculan como $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ (para $i, j \in N$, $i \neq j$)³, es decir, se trata de la matriz de ahorros usual. No obstante, al tener ahora clientes de dos tipos, y sabiendo que los clientes v.c. pueden servir como aparcamiento del tráiler, consideraremos, a mayores, otro tipo de ahorros, \hat{s}_{ij} . Se calculan del siguiente modo:

- $\hat{s}_{ij} = s_{ij}$, para $i, j \in N_1$ o $i, j \in N_2$, e $i \neq j$.
- $\hat{s}_{ij} = c_{j0} + c_{0j} - c_{ij} - c_{ji}$, para $i \in N_1$ y $j \in N_2$. Este ahorro resulta de eliminar la ruta $(0, j, 0)$ e introducirla como sub-ruta que parte de i . Es decir, pasaríamos de tener $(0, j, 0)$ y $(0, \dots, i, \dots, 0)$ a tener $(0, \dots, i, j, i, \dots, 0)$. Ilustramos cómo surge este ahorro en las Figuras 3.4 y 3.5:

²Esta heurística simplifica ligeramente el TTRP que hemos formulado en el Capítulo 1, ya que tal y como está implementada no permite que haya más de una sub-ruta por aparcamiento.

³Además, $s_{ii} = 0$, $\forall i \in N$.

Figura 3.4: Rutas $(0, i, 0)$ y $(0, j, 0)$.Figura 3.5: Ruta $(0, i, j, i, 0)$.

- $\hat{s}_{ij} = c_{i0} + c_{0i} - c_{ij} - c_{ji}$, para $j \in N_1$ e $i \in N_2$. Este ahorro es análogo al anterior, intercambiándose i y j el tipo de cliente.

Algoritmo 1 Heurística basada en los ahorros para el TTRP

```

1: procedure CW_TTRP( $n, n_1, C, d, Q_{truck}, Q_{trailer}$ )
2:   Create  $N_1 = \{1, \dots, n_1\}$  and  $N_2 = \{n_1 + 1, \dots, n\}$ 
3:   Create  $R$  and  $\hat{R}$  ▷ matrices  $n \times 3$  of routes and subroutes, respectively
4:   Create  $S$  and  $\hat{S}$  ▷ matrices  $n \times n$  of classic and modified savings, respectively
5:    $S_m = 1$ 
6:   while  $S_m > 0$  do
7:      $S_m = \text{maxSaving}\{S, \hat{S}\}$  ▷ maximum saving in  $S$  or  $\hat{S}$  matrices
8:     if  $S_m \in S$  then
9:       See Algorithm 2 ▷ classic C&W
10:    else
11:       $[i, j] = \text{returnCoord}(S_m, \hat{S})$ 
12:      if  $i \in N_1$  and  $j \in N_2$  then
13:        See Algorithm 3
14:      else if  $i \in N_2$  and  $j \in N_1$  then
15:        See Algorithm 4
16:      end if
17:    end if
18:  end while
19:  ResultRoute = CreateRoute( $R, \hat{R}$ )
20:  return ResultRoute
21: end procedure

```

Por lo tanto, la matriz \hat{S} será del siguiente modo:

$$\hat{S} = \left(\begin{array}{ccc|ccc} s_{1,1} & \cdots & s_{1,n_1} & \hat{s}_{1,n_1+1} & \cdots & \hat{s}_{1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ s_{n_1,1} & \cdots & s_{n_1,n_1} & \hat{s}_{n_1,n_1+1} & \cdots & \hat{s}_{n_1,n} \\ \hline \hat{s}_{n_1+1,1} & \cdots & \hat{s}_{n_1+1,n_1} & s_{n_1+1,n_1+1} & \cdots & s_{n_1+1,n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \hat{s}_{n,1} & \cdots & \hat{s}_{n,n_1} & s_{n,n_1+1} & \cdots & s_{n,n} \end{array} \right) \quad (3.1)$$

Una vez ya tenemos calculadas las matrices de rutas (R y \hat{R}) y de ahorros (S y \hat{S}), entramos en el bucle principal de la heurística propuesta (líneas 6 – 18, Algoritmo 1), en el que repetiremos una serie de pasos mientras que el ahorro máximo de S y \hat{S} sea mayor que 0. A partir de aquí, tenemos dos grandes casos diferenciados: si S_m (el ahorro máximo) pertenece a S , tenemos una ruta típica del CVRP. Entonces aplicamos el algoritmo clásico del Clarke & Wright, pero teniendo en cuenta el tipo de clientes involucrados en dicho ahorro, pues de ello dependerá qué tipo de ruta vayamos a generar (de camión o de vehículo completo). En el caso contrario, si S_m pertenece a \hat{S} (y no a S), significa que el ahorro actual surge de formar rutas mezcladas de camión y vehículo, y por lo tanto se representarán diferentes casuísticas de las distintas combinaciones que se pueden dar.

El Algoritmo 2 explica detalladamente el primer caso, si $S_m \in S$, el cual inicialmente obtiene las coordenadas i y j de S_m en S , usando la función `returnCoord`. Sean r_i y r_j las rutas en las que se encuentran los clientes i y j , respectivamente. Comprobamos:

- Si alguno de los dos clientes es de tipo t.c., siendo i el último de la ruta r_i y j el primero de la ruta r_j , entonces para poder efectuar la fusión $i - j$, la demanda de los clientes que forman ambas rutas no debe exceder Q_{truck} .
- Si ambos clientes son de tipo v.c., siendo i el último de la ruta r_i y j el primero de la ruta r_j , entonces para poder efectuar la fusión $i - j$, hay que tener en cuenta lo siguiente:
 - Si en alguna de las rutas r_i o r_j existe un cliente de tipo t.c., entonces para poder efectuar la fusión $i - j$, la demanda de los clientes que forman ambas rutas no debe exceder Q_{truck} .

- Por el contrario, si todos los clientes en r_i y r_j son de tipo v.c., se implementa la fusión $i - j$ siempre que las demandas en ambas rutas no excedan $Q_{truck} + Q_{trailer}$.

Hay que tener en cuenta que cuando se realiza una fusión, se eliminan los ahorros $s_{ij} = s_{ji}$, $\hat{s}_{ij} = \hat{s}_{ji}$. Además, también se actualiza la matriz de rutas R , del siguiente modo:

$$R = \begin{pmatrix} 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & i & j \\ \vdots & \vdots & \vdots \\ i & j & 0 \\ \vdots & \vdots & \vdots \\ 0 & n & 0 \end{pmatrix}$$

En otro caso, es decir, si no se lleva a cabo la fusión, solamente eliminamos s_{ij} .

Volviendo al Algoritmo 1, si $S_m \notin S$, esto implica que $S_m \in \hat{S}$. En tal caso, necesitamos de nuevo aplicar la función `returnCoord` (línea 11), esta vez para saber qué clientes i y j de \hat{S} dan lugar al ahorro máximo. Es inmediato percatarse, a la vista de la matriz \hat{S} (3.1) y teniendo en cuenta que le hemos dado preferencia a S , que estos clientes serán de distinto tipo. Veamos cómo proceder cuando $i \in N_1$ y $j \in N_2$, lo cual contemplamos en el Algoritmo 3. Tenemos dos posibilidades:

- Que el cliente j sea el único de su ruta, es decir, $r_j = (0, j, 0)$. En tal caso, tendríamos que calcular la suma de las demandas de todos los clientes en la ruta r_i , sea Q_{iaccum} .
- Si $d_j \leq Q_{truck}$ y $Q_{iaccum} + d_j \leq Q_{truck} + Q_{trailer}$, entonces se implementa la fusión. Mostramos cómo proceder gráficamente en las Figuras 3.6 y 3.7.

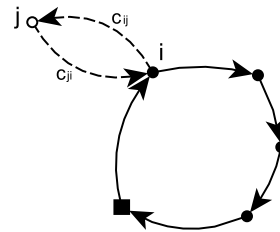
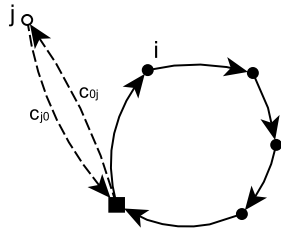


Figura 3.6: Antes de la implementación. Figura 3.7: Después de la implementación.

Cuando esto ocurre, además de borrar los ahorros $\hat{s}_{ij} = \hat{s}_{ji}$ y $s_{ij} = s_{ji}$, debemos actualizar la matriz \hat{R} , contemplando la sub-ruta que acabamos de crear:

$$\hat{R} = \begin{pmatrix} 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ j & i & j \\ \vdots & \vdots & \vdots \\ i & j & i \\ \vdots & \vdots & \vdots \\ 0 & n & 0 \end{pmatrix}$$

- Si no, eliminamos \hat{s}_{ij} .
- Que el cliente j sea el primero de su ruta, pero no el único. Luego $r_j = (0, j, \dots, 0)$. Sean ahora Q_{iaccum} y Q_{jaccum} las sumas de las demandas de todos los clientes en las rutas r_i y r_j , respectivamente. Además, sea l el último cliente de r_j . Así:
 - Si $Q_{jaccum} \leq Q_{truck}$ y $Q_{iaccum} + Q_{jaccum} \leq Q_{truck} + Q_{trailer}$, y se tiene que $c_{0j} + c_{l0} - c_{ij} - c_{li} > 0$, entonces se implementa la fusión. Mostramos cómo proceder gráficamente en las Figuras 3.8 y 3.9.



Figura 3.8: Antes de la implementación. Figura 3.9: Después de la implementación.

Cuando esto ocurre, además de borrar los ahorros $\hat{s}_{ij} = \hat{s}_{ji}$ y $s_{ij} = s_{ji}$, debemos actualizar la matriz \hat{R} , contemplando la sub-ruta que acabamos de crear, y la matriz R , de donde borraremos r_j , que anteriormente era una PTR.

- Si no, eliminamos \hat{s}_{ij} .

Nótese que el caso en el que $i \in N_2$ y $j \in N_1$, presente en el Algoritmo 4, es análogo.

Algoritmo 2 Heurística de Clarke & Wright diferenciando el tipo de cliente

```

1:  $[i, j] = \text{returnCoord}(S_m, S)$ 
2: if  $i \in N_1$  and  $j \in N_1$  then
3:   if  $R_{i3} == 0$  and  $R_{j1} == 0$  then
4:      $Q_{accum} = \text{returnQRuta}_i() + \text{returnQRuta}_j()$ 
5:      $x \rightarrow$  number of t.c. customers of Route_ $i$  and Route_ $j$ 
6:     if  $x == 0$  then
7:       if  $Q_{accum} \leq Q_{truck} + Q_{trailer}$  then
8:          $R_{i3} = j, R_{j1} = i$ 
9:         Delete  $s_{ij}, s_{ji}, \hat{s}_{ij}, \hat{s}_{ji}$ 
10:      end if
11:    else
12:      if  $Q_{accum} \leq Q_{truck}$  then
13:         $R_{i3} = j, R_{j1} = i$ 
14:        Delete  $s_{ij}, s_{ji}, \hat{s}_{ij}, \hat{s}_{ji}$ 
15:      end if
16:    end if
17:  end if
18:  Delete  $s_{ij}$ 
19: else
20:   if  $R_{i3} == 0$  and  $R_{j1} == 0$  then
21:      $Q_{accum} = \text{returnQRuta}_i() + \text{returnQRuta}_j()$ 
22:     if  $Q_{accum} \leq Q_{truck}$  then
23:        $R_{i3} = j, R_{j1} = i$ 
24:       Delete  $s_{ij}, s_{ji}, \hat{s}_{ij}, \hat{s}_{ji}$ 
25:     end if
26:   end if
27:   Delete  $s_{ij}$ 
28: end if

```

Algoritmo 3 Ahorro máximo en \hat{S} : casuística cuando i es v.c. y j es t.c.

```

1: if  $R_{j1} == 0$  and  $R_{j3} == 0$  then
2:    $Q_{iaccum} = \text{returnQRuta\_i}()$ 
3:   if  $d_j \leq Q_{truck}$  and  $Q_{iaccum} + d_j \leq Q_{truck} + Q_{trailer}$  then
4:      $\hat{R}_{i1} = j, \hat{R}_{i3} = j, \hat{R}_{j1} = i, \hat{R}_{j3} = i$ 
5:     Delete  $\hat{s}_{ij}, \hat{s}_{ji}, s_{ij}, s_{ji}$ 
6:   end if
7: else if  $R_{j3} \neq 0$  then
8:    $Q_{iaccum} = \text{returnQRuta\_i}()$ 
9:    $Q_{jaccum} = \text{returnQRuta\_j}()$ 
10:   $l \rightarrow$  last customer of Route_ $j$ 
11:  if  $Q_{jaccum} \leq Q_{truck}$  and  $Q_{iaccum} + Q_{jaccum} \leq Q_{truck} + Q_{trailer}$  then
12:    if  $c_{0j} + c_{l0} - c_{ij} - c_{li} > 0$  then
13:      copyRoute( $j, R, \hat{R}$ ) ▷ copy Route_ $j$  from  $R$  to  $\hat{R}$ 
14:      deleteRoute( $j, R$ ) ▷ delete Route_ $j$  from  $R$ 
15:       $\hat{R}_{j1} = i, \hat{R}_{l3} = i, \hat{R}_{i1} = l, \hat{R}_{i3} = j$ 
16:      Delete  $\hat{s}_{ij}, \hat{s}_{ji}, s_{ij}, s_{ji}$ 
17:    end if
18:  end if
19: end if
20: Delete  $\hat{s}_{ij}$ 

```

Algoritmo 4 Ahorro máximo en \hat{S} : casuística cuando i es t.c. y j es v.c.

```

1: if  $R_{i1} == 0$  and  $R_{i3} == 0$  then
2:    $Q_{jaccum} = \text{returnQRuta\_j}()$ 
3:   if  $d_i \leq Q_{truck}$  and  $Q_{jaccum} + d_i \leq Q_{truck} + Q_{trailer}$  then
4:      $\hat{R}_{j1} = i, \hat{R}_{j3} = i, \hat{R}_{i1} = j, \hat{R}_{i3} = j$ 
5:     Delete  $\hat{s}_{ij}, \hat{s}_{ji}, s_{ij}, s_{ji}$ 
6:   end if
7: else if  $R_{i3} \neq 0$  then
8:    $Q_{iaccum} = \text{returnQRuta\_i}()$ 
9:    $Q_{jaccum} = \text{returnQRuta\_j}()$ 
10:   $k \rightarrow$  last customer of Route_i
11:  if  $Q_{iaccum} \leq Q_{truck}$  and  $Q_{iaccum} + Q_{jaccum} \leq Q_{truck} + Q_{trailer}$  then
12:    if  $c_{0i} + c_{k0} - c_{ji} - c_{kj} > 0$  then
13:      copyRoute( $i, R, \hat{R}$ ) ▷ copy Route_i from  $R$  to  $\hat{R}$ 
14:      deleteRoute( $i, R$ ) ▷ delete Route_i from  $R$ 
15:       $\hat{R}_{i1} = j, \hat{R}_{k3} = j, \hat{R}_{j1} = k, \hat{R}_{j3} = i$ 
16:      Delete  $\hat{s}_{ij}, \hat{s}_{ji}, s_{ij}, s_{ji}$ 
17:    end if
18:  end if
19: end if
20: Delete  $\hat{s}_{ij}$ 

```

En la Figura 3.10, se muestra de manera esquemática el flujo principal de nuestra propuesta heurística, la cual, como hemos explicado, sigue el esquema clásico del algoritmo de los ahorros, pero representando las casuísticas especiales que aparecen en los TTRPs.

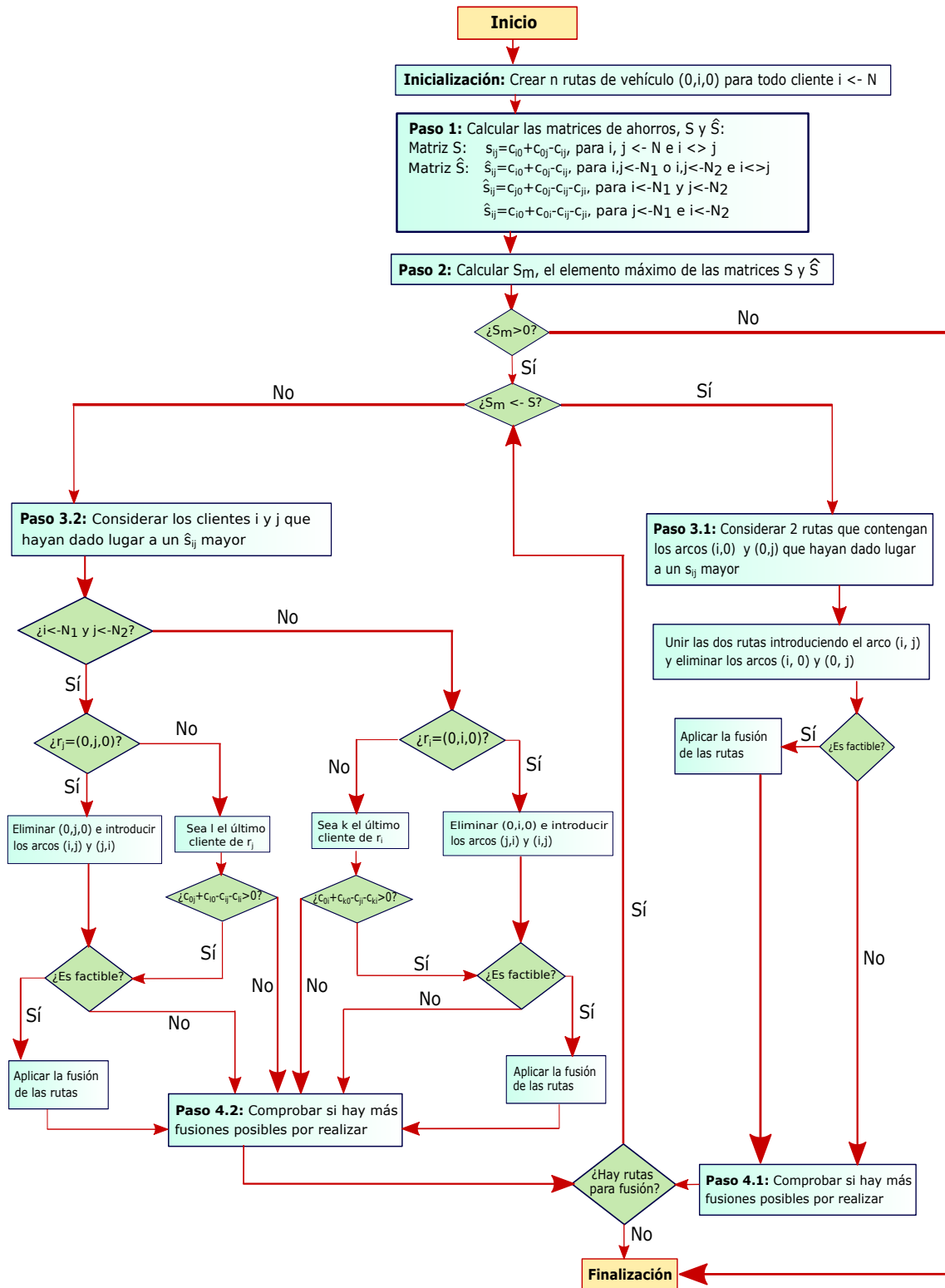


Figura 3.10: Pasos de la heurística CW_TTRP.

3.3. Resultados computacionales de la heurística CW_TTRP

Con el fin de probar nuestro algoritmo heurístico, hemos resuelto los dos ejemplos vistos en el Capítulo 2. Tal y como lo hemos implementado, solo admite vehículos con capacidades homogéneas. De este modo, podemos utilizar para el primer problema los datos presentes en la Subsección 2.2.1. Sin embargo, necesitamos adaptar ligeramente el caso de la cooperativa agrícola visto al final de la Subsección 2.2.3: asumiremos que los camiones y tráileres presentan capacidades de 15300 y 25300 kilogramos, respectivamente. Teniendo esto en cuenta, mostramos en las Tablas 3.1 y 3.2 los resultados obtenidos.

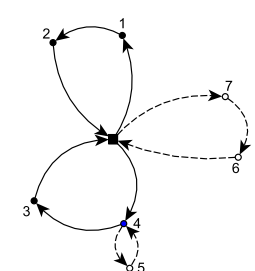
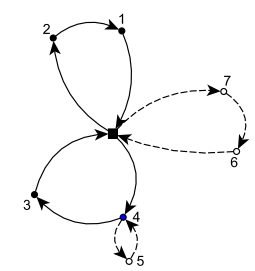
	Objetivo (kilómetros)	Tiempo (segundos)	Rutas
Solución exacta (Gurobi)	1725	0.479964	
CW_TTRP	1725	0.097739	

Tabla 3.1: Comparativa de los resultados computacionales para el primer ejemplo estudiado.

A la vista de la Tabla 3.1 podemos ver que para el primer ejemplo, nuestra heurística CW_TTRP proporciona la solución exacta en un tiempo casi 5 veces menor.

Sin embargo, tal y como refleja la Tabla 3.2, la función objetivo obtenida por CW_TTRP para el ejemplo de la cooperativa agrícola difiere en 7 kilómetros de la óptima. Si bien es cierto que la ruta de camión pura es distinta en ambas soluciones, pues los clientes son visitados en distinto orden, esto no afecta al número de kilómetros.

	Objetivo (kilómetros)	Tiempo (segundos)	Rutas
Solución exacta (Gurobi)	207	13503	
CW_TTRP	214	0.356041	

Tabla 3.2: Comparativa de los resultados computacionales para el problema real de la cooperativa.

La diferencia de distancia entre ambos resultados está presente en la sub-ruta. La solución exacta proporciona la sub-ruta $2 - 9 - 7 - 8 - 2$, con una longitud de 28 km; mientras que CW_TTRP devuelve $2 - 9 - 8 - 7 - 2$, de 35 km. Esto se debe a que, inicialmente, la heurística crea la PTR $0 - 9 - 8 - 7 - 0$ (pues es menos costosa que $0 - 9 - 7 - 8 - 0$) y la implementa en alguna iteración posterior como sub-ruta del cliente 2, siguiendo el procedimiento visto en las Figuras 3.8 y 3.9.

A pesar de no obtener la solución óptima, sí se consigue un resultado cercano al óptimo (empeora un 3.38%), con una configuración de rutas muy similar. Además, el tiempo computacional es considerablemente mejor: mientras que Gurobi tarda casi 4 horas en proporcionar la solución exacta, CW_TTRP devuelve un resultado asequible en menos de medio segundo.

Capítulo 4

TTRP con compartimentos

Hay ciertas ocasiones en las que los clientes demandan diferentes tipos de productos que no se pueden mezclar, o no es conveniente mezclar el producto destinado a clientes distintos (por ejemplo, por cuestiones técnicas). En tal caso, las distintas clases de mercancía deben ser transportadas o bien en vehículos diferentes, o bien en compartimentos separados del mismo vehículo. Para dar cabida a esta última situación, surge el VRP con compartimentos, conocido como MC-VRP (*Multi-Compartment Vehicle Routing Problem*). Las aplicaciones del MC-VRP se pueden encontrar en la distribución de combustible (diferentes tipos de productos petrolíferos transportados dentro del mismo vehículo), distribución de alimentos (por ejemplo, un compartimento refrigerado y un compartimento regular dentro del mismo vehículo), recolección de desechos, recolección de leche en granjas, etc.

En la Sección 4.1 comenzaremos haciendo una revisión bibliográfica del MC-VRP que, como veremos, motivó la posterior extensión al MC-TTRP (*Multi-Compartment Truck and Trailer Routing Problem*), en el cual tanto los camiones como los tráileres estarán compartimentados. Un ejemplo de vehículo multicompartimentado lo podemos ver en la Figura 4.1.

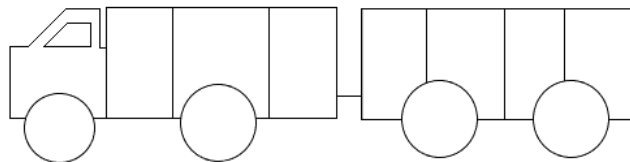


Figura 4.1: Vehículo compartimentado: combinación de camión y tráiler con tres y cuatro tolvas, respectivamente.

En la Sección 4.2 formularemos el MC-TTRP como un problema de programación lineal y entera binaria, extendiendo y modificando en lo necesario la formulación del TTRP de Chao vista en la Subsección 1.1.2. Finalmente, en la Sección 4.3, resolveremos con AMPL algunos de los ejemplos que ya hemos visto, contemplando el uso de vehículos divididos en compartimentos.

4.1. Revisión de problemas con compartimentos

Un caso especial del VRP es aquel en el que los vehículos están divididos en compartimentos (MC-VRP). Los primeros trabajos acerca de este problema se deben a [Brown y Graves \(1981\)](#) y [Brown et al. \(1987\)](#), cuyo objetivo es la distribución de productos petroleros en Estados Unidos. En ellos se considera separadamente el diseño de las rutas y la asignación de productos a compartimentos, y se hace uso de una serie de problemas del viajante. Otro de los artículos precursores del MC-VRP es [van der Bruggen et al. \(1995\)](#). En él se modela un problema de redistribución de productos en una gran compañía petrolera en los Países Bajos, y se propone un algoritmo heurístico para su resolución. Dicho algoritmo, aplicado a los datos de la compañía, generó ahorros con respecto a la política de distribución anterior que tenía la empresa.

[Chajakis y Guignard \(2003\)](#) emplean vehículos con varios compartimentos para el abastecimiento de las llamadas tiendas de conveniencia, o 24 horas; y proponen para la resolución del problema un algoritmo heurístico basado en la relajación lagrangiana. [Avella et al. \(2004\)](#) formulan el problema como un modelo de partición de conjuntos. Basados en él, consideran un algoritmo exacto *branch-and-price* y, finalmente, emplean un algoritmo heurístico de empaquetado/enrutamiento para problemas grandes. [El Fallahi et al. \(2008\)](#) comparan un algoritmo constructivo, un algoritmo memético y una búsqueda tabú, cuando la asignación de los distintos productos a los compartimentos estaba prefijada. Llegaron a la conclusión de que los resultados proporcionados por la búsqueda tabú eran ligeramente mejores, aunque requería un mayor tiempo computacional. [Oppen y Løkketangen \(2008\)](#) proponen un algoritmo de búsqueda tabú para un problema relacionado con el transporte de animales vivos, donde se presentaban vehículos con varios compartimentos y restricciones de inventario.

[Caramia y Guerriero \(2010b\)](#) investigaron un problema de rutas de vehículos con varios compartimentos tal que, como máximo, se podía asignar un tipo de producto a cada compartimento. Además, establecían la restricción adicional de que algunas granjas eran

pequeñas e inaccesibles para los vehículos grandes. Los autores no proporcionaron la formulación de un modelo que abarcara todo el problema, pero, en su lugar, utilizaron para su resolución una heurística en dos etapas. La primera consistía en minimizar el número de vehículos y la asignación de cada cliente a un vehículo. En la segunda fase, se minimizaba el coste total de las rutas. La restricción de inaccesibilidad en este caso, se lleva a cabo asumiendo la existencia de vehículos formados por un camión y un tráiler, de modo que hay algunas granjas que no pueden ser visitadas por el tráiler, en cuyo caso solo son visitadas por el camión. De este modo, se trata del primer trabajo (y, hasta ahora, el único por lo que sabemos) que considera el TTRP con compartimentos, al cual nos referiremos como MC-TTRP en lo sucesivo.

Muyldermans y Pang (2010) utilizaron el algoritmo de ahorros de Clarke & Wright para obtener una solución inicial factible. A continuación, llevaron a cabo una búsqueda local mediante movimientos tomados de la literatura y mejoraron la calidad de la solución obtenida anteriormente mediante una metaheurística basada en la búsqueda local guiada. Estos autores realizaron un análisis de sensibilidad en ciertos parámetros (número de clientes y sus demandas, localización del depósito, capacidad de los vehículos o número de productos) y trabajaron tanto con nuevas instancias como con algunas ya existentes. Por su parte, Derigs et al. (2011) consideran un modelo que involucra a una flota homogénea: todos los vehículos poseen el mismo número de compartimentos, con las mismas capacidades. Presentaron unos *benchmarks* propios y un solver con componentes heurísticos que cubrían una amplia gama de enfoques alternativos para la construcción, búsqueda local, búsqueda de gran vecindad y metaheurísticas, con el objetivo de identificar algoritmos efectivos que obtuviesen soluciones de calidad. Además, una de sus propuestas como trabajo futuro es la extensión de sus métodos al caso de flota heterogénea (vehículos con diferentes compartimentos y capacidades).

Otro trabajo reciente es el de Coelho y Laporte (2015), donde se definen y comparan cuatro categorías del problema de suministro de combustible con multi-compartimentos, introduciendo costes tanto de rutas como de inventario. Proponen dos formulaciones para cada caso y describen un algoritmo de ramificación y corte para resolver casos de período único y de período múltiple, que contienen hasta 50 y 20 clientes, respectivamente. En Archetti et al. (2016) se estudia el problema de entrega de varios productos y se compara el coste de transporte cuando los vehículos se cargan con un único tipo de mercancía o con más de uno. Silvestrin y Ritt (2017) presentan un algoritmo basado en búsqueda tabú para resolver un MC-VRP.

Henke et al. (2019) consideran una variante del MC-VRP que surge en el reciclado de vidrio. Se asume que para la recolección de los contenedores de vidrio se dispone de una flota homogénea de vehículos. De forma individual para cada vehículo, la capacidad puede ser dividida en un cierto número de compartimentos, asignados cada uno a un tipo de vidrio diferente. El objetivo del problema es minimizar la distancia total que deben recorrer los vehículos. Para resolverlo de forma óptima, los autores desarrollan e implementan un algoritmo de ramificación y corte. Ostermeier y Hübner (2018) plantean un MC-VRP para la distribución de alimentos a los supermercados, con una flota de vehículos con compartimentos flexibles. El objetivo de su trabajo es mostrar los beneficios de considerar una flota mixta, formada tanto por vehículos con un único compartimento como por vehículos compartimentados. Para ello tuvieron en cuenta los distintos costos incurridos por el uso de los ambos tipos de vehículos. El problema se resolvió mediante búsqueda de gran vecindad.

Mendoza et al. (2010) extienden el MC-VRP al caso en el que las demandas son estocásticas, dando lugar al MC-VRPSD (*Multi-Compartment Vehicle Routing Problem with Stochastic Demands*). En Mendoza et al. (2011) se propone un conjunto de heurísticas de construcción para resolver este problema, que comprenden versiones estocásticas del vecino más cercano, de mejor inserción y del algoritmo de Clarke & Wright propuesto por Dror y Trudeau (1986), extendidas al caso de multi-compartimentos. Posteriormente, trabajos como Pandelis et al. (2012), Huang (2015) o Goodson (2015) se centran en estudiar y proponer heurísticas para obtener soluciones de este problema.

Un estudio reciente acerca del MC-VRP en el que se presenta una situación real es Guitián de Frutos y Casas-Méndez (to appear).

4.2. Formulación del MC-TTRP

En esta sección vamos a proponer una formulación del MC-TTRP como un problema de programación lineal entera binaria; algo que, hasta este momento, no consta que se haya hecho. Para facilitar la lectura del trabajo y, a su vez, evitar ser repetitivos, omitimos especificar de nuevo todas aquellas variables o parámetros que hayan aparecido con anterioridad. De tal modo, trataremos de complementar la Tabla 1.2 presentando en la Tabla 4.1 las constantes, conjuntos y variables adicionales (así como todas aquellas que haya sido necesario modificar) imprescindibles para la formulación de este nuevo problema.

Constantes y conjuntos	
F	Conjunto de los distintos tipos de pienso
Ht_k	Conjunto de tolvas del camión $k \in K$
Hl_r	Conjunto de tolvas del tráiler $r \in K_1$
Lt	Capacidad legal de los camiones
Ll	Capacidad legal de los tráileres
$d_{if} \geq 0$	Demanda del cliente $i \in N$ de pienso $f \in F$
$Q_{truck_k} \geq 0$	Capacidad del camión $k \in K$
$Q_{trailer_r} \geq 0$	Capacidad del tráiler $r \in K_1$
$Qt_{k,ht} \geq 0$	Capacidad de la tolva $ht \in Ht_k$ del camión $k \in K$
$Ql_{r,hl} \geq 0$	Capacidad de la tolva $hl \in Hl_r$ del tráiler $r \in K_1$
Variables	
$U_{i,f,ht}^k = \begin{cases} 1 & \text{si la tolva } ht \in Ht_k \text{ del camión } k \in K \text{ está cargada con} \\ & \text{mercancía } f \in F \text{ para el cliente } i \in N \\ 0 & \text{en otro caso} \end{cases}$	
$V_{i,f,hl}^r = \begin{cases} 1 & \text{si la tolva } hl \in Hl_r \text{ del tráiler } r \in K_1 \text{ está cargada con} \\ & \text{mercancía } f \in F \text{ para el cliente } i \in N_1 \\ 0 & \text{en otro caso} \end{cases}$	
$ZT_{i,f,ht}^k \in [0, 1]$, proporción de la tolva $ht \in Ht_k$ del camión $k \in K$ cargada con mercancía $f \in F$ para el cliente $i \in N$	
$ZL_{i,f,hl}^r \in [0, 1]$, proporción de la tolva $hl \in Hl_r$ del trailer $r \in K_1$ cargada con mercancía $f \in F$ para el cliente $i \in N_1$	

Tabla 4.1: Constantes, conjuntos y variables adicionales para la formulación del MC-TTRP.

Una vez hemos presentado la notación que vamos a utilizar, ya estamos en condiciones de formular el modelo:

$$\text{minimizar } \left\{ \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} c_{ij} x_{ij}^{kr} + \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n c_{ij} y_{ij}^{klv} \right\} \quad (4.1)$$

sujeto a

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ij}^{kr} + \sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{\substack{l=0 \\ l \neq j}}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = 1, \dots, n_1; \quad (4.2)$$

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ij}^{kr} + \sum_{i=0}^n \sum_{k=1}^{m_k} y_{ij}^{k j v} \leq 2, \quad j = 1, \dots, n_1; \quad v = 1, \dots, n; \quad (4.3)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{ij}^{klv} = 1, \quad j = n_1 + 1, \dots, n; \quad (4.4)$$

$$\sum_{i=0}^n \sum_{k=1}^{m_k} \sum_{l=1}^{n_1} \sum_{v=1}^n y_{i0}^{klv} = 0, \quad (4.5)$$

$$\sum_{j=0}^n y_{lj}^{klv} \leq 1, \quad k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \quad v = 1, \dots, n; \quad (4.6)$$

$$y_{lj}^{klv} \leq \sum_{i=0}^{n_1} \sum_{r=1}^{m_l} x_{il}^{kr}, \quad j = 1, \dots, n; \quad k = 1, \dots, m_l; \quad l = 1, \dots, n_1; \quad v = 1, \dots, n; \quad (4.7)$$

$$y_{ij}^{klv} \leq \sum_{p=0}^n y_{lp}^{klv}, \quad i, j = 0, \dots, n; \quad k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \quad v = 1, \dots, n; \quad (4.8)$$

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^n y_{ij}^{klv} = 0, \quad k = m_l + 1, \dots, m_k; \quad l = 1, \dots, n_1; \quad (4.9)$$

$$\sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^n y_{ij}^{k0v} = 0, \quad k = 1, \dots, m_l; \quad (4.10)$$

$$\sum_{v=1}^n \sum_{j=1}^n y_{0j}^{k0v} \leq 1, \quad k = m_l + 1, \dots, m_k; \quad (4.11)$$

$$\sum_{j=1}^{n_1} \sum_{r=1}^{m_l} x_{0j}^{kr} \leq 1, \quad k = 1, \dots, m_l; \quad (4.12)$$

$$\sum_{j=1}^{n_1} \sum_{k=1}^{m_l} x_{0j}^{kr} \leq 1, \quad r = 1, \dots, m_l; \quad (4.13)$$

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq l}}^n \sum_{f \in F} d_{jf} y_{ij}^{klv} \leq Q_{truckk}, \quad k = 1, \dots, m_l; \quad l = 1, \dots, n_1; \quad v = 1, \dots, n; \quad (4.14)$$

$$\sum_{i=0}^n \sum_{j=1}^n \sum_{f \in F} d_{jf} y_{ij}^{k0v} \leq Q_{truckk}, \quad k = m_l + 1, \dots, m_k; \quad v = 1, \dots, n; \quad (4.15)$$

$$\begin{aligned} & \sum_{i=0}^{n_1} \sum_{j=1}^{n_1} \sum_{f \in F} d_{jf} x_{ij}^{kr} + \\ & + \sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^{n_1} \sum_{v=1}^n \sum_{\substack{f \in F \\ j \neq l}} d_{jf} y_{ij}^{klv} \leq Q_{truckk} + Q_{trailer_r}, \quad k, r = 1, \dots, m_l; \end{aligned} \quad (4.16)$$

$$\begin{aligned} & \sum_{i=1}^{n_1} \sum_{j=0}^{n_1} s_j x_{ij}^{kr} + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{\substack{i=1 \\ i \neq l}}^n \sum_{j=0}^n s_j y_{ij}^{klv} + \sum_{i=0}^{n_1} \sum_{j=0}^{n_1} (c_{ij}/vm) x_{ij}^{kr} + \\ & + \sum_{l=0}^{n_1} \sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (c_{ij}/vm) y_{ij}^{klv} \leq D - s_0, \quad k, r = 1, \dots, m_l; \end{aligned} \quad (4.17)$$

$$\sum_{v=1}^n \sum_{i=0}^n \sum_{j=0}^n (s_j + c_{ij}/vm) y_{ij}^{k0v} \leq D, \quad k = m_l + 1, \dots, m_k; \quad (4.18)$$

$$\sum_{i=0}^n y_{ij}^{klv} = \sum_{p=0}^n y_{jp}^{klv}, \quad j = 0, \dots, n, \quad k = 1, \dots, m_k; \quad l = 0, \dots, n_1; \quad v = 1, \dots, n; \quad (4.19)$$

$$\sum_{i=0}^{n_1} x_{ij}^{kr} = \sum_{p=0}^{n_1} x_{jp}^{kr}, \quad j = 0, \dots, n_1; \quad k, r = 1, \dots, m_l; \quad (4.20)$$

$$\sum_{i \in B} \sum_{j \in B} x_{ij}^{kr} \leq |B| - 1, \quad k, r = 1, \dots, m_l; \quad \forall B \subseteq N_1 : |B| \geq 2; \quad (4.21)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{klv} - \sum_{i \in B \cap N_1} \sum_{j \in N_1 \setminus B} \sum_{r=1}^{m_l} x_{ij}^{kr} \leq |B| - 1, \quad k = 1, \dots, m_l; \quad l = 1, \dots, n_1; \quad v = 1, \dots, n; \quad \forall B \subseteq N : |B| \geq 2; \quad (4.22)$$

$$\sum_{i \in B} \sum_{j \in B} y_{ij}^{k0v} \leq |B| - 1, \quad k = m_l + 1, \dots, m_k; \quad v = 1, \dots, n, \quad \forall B \subseteq N : |B| \geq 2; \quad (4.23)$$

$$\sum_{i=0}^{n_1} \sum_{k=1}^{m_l} \sum_{r=1}^{m_l} x_{ii}^{kr} + \sum_{j=0}^n \sum_{t=1}^{m_k} \sum_{l=0}^{n_1} \sum_{v=1}^n y_{jj}^{tlv} = 0, \quad (4.24)$$

$$\frac{1}{|Ht_k|} \sum_{f \in F} \sum_{ht \in Ht_k} ZT_{j,f,ht}^k \leq \sum_{i=0}^{n_1} \sum_{r=1}^{m_l} x_{ij}^{kr} + \sum_{i=0}^n \sum_{l=1}^{n_1} \sum_{v=1}^n y_{ij}^{klv}, \quad k = 1, \dots, m_l; \quad j = 1, \dots, n_1; \quad (4.25)$$

$$\frac{1}{|Ht_k|} \sum_{f \in F} \sum_{ht \in Ht_k} ZT_{j,f,ht}^k \leq \sum_{i=0}^n \sum_{v=1}^n y_{ij}^{k0v}, \quad k = m_l + 1, \dots, m_k; \quad j = 1, \dots, n_1; \quad (4.26)$$

$$\frac{1}{|Hl_r|} \sum_{f \in F} \sum_{hl \in Hl_r} ZL_{j,f,hl}^r \leq \sum_{i=0}^{n_1} \sum_{k=1}^{m_l} x_{ij}^{kr}, \quad r = 1, \dots, m_l; \quad j = 1, \dots, n_1; \quad (4.27)$$

$$\frac{1}{|Ht_k|} \sum_{f \in F} \sum_{ht \in Ht_k} ZT_{j,f,ht}^k \leq \sum_{i=0}^n \sum_{l=0}^{n_1} \sum_{v=1}^n y_{ij}^{klv}, \quad k = 1, \dots, m_k; \quad j = n_1 + 1, \dots, n; \quad (4.28)$$

$$\sum_{i=0}^{n_1} x_{ij}^{kr} \leq \sum_{f \in F} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k + \sum_{f \in F} \sum_{hl \in Hl_r} Ql_{r,hl} ZL_{j,f,hl}^r, \quad k, r = 1, \dots, m_l; \quad j = 1, \dots, n_1; \quad (4.29)$$

$$\sum_{i=0}^n \sum_{v=1}^n y_{ij}^{k0v} \leq \sum_{f \in F} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k, \quad k = m_l + 1, \dots, m_k; \quad j = 1, \dots, n_1; \quad (4.30)$$

$$\sum_{i=0}^n \sum_{l=1}^{n_1} \sum_{v=1}^n y_{ij}^{klv} \leq \sum_{f \in F} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k, \quad k = 1, \dots, m_l; \quad j = 1, \dots, n_1; \quad (4.31)$$

$$\sum_{i=0}^n \sum_{l=0}^{n_1} \sum_{v=1}^n y_{ij}^{klv} \leq \sum_{f \in F} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k, \quad k = 1, \dots, m_k; \quad j = n_1 + 1, \dots, n; \quad (4.32)$$

$$\sum_{r=1}^{m_l} \sum_{hl \in Hl_r} Ql_{r,hl} ZL_{j,f,hl}^r + \sum_{k=1}^{m_k} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k = d_{jf}, \quad j = 1, \dots, n_1; \quad f \in F; \quad (4.33)$$

$$\sum_{k=1}^{m_k} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{j,f,ht}^k = d_{jf}, \quad j = n_1 + 1, \dots, n; \quad f \in F; \quad (4.34)$$

$$\sum_{i=1}^n \sum_{f \in F} \sum_{ht \in Ht_k} Qt_{k,ht} ZT_{i,f,ht}^k \leq Lt, \quad k = 1, \dots, m_k; \quad (4.35)$$

$$\sum_{i=1}^{n_1} \sum_{f \in F} \sum_{hl \in Hl_r} Ql_{r,hl} ZL_{i,f,hl}^r \leq Ll, \quad r = 1, \dots, m_l; \quad (4.36)$$

$$\sum_{i=1}^n \sum_{f \in F} ZT_{i,f,ht}^k \leq 1, \quad k = 1, \dots, m_k; \quad ht \in Ht_k; \quad (4.37)$$

$$\sum_{i=1}^{n_1} \sum_{f \in F} ZL_{i,f,hl}^r \leq 1, \quad r = 1, \dots, m_l; \quad hl \in Hl_r; \quad (4.38)$$

$$ZT_{i,f,ht}^k - U_{i,f,ht}^k \leq 0, \quad i = 1, \dots, n; \quad f \in F; \quad k = 1, \dots, m_k; \quad ht \in Ht_k; \quad (4.39)$$

$$ZL_{i,f,hl}^r - V_{i,f,hl}^r \leq 0, \quad i = 1, \dots, n_1; \quad f \in F; \quad r = 1, \dots, m_l; \quad hl \in Hl_r; \quad (4.40)$$

$$U_{i,f_1,ht}^k + U_{j,f_2,ht}^k \leq 1, \quad i, j = 1, \dots, n; \quad f_1, f_2 \in F, \quad f_1 \neq f_2; \quad k = 1, \dots, m_k; \quad ht \in Ht_k; \quad (4.41)$$

$$V_{i,f_1,hl}^r + V_{j,f_2,hl}^r \leq 1, \quad i, j = 1, \dots, n_1; \quad f_1, f_2 \in F, \quad f_1 \neq f_2; \quad r = 1, \dots, m_l; \quad hl \in Hl_r; \quad (4.42)$$

$$U_{i,f,ht}^k + U_{j,f,ht}^k \leq 1, \quad i, j = 1, \dots, n; \quad i \neq j; \quad f \in F; \quad k = 1, \dots, m_k; \quad ht \in Ht_k; \quad (4.43)$$

$$V_{i,f,hl}^r + V_{j,f,hl}^r \leq 1, \quad \begin{array}{l} i, j = 1, \dots, n_1; i \neq j; f \in F; \\ r = 1, \dots, m_l; hl \in Hl_r; \end{array} \quad (4.44)$$

$$x_{ij}^{kr} \in \{0, 1\}, \quad i, j = 0, \dots, n_1; k, r = 1, \dots, m_l; \quad (4.45)$$

$$y_{ij}^{klv} \in \{0, 1\}, \quad \begin{array}{l} i, j = 0, \dots, n, k = 1, \dots, m_k; \\ l = 0, \dots, n_1; v = 1, \dots, n; \end{array} \quad (4.46)$$

$$ZT_{i,f,ht}^k \in [0, 1], \quad i = 0, \dots, n; f \in F; k = 1, \dots, m_k; ht \in Ht_k; \quad (4.47)$$

$$ZL_{i,f,hl}^r \in [0, 1], \quad i = 0, \dots, n_1; f \in F; r = 1, \dots, m_l; hl \in Hl_r; \quad (4.48)$$

$$U_{i,f,ht}^k \in \{0, 1\}, \quad i = 0, \dots, n; f \in F; k = 1, \dots, m_k; ht \in Ht_k; \quad (4.49)$$

$$V_{i,f,hl}^r \in \{0, 1\}, \quad i = 0, \dots, n_1; f \in F; r = 1, \dots, m_l; hl \in Hl_r; \quad (4.50)$$

Detallemos el modelo arriba expuesto:

- Las restricciones (4.2) – (4.24) ya han sido explicadas en secciones anteriores del presente trabajo, ya que se centran en la construcción de las rutas y en que estas sean factibles.
- El conjunto de restricciones (4.25) – (4.32), sin embargo, relaciona las rutas con la distribución de la carga en los distintos compartimentos:

(4.25): en una CVR o PVR, si un camión no visita a un cliente de tipo v.c. entonces dicho camión no carga pienso para ese cliente en ninguna de sus tolvas.

(4.26): en una PTR, si un camión no visita a un cliente de tipo v.c. entonces dicho camión no carga pienso para ese cliente en ninguna de sus tolvas.

(4.27): si un tráiler no visita a un cliente de tipo v.c. entonces dicho tráiler no carga pienso para ese cliente en ninguna de sus tolvas.

(4.28): si un camión no visita a un cliente de tipo t.c. entonces no carga pienso para él.

(4.29): si un vehículo completo visita a un cliente v.c. entonces o bien el camión o bien el tráiler que lo conforman están cargados con mercancía para él.

(4.30): en una PTR, si un cliente v.c. es visitado por un camión, entonces dicho camión le reparte mercancía.

(4.31): en una sub-ruta de una ruta completa, si un cliente v.c. está siendo servido por un camión, entonces, en efecto, ese camión está cargado con mercancía para él.

(4.32): si un camión visita a un cliente t.c. entonces le lleva mercancía en alguna de sus tolvas.

- Las restricciones (4.33) – (4.34) modelizan la entrega de las demandas a los clientes:
 - (4.33): todos los clientes v.c. reciben toda su demanda.
 - (4.34): todos los clientes t.c. reciben toda su demanda.
- El conjunto (4.35) – (4.38) restringe la cantidad de carga que se puede transportar:
 - (4.35): ningún camión carga más de lo que está legalmente permitido.
 - (4.36): ningún tráiler carga más de lo que está legalmente permitido.
 - (4.37): ninguna tolva de camión carga por encima de su capacidad.
 - (4.38): ninguna tolva de tráiler carga por encima de su capacidad.
- El conjunto (4.39) – (4.44) modeliza restricciones técnicas en el procedimiento de carga:
 - (4.39): relación lógica entre U y ZT .
 - (4.40): relación lógica entre V y ZL .
 - (4.41): no está permitido mezclar productos de distinto tipo (ni de un mismo socio ni de socios diferentes) en una misma tolva de camión.
 - (4.42): no está permitido mezclar productos de distinto tipo (ni de un mismo socio ni de socios diferentes) en una misma tolva de tráiler.
 - (4.43): no se pueden mezclar productos de clientes distintos en una tolva de camión.
 - (4.44): no se pueden mezclar productos de clientes distintos en una tolva de tráiler.
- Las restricciones (4.45) – (4.50) expresan la naturaleza de las variables involucradas en el modelo:
 - (4.45): las variables x 's son binarias.
 - (4.46): las variables y 's son binarias.
 - (4.47): las variables ZT 's toman valores en $[0, 1]$.
 - (4.48): las variables ZL 's toman valores en $[0, 1]$.
 - (4.49): las variables U 's son binarias.
 - (4.50): las variables V 's son binarias.

Nótese que, si bien es verdad que ciertas restricciones de las anteriores podrían expresarse como una sola, hemos decidido separar algunas de ellas atendiendo al tipo de cliente y/o vehículo que lo abastece, con el fin de facilitar la comprensión del modelo.

4.3. Resolución del MC-TTRP en AMPL

Al igual que hicimos en la Sección 2.2 con las dos versiones del TTRP presentadas hasta ese momento, hemos implementado en AMPL el MC-TTRP que acabamos de explicar. De nuevo, incluimos en el Apéndice B el código fuente del modelo, `MC_TTRP.mod`. Con el fin de testar esta formulación, resolveremos en las dos subsecciones posteriores adaptaciones de los problemas ya vistos en el caso del TTRP, introduciendo diferentes tipos de productos y, en consecuencia, la necesidad de tener vehículos con compartimentos.

4.3.1. Resolución de un ejemplo en AMPL

En primer lugar, presentamos una modificación del ejemplo expuesto en la Subsección 2.2.2. Asumimos ahora que tanto los 3 camiones como los 2 tráileres están compartimentados, y resumimos la capacidad de cada una de sus tolvas en las Tablas 4.2 y 4.3.

La matriz de distancias es la misma que tenemos en la Subsección 2.2.1. Lo que sí varían son las demandas: lo que anteriormente era un vector, pasa ahora a ser una matriz, cuyas filas hacen referencia a los socios y cuyas columnas indican el tipo de producto. En este ejemplo, asumiremos que hay dos productos diferentes y presentamos a continuación la matriz de demandas:

$$d = \begin{pmatrix} 20 & 5 \\ 10 & 10 \\ 10 & 0 \\ 10 & 6 \\ 10 & 4 \\ 0 & 5 \\ 7 & 0 \end{pmatrix}$$

Así, estaríamos diciendo, tal y como recoge la fila 5 de la matriz d , que el cliente 5 demanda 10 toneladas de producto 1 y 4 toneladas de producto 2. El archivo de AMPL en el que hemos introducido estos datos es `MC-TTRP.dat` (Apéndice B).

Camión (n° de tolvas)	Tolva	Capacidad
Camión 1 (2 tolvas)	1	10
	2	8
Camión 2 (2 tolvas)	1	10
	2	5
Camión 3 (3 tolvas)	1	5
	2	5
	3	5

Tráiler (n° de tolvas)	Tolva	Capacidad
Tráiler 1 (3 tolvas)	1	10
	2	10
	3	10
Tráiler 2 (4 tolvas)	1	10
	2	5
	3	10
	4	10

Tabla 4.2: Capacidad de cada tolva de camión.

Tabla 4.3: Capacidad de cada tolva de tráiler.

Hemos resuelto este problema con Gurobi (en NEOS) y, tras un tiempo de ejecución de 1.20634 segundos, se ha obtenido una solución óptima de 1725 kilómetros. En ella aparecen, como ocurría cuando no teníamos compartimentos, los tres tipos de rutas que puede haber en los TTRPs: una ruta de vehículo pura, una ruta de vehículo completa con una sub-ruta, y una ruta de camión pura. La representación gráfica de la solución obtenida es muy similar a la vista en la Figura 2.6; y las variables de decisión involucradas en dicha solución se pueden consultar en la Tabla 4.4.

Ruta	Variables de decisión		
	Relativas a las rutas x, y	Relativas a los compartimentos	
		U, V	ZT, ZL
<u>Ruta 1</u> Camión 1 Tráiler 1	$x_{01}^{11} = 1$ $x_{12}^{11} = 1$ $x_{20}^{11} = 1$	$V_{111}^1 = 1$	$ZL_{111}^1 = 1$
		$V_{113}^1 = 1$	$ZL_{113}^1 = 1$
		$U_{122}^1 = 1$	$ZT_{122}^1 = 0.625$
		$U_{211}^1 = 1$	$ZT_{211}^1 = 1$
		$V_{222}^1 = 1$	$ZL_{222}^1 = 1$
<u>Ruta 2</u> Camión 2 Tráiler 2	$x_{04}^{22} = 1$ $x_{43}^{22} = 1$ $x_{30}^{22} = 1$ $y_{45}^{246} = 1$ $y_{54}^{246} = 1$	$V_{313}^2 = 1$	$ZL_{313}^2 = 1$
		$V_{411}^2 = 1$	$ZL_{411}^2 = 1$
		$V_{422}^2 = 1$	$ZL_{422}^2 = 1$
		$V_{424}^2 = 1$	$ZL_{424}^2 = 0.1$
		$U_{511}^2 = 1$	$ZT_{511}^2 = 1$
		$U_{522}^2 = 1$	$ZT_{522}^2 = 0.8$
<u>Ruta 3</u> Camión 3	$y_{06}^{301} = 1$ $y_{67}^{301} = 1$ $y_{70}^{301} = 1$	$U_{621}^3 = 1$	$ZT_{621}^3 = 1$
		$U_{712}^3 = 1$	$ZT_{712}^3 = 0.4$
		$U_{713}^3 = 1$	$ZT_{713}^3 = 1$

Tabla 4.4: Variables de decisión en la solución al ejemplo.

4.3.2. Resolución de un problema real en AMPL

El problema expuesto en la Subsección 2.2.3 es una simplificación de la situación real en la que se encuentra dicha cooperativa agrícola. Si bien es cierto que las distancias entre los vértices (los 10 socios y el depósito) son las mismas que las allí presentadas, no lo son las demandas. Lo que sucede en la realidad es que cada uno de los clientes solicita diferentes tipos de pienso, en ocasiones porque disponen de distintas especies de animales. Como es lógico, la mercancía que no es del mismo tipo no se puede mezclar. De tal modo, es necesario contar con camiones y tráileres compartimentados. En las Tablas 4.5 y 4.6 presentamos la flota de la que dispone la compañía, donde la capacidad total de los vehículos es la misma que teníamos anteriormente. Estos datos los hemos incluido en el archivo de AMPL `MC-TTRP_piensos10.dat`, presente en el Apéndice B. Ahí también podemos ver las demandas de los clientes, según los 4 tipos de pienso que distribuye la cooperativa, que son:

$$d = \begin{pmatrix} 1000 & 0 & 0 & 2300 \\ 4000 & 0 & 0 & 2041 \\ 1959 & 0 & 4000 & 0 \\ 0 & 951 & 2000 & 0 \\ 0 & 3500 & 1385 & 0 \\ 0 & 3003 & 0 & 0 \\ 516 & 0 & 0 & 2500 \\ 978 & 0 & 3500 & 0 \\ 2000 & 0 & 2513 & 900 \\ 0 & 3490 & 0 & 0 \end{pmatrix}$$

Nótese que además de no poder mezclar en una misma tolva diferentes tipos de pienso (algo que parece evidente), tampoco se puede utilizar un mismo compartimento para abastecer a dos socios distintos. En el modelo del MC-TTRP formulado en la Sección 4.2, se incluyen estas restricciones como (4.43) y (4.44), las cuales se consideran o no atendiendo al tipo de problema que queramos resolver: en nuestro caso, la cooperativa no presenta la tecnología suficiente como para realizar mediciones y decidir cuándo *parar* de abastecer a un cliente y *guardar lo que sobra* para otro.

Hemos resuelto este problema con Gurobi (en NEOS) y, tras un tiempo de ejecución de 21219.9 segundos (5.89 horas), obtuvimos la solución presente en la Tabla 4.7. Ilustramos las rutas óptimas en la Figura 4.2.

Camión (nº de tolvas)	Tolva	Capacidad
Camiones 1 y 2 (8 tolvas)	1, 2, 7	1000
	3	2000
	4	600
	5	2500
	6	3000
	8	4200
Camión 3 (6 tolvas)	1	2000
	2	1000
	3	1500
	4, 5, 6	3500

Tráiler (nº de tolvas)	Tolva	Capacidad
Tráiler 1 (6 tolvas)	1	6500
	2	7000
	3	3500
	4	2000
	5	4300
	6	2000
Tráiler 2 (6 tolvas)	1, 3, 6	2500
	2	7000
	4	2000
	5	4000

Tabla 4.5: Capacidad de cada tolva de camión.

Tabla 4.6: Capacidad de cada tolva de tráiler.

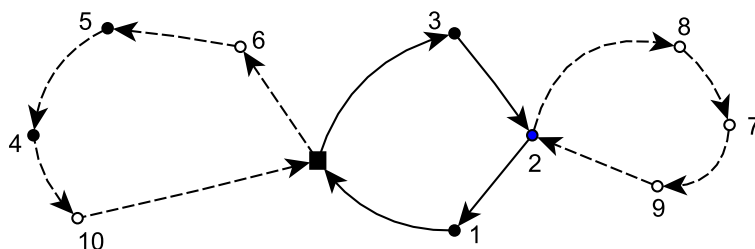


Figura 4.2: Solución del ejemplo real de los piensos (MC-TTRP).

Ruta	Variables de decisión				
	Relativas a las rutas x,y	Relativas a los compartimentos			
		U	ZT	V	ZL
<u>Ruta 1</u> Camión 1 Tráiler 1		$U_{714}^1 = 1$	$ZT_{714}^1 = 0.86$	$V_{116}^1 = 1$	$ZL_{116}^1 = 0.5$
	$x_{03}^{11} = 1$	$U_{741}^1 = 1$	$ZT_{741}^1 = 1$	$V_{141}^1 = 1$	$ZL_{141}^1 = 0.353846$
	$x_{32}^{11} = 1$	$U_{745}^1 = 1$	$ZT_{745}^1 = 0.6$	$V_{212}^1 = 1$	$ZL_{212}^1 = 0.571428$
	$x_{21}^{11} = 1$				
	$x_{10}^{11} = 1$	$U_{812}^1 = 1$	$ZT_{812}^1 = 0.978$	$V_{243}^1 = 1$	$ZL_{243}^1 = 0.583143$
	$y_{28}^{125} = 1$	$U_{838}^1 = 1$	$ZT_{838}^1 = 0.8333$	$V_{314}^1 = 1$	$ZL_{314}^1 = 0.9795$
	$y_{87}^{125} = 1$				
	$y_{79}^{125} = 1$	$U_{913}^1 = 1$	$ZT_{913}^1 = 1$	$V_{335}^1 = 1$	$ZL_{335}^1 = 0.930233$
	$y_{92}^{125} = 1$	$U_{936}^1 = 1$	$ZT_{936}^1 = 0.83766$		
	$U_{947}^1 = 1$	$ZT_{947}^1 = 0.9$			
<u>Ruta 2</u> Camión 3		$U_{422}^3 = 1$	$ZT_{422}^3 = 0.951$		
	$y_{06}^{302} = 1$	$U_{431}^3 = 1$	$ZT_{431}^3 = 1$		
	$y_{65}^{302} = 1$	$U_{526}^3 = 1$	$ZT_{526}^3 = 1$		
	$y_{54}^{302} = 1$				
	$y_{4,10}^{302} = 1$	$U_{533}^3 = 1$	$ZT_{533}^3 = 0.92333$		
	$y_{10,0}^{302} = 1$	$U_{625}^3 = 1$	$ZT_{625}^3 = 0.858$		
		$U_{10,2,4}^3 = 1$	$ZT_{10,2,4}^3 = 0.99714$		

Tabla 4.7: Variables de decisión en la solución al ejemplo real.

Nos hemos planteado estudiar el efecto en el tiempo de computación del tipo de flota (heterogénea o no). Se podría pensar que si en lugar de utilizar vehículos con diferentes capacidades contásemos con una flota homogénea (en cuanto a capacidad total se refiere),

el resultado se obtendría más rápidamente. Sin embargo, podemos ver que eso no tiene por qué ser así.

Supongamos que la compañía agrícola dispone de tres camiones, todos con capacidad 15300, y dos tráileres de capacidad 20000, tal que la configuración de sus tolvas es la presente en las Tablas 4.8 y 4.9. El archivo de datos correspondiente en AMPL para esta situación es `MC-TTRP_piensos10_2.dat`. Si lanzamos este problema en NEOS, no obtenemos solución alguna, pues se excede el tiempo máximo permitido por el servidor.

Por lo tanto, fue necesaria la utilización de una licencia académica tanto de AMPL como del solver Gurobi para conseguir resultados. Mostramos las variables de decisión obtenidas para esta modificación en la Tabla 4.10, e ilustramos las rutas en la Figura 4.3. El tiempo de ejecución para este problema se redujo notablemente al utilizar la licencia, bajando a 6914.64 segundos (1.92 horas).

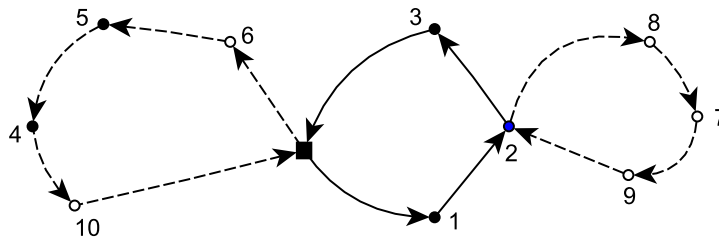


Figura 4.3: Solución del ejemplo real de los piensos (MC-TTRP).

El hecho de que NEOS no proporcione resultados con estos datos, aún siendo aparentemente más sencillos, se puede deber a que ahora, si bien los tráileres tienen la misma capacidad, se ha producido una reducción considerable de la misma con respecto al caso anterior. Por lo tanto, existe una mayor dificultad a la hora de encontrar la configuración óptima de carga de los vehículos. Nótese que en el escenario arriba expuesto, donde los tráileres tenían capacidades de 25300 y 20500, se elegía el primero para la CVR (siendo el segundo también factible) aún quedando cargado en una menor proporción.

Camión (n° de tolvas)	Tolva	Capacidad
Camiones 1 y 2 (8 tolvas)	1, 2, 7	1000
	3	2000
	4	600
	5	2500
	6	3000
	8	4200
Camión 3 (6 tolvas)	1	2000
	2	1000
	3	1800
	4, 5, 6	3500

Tabla 4.8: Capacidad de cada tolva de camión.

Tráiler (n° de tolvas)	Tolva	Capacidad
Tráileres 1 y 2 (6 tolvas)	1	2500
	2	7000
	3	2500
	4	2000
	5	4000
	6	2000

Tabla 4.9: Capacidad de cada tolva de tráiler.

Ruta	Variables de decisión				
	Relativas a las rutas x,y	Relativas a los compartimentos			
		U	ZT	V	ZL
<u>Ruta 1</u> Camión 1 Tráiler 2		$U_{714}^1 = 1$	$ZT_{714}^1 = 0.86$	$V_{116}^2 = 1$	$ZL_{116}^2 = 0.5$
	$x_{01}^{12} = 1$	$U_{745}^1 = 1$	$ZT_{745}^1 = 0.6$	$V_{141}^2 = 1$	$ZL_{141}^2 = 0.92$
	$x_{12}^{12} = 1$	$U_{747}^1 = 1$	$ZT_{747}^1 = 1$	$V_{212}^2 = 1$	$ZL_{212}^2 = 0.571429$
	$x_{23}^{12} = 1$				
	$x_{30}^{12} = 1$	$U_{812}^1 = 1$	$ZT_{812}^1 = 0.978$	$V_{243}^2 = 1$	$ZL_{243}^2 = 0.8164$
	$y_{28}^{1,2,10} = 1$	$U_{838}^1 = 1$	$ZT_{838}^1 = 0.8333$	$V_{314}^2 = 1$	$ZL_{314}^2 = 0.9795$
	$y_{87}^{1,2,10} = 1$				
	$y_{79}^{1,2,10} = 1$	$U_{913}^1 = 1$	$ZT_{913}^1 = 1$	$V_{335}^2 = 1$	$ZL_{335}^2 = 1$
	$y_{92}^{1,2,10} = 1$	$U_{936}^1 = 1$	$ZT_{936}^1 = 0.83766$		
	$U_{941}^1 = 1$	$ZT_{941}^1 = 0.9$			
<u>Ruta 2</u> Camión 3		$U_{422}^3 = 1$	$ZT_{422}^3 = 0.951$		
	$y_{06}^{309} = 1$	$U_{431}^3 = 1$	$ZT_{431}^3 = 1$		
	$y_{65}^{309} = 1$	$U_{525}^3 = 1$	$ZT_{525}^3 = 1$		
	$y_{54}^{309} = 1$				
	$y_{4,10}^{309} = 1$	$U_{533}^3 = 1$	$ZT_{533}^3 = 0.769444$		
	$y_{10,0}^{309} = 1$	$U_{624}^3 = 1$	$ZT_{624}^3 = 0.858$		
		$U_{10,2,6}^3 = 1$	$ZT_{10,2,6}^3 = 0.99714$		

Tabla 4.10: Variables de decisión en la solución al ejemplo real.

Ahora bien, reajustando las capacidades de algunas de las tolvas, hemos probado a resolver este problema cuando añadimos un nuevo cliente de tipo v.c. Para facilitar la lectura de este trabajo y evitar ser repetitivos, omitimos escribir aquí los nuevos datos, que

pueden ser consultados en el Apéndice B (MC-TTRP_piensos11.dat). Claramente, para resolverlo necesitamos hacer uso de la licencia mencionada con anterioridad. Al contar con un nodo más, el número de variables involucradas aumenta considerablemente y esto hace que el tiempo de ejecución sea de 157387 segundos (43.7 horas).

Las cooperativas agrícolas a menudo necesitan abastecer a una gran cantidad de socios, y no resulta viable tardar casi 2 días para resolver un problema con 11 clientes. La cooperativa que ha motivado nuestro estudio puede atender cada día del orden de 20 clientes. Además, la aparición de incidencias podría requerir la organización de las rutas de un día para otro. Esto motiva a plantearse como trabajo futuro la adaptación de la heurística vista en el Capítulo 3 al caso de vehículos multicompartimentados.

Capítulo 5

Conclusiones y trabajo futuro

Como hemos visto a lo largo de este trabajo, el análisis de los problemas de rutas proporciona información vital para diseñar, de manera eficiente, modelos que satisfagan las demandas de los clientes. En particular, se ha realizado una revisión exhaustiva de un campo de la Investigación Operativa con un gran interés metodológico y aplicado: el TTRP.

Uno de los resultados clave de este estudio ha sido el diseño de un nuevo algoritmo heurístico, el CW_TTRP. Este método se presenta como una adaptación de la heurística constructiva Clarke & Wright, una de las más populares para obtener soluciones a distintos VRPs.

Todos los modelos expuestos en este trabajo han sido implementados en AMPL y resueltos con Gurobi. Además, hemos programado en R el algoritmo heurístico anteriormente mencionado. Tal y como hemos podido comprobar en la Sección 3.3, CW_TTRP presenta un comportamiento satisfactorio en la práctica, reduciendo cuantiosamente los tiempos computacionales en comparación con Gurobi.

Asimismo, se ha propuesto la modelización de un problema general no presente en la literatura, el MC-TTRP. Esta variante incorpora vehículos divididos en compartimentos y surge como respuesta a situaciones habituales en las que los clientes demandan distintos tipos de mercancía.

Todo lo anterior ha sido profusamente ilustrado. Para validar estos modelos y algoritmos se han utilizado varios ejemplos propios y un problema real cercano, relativo al reparto de pienso por parte de una cooperativa agrícola gallega.

Aún así, quedan diversos problemas abiertos, que fueron surgiendo durante el desarrollo del trabajo. Algunas líneas de investigación futuras son:

- Probar el algoritmo heurístico CW_TTRP propuesto en la Sección 3.2 con los *benchmarks* de Chao (2002) y Lin et al. (2010) y comparar los resultados con otras heurísticas existentes.
- Proponer un algoritmo de mejora o adaptar una metaheurística para complementar a CW_TTRP.
- Extender nuestra propuesta heurística al caso de vehículos multicompartimentados, tomando como referencia Muyldermans y Pang (2010) con el fin de resolver el modelo propuesto en el Capítulo 4.
- Chequear de forma exhaustiva nuestros algoritmos frente a solvers del estado del arte (principalmente Gurobi).
- Crear un paquete de R que recopile todo lo anterior.
- Estudiar el funcionamiento de los denominados algoritmos evolutivos (Simon 2013) en el TTRP y MC-TTRP, ya que han dado buenos resultados en otros problemas complejos.
- Proponer modelos y soluciones de Teoría de Juegos en el caso de la cooperativa agrícola para repartir los costes de cada ruta entre los diferentes socios.

Apéndice A

VRP

En este apéndice vamos explicar de un modo más exhaustivo el VRP, comenzando por sus elementos principales y siguiendo [Toth y Vigo \(2002\)](#). Posteriormente, detallaremos alguno de los algoritmos de resolución más conocidos para el modelo clásico, que han servido como punto de partida para la creación de nuevos métodos que permitan resolver otros problemas de rutas como el TTRP.

A.1. Elementos principales del VRP

La distribución de productos implica el servicio, en un período de tiempo dado, de un conjunto de clientes mediante un conjunto de vehículos que están situados en uno o más depósitos, conducidos por un conjunto de conductores, y cuyos movimientos se llevan a cabo usando una red de rutas apropiada. En particular, la solución de un VRP consiste en determinar el conjunto de rutas, cada una de ellas recorrida por un único vehículo, que comienzan y terminan en el mismo depósito, de modo que todos los requerimientos de los clientes se cumplen, todas las restricciones operativas se satisfacen, y el coste total de transporte se minimiza. A continuación vamos a describir algunas de las características más comunes de este tipo de problemas de rutas, considerando sus principales componentes (red de rutas, clientes, depósitos, vehículos y conductores), las diferentes restricciones operativas que se pueden imponer para la construcción de las rutas, y las posibles funciones objetivo a optimizar.

La **red de rutas**, usada para el transporte de productos, se describe generalmente mediante un grafo, cuyos arcos representan los caminos y cuyos vértices se corresponden con el depósito y los clientes. Los arcos pueden ser dirigidos o no dirigidos, dependiendo de si los caminos pueden ser recorridos en un solo sentido o ambos, respectivamente. Cada

arco lleva un coste asociado, que generalmente representa su distancia, y un tiempo de viaje.

En cuanto a los **clientes**, sus características conocidas son:

- Vértice del grafo en el que se sitúa dicho cliente.
- Cantidad de producto (demanda), posiblemente de distintos tipos, que debe ser entregada o recogida en dicho cliente.
- Períodos del día (ventanas de tiempo) durante los cuales el cliente puede ser atendido.
- Tiempo requerido para servir o recolectar los bienes en la localización del cliente.
- Conjunto de los vehículos disponibles que pueden ser utilizados para servir a dicho cliente.

En algunas ocasiones no es posible satisfacer completamente la demanda de cada cliente. En estos casos, se puede o bien reducir la cantidad de mercancía a entregar o recoger, o bien dejar a algunos clientes sin atender. Para lidiar con estas situaciones, se pueden asociar a los clientes distintas prioridades o penalizaciones derivadas de la falta parcial o total de servicio.

Todas las rutas empiezan y terminan en uno o más **depósitos**, localizados en los vértices del grafo. Cada depósito está caracterizado con el número y tipos de vehículos asociado con él, así como también con la máxima cantidad de mercancía que puede almacenar. En algunas aplicaciones reales, los clientes están divididos a priori entre todos los depósitos, y los vehículos deben volver al depósito asignado al final de cada ruta. En estos casos, el VRP puede descomponerse en varios problemas independientes, cada uno de ellos asociado a un depósito diferente.

El transporte de la mercancía se lleva a cabo mediante una **flota de vehículos** cuya composición y tamaño puede estar fijada o definirse de acuerdo a los requerimientos de los clientes. Algunas características conocidas de los vehículos son:

- Depósito origen del vehículo, y la posibilidad o no de finalizar el servicio en un depósito distinto.
- Capacidad del vehículo, expresada como el máximo peso, o volumen, o número de palés, que el vehículo puede cargar.
- Posible subdivisión del vehículo en compartimentos, cada uno caracterizado mediante su capacidad y los tipos de productos que puede almacenar.

- Recursos disponibles para las operaciones de carga y descarga.
- Subconjunto de arcos del grafo que pueden ser atravesados por el vehículo.
- Coste asociado a la utilización de dicho vehículo.

Los **conductores** deben satisfacer diversas restricciones relacionadas con sus contratos y regulaciones de las compañías (horario laboral, número y duraciones de los descansos durante el servicio, máximo tiempo de conducción permitido, etc). A la hora de formular un VRP, las restricciones asociadas a los conductores se integran dentro de aquellas relativas a sus correspondientes vehículos.

Las rutas deben satisfacer una serie de **restricciones operativas**, que dependen de la naturaleza de los productos transportados, de la calidad del servicio, y de las características de los clientes y vehículos. Algunas restricciones comunes son: en cada ruta, la carga del vehículo en cuestión no puede exceder su capacidad; los clientes atendidos en una ruta pueden requerir o bien solo la entrega o recogida de mercancía, o bien ambas posibilidades; los clientes solo pueden ser servidos dentro de sus ventanas de tiempo y períodos de trabajo asociados a los conductores de los vehículos que los visitan, etc.

Hay diferentes **funciones objetivo** a considerar en los problemas de rutas. Destacan, entre otras:

- Minimización del coste global del transporte, que depende de la distancia total recorrida (o del tiempo total de viaje) y de los costes fijos asociados a los vehículos utilizados (y sus correspondientes conductores).
- Minimización del número de vehículos (o conductores) requeridos para servir a todos los clientes.
- Minimización de las penalizaciones asociadas a la falta de servicio total o parcial de algunos clientes.
- Maximización de la mercancía total repartida.

En algunas aplicaciones, cada vehículo puede realizar más de una ruta en su correspondiente período de tiempo, o las rutas pueden durar más de un día. Además, a veces es necesario considerar versiones estocásticas del problema: problemas en los que, a priori, solo existe un conocimiento parcial de las demandas de los clientes, o de los costes asociados a los arcos de la red.

A continuación vamos a mostrar alguna de las formulaciones que se han hecho del VRP clásico, el CVRP. En el CVRP, el objetivo es únicamente minimizar la distancia recorrida, entregando a todos los clientes la totalidad de su demanda.

A.2. Formulaciones del CVRP

El CVRP concierne la entrega de mercancía, las demandas de los clientes son determinísticas, conocidas con antelación, y no se pueden dividir. Hay un único depósito, y los vehículos, que solo tienen asociadas restricciones de capacidad, son idénticos. El objetivo es minimizar la distancia total necesaria para servir a todos los clientes.

Sea $G = (N, A)$ un grafo completo, donde $N = \{0, \dots, n\}$ es el conjunto de vértices y A el de arcos. Los vértices $i = 1, \dots, n$ se corresponden con los clientes, mientras que el vértice 0 hace referencia al depósito.

Cada arco $(i, j) \in A$ tiene asociado un coste no negativo c_{ij} , y representa el coste de viaje para ir de i a j . En la mayor parte de las ocasiones, los vértices están asociados con puntos en el plano mediante sus dos coordenadas, y el coste c_{ij} se define como la distancia euclídea entre los dos puntos de vértices i y j .

Cada cliente i tiene una demanda no negativa d_i , y al depósito se le asigna una demanda ficticia, $d_0 = 0$. Se dispone de K vehículos idénticos, todos con capacidad Q , y se asume, para asegurar la factibilidad del problema, que $d_i \leq Q$ para todo $i = 1, \dots, n$.

Empezamos mostrando una formulación del CVRP como un problema de programación lineal entera. Esta modelización del problema de *flujo de vehículos en dos índices* usa $O(n^2)$ variables binarias x_{ij} , que toman el valor 1 si el arco $(i, j) \in N$ pertenece a la solución óptima y el valor 0 en otro caso.

$$\text{minimizar} \quad \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (\text{A.1})$$

sujeto a

$$\sum_{i \in N} x_{ij} = 1, \quad \forall j \in N \setminus \{0\}; \quad (\text{A.2})$$

$$\sum_{j \in N} x_{ij} = 1, \quad \forall i \in N \setminus \{0\}; \quad (\text{A.3})$$

$$\sum_{i \in N} x_{i0} = K, \quad (\text{A.4})$$

$$\sum_{j \in N} x_{0j} = K, \quad (\text{A.5})$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S), \quad \forall S \subseteq N \setminus \{0\}, \quad S \neq \emptyset; \quad (\text{A.6})$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in N. \quad (\text{A.7})$$

Las restricciones (A.2) y (A.3) establecen que exactamente un arco entra y sale de cada cliente, respectivamente. De forma análoga, (A.4) y (A.5) imponen los requisitos que debe satisfacer el depósito: debe haber exactamente K arcos entrantes y salientes. La restricción dada por (A.6) es la restricción de corte de la capacidad, la cual impone la conectividad de la solución y que la demanda de cada ruta no debe superar la capacidad del vehículo. De hecho, cada corte $(N \setminus S, N)$ definido por un conjunto S de clientes está atravesado por un número de arcos no menor que $r(S)$ (el mínimo número de vehículos necesarios para atender a los clientes de S). Finalmente, (A.7) define el dominio de las variables x .

Esta formulación en dos índices, si bien es válida para los modelos más sencillos, presenta carencias cuando queremos extender el VRP a alguna de sus variantes. De hecho, solo es aplicable cuando el coste de la solución se puede expresar como la suma de los costes asociados a los arcos recorridos. Además, no permite saber qué vehículo pasa por cada uno de los arcos en la solución, siendo, por tanto, inadecuada en los casos donde el coste de las rutas depende del tipo de vehículo asignado a cada una de ellas.

Con el fin de suplir estos problemas, se propone una formulación alternativa en tres índices, indicando de modo explícito qué vehículo atraviesa un determinado arco. Esta modelización, naturalmente, involucra un mayor número de variables. En concreto, se tienen $O(n^2K)$ variables binarias x_{ij}^k , que cuentan el número de veces que un vehículo k ($k = 1, \dots, K$) atraviesa el arco $(i, j) \in N$ en la solución óptima. También aparecen $O(nK)$ variables binarias y_i^k ($i \in N, k = 1, \dots, K$) cuyo valor es 1 si el cliente i es atendido por el vehículo k en la solución óptima, y 0 en caso contrario. La formulación del modelo en tres índices viene dada por:

$$\text{minimizar} \quad \sum_{i \in N} \sum_{j \in N} \sum_{k=1}^K c_{ij} x_{ij}^k \quad (\text{A.8})$$

sujeto a

$$\sum_{k=1}^K y_i^k = 1, \quad \forall i \in N \setminus \{0\}; \quad (\text{A.9})$$

$$\sum_{k=1}^K y_0^k = K, \quad (\text{A.10})$$

$$\sum_{j \in N} x_{ij}^k = \sum_{j \in N} x_{ji}^k = y_i^k, \quad \forall i \in N, \quad k = 1, \dots, K; \quad (\text{A.11})$$

$$\sum_{i \in N} d_i y_i^k \leq Q, \quad \forall k = 1, \dots, K; \quad (\text{A.12})$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ij}^k \geq y_h^k, \quad \forall S \subseteq N \setminus \{0\}, \quad h \in S, \quad k = 1, \dots, K; \quad (\text{A.13})$$

$$y_i^k \in \{0, 1\}, \quad \forall i \in N, \quad k = 1, \dots, K; \quad (\text{A.14})$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, j \in N, \quad k = 1, \dots, K. \quad (\text{A.15})$$

A.3. Métodos de resolución

Existen numerosos algoritmos de resolución para el CVRP. Los que se han revisado en este trabajo, se pueden clasificar en dos tipos: las heurísticas clásicas y las metaheurísticas. En la actualidad, la mayor parte de los procedimientos de construcción y mejora para resolver este problema pertenecen a la primera clase. Estos métodos realizan una exploración relativamente limitada del espacio de búsqueda y producen, en general, soluciones de buena calidad en un tiempo computacional razonable. Además, y dada su flexibilidad de adaptación a restricciones presentes en problemas reales, han sido introducidas en numerosos paquetes comerciales. No obstante, no garantizan la optimalidad de las soluciones encontradas. Con el fin de mejorar estos algoritmos surgen las metaheurísticas, que realizan una exploración más pormenorizada del espacio de soluciones: suelen combinar procedimientos sofisticados de búsqueda en la vecindad, esquemas de memoria y agregación de soluciones. Si bien es cierto que los resultados obtenidos por estos algoritmos son de mayor calidad (pues utilizan mecanismos que les permiten escapar de óptimos locales), el tiempo computacional es bastante más elevado.

A continuación expondremos algunas heurísticas clásicas y metaheurísticas para resolver el VRP, y que han servido como punto de partida para la creación de métodos de resolución del TTRP.

A.3.1. Heurísticas clásicas

Las heurísticas clásicas se pueden clasificar en tres categorías:

- Heurísticas de construcción: construyen de forma gradual una solución factible fijando su atención en el coste, pero no contienen una fase de mejora en sí mismas. La más conocida se debe a [Clarke y Wright \(1964\)](#).
- Heurísticas de dos fases: dividen el problema en sus dos componentes naturales; esto es, agrupación de los vértices en rutas factibles y construcción de la ruta en sí. Algunos ejemplos de estos métodos son el algoritmo de barrido y el de [Fisher y Jaikumar \(1981\)](#).
- Heurísticas de mejora: tratan de mejorar cualquier solución factible mediante el intercambio de secuencias de arcos o vértices de una misma ruta o entre rutas diferentes.

Veamos ahora cada uno de los métodos concretos mencionados anteriormente.

Algoritmo de ahorros de Clarke & Wright

El algoritmo de Clarke & Wright es la heurística más conocida para el VRP. Está basado en la noción de los ahorros. Cuando dos rutas $(0, \dots, i, 0)$ y $(0, j, \dots, 0)$ se pueden unir de modo factible en una simple ruta $(0, \dots, i, j, \dots, 0)$, se genera un ahorro en la distancia total de $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. Los pasos del algoritmo, para el cual tenemos versiones paralela y secuencial, son los siguientes:

- **Paso 1** (cálculo de ahorros). Calcular los ahorros $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ para $i, j = 1, \dots, n$ e $i \neq j$. Crear n rutas de vehículo $(0, i, 0)$ para $i = 1, \dots, n$. Ordenar los ahorros de modo no creciente.
- **Versión paralela**

Paso 2 (mejor unión factible). Comenzando desde el principio de la lista de ahorros, ejecutar lo siguiente: dado un ahorro s_{ij} , determinar si existen o no dos rutas, una conteniendo el arco $(0, j)$ y otra el $(i, 0)$, que se pueden juntar de forma factible. En caso afirmativo, combinar ambas rutas eliminando $(0, j)$ y $(i, 0)$ e introduciendo (i, j) .
- **Versión secuencial**

Paso 2 (extensión de la ruta). Considerar ordenadamente cada ruta $(0, i, \dots, j, 0)$. Determinar el primer ahorro s_{ki} o s_{lj} que se puede usar de forma factible para fusionar la ruta actual con otra que contenga el arco $(k, 0)$ o el arco $(0, l)$. Implementar dicha unión y repetir esta operación sobre la ruta generada. Si no existe ninguna

unión factible, considerar la siguiente ruta e repetir las mismas operaciones. Detener el proceso cuando ya no haya ninguna fusión factible.

Algoritmo de barrido

El algoritmo de barrido se utiliza en instancias del VRP en el plano. Inicialmente, se crean clústeres factibles rotando un segmento centrado en el depósito. Para cada uno de los clústeres, se resuelve un TSP, lo cual origina una ruta de vehículo. Algunas implementaciones del algoritmo incluyen una fase de post-optimización, en la cual se intercambian vértices de clústeres adyacentes, y se mejoran las rutas. El algoritmo de barrido se popularizó gracias a Gillett y Miller (1974). Para la implementación del método, se comienza asumiendo que cada vértice i está representado por sus coordenadas polares (θ_i, ρ_i) , donde θ_i es el ángulo y ρ_i la longitud del segmento. Se sigue asignando el valor $\theta_i^* = 0$ a un vértice arbitrario i^* y se calcula el resto de ángulos desde $(0, i^*)$. Después se ordenan los vértices en orden creciente según su θ_i .

- **Paso 1** (inicialización de la ruta). Elegir un vehículo k que no esté siendo utilizado.
- **Paso 2** (construcción de la ruta). Empezando en el vértice con menor ángulo que no pertenezca a la ruta, asignar vértices al vehículo k siempre y cuando no se exceda su capacidad o la longitud máxima de la ruta. Si quedan vértices fuera de la ruta, ir al Paso 1.
- **Paso 3** (optimización de la ruta). Optimizar cada ruta de vehículo separadamente resolviendo el correspondiente TSP (de forma exacta o aproximada).

Algoritmo de Fisher y Jaikumar

El algoritmo de Fisher y Jaikumar es también muy conocido. En lugar de usar un método geométrico para formar los clústeres, se resuelve un problema de asignación generalizado (GAP). Lo describimos a continuación:

- **Paso 1** (selección de la semilla). Elegir vértices semilla j_k de N para inicializar cada clúster k .
- **Paso 2** (asignación de los clientes a las semillas). Calcular el coste d_{ik} de asignar cada cliente i a cada clúster k como $d_{ik} = \min\{c_{0i} + c_{ij_k} + c_{j_k 0}, c_{0j_k} + c_{j_k 0} + c_{i0}\} - (c_{0j_k} - c_{j_k 0})$.
- **Paso 3** (asignación generalizada). Resolver un GAP con costes d_{ij} , pesos de clientes q_i , y capacidad del vehículo Q .
- **Paso 4** (solución del TSP). Resolver un TSP para cada uno de los clústeres obtenidos en la resolución del GAP.

En la heurística de Fisher y Jaikumar el número de rutas K está fijado a priori. Los autores propusieron un método geométrico basado en la partición del plano en K conos de acuerdo a los pesos de los clientes. Los vértices semilla son clientes ficticios localizados a lo largo de las semirrectas que bisecan los conos. Una vez se determinan los clústeres, se resuelven los TSPs de forma óptima usando una aproximación basada en la relajación de restricciones.

A.3.2. Metaheurísticas

A diferencia de las aproximaciones clásicas, las metaheurísticas permiten el empeoramiento de las soluciones e incluso la obtención de soluciones intermedias no factibles durante el proceso de búsqueda. Las metaheurísticas más conocidas para el VRP suelen encontrar mejores óptimos locales que las heurísticas explicadas anteriormente, pero también requieren un mayor tiempo computacional, tal y como ya hemos mencionado.

Vamos a explicar de modo más bien intuitivo, y sin entrar en demasiados detalles, tres tipos de metaheurísticas que han sido aplicadas al VRP: el recocido simulado (SA), el recocido determinístico (DA) y la búsqueda tabú (TS). Estos algoritmos parten de una solución inicial x_1 y se mueven en cada iteración t de x_t a x_{t+1} en la vecindad $N(x_t)$ de x_t , hasta que se satisface una cierta condición de parada. Si $f(x)$ denota el coste de x , entonces $f(x_{t+1})$ no es necesariamente menor que $f(x_t)$. En consecuencia, se deben evitar movimientos cíclicos.

Recocido simulado (SA)

Este algoritmo (Kirkpatrick et al. 1983) está inspirado en el proceso de recocido de sólidos, que utiliza un procedimiento que va disminuyendo la temperatura con lo cual se modifica la estructura del material.

En la iteración t del recocido simulado, se genera una solución x en $N(x_t)$ de forma aleatoria. Si $f(x) \leq f(x_t)$, entonces se fija x_{t+1} como x ; en otro caso:

$$x_{t+1} := \begin{cases} x & \text{con probabilidad } p_t, \\ x_t & \text{con probabilidad } 1 - p_t, \end{cases}$$

donde p_t es normalmente una función decreciente en t y en $f(x) - f(x_t)$. Se suele definir como:

$$p_t = \exp\left(-\frac{f(x) - f(x_t)}{\theta_t}\right),$$

donde θ_t denota lo que se conoce como *temperatura* en la iteración t . Normalmente, θ_t es una función decreciente en t : en la etapa inicial se fija un valor $\theta_1 > 0$ que, tras T iteraciones, se multiplica por un factor α ($0 < \alpha < 1$), de modo que la probabilidad de aceptar una solución peor decrece según aumenta el tiempo. Hay tres reglas de parada comunes: (1) que el valor f^* del correspondiente x^* no haya decrecido al menos un ϕ_1 % en, como mínimo, k_1 ciclos consecutivos de T iteraciones; (2) que el número de movimientos aceptados haya sido menor que el ϕ_2 % de T en k_2 ciclos consecutivos de T iteraciones; y (3) que hayan sido ejecutados k_3 ciclos de T iteraciones.

Una variante del recocido simulado es el **recocido determinístico** (DA). La única diferencia es que ahora se usa una regla determinística para la aceptación o no de un movimiento. Existen dos implementaciones habituales de este algoritmo: *threshold accepting* y *record-to-record travel*, introducidas por Dueck y Scheuer (1990) y Dueck (1993), respectivamente. En la iteración t de la primera implementación, se acepta la solución x_{t+1} si $f(x_{t+1}) < f(x_t) + \theta_1$, siendo θ_1 un parámetro de control, conocido como desviación. En *record-to-record travel*, sin embargo, una solución x_{t+1} será aceptada cuando $f(x_{t+1}) < \theta_2 f(x_t)$, donde θ_2 es un parámetro de control ligeramente mayor que 1.

Búsqueda tabú (TS)

En la búsqueda tabú, igual que en el recocido simulado, se analizan secuencias de soluciones. No obstante, ahora en la iteración t el siguiente movimiento se hace al mejor vecino de la solución actual, x_t , aún si esto resulta en un empeoramiento de la función objetivo. Para evitar movimientos cíclicos, se prohíbe volver a las soluciones recientemente examinadas (se crea lo que se conoce como *lista tabú*), durante un número de iteraciones (*tiempo tabú*). El mecanismo básico de la búsqueda tabú se puede mejorar mediante el uso de herramientas computacionales, tales como estrategias de diversificación o intensificación.

Existen otros algoritmos que no citamos aquí. Entre ellos, cabe citar los denominados algoritmos evolutivos por el interés que recientemente han despertado de la mano de la denominada inteligencia artificial (véase Simon 2013).

Apéndice B

Código de AMPL

TTRP_Chao.mod

```
set N ordered;
set N1 ordered;
set Truck;
set Trailer;

param n:= card(N);
set index_N ordered := {1..((2**(n))-1)};
set POW_N {k in index_N} ordered := {i in N: (k div 2**(ord(i) - 1)) mod
    2 = 1};

param n1:= card(N1);
set index_N1 ordered := {1..((2**(n1))-1)};
set POW_N1 {k in index_N1} := {i in N1: (k div 2**(ord(i) - 1)) mod 2 =
    1};

param D;
param vm;
param C {N union {0}, N union {0}};
param Qtruck;
param Qtrailer;
param d{i in N};
param s{i in N union {0}};

var X {i in N1 union {0}, j in N1 union {0}, k in Trailer} binary;
var Y {i in N union {0}, j in N union {0}, k in Truck, l in N1 union
    {0}, v in N} binary;
```

```

#Funcion objetivo:
minimize Cost: sum{i in N1 union {0}, j in N1 union {0}, k in Trailer}
    C[i,j] * X[i,j,k]
+ sum{i in N union {0}, j in N union {0}, k in Truck, l in N1 union {0},
    v in N} C[i,j] * Y[i,j,k,l,v];

#Restricciones:

subject to 1_VC_ServedOnce {j in N1}:
sum{i in N1 union {0}, k in Trailer} X[i,j,k] + sum{i in N union {0},
    k in Truck, l in N1 union {0}, v in N: l!=j} Y[i,j,k,l,v] = 1;

subject to 2_TC_SubtourOrPTR {j in N diff N1}:
sum{i in N union {0}, k in Truck, l in N1 union {0}, v in N} Y[i,j,k,l,v
    ] = 1;

subject to 3_DepotSubtourVC:
sum{i in N union {0}, k in Truck, l in N1, v in N} Y[i,0,k,l,v] = 0;

subject to 4_LeaveAtMostOnce{k in Truck, l in N1 union {0}, v in N}:
sum{j in N union {0}} Y[l,j,k,l,v] <= 1;

subject to 5_VCRoot {j in N, k in Trailer, l in N1, v in N}:
Y[l,j,k,l,v] <= sum{i in N1 union {0}} X[i,l,k];

subject to 6_SubtourExistence {i in N union {0}, j in N union {0}, k in
    Trailer, l in N1 union {0}, v in N}:
Y[i,j,k,l,v] <= sum{p in N union {0}} Y[l,p,k,l,v];

subject to 7_NoMainTour {i in N union {0}, j in N union {0}, k in Truck
    diff Trailer, l in N1, v in N}:
Y[i,j,k,l,v]=0;

subject to 8_UniquePTR {k in Truck diff Trailer}:
sum{v in N, j in N} Y[0,j,k,0,v] <= 1;

subject to 9_VCAAtMostOneMT {k in Trailer}:
sum{j in N1} X[0,j,k] <= 1;

subject to 10_TruckCapacity {k in Truck, l in N1 union {0}, v in N}:
sum{i in N union {0}, j in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck;
subject to 11_VehicleCapacity {k in Trailer}:
sum{i in N1 union {0}, j in N1} d[j]*X[i,j,k] + sum{l in N1 union {0},
    v in N, i in N union {0}, j in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck
    + Qtrailer;

```

```

subject to 12_VehicleUtility {k in Trailer}:
sum{i in N1 union {0}, j in N1} s[j]*X[i,j,k] + sum{l in N1 union {0},
  v in N, i in N union {0}, j in N: j!=1} s[j]*Y[i,j,k,l,v] +
sum{i in N1 union {0}, j in N1 union {0}} (C[i,j]/vm)*X[i,j,k] +
  sum{l in N1 union {0}, v in N, i in N union {0}, j in N union {0}}
  (C[i,j]/vm)*Y[i,j,k,l,v] <= D-s[0];

subject to 13_TruckUtility {k in Truck diff Trailer}:
sum{v in N, i in N union {0}, j in N union {0}} (s[j]+C[i,j]/vm)*Y[i,j,k
,0,v] <= D;

subject to 14_FlowConstTR {j in N union {0}, k in Truck, l in N1 union
{0}, v in N}:
sum{i in N union {0}} Y[i,j,k,l,v] = sum {p in N union {0}}Y[j,p,k,l,v];

subject to 15_FlowConsVC {j in N1 union {0}, k in Trailer}:
sum{i in N1 union {0}} X[i,j,k] = sum{p in N1 union {0}} X[j,p,k];

subject to 16_SubtourElimination {k in Trailer, z in index_N1: card(POW_
N1[z]) >= 2}:
sum{i in POW_N1[z], j in POW_N1[z]} X[i,j,k] <= card(POW_N1[z]) - 1;

subject to 17_SubtourElimination {k in Trailer, l in N1 union {0}, v in
N, z in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,l,v] - sum{i in POW_N[z] inter
N1, j in N1 diff POW_N[z]} X[i,j,k] <= card(POW_N[z]) - 1;

subject to 18_SubtourElimination {k in Truck diff Trailer, v in N, z in
index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,0,v] <= card(POW_N[z]) - 1;

subject to ExtraRes:
sum{i in N1 union {0}, k in Trailer} X[i,i,k] + sum{j in N union {0},
  r in Truck, l in N1 union {0}, v in N} Y[j,j,r,l,v] = 0;

```

TTRP_Chao.dat

```

set N := 1 2 3 4 5 6 7;
set N1 := 1 2 3 4;
set Truck := 1 2 3;
set Trailer := 1 2;

```

```

param Qtruck := 15;
param Qtrailer:= 30;

param D := 600;
param vm := 1.3333333;

param C:
  0   1   2   3   4   5   6   7   :=
  0   0  130 125 150 190 360 250 245
  1  130  0  125 280 335 460 320 260
  2  125 125  0  210 315 440 360 345
  3  150 280 210  0  165 280 310 370
  4  190 335 315 165  0  100 205 310
  5  360 460 440 280 100  0  260 390
  6  250 320 360 310 205 260  0  145
  7  245 260 345 370 310 390 145  0 ;

param d :=
  1  25
  2  20
  3  14
  4  16
  5  14
  6  6
  7  7;

param s :=
  0  30
  1  10
  2  8
  3  5
  4  6
  5  5
  6  12
  7  9;

```

TTRP_Chao.run

```

solve;
display _varname, _var;
display _total_solve_elapsed_time;

```

TTRP_het.mod

```

set N ordered;
set N1 ordered;
set Truck;
set Trailer;

param n:= card(N);
set index_N ordered := {1..((2**(n))-1)};
set POW_N {k in index_N} ordered := {i in N: (k div 2**(ord(i) - 1)) mod
    2 = 1};

param n1:= card(N1);
set index_N1 ordered := {1..((2**(n1))-1)};
set POW_N1 {k in index_N1} := {i in N1: (k div 2**(ord(i) - 1)) mod 2 =
    1};

param D;
param vm;
param C {N union {0}, N union {0}};
param Qtruck {k in Truck};
param Qtrailer {r in Trailer};
param d{i in N};
param s{i in N union {0}};

var X {i in N1 union {0}, j in N1 union {0}, k in Trailer, r in Trailer}
    binary;
var Y {i in N union {0}, j in N union {0}, k in Truck, l in N1 union
    {0}, v in N} binary;

#Funcion objetivo:

minimize Cost: sum{i in N1 union {0}, j in N1 union {0}, k in Trailer,
    r in Trailer} C[i,j] * X[i,j,k,r]
+ sum{i in N union {0}, j in N union {0}, k in Truck, l in N1 union {0},
    v in N} C[i,j] * Y[i,j,k,l,v];

#Restricciones:

subject to 1_VC_ServedOnce {j in N1}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck, l in N1 union {0}, v in N: l!=j} Y[i,j,k,l
    ,v] = 1;

```

```

subject to 1_VC_root {j in N1, v in N}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck} Y[i,j,k,j,v] <= 2;

subject to 2_TC_SubtourOrPTR {j in N diff N1}:
sum{i in N union {0}, k in Truck, l in N1 union {0}, v in N} Y[i,j,k,l,v
    ] = 1;

subject to 3_DepotSubtourVC:
sum{i in N union {0}, k in Truck, l in N1, v in N} Y[i,0,k,l,v] = 0;

subject to 4_LeaveAtMostOnce{k in Truck, l in N1 union {0}, v in N}:
sum{j in N union {0}} Y[l,j,k,l,v] <= 1;

subject to 5_VCRoot {j in N, k in Trailer, l in N1, v in N}:
Y[l,j,k,l,v] <= sum{i in N1 union {0}, r in Trailer} X[i,l,k,r];

subject to 6_SubtourExistence {i in N union {0}, j in N union {0}, k in
    Truck, l in N1 union {0}, v in N}:
Y[i,j,k,l,v] <= sum{p in N union {0}} Y[l,p,k,l,v];

subject to 7_NoMainTour {k in Truck diff Trailer, l in N1}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,l,v] = 0;

subject to 7_TrailerNotPTR {k in Trailer}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,0,v] = 0;

subject to 8_UniquePTR {k in Truck diff Trailer}:
sum{v in N, j in N} Y[0,j,k,0,v] <= 1;

subject to 9_TruckAtMostOneMT {k in Trailer}:
sum{j in N1, r in Trailer} X[0,j,k,r] <= 1;

subject to 9_TrailerAtMostOneMT{r in Trailer}:
sum{j in N1, k in Trailer} X[0,j,k,r] <= 1;

subject to 10_TruckCapacity1 {k in Trailer, l in N1, v in N}:
sum{i in N, j in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck[k];

subject to 10_TruckCapacity2 {k in Truck diff Trailer, v in N}:
sum{i in N union {0}, j in N} d[j]*Y[i,j,k,0,v] <= Qtruck[k];

subject to 11_VehicleCapacity {k in Trailer, r in Trailer}:
sum{i in N1 union {0}, j in N1} d[j]*X[i,j,k,r] + sum{i in N, j in N, l
    in N1, v in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck[k] + Qtrailer[r];

```



```

subject to 12_VehicleUtility {k in Trailer, r in Trailer}:
sum{i in N1, j in N1 union {0}} s[i]*X[i,j,k,r] + sum{l in N1 union {0},
    v in N, i in N, j in N union {0}: i!=l} s[i]*Y[i,j,k,l,v] +
sum{i in N1 union {0}, j in N1 union {0}} (C[i,j]/vm)*X[i,j,k,r] + sum{l
    in N1 union {0}, v in N, i in N union {0}, j in N union {0}}
    (C[i,j]/vm)*Y[i,j,k,l,v] <= D-s[0];

subject to 13_TruckUtility {k in Truck diff Trailer}:
sum{v in N, i in N union {0}, j in N union {0}} (s[i]+C[i,j]/vm)*Y[i,j,k
    ,0,v] <= D;

subject to 14_FlowConstR {j in N union {0}, k in Truck, l in N1 union
    {0}, v in N}:
sum{i in N union {0}} Y[i,j,k,l,v] = sum{p in N union {0}} Y[j,p,k,l,v];

subject to 15_FlowConstVC{j in N1 union {0}, k in Trailer, r in Trailer}:
sum{i in N1 union {0}} X[i,j,k,r] = sum{p in N1 union {0}} X[j,p,k,r];

subject to 16_SubtourEliminationPVRorMT {k in Trailer, r in Trailer, z
    in index_N1: card(POW_N1[z]) >= 2}:
sum{i in POW_N1[z], j in POW_N1[z]} X[i,j,k,r] <= card(POW_N1[z]) - 1;

subject to 17_SubtourEliminationSubtour {k in Trailer, l in N1, v in N,
    z in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]}Y[i,j,k,l,v]-sum{i in POW_N[z] inter N1
    , j in N1 diff POW_N[z], r in Trailer} X[i,j,k,r]<=card(POW_N[z])-1;

subject to 18_SubtourEliminationPTR {k in Truck diff Trailer, v in N, z
    in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,0,v] <= card(POW_N[z]) - 1;

subject to ExtraRes:
sum{i in N1 union {0}, k in Trailer, t in Trailer} X[i,i,k,t] + sum{j in
    N union {0}, r in Truck, l in N1 union {0}, v in N} Y[j,j,r,l,v] =
    0;

```

TTRP_het.dat

```

set N := 1 2 3 4 5 6 7;
set N1 := 1 2 3 4;
set Truck := 1 2 3;
set Trailer := 1 2;

```

```

param Qtruck :=
1  18
2  15
3  15;

param Qtrailer :=
1  30
2  35;

param D := 600;
param vm:= 1.3333333;

param C:      0    1    2    3    4    5    6    7    :=
0      0   130  125  150  190  360  250  245
1     130    0   125  280  335  460  320  260
2     125  125    0   210  315  440  360  345
3     150  280  210    0   165  280  310  370
4     190  335  315  165    0   100  205  310
5     360  460  440  280  100    0   260  390
6     250  320  360  310  205  260    0   145
7     245  260  345  370  310  390  145    0 ;

param d      :=
1   25
2   20
3   14
4   16
5   14
6    6
7    7;

param s      :=
0   30
1   10
2    8
3    5
4    6
5    5
6   12
7    9;

```

CVRP_piensos.mod

```

set N ordered;
set N1 := {};
set Truck;
set Trailer := {};

param n:= card(N);
set index_N ordered := {1..((2**(n))-1)};
set POW_N {k in index_N} ordered := {i in N: (k div 2**(ord(i) - 1)) mod
    2 = 1};

param n1:= card(N1);
set index_N1 ordered := {1..((2**(n1))-1)};
set POW_N1 {k in index_N1} := {i in N1: (k div 2**(i - 1)) mod 2 = 1};

param D;
param vm;
param C {N union {0}, N union {0}};
param Qtruck {k in Truck};
param Qtrailer {r in Trailer};
param d{i in N};
param s{i in N union {0}};

var X {i in N1 union {0}, j in N1 union {0}, k in Trailer, r in Trailer}
    binary;
var Y {i in N union {0}, j in N union {0}, k in Truck, l in N1 union
    {0}, v in N} binary;

#Funcion objetivo:

minimize Cost: sum{i in N1 union {0}, j in N1 union {0}, k in Trailer,
    r in Trailer} C[i,j] * X[i,j,k,r]
+ sum{i in N union {0}, j in N union {0}, k in Truck, l in N1 union {0},
    v in N} C[i,j] * Y[i,j,k,l,v];

#Restricciones:

subject to 1_VC_ServedOnce {j in N1}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck, l in N1 union {0}, v in N: l!=j} Y[i,j,k,l
    ,v] = 1;

```

```

subject to 1_VC_root {j in N1, v in N}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck} Y[i,j,k,j,v] <= 2;

subject to 2_TC_SubtourOrPTR {j in N diff N1}:
sum{i in N union {0}, k in Truck, l in N1 union {0}, v in N} Y[i,j,k,l,v
    ] = 1;

subject to 3_DepotSubtourVC:
sum{i in N union {0}, k in Truck, l in N1, v in N} Y[i,0,k,l,v] = 0;

subject to 4_LeaveAtMostOnce{k in Truck, l in N1 union {0}, v in N}:
sum{j in N union {0}} Y[l,j,k,l,v] <= 1;

subject to 5_VCRoot {j in N, k in Trailer, l in N1, v in N}:
Y[l,j,k,l,v] <= sum{i in N1 union {0}, r in Trailer} X[i,l,k,r];

subject to 6_SubtourExistence {i in N union {0}, j in N union {0}, k in
    Truck, l in N1 union {0}, v in N}:
Y[i,j,k,l,v] <= sum{p in N union {0}} Y[l,p,k,l,v];

subject to 7_NoMainTour {k in Truck diff Trailer, l in N1}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,l,v] = 0;

subject to 7_TrailerNotPTR {k in Trailer}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,0,v] = 0;

subject to 8_UniquePTR {k in Truck diff Trailer}:
sum{v in N, j in N} Y[0,j,k,0,v] <= 1;

subject to 9_TruckAtMostOneMT {k in Trailer}:
sum{j in N1, r in Trailer} X[0,j,k,r] <= 1;

subject to 9_TrailerAtMostOneMT{r in Trailer}:
sum{j in N1, k in Trailer} X[0,j,k,r] <= 1;

subject to 10_TruckCapacity1 {k in Trailer, l in N1, v in N}:
sum{i in N, j in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck[k];

subject to 10_TruckCapacity2 {k in Truck diff Trailer, v in N}:
sum{i in N union {0}, j in N} d[j]*Y[i,j,k,0,v] <= Qtruck[k];

subject to 11_VehicleCapacity {k in Trailer, r in Trailer}:
sum{i in N1 union {0}, j in N1} d[j]*X[i,j,k,r] + sum{i in N, j in N,
    l in N1, v in N: j!=1} d[j]*Y[i,j,k,l,v] <= Qtruck[k] + Qtrailer[r];

```

```

subject to 12_VehicleUtility {k in Trailer, r in Trailer}:
sum{i in N1, j in N1 union {0}} s[i]*X[i,j,k,r] + sum{l in N1 union {0},
    v in N, i in N, j in N union {0}: i!=l} s[i]*Y[i,j,k,l,v] +
sum{i in N1 union {0}, j in N1 union {0}} (C[i,j]/vm)*X[i,j,k,r] +
    sum{l in N1 union {0}, v in N, i in N union {0}, j in N union {0}}
    (C[i,j]/vm)*Y[i,j,k,l,v] <= D-s[0];

subject to 13_TruckUtility {k in Truck diff Trailer}:
sum{v in N, i in N union {0}, j in N union {0}} (s[i]+C[i,j]/vm)*Y[i,j,k
    ,0,v] <= D;

subject to 14_FlowConstR {j in N union {0}, k in Truck, l in N1 union
    {0}, v in N}:
sum{i in N union {0}} Y[i,j,k,l,v] = sum{p in N union {0}} Y[j,p,k,l,v];

subject to 15_FlowConstVC{j in N1 union {0}, k in Trailer, r in Trailer}:
sum{i in N1 union {0}} X[i,j,k,r] = sum{p in N1 union {0}} X[j,p,k,r];

subject to 16_SubtourEliminationPVRorMT {k in Trailer, r in Trailer, z
    in index_N1: card(POW_N1[z]) >= 2}:
sum{i in POW_N1[z], j in POW_N1[z]} X[i,j,k,r] <= card(POW_N1[z]) - 1;

subject to 17_SubtourEliminationSubtour {k in Trailer, l in N1, v in N,
    z in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,l,v]-sum{i in POW_N[z] inter N1
    , j in N1 diff POW_N[z], r in Trailer} X[i,j,k,r]<=card(POW_N[z])-1;

subject to 18_SubtourEliminationPTR {k in Truck diff Trailer, v in N, z
    in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,0,v] <= card(POW_N[z]) - 1;

subject to ExtraRes:
sum{i in N1 union {0}, k in Trailer, t in Trailer} X[i,i,k,t] + sum{j in
    N union {0}, r in Truck, l in N1 union {0}, v in N} Y[j,j,r,l,v]=0;

```

CVRP_piensos10.dat

```

set N := 1 2 3 4 5 6 7 8 9 10;
set Truck := 1 2 3;
param Qtruck :=
1 15300
2 15300
3 15000;

```

```

param D := 480;
param vm:= 1;

param C:
  0   1   2   3   4   5   6   7   8   9   10 :=
  0   0  21  20  17  65  63  60  19  22  24  60
  1  21  0   4   6  60  58  55  15  18  20  55
  2  20  4   0   4  59  56  53  13   8  12  53
  3  17  6   4   0  57  54  52  11  13  16  52
  4  65  60  59  57   0   3   7  66  69  71   6
  5  63  58  56  54   3   0   4  64  66  69   3
  6  60  55  53  52   7   4   0  61  64  66   2
  7  19  15  13  11  66  64  61   0   3   5  61
  8  22  18   8  13  69  66  64   3   0   7  64
  9  24  20  12  16  71  69  66   5   7   0  66
 10  60  55  53  52   6   3   2  61  64  66   0;

param d :=
  1  3300
  2  6041
  3  5959
  4  2951
  5  4885
  6  3003
  7  3016
  8  4478
  9  5413
 10  3490;

param s :=
  0  0
  1  0
  2  0
  3  0
  4  0
  5  0
  6  0
  7  0
  8  0
  9  0
 10  0;

```

TTRP_piensos10.dat

```

set N := 1 2 3 4 5 6 7 8 9 10;
set N1 := 1 2 3 4 5;

set Truck := 1 2 3;
set Trailer := 1 2;

param Qtruck :=
1 15300
2 15300
3 15000;
param Qtrailer:=
1 25300
2 20500;

param D := 480;
param vm:= 1;

param C:      0    1    2    3    4    5    6    7    8    9    10    :=
  0    0    21   20   17   65   63   60   19   22   24   60
  1    21   0    4    6    60   58   55   15   18   20   55
  2    20   4    0    4    59   56   53   13    8   12   53
  3    17   6    4    0    57   54   52   11   13   16   52
  4    65   60   59   57    0    3    7   66   69   71    6
  5    63   58   56   54    3    0    4   64   66   69    3
  6    60   55   53   52    7    4    0   61   64   66    2
  7    19   15   13   11   66   64   61    0    3    5   61
  8    22   18    8   13   69   66   64    3    0    7   64
  9    24   20   12   16   71   69   66    5    7    0   66
 10    60   55   53   52    6    3    2   61   64   66    0;

param d      :=
  1    3300
  2    6041
  3    5959
  4    2951
  5    4885
  6    3003
  7    3016
  8    4478
  9    5413
 10    3490;

```

```

param s :=
    0  0
    1  0
    2  0
    3  0
    4  0
    5  0
    6  0
    7  0
    8  0
    9  0
    10 0;

```

MC-TTRP.mod

```

set N ordered;
set N1 ordered;
set Truck;
set Trailer;

set F;
set Ht{k in Truck} ordered;
set Hl{r in Trailer} ordered;

param n:= card(N);
set index_N ordered := {1..((2**(n))-1)};
set POW_N {k in index_N} ordered := {i in N: (k div 2**(ord(i) - 1)) mod
    2 = 1};

param n1:= card(N1);
set index_N1 ordered := {1..((2**(n1))-1)};
set POW_N1 {k in index_N1} ordered := {i in N1: (k div 2**(ord(i) - 1))
    mod 2 = 1};

param D;
param vm;
param C {N union {0}, N union {0}};
param Qtruck {k in Truck};
param Qtrailer {r in Trailer};
param d{i in N, f in F};
param s{i in N union {0}};

```



```

param Qt {k in Truck, ht in Ht[k]};
param Ql {r in Trailer, hl in Hl[r]};
param Lt;
param Ll;

var X {i in N1 union {0}, j in N1 union {0}, k in Trailer, r in Trailer}
    binary;
var Y {i in N union {0}, j in N union {0}, k in Truck, l in N1 union
    {0}, v in N} binary;
var ZT {i in N, f in F, k in Truck, ht in Ht[k]} >= 0, <= 1;
var ZL {i in N1, f in F, r in Trailer, hl in Hl[r]} >= 0, <= 1;
var U {i in N, f in F, k in Truck, ht in Ht[k]} binary;
var V {i in N1, f in F, r in Trailer, hl in Hl[r]} binary;

#Funcion objetivo:
minimize Cost: sum{i in N1 union {0}, j in N1 union {0}, k in Trailer, r
    in Trailer} C[i,j] * X[i,j,k,r]
+ sum{i in N union {0}, j in N union {0}, k in Truck, l in N1 union {0},
    v in N} C[i,j] * Y[i,j,k,l,v];

#Restricciones:
subject to 1_VC_ServedOnce {j in N1}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck, l in N1 union {0}, v in N: l!=j} Y[i,j,k,l
    ,v] = 1;

subject to 1_VC_root {j in N1, v in N}:
sum{i in N1 union {0}, k in Trailer, r in Trailer} X[i,j,k,r] + sum{i in
    N union {0}, k in Truck} Y[i,j,k,j,v] <= 2;

subject to 2_TC_SubtourOrPTR {j in N diff N1}:
sum{i in N union {0}, k in Truck, l in N1 union {0}, v in N} Y[i,j,k,l,v
    ] = 1;

subject to 3_DepotSubtourVC:
sum{i in N union {0}, k in Truck, l in N1, v in N} Y[i,0,k,l,v] = 0;

subject to 4_LeaveAtMostOnce{k in Truck, l in N1 union {0}, v in N}:
sum{j in N union {0}} Y[l,j,k,l,v] <= 1;

subject to 5_VCRoot {j in N, k in Trailer, l in N1, v in N}:
Y[l,j,k,l,v] <= sum{i in N1 union {0}, r in Trailer} X[i,l,k,r];

subject to 6_SubtourExistence {i in N union {0}, j in N union {0}, k in
    Truck, l in N1 union {0}, v in N}:

```

```

Y[i,j,k,l,v] <= sum{p in N union{0}} Y[l,p,k,l,v];

subject to 7_NoMainTour {k in Truck diff Trailer, l in N1}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,l,v] = 0;

subject to 7_TrailerNotPTR {k in Trailer}:
sum{i in N union {0}, j in N union {0}, v in N} Y[i,j,k,0,v] = 0;

subject to 8_UniquePTR {k in Truck diff Trailer}:
sum{v in N, j in N} Y[0,j,k,0,v] <= 1;

subject to 9_VCAAtMostOneMT {k in Trailer}:
sum{j in N1, r in Trailer} X[0,j,k,r] <= 1;

subject to 9_TrailerAtMostOneMT{r in Trailer}:
sum{j in N1, k in Trailer} X[0,j,k,r] <= 1;

subject to 10_TruckCapacity1 {k in Trailer, l in N1, v in N}:
sum{i in N, j in N: j!=1} (sum{f in F}d[j,f])*Y[i,j,k,l,v] <= Qtruck[k];

subject to 10_TruckCapacity2 {k in Truck diff Trailer, v in N}:
sum{i in N union {0}, j in N}(sum{f in F}d[j,f])*Y[i,j,k,0,v] <= Qtruck[k];

subject to 11_VehicleCapacity {k in Trailer, r in Trailer}:
sum{i in N1 union {0}, j in N1} (sum{f in F}d[j,f])*X[i,j,k,r] + sum{i
    in N, j in N, l in N1, v in N: j!=1}(sum{f in F}d[j,f])*Y[i,j,k,l,v]
    <= Qtruck[k] + Qtrailer[r];

subject to 12_VehicleUtility {k in Trailer, r in Trailer}:
sum{i in N1, j in N1 union {0}} s[i]*X[i,j,k,r] + sum{l in N1 union {0},
    v in N, i in N, j in N union {0}: i!=1} s[i]*Y[i,j,k,l,v] +
sum{i in N1 union {0}, j in N1 union {0}} (C[i,j]/vm)*X[i,j,k,r] +
sum{l in N1 union {0}, v in N, i in N union {0}, j in N union {0}}
(C[i,j]/vm)*Y[i,j,k,l,v] <= D-s[0];

subject to 13_TruckUtility {k in Truck diff Trailer}:
sum{v in N, i in N union {0}, j in N union {0}} (s[i]+C[i,j]/vm)*Y[i,j,k
    ,0,v] <= D;

subject to 14_FlowConstR {j in N union {0}, k in Truck, l in N1 union
    {0}, v in N}:
sum{i in N union {0}} Y[i,j,k,l,v] = sum{p in N union {0}} Y[j,p,k,l,v];

subject to 15_FlowConsVC{j in N1 union {0}, k in Trailer, r in Trailer}:
sum{i in N1 union {0}} X[i,j,k,r] = sum{p in N1 union {0}} X[j,p,k,r];

```

```

subject to 16_SubtourEliminationPVRorMT {k in Trailer, r in Trailer,
    z in index_N1: card(POW_N1[z]) >= 2}:
sum{i in POW_N1[z], j in POW_N1[z]} X[i,j,k,r] <= card(POW_N1[z]) - 1;

subject to 17_SubtourEliminationSubtour {k in Trailer, l in N1, v in N,
    z in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,l,v] - sum{i in POW_N[z] inter N1
    , j in N1 diff POW_N[z], r in Trailer} X[i,j,k,r] <= card(POW_N[z]) - 1;

subject to 18_SubtourEliminationPTR {k in Truck diff Trailer, v in N,
    z in index_N: card(POW_N[z]) >= 2}:
sum{i in POW_N[z], j in POW_N[z]} Y[i,j,k,0,v] <= card(POW_N[z]) - 1;

subject to ExtraRes{i in N1 union {0}, j in N union {0}, k in Trailer,
    r in Truck, t in Trailer, l in N1 union {0}, v in N}:
X[i,i,k,t] + Y[j,j,r,l,v] = 0;

subject to comp1{k in Trailer, j in N1}:
1/card(Ht[k])*(sum{f in F, ht in Ht[k]} ZT[j,f,k,ht]) <=
sum{i in N1 union {0}, r in Trailer} X[i,j,k,r] + sum{i in N union {0},
    l in N1, v in N} Y[i,j,k,l,v];

subject to comp1_2{k in Truck diff Trailer, j in N1}:
1/card(Ht[k])*(sum{f in F, ht in Ht[k]} ZT[j,f,k,ht]) <= sum{i in N
    union {0}, v in N} Y[i,j,k,0,v];

subject to comp1_3 {j in N1, r in Trailer}:
1/card(Hl[r])*(sum{f in F, hl in Hl[r]} ZL[j,f,r,hl]) <= sum{i in N1
    union {0}, k in Trailer} X[i,j,k,r];

subject to comp2{k in Truck, j in N diff N1}:
1/card(Ht[k])*(sum{f in F, ht in Ht[k]} ZT[j,f,k,ht]) <= sum{i in N
    union {0}, l in N1 union {0}, v in N} Y[i,j,k,l,v];

subject to comp3{k in Trailer, r in Trailer, j in N1}:
sum{i in N1 union {0}} X[i,j,k,r] <=
sum{f in F, ht in Ht[k]} Qt[k,ht]*ZT[j,f,k,ht] + sum{f in F, hl in Hl[r]}
    Ql[r,hl]*ZL[j,f,r,hl];

subject to comp3_2{k in Truck diff Trailer, j in N1}:
sum{i in N union {0}, v in N} Y[i,j,k,0,v] <= sum{f in F, ht in Ht[k]}
    Qt[k,ht]*ZT[j,f,k,ht];
subject to comp3_3{k in Trailer, j in N1}:

```

```

sum{i in N union {0}, l in N1, v in N: j!=1} Y[i,j,k,l,v] <= sum{f in F,
  ht in Ht[k]} Qt[k,ht]*ZT[j,f,k,ht];

subject to comp4{j in N diff N1, k in Truck}:
sum{i in N union {0}, l in N1 union {0}, v in N}Y[i,j,k,l,v] <= sum{f in
  F, ht in Ht[k]} Qt[k,ht]*ZT[j,f,k,ht];

subject to comp5{j in N1, f in F}:
sum{r in Trailer, hl in Hl[r]} Ql[r,hl]*ZL[j,f,r,hl] + sum{k in Truck,
  ht in Ht[k]} Qt[k,ht]*ZT[j,f,k,ht] = d[j,f];

subject to comp6{j in N diff N1, f in F}:
sum{k in Truck, ht in Ht[k]} Qt[k,ht]*ZT[j,f,k,ht] = d[j,f];

subject to comp8{k in Truck}:
sum{i in N, f in F, ht in Ht[k]} Qt[k,ht]*ZT[i,f,k,ht] <= Lt;

subject to comp9{r in Trailer}:
sum{i in N1, f in F, hl in Hl[r]} Ql[r,hl]*ZL[i,f,r,hl] <= Ll;

subject to comp16{k in Truck, ht in Ht[k]}:
sum{i in N, f in F} ZT[i,f,k,ht] <= 1;

subject to comp17{r in Trailer, hl in Hl[r]}:
sum{i in N1, f in F} ZL[i,f,r,hl] <= 1;

subject to comp18{i in N, f in F, k in Truck, ht in Ht[k]}:
ZT[i,f,k,ht] - U[i,f,k,ht] <= 0;

subject to comp19{i in N1, f in F, r in Trailer, hl in Hl[r]}:
ZL[i,f,r,hl] - V[i,f,r,hl] <= 0;

subject to comp20{i in N, j in N, f1 in F, f2 in F, k in Truck, ht in Ht
  [k]: f1!=f2}:
U[i,f1,k,ht] + U[j,f2,k,ht] <= 1;

subject to comp21{i in N1, j in N1, f1 in F, f2 in F, r in Trailer, hl
  in Hl[r]: f1!=f2}:
V[i,f1,r,hl] + V[j,f2,r,hl] <= 1;

subject to comp24{i in N, j in N, f in F, k in Truck, ht in Ht[k]:i!=j}:
U[i,f,k,ht] + U[j,f,k,ht] <= 1;

subject to comp25{i in N1,j in N1,f in F,r in Trailer,hl in Hl[r]:i!=j}:
V[i,f,r,hl] + V[j,f,r,hl] <= 1;

```

MC-TTRP.dat

```

set N := 1 2 3 4 5 6 7;
set N1 := 1 2 3 4;

set Truck := 1 2 3;
set Trailer := 1 2;

param Qtruck :=
1 18
2 15
3 15;

param Qtrailer :=
1 30
2 35;

param Lt := 60;
param Ll := 60;

set F:= 1 2;

set Ht[1] := 1 2;
set Ht[2] := 1 2;
set Ht[3] := 1 2 3;

set Hl[1] := 1 2 3;
set Hl[2] := 1 2 3 4;

param Qt      :=
    [1,*]  1  10    2  8
    [2,*]  1  10    2  5
    [3,*]  1   5    2  5    3  5;

param Ql:=
    [1,*]  1  10    2  10    3  10
    [2,*]  1  10    2   5    3  10    4  10;

param D := 600;
param vm := 1.3333333;

```

```

param C:    0    1    2    3    4    5    6    7    :=
0    0    130  125  150  190  360  250  245
1    130   0   125  280  335  460  320  260
2    125  125   0   210  315  440  360  345
3    150  280  210   0   165  280  310  370
4    190  335  315  165   0   100  205  310
5    360  460  440  280  100   0   260  390
6    250  320  360  310  205  260   0   145
7    245  260  345  370  310  390  145   0;

param d:    1    2    :=
1    20    5
2    10   10
3    10    0
4    10    6
5    10    4
6     0    5
7     7    0;

param s      :=
0    30
1    10
2     8
3     5
4     6
5     5
6    12
7     9;

```

MC-TTRP_piensos10.dat

```

set N := 1 2 3 4 5 6 7 8 9 10;
set N1 := 1 2 3 4 5;

set Truck := 1 2 3;
set Trailer := 1 2;

param Qtruck :=
1 15300
2 15300
3 15000;

```

```

param Qtrailer:=
1 25300
2 20500;

param Lt := 18000;
param Ll := 30000;

set F:= 1 2 3 4;

set Ht[1] := 1 2 3 4 5 6 7 8;
set Ht[2] := 1 2 3 4 5 6 7 8;
set Ht[3] := 1 2 3 4 5 6;

set Hl[1] := 1 2 3 4 5 6;
set Hl[2] := 1 2 3 4 5 6;

param Qt:=
[1,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
[2,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
[3,*] 1 2000 2 1000 3 1500 4 3500 5 3500 6 3500;

param Ql:=
[1,*] 1 6500 2 7000 3 3500 4 2000 5 4300 6 2000
[2,*] 1 2500 2 7000 3 2500 4 2000 5 4000 6 2500;

param D := 480;

param vm := 1;

param C:
0 1 2 3 4 5 6 7 8 9 10 :=
0 0 21 20 17 65 63 60 19 22 24 60
1 21 0 4 6 60 58 55 15 18 20 55
2 20 4 0 4 59 56 53 13 8 12 53
3 17 6 4 0 57 54 52 11 13 16 52
4 65 60 59 57 0 3 7 66 69 71 6
5 63 58 56 54 3 0 4 64 66 69 3
6 60 55 53 52 7 4 0 61 64 66 2
7 19 15 13 11 66 64 61 0 3 5 61
8 22 18 8 13 69 66 64 3 0 7 64
9 24 20 12 16 71 69 66 5 7 0 66
10 60 55 53 52 6 3 2 61 64 66 0;

```

```

param d:      1      2      3      4      :=
  1  1000      0      0      2300
  2  4000      0      0      2041
  3  1959      0      4000      0
  4      0      951      2000      0
  5      0      3500      1385      0
  6      0      3003      0      0
  7      516      0      0      2500
  8      978      0      3500      0
  9      2000      0      2513      900
 10      0      3490      0      0;

param s      :=
  0  0
  1  0
  2  0
  3  0
  4  0
  5  0
  6  0
  7  0
  8  0
  9  0
 10 0;

```

MC-TTRP_piensos10_2.dat

```

set N      := 1 2 3 4 5 6 7 8 9 10;
set N1     := 1 2 3 4 5;
set Truck  := 1 2 3;
set Trailer := 1 2;

param Qtruck :=
  1 15300
  2 15300
  3 15300;
param Qtrailer :=
  1 20000
  2 20000;

param Lt   := 20000;
param Ll   := 25000;

```



```

set F:= 1 2 3 4;

set Ht[1] := 1 2 3 4 5 6 7 8;
set Ht[2] := 1 2 3 4 5 6 7 8;
set Ht[3] := 1 2 3 4 5 6;

set Hl[1] := 1 2 3 4 5 6;
set Hl[2] := 1 2 3 4 5 6;

param Qt:=
  [1,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
  [2,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
  [3,*] 1 2000 2 1000 3 1800 4 3500 5 3500 6 3500;

param Ql:=
  [1,*] 1 2500 2 7000 3 2500 4 2000 5 4000 6 2000
  [2,*] 1 2500 2 7000 3 2500 4 2000 5 4000 6 2000;

param D := 480;
param vm := 1;

param C:
  0 1 2 3 4 5 6 7 8 9 10 :=
  0 0 21 20 17 65 63 60 19 22 24 60
  1 21 0 4 6 60 58 55 15 18 20 55
  2 20 4 0 4 59 56 53 13 8 12 53
  3 17 6 4 0 57 54 52 11 13 16 52
  4 65 60 59 57 0 3 7 66 69 71 6
  5 63 58 56 54 3 0 4 64 66 69 3
  6 60 55 53 52 7 4 0 61 64 66 2
  7 19 15 13 11 66 64 61 0 3 5 61
  8 22 18 8 13 69 66 64 3 0 7 64
  9 24 20 12 16 71 69 66 5 7 0 66
  10 60 55 53 52 6 3 2 61 64 66 0;

param d:
  1 2 3 4 :=
  1 1000 0 0 2300
  2 4000 0 0 2041
  3 1959 0 4000 0
  4 0 951 2000 0
  5 0 3500 1385 0
  6 0 3003 0 0
  7 516 0 0 2500
  8 978 0 3500 0
  9 2000 0 2513 900
  10 0 3490 0 0;

```

```
param s :=
    0 0
    1 0
    2 0
    3 0
    4 0
    5 0
    6 0
    7 0
    8 0
    9 0
    10 0;
```

MC-TTRP_piensos11.dat

```
set N := 1 2 3 4 5 6 7 8 9 10 11;
set N1 := 1 2 3 4 5 11;

set Truck := 1 2 3;
set Trailer := 1 2;

param Qtruck :=
    1 15300
    2 15300
    3 15300;

param Qtrailer:=
    1 20000
    2 20000;

param Lt := 20000;
param Ll := 25000;

set F:= 1 2 3 4;

set Ht[1] := 1 2 3 4 5 6 7 8;
set Ht[2] := 1 2 3 4 5 6 7 8;
set Ht[3] := 1 2 3 4 5 6;

set Hl[1] := 1 2 3 4 5 6;
set Hl[2] := 1 2 3 4 5 6;
```

```

param Qt:=
  [1,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
  [2,*] 1 1000 2 1000 3 2000 4 600 5 2500 6 3000 7 1000 8 4200
  [3,*] 1 2000 2 1000 3 1800 4 3500 5 3500 6 3500;

param Ql:=
  [1,*] 1 2500 2 4000 3 2200 4 4000 5 2000 6 5300
  [2,*] 1 2500 2 7000 3 2500 4 2000 5 4000 6 2000;

param D := 480;

param vm := 1;

param C: 0 1 2 3 4 5 6 7 8 9 10 11 :=
  0 0 21 20 17 65 63 60 19 22 24 60 5
  1 21 0 4 6 60 58 55 15 18 20 55 26
  2 20 4 0 4 59 56 53 13 8 12 53 24
  3 17 6 4 0 57 54 52 11 13 16 52 21
  4 65 60 59 57 0 3 7 66 69 71 6 66
  5 63 58 56 54 3 0 4 64 66 69 3 64
  6 60 55 53 52 7 4 0 61 64 66 2 61
  7 19 15 13 11 66 64 61 0 3 5 61 24
  8 22 18 8 13 69 66 64 3 0 7 64 26
  9 24 20 12 16 71 69 66 5 7 0 66 29
  10 60 55 53 52 6 3 2 61 64 66 0 61
  11 5 26 24 21 66 64 61 24 26 29 61 0;

param d: 1 2 3 4 :=
  1 1000 0 0 2300
  2 4000 0 0 2041
  3 1959 0 4000 0
  4 0 951 2000 0
  5 0 3500 1385 0
  6 0 3003 0 0
  7 516 0 0 2500
  8 978 0 3500 0
  9 2000 0 2513 900
  10 0 3490 0 0
  11 0 0 5283 0;

```

```
param s :=  
  0  0  
  1  0  
  2  0  
  3  0  
  4  0  
  5  0  
  6  0  
  7  0  
  8  0  
  9  0  
 10  0  
 11  0;
```

Apéndice C

Algoritmo de ahorros para un CVRP

En este apéndice mostramos las iteraciones del algoritmo de Clarke & Wright para el CVRP presente en la Sección 2.2.3. Antes de nada, tal y como vemos en la Figura C.1, debemos crear las rutas de camión $(0, i, 0)$ para todo cliente $i \in \{1, \dots, 10\}$. Esta configuración de rutas inicial tiene un coste de 742 kilómetros.

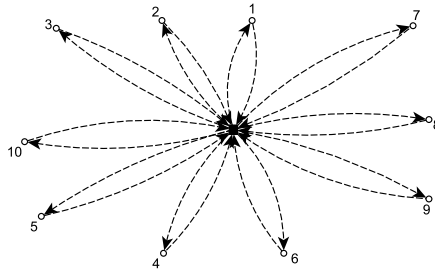


Figura C.1: Creamos las rutas $(0, i, 0)$ para todo cliente i .

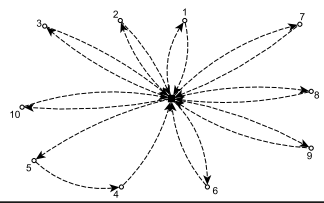
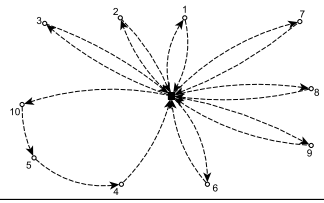
A continuación, presentamos en la matriz S (C.1) los ahorros $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ para todo $i, j \in \{1, \dots, 10\}$.

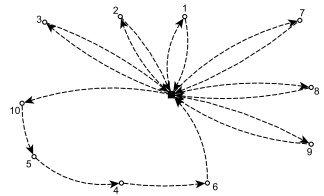
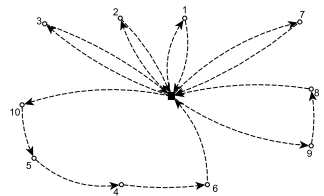
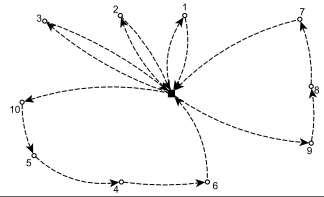
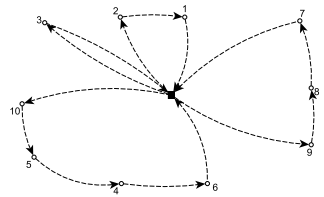
$$S = \begin{pmatrix} 0 & 37 & 32 & 26 & 26 & 26 & 25 & 25 & 25 & 26 \\ 37 & 0 & 33 & 26 & 27 & 27 & 26 & 34 & 32 & 27 \\ 32 & 33 & 0 & 25 & 26 & 25 & 25 & 26 & 25 & 25 \\ 26 & 26 & 25 & 0 & 125 & 118 & 18 & 18 & 18 & 119 \\ 26 & 27 & 26 & 125 & 0 & 119 & 18 & 19 & 18 & 120 \\ 26 & 27 & 25 & 118 & 119 & 0 & 18 & 18 & 18 & 118 \\ 25 & 26 & 25 & 18 & 18 & 18 & 0 & 38 & 38 & 18 \\ 25 & 34 & 26 & 18 & 19 & 18 & 38 & 0 & 39 & 18 \\ 25 & 32 & 25 & 18 & 18 & 18 & 38 & 39 & 0 & 18 \\ 26 & 27 & 25 & 119 & 120 & 118 & 18 & 18 & 18 & 0 \end{pmatrix} \quad (\text{C.1})$$

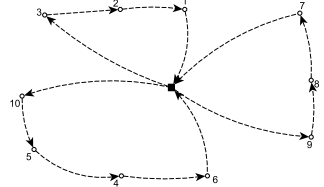
Las columnas de la Tabla C.1, donde aparecen todas las iteraciones de la heurística C&W, indican lo siguiente:

- *iter*: iteración del algoritmo.
- S_m : ahorro máximo.
- Clientes $i - j$: clientes i y j que dan lugar a S_m .
- Ruta i : ruta a la que pertenece el cliente i .
- Ruta j : ruta a la que pertenece el cliente j .
- ¿Unión?: toma el valor Sí o No, dependiendo de si la unión de ambas rutas es factible.
- Coste: coste de la configuración de rutas en dicha iteración.
- Dibujo rutas: representación gráfica de la configuración de rutas en dicha iteración.

Para evitar ser repetitivos, dejaremos las columnas “Coste” y “Dibujo de rutas” en blanco cuando en una iteración no se produzca una fusión, ya que en ese caso el contenido de dichas columnas coincide con el de la iteración anterior.

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
1	125	5 - 4	(0, 5, 0)	(0, 4, 0)	Sí	617	
2	120	10 - 5	(0, 10, 0)	(0, 5, 4, 0)	Sí	497	
3	119	10 - 4	(0, 10, 5, 4, 0)	(0, 10, 5, 4, 0)	No		
4	119	6 - 5	(0, 6, 0)	(0, 10, 5, 4, 0)	No		
5	119	6 - 5	(0, 6, 0)	(0, 10, 5, 4, 0)	No		
6	119	4 - 10	(0, 10, 5, 4, 0)	(0, 10, 5, 4, 0)	No		

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
7	118	6 - 4	(0, 6, 0)	(0, 10, 5, 4, 0)	No		
8	118	4 - 6	(0, 10, 5, 4, 0)	(0, 6, 0)	Sí	379	
9	118	10 - 6	(0, 10, 5, 4, 6, 0)	(0, 10, 5, 4, 6, 0)	No		
10	118	6 - 10	(0, 10, 5, 4, 6, 0)	(0, 10, 5, 4, 6, 0)	No		
11	39	9 - 8	(0, 9, 0)	(0, 8, 0)	Sí	340	
12	38	8 - 7	(0, 9, 8, 0)	(0, 7, 0)	Sí	302	
13	38	9 - 7	(0, 9, 8, 7, 0)	(0, 9, 8, 7, 0)	No		
14	38	7 - 9	(0, 9, 8, 7, 0)	(0, 9, 8, 7, 0)	No		
15	37	2 - 1	(0, 2, 0)	(0, 1, 0)	Sí	265	
16	34	8 - 2	(0, 9, 8, 7, 0)	(0, 2, 1, 0)	No		
17	34	2 - 8	(0, 2, 1, 0)	(0, 9, 8, 7, 0)	No		

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
18	33	3 - 2	(0, 3, 0)	(0, 2, 1, 0)	Sí	232	
19	32	3 - 1	(0, 3, 2, 1, 0)	(0, 3, 2, 1, 0)	No		
20	32	9 - 2	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
21	32	1 - 3	(0, 3, 2, 1, 0)	(0, 3, 2, 1, 0)	No		
22	32	2 - 9	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
23	27	5 - 2	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
24	27	6 - 2	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
25	27	10 - 2	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
26	27	2 - 5	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
27	27	2 - 6	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
28	27	2 - 10	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
29	26	4 - 1	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
30	26	5 - 1	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
31	26	6 - 1	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
32	26	10 - 1	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
33	26	4 - 2	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
34	26	7 - 2	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
35	26	5 - 3	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
36	26	8 - 3	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
37	26	1 - 4	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
38	26	2 - 4	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
39	26	1 - 5	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
40	26	3 - 5	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
41	26	1 - 6	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
42	26	2 - 7	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
43	26	3 - 8	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
44	26	1 - 10	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
45	25	7 - 1	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
46	25	8 - 1	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
47	25	9 - 1	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
48	25	4 - 3	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
49	25	6 - 3	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
50	25	7 - 3	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
51	25	9 - 3	(0, 9, 8, 7, 0)	(0, 3, 2, 1, 0)	No		
52	25	10 - 3	(0, 10, 5, 4, 6, 0)	(0, 3, 2, 1, 0)	No		
53	25	3 - 4	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
54	25	1 - 7	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
55	25	1 - 8	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
56	25	1 - 9	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
57	25	3 - 9	(0, 3, 2, 1, 0)	(0, 9, 8, 7, 0)	No		
58	25	3 - 10	(0, 3, 2, 1, 0)	(0, 10, 5, 4, 6, 0)	No		
59	19	8 - 5	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
60	19	5 - 8	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
61	18	7 - 4	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
62	18	8 - 4	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
63	18	9 - 4	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
64	18	7 - 5	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
65	18	9 - 5	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
66	18	7 - 6	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
67	18	8 - 6	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
68	18	9 - 6	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
69	18	4 - 7	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
70	18	5 - 7	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
71	18	6 - 7	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
72	18	10 - 7	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
73	18	4 - 8	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
74	18	6 - 8	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
75	18	10 - 8	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
76	18	4 - 9	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
77	18	5 - 9	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		

<i>iter</i>	S_m	Clientes $i - j$	Ruta i	Ruta j	¿Unión?	Coste	Dibujo rutas
78	18	6 – 9	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
79	18	10 – 9	(0, 10, 5, 4, 6, 0)	(0, 9, 8, 7, 0)	No		
80	18	7 – 10	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
81	18	8 – 10	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
82	18	9 – 10	(0, 9, 8, 7, 0)	(0, 10, 5, 4, 6, 0)	No		
83	0						

Tabla C.1: Iteraciones del algoritmo de Clarke & Wright.

Lista de acrónimos

ABHC	Attribute Based Hill Climber
AC	Aspiration Criterion
ACTSP	Travelling Salesman Problem under Accessibility Constraints
AMPL	A Modeling Language for Mathematical Programming
CAP	Customer-route Assignment Problem
CMT	Christofides, Migozzi and Toth
CVR	Complete Vehicle Route
CVRP	Capacitated Vehicle Routing Problem
DA	Deterministic Annealing
FTB	Frequency-Based Tabu restriction
GAP	Generalized Assignment Problem
GTTRP	Generalized Truck and Trailer Routing Problem
LNS	Large Neighborhood Search
LP	Linear Program
LS	Local Search
MC-TTRP	Multi-Compartment Truck and Trailer Routing Problem
MC-VRP	Multi-Compartment Vehicle Routing Problem
MC-VRPSD	Multi-Compartment Vehicle Routing Problem with Stochastic Demands
MDTTRP	Multi-depot Truck and Trailer Routing Problem
MT	Main Tour
OPD	One-Point Descent movement
OPT	One-Point Tabu search movement
OTB	Objective-Based Tabu restriction
PACVRP	Partial Accesibility Constrained Vehicle Routing Problem

PTR	Pure Truck Route
PVR	Pure Vehicle Route
RDP	Route Definition Problem
RGAP	Relaxed Generalized Assignment Problem
RRT	Record-to-Record Travel
SA	Simulated Annealing
SDVRP	Split-Delivery Vehicle Routing Problem
SVRP	Site-Dependent Vehicle Routing Problem
TPD	Two-Point Descent exchange
TPT	Two-Point Tabu Search exchange
TS	Tabu Search
TSP	Travelling Salesman Problem
TTRP	Truck and Trailer Routing Problem
TTRPTW	Truck and Trailer Routing Problem with Time Windows
VRP	Vehicle Routing Problem
VRPPD	Vehicle Routing Problem with Pickup and Delivery
VRPSD	Vehicle Routing Problem with Stochastic Demands
VRPT	Vehicle Routing Problem with Trailers
VRPTW	Vehicle Routing Problem with Time Windows

Bibliografía

- Archetti C, Campbell AM, Speranza MG (2016) Multicommodity vs. single-commodity routing. *Transportation Science* 50:461-472.
- Avella P, Boccia M, Sforza A (2004) Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research* 152:170-179.
- Brown GG, Ellis C, Graves GW (1987) Real-time, wide area dispatch of mobil tank trucks. *Interfaces* 17: 107-120.
- Brown GG, Graves GW (1981) Real-time dispatch of petroleum tank trucks. *Management Science* 27:19-32.
- Caramia M, Guerriero F (2010a) A heuristic approach for the truck and trailer routing problem. *Journal of Operational Research Society* 61:1168-1180.
- Caramia M, Guerriero F (2010b) A milk collection problem with incompatibility constraints. *Interfaces* 40:130-143.
- Chajakis ED, Guignard M (2003) Scheduling deliveries in vehicles with multiple compartments. *Journal of Global Optimization* 26:43-78.
- Chao IM (2002) A tabu search method for the truck and trailer routing problem. *Computers & Operations Research* 29:33-51.
- Chao IM, Golden BL, Wasil EA (1995) An improved heuristic for the period vehicle routing problem. *Networks* 26:25-44.
- Chao IM, Golden BL, Wasil EA (1998) A New Algorithm for the Site-Dependent Vehicle Routing Problem. En: *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristics Search: Interfaces in Computer Science and Operations Research* (ed) D. L. Woodruff. Springer US, Boston, pp 301-312.

- Christofides N, Mingozzi A, Toth P (1979) The vehicle routing problem. En: Christofides N, Mingozzi A, Toth P, Sandi C (ed) *Combinatorial Optimization*. Wiley, Chichester, pp 315-338.
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12:568-581.
- Coelho LC, Laporte G (2015) Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research* 242:854-864.
- Croes GA (1958) A method for solving traveling salesman problems. *Operations Research* 6:791-812.
- Dantzig GB, Fulkerson RD, Johnson SM (1954) Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America* 2:393-410.
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Science* 6:80-91.
- Derigs U, Gottlieb J, Kalkoff J, Piesche M, Rothlauf F, Vogel U (2011) Vehicle routing with compartments: Applications, modelling and heuristics. *OR Spectrum* 33:885-914.
- Derigs U, Pullmann M, Vogel U (2013) Truck and trailer routing – Problems, heuristics and computational experience. *Computers & Operations Research* 40:536-546.
- Drexl M (2007) A branch-and-price algorithm for the truck-and-trailer routing problem. Technical Report, RWTH Aachen University, Germany.
- Dror M, Trudeau P (1986) Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research* 23:228-235.
- Dueck G (1993) New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics* 104:86-92.
- Dueck G, Scheuer T (1990) Threshold accepting: A general purpose optimization algorithm. *Journal of Computational Physics* 90:161-175.
- El Fallahi A, Prins C, Wolfler Calvo, R (1986) A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 35:1725-1741.

- Fisher ML, Jaikumar R (1981) A generalized assignment heuristic for vehicle routing. *Networks* 11:109-124.
- Gerdessen JC (1993) Vehicle routing problem with trailers. Technical Note 93-09, Department of Mathematics, Wageningen Agricultural University, The Netherlands.
- Gerdessen JC (1996) Vehicle routing problem with trailers. *European Journal of Operational Research* 93:135-147.
- Gillett BE, Miller LR (1974) A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 22:340-349.
- Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research* 13:533-549.
- Goodson JC (2015) A priori policy evaluation and cyclic-order-based simulated annealing for the multicompartment vehicle routing problem with stochastic demands. *European Journal of Operational Research* 241:361-369.
- Gutián de Frutos RM, Casas-Méndez BV (To appear) Routing problems in agricultural cooperatives: A model for optimization of transport vehicle logistics. *IMA Journal of Management Mathematics* doi:10.1093/imaman/dpy010.
- Henke T, Speranza MG, Wäscher, G (2019) A branch-and-cut algorithm for the multicompartment vehicle routing problem with flexible compartment sizes. *Annals of Operations Research* 275:321-338.
- Huang SH (2015) Solving the multi-compartment capacitated location routing problem with pickup-delivery routes and stochastic demands. *Computers & Industrial Engineering* 87:104-113.
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671-680.
- Lasdon LS (1970) Optimization theory for large systems. Macmillan Company, New York.
- Lenstra JK, Rinnooy Kan AHG (1981) Complexity of vehicle routing and scheduling problems. *Networks* 11:221-227.
- Lin S (1965) Computer solutions to the traveling salesman problem. *Bell System Technical Journal* 44:2245-2269.

- Lin SW, Yu VF, You SC (2009) Solving the truck and trailer routing problem based on a simulated annealing heuristic. *Computers & Operations Research* 36:1683-1692.
- Lin SW, Yu VF, Chou SY (2010) A note on the truck and trailer routing problem. *Expert Systems with Applications* 37:899-903.
- Lin SW, Yu VF, Lu CC (2011) A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications* 38:15244-15252.
- Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco, N (2010) A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. *Computers & Operations Research* 37:1886-1898.
- Mendoza JE, Castanier B, Guéret C, Medaglia AL, Velasco N (2011) Constructive heuristics for the multicompartment vehicle routing problem with stochastic demands. *Transportation Science* 45:346-363.
- Muyldermans L, Pang G (2010) On the benefits of co-collection: Experiments with a multi-compartment vehicle routing algorithm. *European Journal of Operational Research* 206:93-103.
- Oppen J, Løkketangen A (2008) A tabu search approach for the livestock collection problem. *Computers & Operations Research* 35:3213-3229.
- Or I (1976) Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking. Thesis, Northwestern University.
- Ostermeier M, Hübner A (2018) Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research* 269:682-694.
- Pandelis DG, Kyriakidis EG, Dimitrakos TD (2012) Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research* 217:324-332.
- Parragh SN, Cordeau JF (2017) Branch-and-price and adaptive large neighborhood search for the truck and trailer routing problem with time windows. *Computers and Operations Research* 83:28-44.
- Raff S (1983) Routing and scheduling of vehicles and crews: the state of the art. *Computers & Operations Research* 10:63-211.
- Rochat Y, Taillard ED (1995) Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1:147-167.

- Rothenbächer AK, Drexel M, Irnich S (2018) Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. *Transportation Science* 52:1035-1296.
- Scheuerer S (2004) Neue Tabusuche-Heuristiken für die logistische Tourenplanung bei restringierendem Anhängereinsatz, mehreren Depots und Planungsperioden. Tesis, Universität Regensburg.
- Scheuerer S (2006) A tabu search heuristic for the truck and trailer routing problem. *Computers & Operations Research* 33:894-909.
- Semet F (1995) A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research* 61:45-65.
- Semet F, Taillard ED (1993) Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research* 41:469-488.
- Silvestrin PV, Ritt M (2017) An iterated tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research* 81:192-202.
- Simon D (2013) *Evolutionary Optimization Algorithms*. John Wiley & Sons, New Jersey.
- Toth P, Vigo D (2002) *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, New Jersey.
- van der Bruggen R, Gruson L, Salomon M (1995) Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research* 81:460-473.