



Universidad  
de La Laguna

---

# Problemas de Rutas de Vehículos Algunas de sus variantes más conocidas

*Vehicle Routing Problems  
Some of its most popular variants*

Carlos Rafael Pérez Cabrera

*Trabajo de Fin de Grado*

Matemáticas, Estadística e Investigación Operativa

Facultad de Ciencias. Sección de Matemáticas

Universidad de La Laguna

---

La Laguna, 13 de julio de 2016

Dr. D. **Juan José Salazar González**, con N.I.F. 43.356.435D profesor Catedrático de Universidad adscrito al Departamento de Matemáticas, Estadística e Investigación Operativa de la Universidad de La Laguna

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Problemas de Rutas de Vehículos. Algunas de sus variantes más conocidas”*

ha sido realizada bajo su dirección por D. **Carlos Rafael Pérez Cabrera**, con N.I.F. 42.199.236-D.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 13 de julio de 2016

## Agradecimientos

A Juan José Salazar González por toda su ayuda durante el desarrollo de este  
proyecto.  
A mi madre y hermana por su apoyo para llegar hasta aquí.  
Y a Ana.

## Resumen

*El proceso de transporte de mercancías está presente en numerosos servicios, llegando a ser una parte esencial en muchos casos, como en las compras por Internet. Por ello, es necesario optimizar los costes derivados de este proceso; ahí reside la importancia de los Problemas de Rutas de Vehículos, los cuales se encargan del diseño óptimo de rutas para una flota de vehículos que debe satisfacer las demandas de un conjunto dado de clientes.*

*La diversidad de estos problemas es muy amplia, pues existen infinidad de variantes del problema de ruta de vehículos, cada una de ellas enfocada a resolver un problema concreto. En general, estos problemas tratan de buscar un conjunto de rutas, de forma que se satisfaga a todos los clientes y se minimicen los costes. Dependiendo del problema concreto que se trate de resolver, se añaden limitaciones como pueden ser la capacidad máxima de los vehículos o ventanas de tiempo para los clientes.*

*En este Trabajo de Fin de Grado, describiremos algunas de las principales variantes del VRP, atendiendo a si se restringe o no, la capacidad de los vehículos, si es necesario atender a los clientes en un intervalo de tiempo concreto, si a los clientes se les permite enviar y recibir cierta cantidad de producto y, por último, si los clientes se dividen en dos subconjuntos, de recogida y entrega.*

*Se han propuesto distintas formulaciones que permiten resolver estos problemas. Además describimos los algoritmos necesarios para su implementación y el código correspondiente, para algunas de estas formulaciones, en un lenguaje denominado GMPL. Finalmente, se presentan los resultados computacionales obtenidos con un software matemático libre.*

**Palabras clave:** VRP sin capacidad, CVRP, VRPTW, VRPPD, VRPB, GUSEK.

## Abstract

The process of transport of goods is present in numerous services, becoming an essential part in many cases, such as internet purchases. For this reason, it is necessary to optimise costs deriving from this process; thus the importance of the Vehicle Routing Problems, which are responsible for the optimum design of routes for a fleet of vehicles which must satisfy the demands of a given group of customers.

The diversity of these problems is very great since there is an infinity of variants of the problem of vehicle routes, each one of which is focussed on solving a specific problem. In general, these problems attempt to find a set of routes, in such a manner that all the customers are satisfied and costs are minimised. Depending on the specific problem which one is attempting to solve, limitations are added such as the maximum capacity of the vehicles or time windows for customers.

In this End-of-Degree project, we shall describe some of the main variants of the VRP, looking at whether the capacity of the vehicles is restricted or not, whether it is necessary to deal with the customers in a specific period of time, whether the customers are allowed to send and receive a certain amount of product and, finally, whether the customers are divided into two subsets, of collection and delivery.

Different formulations have been proposed which make it possible to solve these problems. Furthermore, we describe the algorithms necessary for implementation and the corresponding code, for some of the formulations, in a language called GMPL. Finally, the computational results obtained with free mathematical software are presented.

**Keywords:** *VRP without capacity* , *CVRP*, *VRPTW*, *VRPPD*, *VRPB*, *GUSEK*.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del problema . . . . .	2
1.2. Objetivos y notación del problema . . . . .	4
<b>2. Variantes del VRP</b>	<b>7</b>
2.1. VRP sin capacidad . . . . .	7
2.1.1. Formulación de dos índices . . . . .	8
2.1.2. Formulación de tres índices . . . . .	9
2.1.3. Formulación con variables de flujo . . . . .	10
2.1.4. Formulación con variables de multiflujo . . . . .	10
2.2. VRP con capacidad limitada (CVRP) . . . . .	11
2.2.1. Formulación de dos índices . . . . .	12
2.2.2. Formulación de tres índices . . . . .	12
2.2.3. Formulación con variables de flujo . . . . .	13
2.2.4. Formulación con variables de multiflujo . . . . .	13
2.3. VRP con ventanas de tiempo (VRPTW) . . . . .	13
2.3.1. Formulación de tres índices . . . . .	14
2.3.2. Formulación con variables de flujo . . . . .	14
2.3.3. Formulación con variables de multiflujo . . . . .	15
2.4. VRP con recogida y entrega (VRPPD) . . . . .	15
2.4.1. Formulación de tres índices . . . . .	16
2.5. VRP con Backhauls (VRPB) . . . . .	17
2.5.1. Formulación de tres índices . . . . .	18
<b>3. Implementaciones en GUSEK</b>	<b>20</b>
3.1. Implementación del VRP sin capacidad . . . . .	20
3.2. Implementación del CVRP . . . . .	22
3.3. Implementación del VRPTW . . . . .	23
3.4. Implementación del VRPPD . . . . .	23
3.5. Implementación del VRPB . . . . .	24
<b>4. Resultados computacionales</b>	<b>25</b>
4.1. Resultados del VRP sin capacidad . . . . .	26

4.2. Resultados del CVRP . . . . .	27
4.3. Resultados del VRPTW . . . . .	28
4.4. Resultados del VRPPD . . . . .	29
4.5. Resultados del VRPB . . . . .	30
<b>5. Conclusiones</b>	<b>32</b>
<b>Bibliografía</b>	<b>34</b>

# Capítulo 1

## Introducción

El Problema de Rutas de Vehículos (VRP, por sus siglas en inglés) es uno de los problemas de optimización combinatoria más importantes y estudiados, con cientos de artículos publicados en la teoría y aplicaciones. Éste se encarga del diseño óptimo de rutas para ser utilizadas por una flota de vehículos con el fin de servir a un conjunto de clientes. La importancia de estos problemas no reside solamente en su dificultad, sino también en el gran número de aplicaciones reales que existen. Algunos ejemplos son el reparto de correo, reparto de mercancía, transporte escolar o recogida de basuras; pero estos problemas no solo pueden ser aplicados en logística y distribución, pues también podemos encontrar aplicaciones de estos problemas, por ejemplo, en la industria o la genética. En general, el proceso de transporte de mercancías se encuentra presente en un gran número de los sistemas de producción, el cual supone una parte considerable (entre el 10 % y el 20 %) del coste final del producto. Por lo tanto, la utilización de estos problemas da lugar a un ahorro (generalmente entre el 5 % y el 20 %) del coste final del producto.

Una motivación para haber realizado este proyecto ha sido cursar la asignatura de *Programación Combinatoria* donde pude conocer la utilidad de las técnicas de investigación operativa. Además, he realizado las prácticas externas en el departamento de logística de la empresa SEUR, donde era necesaria la aplicación de los problemas de rutas de vehículos, lo que ha inspirado finalmente el tema de esta memoria.

Este trabajo consta de 5 capítulos. En el primero realizamos una introducción a los problemas de rutas de vehículos, donde se describe el problema y los objetivos y notación de éste. En el segundo capítulo se presentan y explican detalladamente cinco variantes del VRP, además se proponen distintas formulaciones para éstas. En el tercer capítulo describimos el software matemático que vamos a utilizar, e implementamos uno de los algoritmos de cada variante y, seguidamente, en el cuarto capítulo se presentan los resultados computacionales obtenidos con estas implementaciones.

## 1.1. Descripción del problema

Los problemas de rutas de vehículos tratan de resolver problemas como el que sigue: "una empresa debe repartir cierto producto entre sus clientes y desea encontrar la ruta (o rutas) de menor coste que, partiendo desde el depósito (o depósitos), visite a cada cliente satisfaciendo su demanda y regrese al depósito". La distribución de los productos se refiere al servicio, en un período de tiempo determinado, de un conjunto de clientes por un conjunto de vehículos, que se encuentra en uno o más depósitos, la cual se realiza por un conjunto de conductores utilizando una red de carreteras apropiada. En particular, la solución de un VRP proporciona un conjunto de rutas, cada una realizada por un único vehículo que se inicia y termina en su propio depósito, de tal manera que se cumplan todos los requisitos de los clientes, se satisfagan todas las restricciones operacionales y el costo de transporte global se reduzca al mínimo.

La red de carreteras utilizada para la distribución de los productos, se describe generalmente a través de un gráfico, el cual está formado por un conjunto de vértices y un conjunto de arcos. Los arcos del gráfico representan los tramos de carretera, y los vértices, los cruces de estos tramos, los depósitos y los clientes. Los arcos (y por ello los gráficos correspondientes) pueden ser dirigidos o no dirigidos, dependiendo de si los tramos de carretera tienen un único sentido (por ejemplo, autopistas o calles de un solo sentido) o son de doble sentido, respectivamente. A cada uno de los arcos se le asocia un coste que generalmente representa el recorrido de éste, o el tiempo que se tarda en atravesar dicho arco.

Las rutas que se realizan para atender a los clientes comienzan y terminan en uno o más depósitos, situados en los vértices del gráfico de la carretera. Cada depósito se caracteriza por el número y tipo de vehículos del mismo y por el importe global de los bienes que se pueden tratar. En algunas aplicaciones del mundo real, los clientes se reparten entre los depósitos y los vehículos tienen que volver a su depósito de origen al final de cada ruta. En estos casos, el VRP en general se puede descomponer en varios problemas independientes, cada uno asociado con un depósito diferente.

En cuanto a los clientes, que están representados gráficamente por los vértices del grafo, se les asocia una cantidad de producto que debe ser entregada y/o recogida. También, podría ser necesario servirles en un intervalo de tiempo concreto (ventana de tiempo). Habría que considerar un tiempo de carga o descarga de la mercancía que dependerá del vehículo y sus características. Generalmente, los clientes son visitados una única vez, a no ser, por ejemplo, que sean servidos por más de un vehículo.

Las rutas deben satisfacer varias limitaciones operativas, que dependen de la naturaleza de las mercancías transportadas, de la calidad del nivel de servicio, y de las características de los clientes y los vehículos. Algunas limitaciones operativas típicas son las siguientes: a lo largo de cada ruta, la carga actual del vehículo asociado no puede exceder la capacidad de los vehículos; los clientes que se sirven en una ruta pueden pedir más que la entrega o la recogida de los bienes, o pueden existir dos posibilidades; y los clientes se pueden servir

sólo dentro de sus ventanas de tiempo y de los períodos de trabajo de los conductores asociados con el vehículo. También se pueden dar otras restricciones de precedencia, como cuando a los clientes se les permite enviar y recibir cierta cantidad de producto entre ellos, éste es el caso, por ejemplo, de los llamados problemas de recogida y entrega (pickup and delivery), y cuando se sirven clientes de diferente tipo en una misma ruta, esta situación se da, por ejemplo, en los problemas de *backhauls*.

Este problema es NP-duro por lo que, a día de hoy, no se ha encontrado ningún algoritmo que consiga resolverlos en un tiempo polinómico. No obstante, su importancia no se debe únicamente a su complejidad de resolución, sino a la gran variedad de situaciones prácticas de la vida cotidiana en las que pueden ser aplicados. El éxito en la utilización de técnicas de investigación operativa se debe al desarrollo de los sistemas informáticos, tanto desde el punto de vista del software, como del hardware, y la creciente integración de sistemas informáticos en procesos productivos y comerciales. Otro motivo de este éxito se debe al desarrollo de modelos y herramientas algorítmicas que se han ido implementando en los últimos años.

Los primeros que se introdujeron en el VRP fueron Dantzig y Ramser [4] en 1959, cuando estudiaron la aplicación real en la distribución de gasolina para estaciones de carburante. Proponían una solución basada en una formulación de programación lineal que daba lugar a una solución casi óptima. El objetivo era encontrar una forma de asignar los camiones a las estaciones de servicio de manera que se satisficieran las demandas de éstas y la distancia recorrida por la flota de camiones fuese mínima. Años más tarde, en 1964, Clarke y Wright [3] propusieron un algoritmo que mejoraba el enfoque de Dantzig y Ramser.

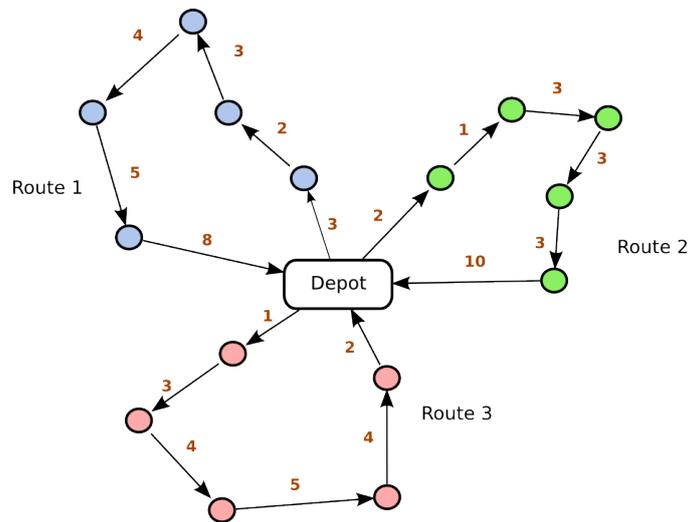


Figura 1.1: Grafo VRP

## 1.2. Objetivos y notación del problema

El VRP se define de la siguiente forma. Dado un conjunto de vehículos, localizados en uno o varios depósitos, se debe atender a un conjunto de clientes distribuidos geográficamente. Luego, el objetivo más habitual es satisfacer la demanda de dicho conjunto de clientes con un coste mínimo, hallando la ruta (o rutas) óptima que comienza y termina en el depósito.

Los objetivos en los VRPs pueden ser varios:

- Minimizar el coste global del transporte, teniendo en cuenta los costes o tiempo de desplazamiento.
- Minimizar el número de vehículos necesario para satisfacer a todos los clientes.
- Minimizar las penalizaciones que se dan cuando un cliente se queda sin servicio.
- Minimizar el tiempo de espera de los vehículos cuando el problema tiene ventanas de tiempo.
- También, el objetivo podría ser maximizar, o una combinación de estos objetivos.

En primer lugar, para determinar el coste global de las rutas es necesario conocer el coste de cada conexión y el tiempo que se tarda en viajar entre cualquier par de nodos. Pero, normalmente no existen todas las conexiones entre nodos, es decir, el grafo de la red de carreteras es disperso. En estos casos generalmente se transforma en un grafo completo, hallando el coste entre todos los nodos no conectados. Esto se hace buscando el camino mínimo entre un nodo y otro, pasando por otros vértices. De igual forma, el tiempo de viaje entre estos dos nodos se define como la suma de tiempo tardado en realizar este camino mínimo.

Por otra parte, se debe establecer si la red de carreteras es unidireccional o por el contrario tiene dos direcciones, pues el problema será asimétrico o simétrico, respectivamente.

Consideremos el grafo  $G = (V, A)$ , donde  $V = \{0, 1, \dots, n\}$  son los clientes y depósitos y  $A$  el conjunto de arcos que unen estos nodos. Los costes de cada arco  $(i, j) \in A$ ,  $c_{ij}$ , son mayores o iguales que 0, y representan el coste de viajar del nodo  $i$  al  $j$ , para todo  $i, j \in V$ .

Otras notaciones importantes son las siguientes. A cada cliente  $i$  se le asocia una demanda  $d_i$ . Para los depósitos estas demandas son ficticias, y se denotan como  $d_0 = 0$ . La cantidad de vehículos disponibles se denota por  $K$  y la capacidad de éstos por  $C$ . De esta forma es necesario asumir que  $d_i \leq C$ ,  $\forall i \in V$ . También se podría considerar el caso en que las capacidades de los vehículos sean distintas para cada uno de ellos y, por tanto,  $C$  dependerá de cada uno,  $C_k$ ,  $k = 1, \dots, K$ . Para un conjunto de  $S$  clientes se define,  $r(S)$ , como el número mínimo de vehículos necesarios para satisfacer a todos los clientes

pertenecientes a  $S$ . Con relación a los vehículos se debe asumir que  $K$  es siempre mayor que el número mínimo necesario de vehículos,  $K_{min}$ . El cálculo de este número requiere la solución de un Bin Packing Problem. El BPP también es un problema NP-duro y consiste en determinar el número mínimo de cajas, con un volumen determinado, que son necesarias para transportar ciertos objetos de los que se conoce su volumen.  $K_{min}$  puede ser sustituido por  $\lceil d(V)/C \rceil$ , que es una cota mínima de la solución del BPP.

Por otro lado, cada arco  $(i, j) \in A$  tiene un tiempo de recorrido asociado  $t_{ij}$ , que normalmente está relacionado con el coste  $c_{ij}$ . Cada cliente  $i$  tiene también asociado un tiempo de descarga  $s_i$ , durante el cual el vehículo está estacionado sirviendo al cliente  $i$ . Para ampliar la información acerca de los problemas de rutas de vehículos se pueden consultar los libros [2] y [9]

Finalmente, detallaremos la notación que vamos a utilizar en los modelos que definiremos en el siguiente capítulo:

- $n \in \mathbb{N} \equiv$  Número de nodos del grafo.
- $V = \{0, 1, \dots, n\} \equiv$  Conjunto de nodos.
- $A \equiv$  Conjunto de arcos que unen los nodos de  $V$ .
- $K \in \mathbb{N} \equiv$  Número de vehículos disponibles en el depósito.
- $C \in \mathbb{R}^+ \equiv$  Capacidad máxima de los vehículos.
- $T \in \mathbb{R}^+ \equiv$  Valor grande que representa el máximo tiempo de una ruta.
- $c_a \in \mathbb{R}^+, \forall a \in A \equiv$  Costo asociado al arco  $a = (i, j) \in A$
- $d_i \in \mathbb{R}^+, \forall i \in V \equiv$  Demanda del nodo  $i \in V$ . Generalmente, el depósito no tiene demanda.
- $s_i \in \mathbb{R}^+, \forall i \in V \equiv$  Tiempo de servicio del nodo  $i$ .
- $t_a \in \mathbb{R}^+, \forall a \in A \equiv$  Tiempo que se tarda en atravesar el arco  $a$ .
- $[a_i, b_i], \forall i \in V \equiv$  Ventana de tiempo del nodo  $i$ .
- $x_a \in \{0, 1\}, \forall a \in A \equiv$  Variable binaria que toma el valor 1 si el arco  $a$  está en la solución, y 0 en otro caso.
- $x_a^k \in \{0, 1\}, \forall a \in A, k = 1, \dots, K \equiv$  Variable binaria que toma el valor 1 si el arco  $a$  es atravesado por el vehículo  $k$ , y 0 en otro caso.
- $y_i^k \in \{0, 1\}, \forall i \in V, k = 1, \dots, K \equiv$  Variable binaria que toma el valor 1 si el vértice  $i$  es visitado por el vehículo  $k$ , 0 en otro caso.
- $f_a \in \mathbb{R}^+, \forall a \in A \equiv$  Variable de flujo que indica la carga del vehículo mientras atraviesa el arco  $a$ .

- $f_a^k \in \mathbb{R}^+, \forall a \in A, k = 1, \dots, K \equiv$  Variable de flujo que indica la carga del vehículo  $k$  mientras atraviesa el arco  $a$ .
- $u_i \in \mathbb{R}^+, \forall i \in V \equiv$  Variable continua que representa la carga del vehículo después de visitar al cliente  $i$ .
- $u_i^k \in \mathbb{R}^+, \forall i \in V, k = 1, \dots, K \equiv$  Variable continua que representa la carga del vehículo  $k$  después de visitar al cliente  $i$ .
- $w_i \in \mathbb{R}^+, \forall i \in V \equiv$  Variable que denota el momento en que comienza el servicio del nodo  $i$ .
- $w_i^k \in \mathbb{R}^+, \forall i \in V, k = 1, \dots, K \equiv$  Variable que denota el momento en que comienza el servicio del nodo  $i$  cuando es servido por el vehículo  $k$ .
- $\delta^+(i), \forall i \in V \equiv$  Conjunto de vértices  $j$  tal que el arco  $(i, j) \in A$ , es decir, los vértices que son directamente alcanzables desde  $i$ .
- $\delta^-(i), \forall i \in V \equiv$  Conjunto de vértices  $j$  tal que el arco  $(j, i) \in A$ , es decir, los vértices desde los cuales  $i$  es directamente alcanzable.

## Capítulo 2

# Variantes del VRP

En este capítulo describiremos las diferentes variantes de los VRPs para el caso de un solo depósito y una flota idéntica de vehículos con igual capacidad limitada. Siendo los más conocidos los VRP *con capacidad limitada* (CVRP), VRP *con ventanas de tiempo* (VRPTW), VRP *con recogida y entrega* (VRPPD) y VRP *con Backhauls* (VRPB). En la Figura 2.1 se muestran las relaciones entre estos tipos de problemas.

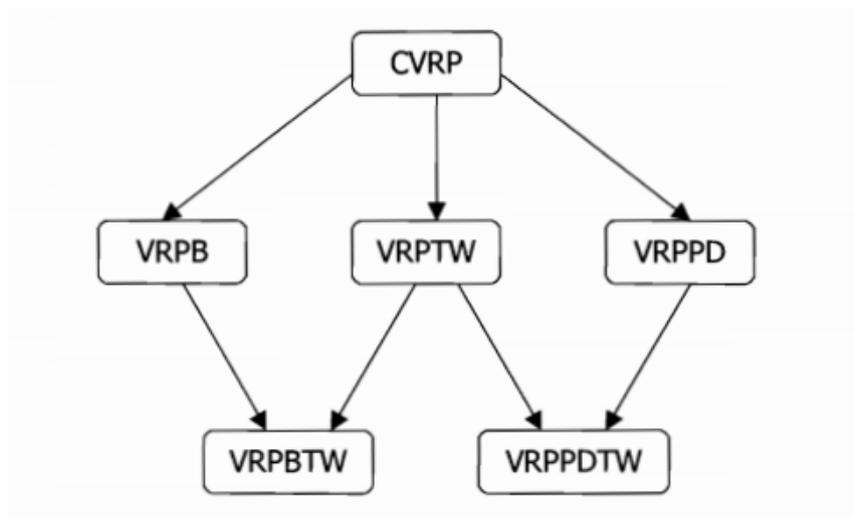


Figura 2.1: Relación variantes de los VRPs

### 2.1. VRP sin capacidad

Antes de describir el VRP sin capacidad, cabe destacar que el *Problema del Viajante* (TSP, por sus siglas en inglés) es un caso especial del VRP sin capacidad en el que solo se utiliza un medio de transporte, ( $K = 1$ ). En el TSP, se tiene una red de nodos, que pueden ser ciudades o simplemente lugares de una ciudad. Se parte de un lugar inicial, y

deben recorrerse todos sin pasar más de una vez por cada uno, volviendo al lugar inicial. Para cada arco, se tiene un valor  $c_a$ , el cual indica la distancia o el costo de atravesar el arco  $a = (i, j)$ . Se trata de encontrar en qué orden recorrer los nodos de la red, de tal modo que se minimice la distancia total recorrida. Se puede consultar el libro de G. Gutin y A.P. Punnen [5] para más información acerca de este problema.

En el caso del VRP sin capacidad se dispone de más de un medio de transporte. Los clientes se suponen conocidos, así como las distancias entre ellos y las demandas de éstos. Consideremos que los vehículos pasarán por todos los clientes recogiendo una cantidad de un tipo de producto específico. Por lo tanto, el objetivo de este problema es encontrar un conjunto de rutas con coste mínimo que satisfagan, por un lado, que cada cliente debe ser visitado una única vez por un único vehículo y, además, que todas las rutas deben comenzar y terminar en el depósito.

Basándose en la teoría de grafos definiremos el problema. Sea  $G = (V, A)$  un grafo, donde  $V = \{0, 1, \dots, n\}$  es el conjunto de nodos (o clientes) del grafo, con el depósito situado en el nodo 0, y  $A$  es el conjunto de arcos que unen los nodos de  $V$ . Cada cliente  $i$  se asocia con una demanda no negativa  $d_i$ . Para cada arco  $a = (i, j)$ ,  $i \neq j$ , existe un coste no negativo asociado  $c_a$ , el cual se suele interpretar como el coste que conlleva atravesar el arco  $a = (i, j)$ , es decir, viajar desde  $i$  hasta  $j$ . Generalmente, estos costos vienen dados en forma de matriz  $C$  y, cuando ésta es simétrica, normalmente  $A$  es reemplazado por un conjunto  $E$  de arcos no dirigidos. Por último, supondremos que hay  $K$  vehículos en el depósito.

Como ya hemos comentado, el VRP sin capacidad, definido de esta forma, es equivalente al TSP con varios viajantes. Luego, no es de extrañar que se haya clasificado también como un problema NP-duro.

En primer lugar, describiremos una formulación de dos índices para el VRP sin capacidad. Esta clase de formulación es muy utilizada para variantes sencillas del VRP (VRP sin capacidad o CVRP), pero generalmente es inadecuada para variantes más complejas del VRP. Concretamente solo puede ser utilizada cuando el coste total de la solución pueda ser expresado como suma de los costes asociados a los arcos visitados. Por ello, este tipo de formulación no es adecuada para problemas donde el coste de la solución dependa de la secuencia global de las ciudades o del tipo de vehículo que se asigne a cada ruta, es decir, no hay una forma directa de saber que un vehículo atraviesa un arco en particular. Por tanto, definiremos también una formulación de tres índices de forma que se indique explícitamente el vehículo que recorre cada arco. Finalmente, describiremos formulaciones para el VRP sin capacidad utilizando variables de flujo. Para definir estas formulaciones nos hemos basado en uno de los modelos que se presentan en Toth y Vigo [8].

### 2.1.1. Formulación de dos índices

Para introducir la *formulación de dos índices*, necesitamos la variable binaria  $x_a$ , la cual toma el valor 1 si el arco  $a = (i, j) \in A$  se encuentra en la solución óptima y 0 en

otro caso. El modelo quedaría de la siguiente forma:

$$\min \sum_{a \in A} c_a x_a \quad (2.1)$$

sujeto a:

$$\sum_{a \in \delta^+(i)} x_a = \sum_{a \in \delta^-(i)} x_a = 1 \quad \forall i \in V \setminus \{0\}, \quad (2.2)$$

$$\sum_{a \in \delta^+(0)} x_a = \sum_{a \in \delta^-(0)} x_a = K, \quad (2.3)$$

$$\sum_{a \in \delta^-(S)} x_a \geq 1 \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (2.4)$$

$$x_a \in \{0, 1\} \quad \forall i, j \in V. \quad (2.5)$$

Las restricciones (2.2) imponen que sólo un arco entre y salga de cada vértice. De una forma similar, (2.3) imponen que al depósito regresen y salgan  $K$  vehículos. Por otro lado, las restricciones (2.4), evitan los subciclos. Finalmente, (2.5) indica que las variables de decisión son binarias.

### 2.1.2. Formulación de tres índices

Una forma de solventar los problemas de la *formulación de dos índices* es indicar explícitamente el vehículo que recorre cada arco. Una de las múltiples formas de hacerlo es recurrir a una *formulación de tres índices*. Dicha formulación usa variables binarias  $x_a^k$ , que toman el valor 1 si el arco  $a = (i, j)$  es atravesado por el vehículo  $k$  en la solución, y 0 en otro caso. Y, también, variables binarias  $y_i^k$  ( $i \in V; k = 1, \dots, K$ ), que toman el valor 1 si el vértice  $i$  es visitado por el vehículo  $k$  en la solución, y 0 en otro caso. Entonces, el modelo se define de la siguiente forma:

$$\min \sum_{k \in K} \sum_{a \in A} c_a x_a^k \quad (2.6)$$

sujeto a:

$$\sum_{k=1}^K y_i^k = 1 \quad \forall i \in V \setminus \{0\}, \quad (2.7)$$

$$y_0^k = 1 \quad \forall k = 1, \dots, K, \quad (2.8)$$

$$\sum_{a \in \delta^+(i)} x_a^k = \sum_{a \in \delta^-(i)} x_a^k = y_i^k \quad \forall i \in V, k = 1, \dots, K, \quad (2.9)$$

$$\sum_{a \in \delta^+(S)} x_a^k \geq y_h^k \quad \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, \dots, K, \quad (2.10)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in V, k = 1, \dots, K, \quad (2.11)$$

$$x_a^k \in \{0, 1\} \quad \forall a \in A, k = 1, \dots, K. \quad (2.12)$$

Las restricciones (2.7) y (2.8) imponen que cada cliente sea visitado por un único vehículo y que deben salir  $K$  vehículos del depósito, respectivamente. Por otro lado, las desigualdades (2.9) exigen que si un vehículo entra en un vértice también debe salir de éste, para así conservar el flujo de vehículos. Las ecuaciones (2.10) son las restricciones de rotura de subciclos, para evitar que se generen subciclos que no pasen por el depósito. Por último, (2.11) y (2.12) definen ambas variables de decisión.

### 2.1.3. Formulación con variables de flujo

En esta ocasión, para definir el modelo del VRP sin capacidad, utilizaremos variables de flujo,  $f_a$ ,  $a = (i, j) \in A$ , que indican la carga del vehículo mientras recorre el arco  $a$ . El problema quedaría de la siguiente manera:

$$\min \sum_{a \in A} c_a x_a \quad (2.13)$$

sujeto a:

$$\sum_{a \in \delta^+(i)} x_a = \sum_{a \in \delta^-(i)} x_a = 1 \quad \forall i \in V \setminus \{0\}, \quad (2.14)$$

$$\sum_{a \in \delta^+(0)} x_a = \sum_{a \in \delta^-(0)} x_a = K, \quad (2.15)$$

$$\sum_{a \in \delta^+(i)} f_a - \sum_{a \in \delta^-(i)} f_a = d_i \quad \forall i \in V \setminus \{0\}, \quad (2.16)$$

$$x_a \in \{0, 1\} \quad \forall a \in A. \quad (2.17)$$

Las restricciones (2.14) indican que solo un arco entra y sale de cada vértice, y (2.15) imponen que del depósito salen y llegan exactamente  $K$  vehículos. Por otra parte, las ecuaciones de flujo de mercancía (2.16), exigen que la carga del vehículo al llegar a un vértice cualquiera menos la carga a salir del él sea exactamente la demanda de dicho vértice,  $d_i$ . Y con (2.17) hacemos que las variables de decisión sean binarias.

### 2.1.4. Formulación con variables de multifujo

Ahora definiremos un modelo similar al anterior, pero utilizando variables multifujo,  $f_a^k$ , que indican la carga del vehículo  $k$  a lo largo del arco  $a = (i, j) \in A$ . En este caso, se indica que vehículo está atravesando cada arco. Luego, el modelo quedaría definido de esta forma:

$$\min \sum_{k \in K} \sum_{a \in A} c_a x_a^k \quad (2.18)$$

sujeto a:

$$\sum_{a \in \delta^+(i)} x_a^k = \sum_{a \in \delta^-(i)} x_a^k = y_i^k \quad \forall i \in V, k = 1, \dots, K, \quad (2.19)$$

$$\sum_{k=1}^K y_i^k = 1 \quad \forall i \in V \setminus \{0\}, \quad (2.20)$$

$$y_0^k = 1 \quad \forall k = 1, \dots, K, \quad (2.21)$$

$$\sum_{a \in \delta^+(i)} f_a^k - \sum_{a \in \delta^-(i)} f_a^k = d_i y_i^k \quad \forall i \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.22)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in V, k = 1, \dots, K, \quad (2.23)$$

$$x_a \in \{0, 1\} \quad \forall a \in A. \quad (2.24)$$

Las restricciones (2.19) indican si un vehículo entra en un vértice también debe salir de éste. Las restricciones (2.20) y (2.21) denotan que cada cliente debe ser visitado por un único vehículo, y que del depósito deben salir exactamente  $K$  vehículos, respectivamente. Por otro lado, las ecuaciones de flujo de mercancía (2.22), exigen que la carga de cada vehículo al llegar a un vértice cualquiera menos la carga a salir del él sea exactamente la demanda de dicho vértice,  $d_i$ , si dicho vehículo visita ese vértice, de lo contrario la resta de las cargas es igual a 0. Finalmente, (2.23) y (2.24) indican que ambas variables de decisión son binarias.

## 2.2. VRP con capacidad limitada (CVRP)

El VRP con *capacidad limitada* es un VRP en el que la flota de vehículos tiene una capacidad determinada, la cual es común para todos los vehículos. El objetivo sigue siendo el mismo pero debemos añadir la restricción de capacidad: la suma de las demandas de los clientes que se visitan en cada ruta no debe exceder la capacidad del vehículo.

Formalmente, el CVRP se define de la siguiente forma. Sea  $G = (V, A)$  un grafo, donde  $V = \{0, 1, \dots, n\}$  en el conjunto de nodos, o clientes, con el depósito situado en el nodo 0, y  $A$  es el conjunto de arcos que unen los nodos de  $V$ . Para cada arco  $(i, j), i \neq j$ , existe un coste no negativo asociado  $c_a$ , el cual se suele interpretar como el coste que conlleva atravesar el arco  $a = (i, j)$ , es decir, viajar desde  $i$  hasta  $j$ . El uso de arcos de la forma  $(i, i)$  no está permitido, por lo que se suele imponer fijando  $c_{ii} = \infty$  para todo  $i \in V$ . Si el grafo  $G$  es dirigido, entonces la matriz de costes es asimétrica y el CVRP recibe el nombre de CVRP asimétrico. Si, por el contrario,  $c_{ij} = c_{ji}$  para todo  $(i, j) \in A$ , el problema se denomina CVRP simétrico (SCVRP) y el conjunto  $A$  es sustituido por un conjunto  $E$  de arcos no dirigidos.

También, a cada cliente  $i$  ( $i = 1, \dots, n$ ) se le asocia una demanda  $d_i \geq 0$  que debe ser recogida. Al depósito se le asigna una demanda ficticia  $d_0 = 0$ . Dado un conjunto  $S \subseteq V$ , se define  $d(S) = \sum_{i \in S} d_i$  como la demanda total del conjunto.

Un conjunto de  $K$  vehículos iguales, todos con la misma capacidad  $C$ , se encuentran disponibles en el depósito. Para asegurar la viabilidad, se asume que  $d_i \leq C$ , para todo  $i \in V$  y que  $K$  no es más pequeño que  $K_{min}$ . El valor  $K_{min}$  se determina a través del

problema de cubicación (Bin Packing Problem) asociado al CVRP, el cual es difícil de hallar. Por tanto, este número mínimo de vehículos se puede aproximar en función de la demanda total de los clientes tal que,  $K_{min} = \lceil d(V \setminus \{0\})/C \rceil$ .

Para el problema del CVRP adaptaremos los 4 modelos propuestos anteriormente para el VRP sin capacidad, formulaciones de 2 y 3 índices, y las formulaciones utilizando variables de flujo y multifujo, teniendo en cuenta la capacidad limitada,  $C$ , de los vehículos.

### 2.2.1. Formulación de dos índices

La formulación de dos índices para el CVRP sería la propuesta para el VRP sin capacidad, (2.1)-(2.5), modificando levemente la restricción (2.4) de la siguiente manera:

$$\sum_{a \in \delta^+(S)} x_a \geq r(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (2.25)$$

donde las restricciones de capacidad y conexión (2.25), determinan que cada partición  $(S, V \setminus S)$  debe ser atravesada por un número de arcos que no puede ser inferior a  $r(S)$  (número mínimo de vehículos necesarios para servir al conjunto  $S$ ). No obstante, la restricción (2.25) tiene un cardinal que crece exponencialmente con  $n$ , lo que hace prácticamente imposible resolver directamente este problema. Sin embargo, podemos sortear este contra-tiempo definiendo otra familia de restricciones con cardinal polinómico, como proponen en [8].

$$u_j \geq u_i + d_j x_{ij} - C(1 - x_{ij}) \quad \forall i, j \in V \setminus \{0\}, \quad (2.26)$$

$$d_i \leq u_i \leq C \quad \forall i \in V \setminus \{0\}, \quad (2.27)$$

donde  $u_i, i \in V \setminus \{0\}$ , es una variable continua adicional que representa la carga del vehículo después de visitar al cliente  $i$ . Se observa que las restricciones (2.26) y (2.27) imponen las limitaciones de capacidad y conexión del CVRP.

### 2.2.2. Formulación de tres índices

La formulación de tres índices para el CVRP, es la misma que definimos para el VRP sin capacidad, (2.6)-(2.12), añadiendo una nueva restricción que impone la limitación de capacidad de los vehículos:

$$\sum_{i \in V} d_i y_i^k \leq C \quad \forall k = 1, \dots, K. \quad (2.28)$$

De forma similar a la formulación de dos índices, podemos reemplazar las ecuaciones (2.10) por la siguiente familia de restricciones:

$$u_j^k \geq u_i^k + d_j x_{ij}^k - C(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.29)$$

$$d_i y_i^k \leq u_i^k \leq C y_i^k \quad \forall i \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.30)$$

donde  $u_i^k, i \in V \setminus \{0\}, k = 1, \dots, K$ , es una variable continua que representa la carga del vehículo  $k$  después de visitar al cliente  $i$ . Se puede observar que estas restricciones también reemplazan las restricciones de capacidad (2.28).

### 2.2.3. Formulación con variables de flujo

La formulación para el CVRP utilizando variables de flujo se corresponde con la propuesta para el VRP sin capacidad, (2.13)-(2.17), incorporando una nueva restricción de capacidad:

$$0 \leq f_a \leq Cx_a \quad \forall a \in A. \quad (2.31)$$

### 2.2.4. Formulación con variables de multifujo

La formulación para el CVRP utilizando variables de multifujo se corresponde con la propuesta para el VRP sin capacidad, (2.18)-(2.24), añadiendo una nueva restricción de capacidad:

$$0 \leq f_a^k \leq Cx_a^k \quad \forall a \in A, k = 1, \dots, K. \quad (2.32)$$

## 2.3. VRP con ventanas de tiempo (VRPTW)

El VRP *con ventanas de tiempo* (en inglés, VRP with Time Windows) es la extensión del CVRP en la cual se imponen limitaciones de capacidad y cada cliente  $i$  se asocia con un intervalo de tiempo  $[a_i, b_i]$ , llamado *ventana de tiempo*. Por lo tanto, además de conocer las demandas de los clientes y las distancias entre ellos, ahora es necesario conocer también la ventana de tiempo de cada uno de los clientes, el instante de tiempo en el que los vehículos salen del depósito, el tiempo necesario para viajar de un cliente a otro,  $t_{ij}$ , y un tiempo de servicio adicional,  $s_i$ , para cada cliente  $i$ . El servicio de cada cliente debe comenzar dentro de la ventana de tiempo asociada, y el vehículo debe detenerse en la ubicación del cliente por  $s_i$  instantes de tiempo. Además, en caso de llegar más temprano a la localización del cliente  $i$ , generalmente se le permite al vehículo esperar hasta el instante de tiempo  $a_i$ , es decir, hasta que se pueda servir al cliente dentro de su intervalo.

Formalmente, definimos el VRPTW de la siguiente forma. Sea  $G = (V, A)$  un grafo, donde  $V = \{0, 1, \dots, n\}$  es el conjunto de nodos, o clientes, con el depósito situado en el nodo 0, y  $A$  es el conjunto de arcos que unen los nodos de  $V$ . Para cada arco  $a = (i, j)$ ,  $i \neq j$ , existe un coste no negativo asociado  $c_a$ . También existe una cantidad,  $t_{ij}$ , asociada a cada arco, que equivale al tiempo necesario para atravesar el arco  $(i, j)$ . Por otra parte, a cada nodo se le asocia una demanda,  $d_i$ , que debe ser recogida, un tiempo de servicio,  $s_i$ , necesario para servir a cada cliente, y una ventana de tiempo,  $[a_i, b_i]$ , de tal modo que el vehículo debe llegar al nodo  $i$  en ese intervalo de tiempo. Normalmente, las matrices de costes y tiempo de viaje coinciden, y las ventanas de tiempo son definidas asumiendo que todos los vehículos salen del depósito en el instante de tiempo 0.

El VRPTW consiste en encontrar una colección de exactamente  $K$  rutas simples con un coste mínimo, y tal que:

- cada ruta visita el vértice del depósito;

- cada vértice del cliente es visitado por exactamente una ruta;
- la suma de las demandas de los vértices visitados por una ruta no exceda la capacidad del vehículo,  $C$ ; y
- para cada cliente  $i$ , el servicio comienza dentro de la ventana de tiempo,  $[a_i, b_i]$ , y el vehículo se detiene por un instante de tiempo  $s_i$ .

El VRPTW es NP-duro, ya que generaliza el CVRP, que surge cuando  $a_i = 0, b_i = +\infty$  para cada cliente  $i \in V \setminus \{0\}$ . Por otra parte, el llamado TSP *con ventanas de tiempo* (TS-PTW) es un caso especial del VRPTW en el cual  $C \geq d(V)$  y  $K = 1$ .

Antes de definir los modelos para el VRPTW aclaremos nuevamente la notación del problema. Sea el grafo  $G = (V, A)$ , donde  $V$  es el conjuntos de nodos, y el depósito está representado por el nodo 0, y  $A$  el conjuntos de arcos que une dichos nodos. Todas las rutas factibles comienzan y terminan en el nodo 0. Solo existen soluciones factibles si  $a_0 = E \leq \min_{i \in V \setminus \{0\}} b_i - t_{0i}$  y  $b_0 = L \geq \max_{i \in V \setminus \{0\}} a_i + s_i + t_{i0}$ . Además, la demanda y tiempo de servicio del depósito son ambos igual a 0.

A continuación, definiremos una formulación de tres índices y otras formulaciones con variables de flujo y multifujo para el VRPTW.

### 2.3.1. Formulación de tres índices

El modelo matemático para el VRPTW que definiremos utiliza tres tipos de variables: variables de flujo  $x_a^k$ ,  $a = (i, j) \in A$ ,  $k = 1, \dots, K$ , que toman el valor 1 si el arco  $(i, j)$  es atravesado por el vehículo  $k$  y 0 en otro caso, las variables  $y_i^k$ ,  $i \in V$ ,  $k = 1, \dots, K$  que toman el valor 1 si el vehículo  $k$  visita al cliente  $i$  y 0 en otro caso, y variables de tiempo  $w_i^k$ ,  $i \in V$ ,  $k = 1, \dots, K$ , que denotan el momento en que comienza el servicio del cliente  $i$  cuando es servido por el vehículo  $k$ .

La formulación de tres índices para el VRPTW se corresponde con la propuesta inicialmente para el VRP sin capacidad, (2.6)-(2.12). Además, debemos añadir la restricción de capacidad (2.28), que definimos anteriormente para la formulación de tres índices del CVRP, y las siguientes restricciones temporales:

$$w_j^k \geq w_i^k + (s_i + t_{ij})x_{ij}^k - T(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.33)$$

$$a_i y_i^k \leq w_i^k \leq b_i y_i^k \quad \forall i \in V, k = 1, \dots, K. \quad (2.34)$$

Con estas restricciones temporales se evitan los subciclos, y por tanto no es necesario utilizar la restricción (2.10). Las desigualdades (2.34) denotan que cada cliente debe ser visitado dentro su ventana de tiempo.

### 2.3.2. Formulación con variables de flujo

En esta formulación utilizaremos variables de flujo  $f_a$ , que indican la carga del vehículo mientras atraviesa el arco  $a = (i, j) \in A$ , y las variables  $w_i$ , que indican el instante de

tiempo en el que se visita el vértice  $i$ .

En este caso, la formulación del problema sería la definida en un principio para el VRP sin capacidad, (2.13)-(2.17), incluyendo la restricción de capacidad (2.31), definida anteriormente para la formulación con variables de flujo del CVRP, además de las siguientes restricciones temporales:

$$w_j \geq w_i + (s_i + t_{ij})x_{ij} - T(1 - x_{ij}) \quad \forall i, j \in V \setminus \{0\}, \quad (2.35)$$

$$a_i \leq w_i \leq b_i \quad \forall i \in V. \quad (2.36)$$

A diferencia de la formulación de tres índices, en este modelo no es posible saber que vehículo atraviesa un arco en particular. Las desigualdades (2.35) y (2.36) sirven para que cada cliente sea visitado dentro de su ventana de tiempo.

### 2.3.3. Formulación con variables de multiflujo

Para formular el problema utilizaremos variables de multiflujo  $f_a^k$ , que indican la carga del vehículo  $k$  mientras atraviesa el arco  $a = (i, j) \in A$ , y las variables  $w_i^k$ , que indican el instante de tiempo en el que el vehículo  $k$  visita el vértice  $i$ .

La formulación del problema sería la definida inicialmente para el VRP sin capacidad, (2.18)-(2.24), incluyendo la restricción de capacidad (2.32), descrita anteriormente para la formulación con variables de multiflujo del CVRP, además de las siguientes restricciones:

$$w_j^k \geq w_i^k + (s_i + t_{ij})x_{ij}^k - T(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.37)$$

$$a_i y_i^k \leq w_i^k \leq b_i y_i^k \quad \forall i \in V, k = 1, \dots, K. \quad (2.38)$$

Con las desigualdades (2.37) y (2.38) conseguimos que cada cliente sea visitado dentro de su ventana de tiempo.

## 2.4. VRP con recogida y entrega (VRPPD)

El VRP *con recogida y entrega* (en inglés, VRP with Pickup and Delivery) es, nuevamente, una extensión del CVRP. En este caso, a los clientes se les permite enviar y recibir cierta cantidad de producto entre ellos. Por lo tanto, se deben conocer las cantidades que desean enviar y recibir y el destino del producto, y por ello, se definen dos cantidades  $d_i$  y  $-d_i$ , asociadas a cada cliente  $i$ , las cuales representan la demanda de los productos para ser recogidos y entregados en el cliente  $i$ , respectivamente. Se supone que, en cada nodo de un cliente, la entrega se realiza antes que la recogida, por lo tanto, la carga de un vehículo antes de llegar a un cliente en concreto, se define como la carga inicial menos todas las demandas ya entregadas más todas las demandas ya recogidas.

Formalmente, definimos el VRPPD de la siguiente forma. Sea  $G = (V, A)$  un grafo, donde  $V = \{0, 1, \dots, n\}$  en el conjunto de nodos, o clientes, con el depósito situado en el nodo

0, y  $A$  es el conjunto de arcos que unen los nodos de  $V$ . Para cada arco  $a = (i, j), i \neq j$ , existe un coste no negativo asociado  $c_a$ . Para cada nodo, existe una demanda asociada,  $d_i$ , que será la cantidad recogida, y  $-d_i$ , que será la cantidad que deberá ser repartida. Y finalmente, se define  $O_i$ , que denota el vértice que es el origen de la demanda de entrega, y  $D_i$ , que denota el vértice que es el destino de la demanda de recogida.

El VRPPD consiste en encontrar una colección de exactamente  $K$  rutas simples con coste mínimo, y tal que:

- cada ruta visita el vértice del depósito;
- cada vértice del cliente es visitado por exactamente una ruta;
- la carga actual del vehículo a lo largo de la ruta debe ser no negativa y nunca podrá ser superior a la capacidad del vehículo,  $C$ ;
- para cada cliente  $i$ , el cliente  $O_i$ , cuando sea diferente del depósito, debe ser servido en la misma ruta y antes del cliente  $i$ ;
- para cada cliente  $i$ , el cliente  $D_i$ , cuando sea diferente del depósito, debe ser servido en la misma ruta y después del cliente  $i$ .

A menudo, el origen y destino de las demandas son comunes (por ejemplo, si están asociados con el depósito, como el CVRP), y por lo tanto no es necesario que se indique explícitamente. Este problema es conocido como el VRP *con recogida y entrega simultanea* (VRPSPD). Nuevamente, el VRPPD es NP-duro, pues generaliza el CVRP, que se da cuando  $O_i = D_i$  y  $-d_i = 0$ , para todo  $i \in V$ . Por último, el llamado TSP *con recogida y entrega* (TSPPD) es el caso especial del VRPPD en el cual  $K = 1$ . Para ampliar la información acerca del VRPPD, se puede consultar S.N. Parragh, K. Doerner and R.F. Hartl [7].

### 2.4.1. Formulación de tres índices

Para definir una formulación de tres índices para el VRPPD, es necesario introducir los siguientes conjuntos,  $P$  y  $D$ , conjuntos de nodos de recogida y conjunto de nodos de entrega, respectivamente. Además,  $n_p$  es el número de nodos de recogida,  $i = 1, \dots, n_p$ , y  $n_d$  es el número de nodos de entrega,  $i = n_p + 1, \dots, n_p + n_d$ , donde  $n = n_p + n_d$  es el número total de nodos. Para definir el modelo, consideraremos que el origen y destino de las demandas de todos los clientes es el depósito. Por lo tanto, el modelo se correspondería con la formulación de tres índices propuesta para el VRP sin capacidad, (2.6)-(2.12), incluyendo los siguientes conjuntos de restricciones:

$$w_j^k \geq w_i^k + (s_i + t_{ij})x_{ij}^k - T(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.39)$$

$$u_j^k \geq u_i^k + d_j x_{ij}^k - C(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.40)$$

$$\max\{0, d_i\}y_i^k \leq u_i^k \leq \min\{C, C + d_i\}y_i^k \quad \forall i \in V, k = 1, \dots, K, \quad (2.41)$$

Se introducen variables de tiempo en el conjunto de restricciones (2.39) para asegurar que no ocurran subciclos y, por tanto, podemos excluir las restricciones (2.10). Por otro lado, los conjuntos de restricciones (2.40) y (2.41) garantizan que la capacidad de los vehículos no se sobrepase durante el recorrido. Cabe destacar que las desigualdades (2.39) son las mismas que las propuestas para la formulación de tres índices del VRPTW, (2.33). De forma similar, la restricciones (2.40) y (2.41) son una variación de las propuestas para la formulación de tres índices del CVRP, (2.29) y (2.30), respectivamente, pero teniendo en cuenta que ahora a los clientes no solo se les recoge un tipo de mercancía, sino que también ésta puede ser entregada a los clientes.

Si cada punto de recogida estuviera asociado a un punto de entrega, se daría que  $n_p = n_d$ . Consideremos que el cliente  $n_p + i$  es el destino del producto que sale del cliente  $i$ . En este caso, además de (2.39)-(2.41), también sería necesario añadir dos conjuntos de restricciones más. En primer lugar, tanto el origen como el destino de una solicitud deben ser atendidos por el mismo vehículo:

$$\sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^+(n_p+i)} x_a^k = 0 \quad \forall i \in P, k = 1, \dots, K, \quad (2.42)$$

además, la entrega solo puede ocurrir después de la recogida, es decir:

$$w_i^k \leq w_{n_p+i}^k \quad \forall i \in P, k = 1, \dots, K. \quad (2.43)$$

También podríamos añadir ventanas de tiempo al problema con el siguiente conjunto de restricciones:

$$a_i y_i^k \leq w_i^k \leq b_i y_i^k \quad \forall i \in V, k = 1, \dots, K. \quad (2.44)$$

## 2.5. VRP con Backhauls (VRPB)

El VRP con *Backhauls* es la extensión del CVRP en la cual el conjunto de clientes  $V \setminus \{0\}$  está dividido en dos subconjuntos. El primero de ellos,  $L$ , contiene  $m$  clientes *Linehauls*, a los que debe ser servida una cantidad de producto. El segundo subconjunto,  $B$ , contiene  $n - m$  clientes *Backhauls*, a los que debe ser recogida una cantidad de producto dada. Los clientes se enumeran de forma que  $L = \{1, \dots, m\}$  y  $B = \{m + 1, m + 2, \dots, n\}$ . En el VRPB existe una relación de precedencia entre los diferentes tipos de clientes, así, cuando en una ruta se sirva a ambos tipos de clientes, todos los clientes *Linehaul* deben ser servidos antes que los *Backhaul*. Se define una demanda  $d_i$ , la cual será entregada o recogida en función del tipo de cliente que se trate. Al depósito se le asigna una demanda ficticia  $d_0 = 0$ .

Formalmente, definimos el VRPB de la siguiente forma. Sea  $G = (V, A)$  un grafo, donde  $V = \{0, 1, \dots, m, m + 1, \dots, n\}$  es el conjunto de nodos, o clientes, con el depósito situado en el nodo 0, y  $A$  es el conjunto de arcos que unen los nodos de  $V$ . El conjunto  $V \setminus \{0\}$  está dividido en los dos subconjuntos  $L$  y  $B$ , como comentamos anteriormente. Para cada

arco  $a = (i, j), i \neq j$ , existe un coste no negativo asociado  $c_a$ , y para cada cliente  $i$  existe una demanda  $d_i$ . Ahora, si sabemos que existe una flota de exactamente  $K$  vehículos en el depósito, y siendo  $K_L$  y  $K_B$  el número mínimo de vehículo necesarios para servir a los clientes *Linehaul* y *Backhaul*, respectivamente, para asegurar la viabilidad del problema, es necesario suponer que  $K$  no es más pequeño que el número mínimo de vehículos necesarios para servir a todos los clientes, es decir,  $K \geq \max\{K_L, K_B\}$ .

El VRPB consiste en encontrar una colección de exactamente  $K$  rutas simples con coste mínimo, y tal que:

- cada ruta visita el vértice del depósito;
- cada vértice del cliente es visitado únicamente por una ruta;
- la demanda total de los clientes *Linehaul* y *Backhaul* visitados en cada ruta no exceda, de forma separada, la capacidad del vehículo,  $C$ ;
- en cada ruta todos los clientes *Linehaul* preceden a los clientes *Backhaul*, si los hay.

Una vez más, el VRPB es NP-duro, pues no es más que una generalización de la versión básica del CVRP, que surge cuando  $B = \emptyset$ . Por otra parte, los llamados TSP con Backhauls son un caso especial del VRPB en el cual  $C \geq \max\{d(L), d(B)\}$  y  $K = 1$ .

### 2.5.1. Formulación de tres índices

Para introducir una formulación de tres índices para el VRPB debemos tener en cuenta que, en cada ruta, todos los clientes *Linehauls* tienen que ser visitados antes que los clientes *Backhauls*. Además, la demanda total de ambos tipos de clientes, por separado, no debe exceder la capacidad del vehículo. Por lo tanto, el modelo para este problema se correspondería con la formulación de tres índices propuesta para el VRP sin capacidad, (2.6)-(2.12). A esta formulación debemos añadir los conjuntos de restricciones que nos garanticen la precedencia de los clientes *Linehauls*, y que no se exceda la capacidad del vehículo, como citamos al principio:

$$w_j^k \geq w_i^k + (s_i + t_{ij})x_{ij}^k - T(1 - x_{ij}^k) \quad \forall i, j \in V \setminus \{0\}, k = 1, \dots, K, \quad (2.45)$$

$$w_i^k \leq w_j^k \quad \forall i \in L, j \in B, k = 1, \dots, K, \quad (2.46)$$

$$\sum_{i \in L} d_i y_i^k \leq C \quad \forall k = 1, \dots, K, \quad (2.47)$$

$$\sum_{i \in B} d_i y_i^k \leq C \quad \forall k = 1, \dots, K, \quad (2.48)$$

$$\sum_{i \in L} \sum_{j \in B} x_{ij}^k = 1 \quad \forall k = 1, \dots, K, \quad (2.49)$$

$$\sum_{j \in L} \sum_{i \in B} x_{ij}^k = 0 \quad \forall k = 1, \dots, K. \quad (2.50)$$

Con el conjunto de desigualdades (2.45) se garantiza que no se generen subciclos, por lo que no sería necesario utilizar (2.10). Las restricciones (2.46) imponen que, en cada ruta, todos los clientes *Linehauls* sean visitados antes que cualquier cliente *Backhaul*. Por otro lado, los conjuntos de restricciones (2.47) y (2.48) garantizan que la capacidad de los vehículos no se sobrepase en ninguna de las rutas, en concreto, aseguran que la suma total de las demandas de los clientes *Linehauls* y *Backhauls* visitados en cada ruta, no sea superior, por separado, a la capacidad de los vehículos. Por último, las restricciones (2.49) y (2.50) nos aseguran que solo existe un único arco que vaya desde el subconjunto de clientes *Linehauls* al subconjunto de clientes *Backhauls* y, por el contrario, no existe ningún arco que vaya desde el subconjunto de clientes *Backhauls* al subconjunto de clientes *Linehauls*.

## Capítulo 3

# Implementaciones en GUSEK

GUSEK (GLPK Under Scite Extended Kit) es una interfaz libre para windows que utiliza un motor de optimización para resolver modelos matemáticos lineales. Esta herramienta es GLPK (GNU Linear Programming Kit), y es una alternativa gratuita a programas como XPRESS o CEPLEX. Tanto XPRESS como CEPLEX son programas más potentes a la hora de resolver problemas muy complejos con gran cantidad de variables, pero la desventaja de ambos es que su licencia es de pago.

Tiene un lenguaje de modelado denominado GMPL [1]. El GNU Linear Programming Kit (GLPK) es un software desarrollado por Andrew O. Makhorin del instituto de Aviación de Moscú, cuya primera versión pública fue en octubre del 2000. Se trata de un software para la resolución de problemas a gran escala de programación lineal (LP), programación entera mixta (MIP), y otros problemas relacionados. Es un conjunto de rutinas escritas en ANSI C y organizadas en forma de una librería.

Los problemas pueden ser modelados en el idioma GNU MathProg, que comparte muchas partes de la sintaxis con AMPL, y resueltos con GLPSOL que se trata de un solucionador independiente. GLPK también puede ser utilizado como una biblioteca de C. GLPK utiliza el método simplex revisado y el método del punto interior primal-dual para problemas no enteros y el algoritmo *Branch-and-Bound* junto con el algoritmo de cortes enteros mixtos de Gomory para problemas (mixto) enteros.

A continuación, hemos implementado en GUSEK una formulación de cada una de las variantes del VRP descritas en el capítulo anterior.

### 3.1. Implementación del VRP sin capacidad

El modelo del VRP sin capacidad que hemos elegido para implementar en el citado software matemático (GUSEK), es la formulación de tres índices descrita en 2.1.2. Para implementar este modelo es necesario definir, como mostramos a continuación, los parámetros, conjuntos de valores y variables de decisión que se utilizan en esta formulación.

```

*****
* /* VRP sin capacidad (Formulación de tres índices) *
*****

param n, integer, >= 3;
/* Número de nodos */

param m, integer, >=0;
/* Número de vehículos homogéneos con igual capacidad*/

param q, integer >=0;
/*Capacidad máxima de los vehículos*/

set V:= 1..n;
/* Conjuntos de nodos, donde 1 indica el depósito*/

set A, within V cross V, default V cross V;
/* Conjunto de arcos que unen los nodos de V */

set K := 1..m;
/*Flota de vehículos*/

param d{i in V};
/*demanda de cada cliente i*/

param c{i in V, j in V};
/*Costes de cada arco (i,j) */

var x{(i,j) in A, k in K}, binary;
/* Variable binaria que toma el valor 1 si el arco (i,j)
es atravesado por el vehículo k, y 0 en otro caso */

var y{i in V, k in K}, binary;
/*variable binaria que toma el valor 1 si
el cliente i es visitado por el vehículo k, y 0 en otro caso*/

var u{i in V, k in K};
/*Variable que indica la carga después de visitar al cliente i*/

```

Como se puede comprobar el lenguaje puede resultar intuitivo. Se utiliza la palabra reservada *param* para definir los parámetros como  $n$  (número de nodos),  $m$  (número de vehículos),  $q$  (capacidad de los vehículos),  $d_i$  (demanda de cada cliente) y  $c_{ij}$  (coste de cada arco). Para introducir los conjuntos de valores utilizamos la palabra reservada *set*, estos son,  $V$  (conjunto de nodos),  $A$  (conjunto de arcos) y  $K$  (conjunto de vehículos). Por último, para las variables de decisión, se usa la palabra reservada *var* para introducir las variables  $x_{ijk}$ ,  $y_{ik}$  y  $u_{ik}$ .

Una vez definidos los parámetros, conjuntos y variables del modelo, únicamente faltaría introducir la función objetivo y los conjuntos de restricciones que conforman la formulación del VRP sin capacidad descrita en 2.1.2.

```

minimize total: sum{k in K, (i,j) in A: i!=j} c[i,j] * x[i,j,k];

/* El objetivo es satisfacer a todos los clientes de forma que el coste sea mínimo */

s.t. visit{i in V: i!=1}: sum{k in K} y[i,k] = 1;

/*Cada cliente debe ser visitado por un único vehículo*/

s.t. vehi{k in K}: y[1,k] = 1;
/*Todos los vehículos deben salir del depósito*/

s.t. enter{i in V, k in K}: sum{j in V: i!=j} x[i,j,k] = y[i,k];
s.t. leave{i in V, k in K}: sum{j in V: i!=j} x[j,i,k] = y[i,k];

/* Cada cliente es visitado una única vez*/

s.t. pos{i in V, j in V, k in K: i!=1 and j!=1 and j!=i}:
    u[j,k] >= u[i,k] + x[i,j,k] - (n-1)*(1-x[i,j,k]) + (n-2)*x[j,i,k]

/* No se generan subciclos que no pasen por el depósito*/

```

Para finalizar la implementación del modelo, se introduce la función objetivo utilizando la palabra reservada *minimize* y, a continuación, definiendo dicha función. Y, por último, la sección de restricciones se introduce, cada una de ellas, utilizando inicialmente el comando *s.t.*. Tanto la función objetivo como las restricciones incluyen un rótulo con el cual se podrá identificar en la solución.

Con esto quedaría implementado el modelo del VRP sin capacidad en el programa GUSEK. Ahora, tan solo sería necesario añadir los datos que correspondan con el problema que se desee resolver y el programa devolverá una solución óptima, si la hay.

### 3.2. Implementación del CVRP

Para la implementación del CVRP en GUSEK hemos elegido la formulación de tres índices descrita en 2.2.2. Como describimos en el capítulo anterior esta formulación del CVRP es igual que la descrita para el VRP sin capacidad incluyendo los nuevos conjuntos de restricciones de capacidad. Por lo tanto, este modelo se implementaría en el programa matemático GUSEK de una forma muy similar a la que acabamos de describir para el VRP sin capacidad. Solo tendríamos que incluir los nuevos conjuntos de restricciones, tal como mostramos a continuación.

```

s.t. carg{k in K, i in V, j in V: i!=j and i!=1 and j!=1}:
    u[j,k] >= u[i,k] + d[j]*x[i,j,k] - q*(1-x[i,j,k]);

```

```
s.t. capp{k in K, i in V}: d[i] <= u[i,k] <= q;
```

```
\*Estas restricciones evitan los subciclos además garantizan las condiciones de capacidad*/
```

### 3.3. Implementación del VRPTW

Para implementar el VRPTW en GUSEK hemos elegido la formulación de tres índices que describimos en 2.3.1. De nuevo, como comentamos en el capítulo anterior, dicha formulación del VRPTW es casi idéntica a la formulación de tres índices propuesta para el VRP sin capacidad, con la diferencia de que debemos incluir una serie de nuevos conjuntos de restricciones. Entonces, el modelo del VRPTW se implementaría en el programa matemático GUSEK de un modo muy similar al que describimos para el VRP sin capacidad. Pues, solo tendríamos que añadir, como mostramos seguidamente, los nuevos conjuntos de restricciones del problema.

```
s.t. windows1{i in V, k in K: i!=0}: (a[i] * y[i,k]) <= w[i,k];
```

```
s.t. windows2{i in V, k in K: i!=0}: w[i,k] <= (b[i] * y[i,k]);
```

```
/* Cada cliente i debe ser vistado dentro de su ventana de tiempo*/
```

```
s.t. subcicl1{i in V, j in V, k in K: i!=j and j!=0}: w[j,k] >= w[i,k] +
      (s[i] + c[i,j]) * x[i,j,k] - (M * (1 - x[i,j,k]));
```

```
/* Con esta restricción de presedencia se evitan subciclos*/
```

```
s.t. cap{k in K}: sum{i in V: i!=0} (d[i] * y[i,k]) <= q;
```

```
/* La suma total de las demandas de los clientes que se visitan en cada
ruta no debe exceder la capacidad de los vehículos*/
```

### 3.4. Implementación del VRPPD

En este caso, hemos implementado el modelo del VRPPD descrito en 2.4.1. Como describimos en el capítulo anterior, la formulación del VRPPD es semejante a la que definimos para el VRP sin capacidad añadiendo una serie de nuevos conjuntos de restricciones. Por ello, este modelo quedaría implementado en el programa matemático GUSEK de una forma similar a la descrita para el VRP sin capacidad. Tan solo sería necesario incorporar, tal como mostramos a continuación, los nuevos conjuntos de restricciones del problema.

```
s.t. subcicl{i in V, j in V, k in K: i!=j and j!=0 and i!=0}: w[j,k] >= w[i,k] +
      (s[i] + c[i,j]) * x[i,j,k] - (M * (1 - x[i,j,k]));
```

```
/* Con esta restricción de presedencia se evitan subciclos*/
```

```

s.t. carg{i in V, j in V, k in K: i!=j}:
    u[j,k] >= u[i,k] + d[i]*x[i,j,k] - q*(1-x[i,j,k]);

s.t. capacity{i in V, k in K: i!=0}: max(0,d[i]) <= u[i,k] <= min(q, q + d[i]);

/*Estas restricciones garantizan las condiciones de capacidad*/

```

### 3.5. Implementación del VRPB

Para el VRPB hemos implementado en el programa matemático GUSEK la formulación que describimos en 2.5.1. Nuevamente, los parámetros, conjuntos de valores y variables de decisión del modelo se implementan de un modo similar a los problemas anteriores. Una vez más, el modelo para el VRPB es el descrito en el capítulo anterior para la formulación de tres índices del VRP sin capacidad, añadiendo una serie de nuevos conjuntos de restricciones. Por lo tanto, la implementación de este modelo en el programa GUSEK es muy similar a la descrita para el VRP sin capacidad, pues tan solo habría que añadir, como mostramos a continuación, los nuevos conjuntos de restricciones.

```

s.t. subcicl{i in V, j in V, k in K: i!=j and j!=0 and i!=0}: w[j,k] >= w[i,k] +
    (s[i] + c[i,j]) * x[i,j,k] - (M * (1 - x[i,j,k]));

/* Con esta restricción de presedencia se evitan subciclos */

s.t. cargLine{k in K}: sum{i in L} y[i,k]* d[i] <= q;

s.t. cargBack{k in K}: sum{i in B} y[i,k]*(-d[i]) <= q;

/* La carga cuando se sirve a clientes Linehaults y Backhaults no
excede, por separado, la capacidad del vehículo. */

s.t. preced{k in K, i in L, j in B}: w[i,k] <= w[j,k];

/* Todos los cliente Linehaults se sirven antes que los Backhaults */

s.t. L_a_B{k in K}: sum{i in L, j in B} x[i,j,k] =1;

s.t. B_a_L{k in K}: sum{i in B, j in L} x[i,j,k] =0;

/* Solo existe un arco que va de un cliente Linehaul a uno Backhaul
y no a la inversa. */

```

## Capítulo 4

# Resultados computacionales

En este capítulo presentaremos en tablas los resultados computacionales obtenidos con la formulación de cada variante del VRP que escogimos en el capítulo anterior. Para analizar los resultados se han escogido diversos ejemplos utilizando diferente número de nodos, vehículos y capacidad. La razón de no resolver ejemplos con más nodos o vehículos se debe al tiempo de espera de ejecución de éstos, pues a medida que aumentamos la cantidad de nodos, el tiempo de ejecución crece exponencialmente. Como comentamos en el capítulo anterior, con herramientas como *Xpress* o *CPLEX* los tiempos de ejecución disminuirían considerablemente, con lo que sería posible realizar ejemplos de mayor tamaño en tiempos razonables.

En lo referente al equipo informático, hemos usado un ordenador con un procesador *intel core i7-6500U 2.5GHz with Turbo Boost up to 3.1GHz* y memoria *RAM DDR3 de 8 GB*.

Debido a la gran cantidad de ejemplos que hemos realizado, se considera que el problema a tardado "demasiado" si el tiempo de ejecución es superior a una hora. Por ello se ha puesto un tiempo límite de ejecución, tras el cual, el programa devuelve la mejor solución factible que ha encontrado hasta el momento, la cual no tiene porque ser la óptima.

Antes de presentar las distintas tablas de resultados de cada variante del VRP, explicaremos que datos aparecen en cada columna. En la tres primeras se recogen la cantidad de nodos, vehículos y la capacidad (esta última columna se excluye en el caso del VRP sin capacidad) de cada problema, respectivamente. El número de nodos de cada ejemplo representa los clientes y depósitos y, como explicamos al inicio del capítulo dos, describimos las variantes del VRP para el caso de un solo depósito, por lo que, uno de los nodos equivale al depósito y el resto se corresponde con los clientes. Por otro lado, en la columna con título *Rows* se expone el número de restricciones que contiene cada ejemplo y en *Columns* el número de variables. En la columna de nombre *Iter* se muestra el número de iteraciones que se realizaron hasta encontrar la solución. La columna *UB*, representa el valor de la cota superior de cada problema y, en *Opt*, se indica si dicho valor es el óptimo o no. La columna titulada *Diff* representa la diferencia porcentual entre el valor obtenido

y la mejor solución que podría llegar a encontrarse (si dicho valor es el óptimo esta diferencia será del 0%). Finalmente, en las dos últimas columnas, *Time* y *Mem*, se recogen los tiempos de ejecución, en segundos, y la memoria, en megabytes (MB), utilizada en la ejecución, respectivamente.

#### 4.1. Resultados del VRP sin capacidad

Para el VRP sin capacidad, la formulación escogida para ser implementada en la herramienta matemática GUSEK, ha sido la formulación de tres índices, como detallamos en el capítulo tres. Con el propósito de analizar el comportamiento de dicha formulación a la hora de resolver problemas concretos, se han generado diversos ejemplos con diferente número de nodos y vehículos. En la tabla 4.1, que veremos a continuación, se recogen los resultados obtenidos de diferentes ejemplos que tienen de 21 a 25 nodos y utilizan de 2 a 5 vehículos. En cada ejemplo, las demandas de los clientes toman valores comprendidos en el intervalo  $(0, 25]$ , y son distintas para cada uno de ellos. La suma total de las demandas de los clientes en ningún caso es inferior a la capacidad de los vehículos, pues de lo contrario, se trataría de resolver un TSP. Los costes (o distancia) entre nodos de cada ejemplo, toman valores recogidos en el intervalo  $(0, 100)$  pero, al contrario que las demandas, los costes son iguales en cada uno de ellos, es decir, el depósito y los clientes, en cada ejemplo, tienen las mismas localizaciones y, por lo tanto, las mismas distancias entre ellos.

Nodos	K	Rows	Columns	Iter	UB	Opt	Diff	Time	Mem
21	2	867	922	14951	477	Si	0.0	4.7	3.7
	3	1290	1383	35929	528	Si	0.0	16.2	7.0
	4	1713	1844	165931	584	Si	0.0	94.5	20.6
	5	2136	2305	125863	641	Si	0.0	81.4	19.7
22	2	952	1010	20716	477	Si	0.0	7.3	4.9
	3	1417	1515	214667	528	Si	0.0	103.2	25.9
	4	1882	2020	468815	584	Si	0.0	246.6	53.0
	5	2347	2525	799514	641	Si	0.0	574.3	76.9
23	2	1041	1102	84890	490	Si	0.0	30.4	10.4
	3	1550	1653	471468	545	Si	0.0	246.2	66.3
	4	2059	2204	923329	601	Si	0.0	603.3	103.1
	5	2568	2755	1771658	658	Si	0.0	1598.1	182.1
24	2	1134	1198	72648	497	Si	0.0	27.6	8.2
	3	1689	1797	434854	542	Si	0.0	249.2	74.1
	4	2244	2396	2973204	597	No	1.7	+3600.0	508.6
	5	2799	2995	2416572	653	No	2.5	+3600.0	670.2
25	2	1231	1298	585705	498	Si	0.0	230.8	66.8
	3	1834	1947	1977977	543	Si	0.0	1543.2	210.7
	4	2437	2596	1649553	598	No	3.3	+3600.0	500.0
	5	3040	3245	1459377	654	No	3.2	+3600.0	523.2

Cuadro 4.1: Resultados computacionales del VRP sin capacidad.

En el cuadro 4.1, se puede observar como según crece el número de nodos y vehículos de

cada problema, los tiempos de ejecución, así como la memoria utilizada, aumentan notablemente. Lo cual podría parecer lógico, pues, a más nodos y vehículos tenga el problema, más restricciones y variables lo conformarán, lo que hace mayor el problema y, por tanto, más difícil de resolver.

## 4.2. Resultados del CVRP

En esta ocasión, para el CVRP, se han generado numerosos ejemplos con el objetivo de analizar como se comporta la formulación de tres índices, que hemos propuesto en este trabajo, a la hora de resolver un problema concreto. Hemos creado diferentes ejemplos que tienen entre 15 y 20 nodos, que utilizan de 2 a 5 vehículos, y que cuentan con capacidades que van desde 50 hasta 200 unidades de producto que pueden transportar. Las demandas de los clientes elegidas toman valores comprendidos en el intervalo  $(0, 25]$ , y éstas son distintas en cada caso particular. La capacidad de los vehículos será siempre menor a la suma de las demandas de todos los clientes, pues de nos ser así estaríamos hablando del TSP. Finalmente, los costes, o distancias entre los nodos, nuevamente, son los mismos para todos los ejemplos que hemos realizado.

Nodos	K	C	Rows	Columns	Iter	UB	Opt	Diff	Time	Mem
15	3	100	741	720	1785950	530	Si	0.0	1080.4	201.0
		150	741	720	80864	494	Si	0.0	30.5	12.4
	4	100	983	960	2292176	574	Si	0.0	1856.4	331.3
		180	983	960	389040	550	Si	0.0	229.7	48.5
17	3	100	941	918	2688152	536	No	3.0	+3600.0	468.2
		130	941	918	2411092	520	Si	0.0	2383.2	316.3
		150	941	918	88664	515	Si	0.0	19.4	11.0
	4	130	1249	1224	761653	571	Si	0.0	445.4	107.1
		150	1249	1224	417798	571	Si	0.0	145.0	77.1
	5	180	1249	1224	129038	559	Si	0.0	93.6	21.0
		150	1557	1530	1802959	615	Si	0.0	2440.7	349.6
		180	1557	1530	305455	615	Si	0.0	237.3	61.4
18	3	130	1050	1026	2880851	526	No	3.2	+3600.0	583.9
		150	1050	1026	411257	517	Si	0.0	244.1	58.7
		180	1050	1026	56656	505	Si	0.0	32.2	9.9
	4	150	1394	1368	3036685	571	No	1.2	+3600.0	595.7
		180	1394	1368	605262	559	Si	0.0	482.4	117.7
19	2	150	783	760	875214	478	Si	0.0	520.8	93.5
		180	783	760	210646	466	Si	0.0	130.4	23.1
	3	150	1165	1140	436437	528	Si	0.0	286.4	56.0
		180	1165	1140	367358	517	Si	0.0	252.7	47.1
	4	180	1547	1520	1121748	573	Si	0.0	1234.1	156.2
		200	1547	1520	1121748	573	Si	0.0	1234.1	156.2
20	3	150	1286	1260	1666090	529	Si	0.0	1100.4	204.0
		180	1286	1260	407782	518	Si	0.0	294.4	53.7
	4	200	1708	1680	782496	574	Si	0.0	749.4	103.9

Cuadro 4.2: Resultados computacionales del CVRP

Como podemos observar en el cuadro 4.2, de igual modo que para el VRP sin capacidad, a medida que aumentamos el número de nodos y vehículos del problema, los tiempos de ejecución crecen considerablemente. También es notable, en comparación con el VRP sin capacidad, que para cantidades similares de nodos y vehículos, los tiempos computacionales del CVRP son bastantes más elevados. Esto se debe a que el CVRP incluye restricciones de

capacidad, con lo que los problemas de esta variante del VRP poseen un mayor número de restricciones para instancias similares. Se observa que la capacidad influye en estos tiempos de ejecución, pues con capacidades más pequeñas los problemas parecen tardar más, en cambio, con mayor capacidad los tiempos se reducen. Podemos ver también como para instancias con igual número de nodos y vehículos, cuando la capacidad es más elevada, la solución mejora en la mayoría de los casos.

### 4.3. Resultados del VRPTW

Ahora, para la formulación de tres índices del VRPTW, que ha sido implementada en GUSEK, se han generado diversos ejemplos variando el número de nodos, vehículos y capacidad de éstos. Con el objetivo de analizar el comportamiento de dicha formulación, se han recogido los resultados de cada uno de los ejemplos propuestos en la tabla 4.3, que presentamos a continuación. Estos ejemplos tienen de 15 a 19 nodos, utilizan de 2 a 4 vehículos y cuentan con capacidades desde 100 a 150 unidades de producto que puede transportar. Las ventanas de tiempo de cada cliente son intervalos de tiempo comprendidos dentro del período total de servicio, y los expresamos en minutos. Estas ventanas de tiempo varían para cada cliente en las distintas instancias del problema. La demandas, nuevamente, son distintas en cada ejemplo y toman valores comprendidos entre  $(0, 25]$ . La suma total de las demandas de los clientes es mayor que la capacidad de los vehículos. Finalmente, los costes, que en este caso son equivalentes a los tiempos que se tarda en viajar de un nodo a otro, van de 0 a 30, representados en minutos, y son los mismos en cada ejemplo.

Nodos	K	C	Rows	Columns	Iter	UB	Opt	Diff	Time	Mem
15	2	100	527	480	65691	157	Si	0.0	8.6	7.4
		120	527	480	215633	157	Si	0.0	28.0	21.5
	4	100	1039	960	501030	185	Si	0.0	139.0	71.0
		120	1039	960	418832	185	Si	0.0	104.5	59.7
16	2	100	594	544	237146	149	Si	0.0	34.0	21.9
		120	594	544	210254	125	Si	0.0	30.9	19.2
	3	100	883	816	477213	165	Si	0.0	153.8	54.6
		120	883	816	508638	159	Si	0.0	111.9	49.7
	4	100	1172	1088	543635	181	Si	0.0	159.2	83.2
		120	1172	1088	527363	147	Si	0.0	139.7	63.6
17	2	100	665	612	1852146	135	Si	0.0	745.8	165.9
		150	665	612	893719	128	Si	0.0	383.9	98.8
	3	100	989	918	1378575	168	Si	0.0	511.8	157.6
		150	989	918	473987	140	Si	0.0	132.6	69.2
	4	100	1313	1224	3546150	153	No	3.9	+3600.0	921.8
		150	1313	1224	1818503	153	Si	0.0	1306.8	283.3
19	2	100	819	760	4622368	141	No	5.7	+3600.0	657.5
		150	819	760	2853018	131	Si	0.0	1586.6	313.0
	3	100	1219	1140	3500256	143	Si	0.0	2591.9	428.6
		150	1219	1140	3315172	141	Si	0.0	3347.0	451.9
	4	100	1619	1520	3883586	158	No	9.5	+3600.0	1360.8
		150	1619	1520	3507725	154	No	4.5	+3600.0	1007.4

Cuadro 4.3: Resultados computacionales del VRPTW

En los resultados recogidos en el cuadro 4.3, vemos nuevamente como los tiempos computacionales aumentan notablemente a medida que crece el número de nodos y vehículos del problema. De igual modo que para el CVRP, la capacidad también influye en los tiempos de ejecución, pues notamos que en general, con menores capacidades, el programa requiere un mayor esfuerzo para lograr una solución óptima. Cabe señalar que las ventanas de tiempo introducidas repercuten de un modo destacado en la búsqueda de esta solución. En comparación con el CVRP, podemos observar que con esta formulación, el número de restricciones de las distintas instancias es más elevado, pero la cantidad de variables es la misma.

#### 4.4. Resultados del VRPPD

En cuanto al VRPPD, hemos utilizado la formulación que se propuso en 2.4.1 y que implementamos en GUSEK así como se explica en 3.4, con el objetivo de estudiar el comportamiento de ésta. Para ellos se han analizado numerosas instancias del problema, variando los nodos, vehículos y capacidad en cada ejemplo. Éstos tienen entre 15 y 19 nodos, utilizan de 2 a 4 vehículos y poseen capacidades de entre 50 y 75 unidades de producto que pueden transportar. Las demandas de los clientes que se han utilizado varían en cada ejemplo, tanto en las cantidades, como en el tipo, es decir, si la demanda se trata de una cantidad de producto que hay que entregar, o por el contrario recoger. Por ello, las demandas en cada uno de los ejemplos toman valores comprendidos en el intervalo  $[-25, 25]$ , siendo estos valores negativos cuando se trata de una entrega, y positivos en el caso de una recogida. Para terminar, los costes introducidos varían dentro del intervalo  $(0, 100)$ , pero se utilizan los mismos en cada ejemplo.

Nodos	K	C	Rows	Columns	Iter	UB	Opt	Diff	Time	Mem
13	2	50	695	414	13826	431	Si	0.0	1.8	2.4
		75	695	414	15216	431	Si	0.0	2.4	2.8
	3	50	1036	621	40643	499	Si	0.0	7.9	6.8
		75	1036	621	56775	499	Si	0.0	10.5	7.7
15	2	50	921	538	216504	462	Si	0.0	35.9	18.7
		75	921	538	247101	462	Si	0.0	37.3	16.5
	3	50	1374	807	1354218	529	Si	0.0	605.3	133.7
		75	1374	807	1204914	519	Si	0.0	473.1	120.4
17	2	50	1179	678	249153	472	Si	0.0	55.9	25.0
		75	1179	678	37427	459	Si	0.0	8.3	5.5
	3	50	1760	1017	2284068	528	Si	0.0	1490.1	258.2
		75	1760	1017	927440	515	Si	0.0	383.4	100.0
	4	50	2341	1356	3219885	596	No	3.0	+3600.0	731.8
		75	2341	1356	3424677	583	No	2.7	+3600.0	622.4
19	2	50	1469	834	587885	474	Si	0.0	192.7	60.3
		75	1469	834	419641	469	Si	0.0	121.7	53.9
	3	50	2194	1251	3388584	536	No	2.4	+3600.0	638.1
		75	2194	1251	3537579	536	No	3.7	+3600.0	652.8
	4	50	2919	1668	3346420	611	No	7.4	+3600.0	1229.7
		75	2919	1668	3333424	614	No	7.2	+3600.0	1065.2

Cuadro 4.4: Resultados computacionales del VRPPD

Como observamos en el cuadro 4.4, los tiempos computacionales aumentan considerablemente a medida que crecen los nodos y vehículos del ejemplo, de igual modo que para las otras formulaciones que estudiamos anteriormente. Cabe destacar que el mayor uso de vehículos complica significativamente la resolución del problema, pues los tiempos de ejecución para ejemplos con igual número de nodos son relativamente bajos cuando utilizan 2 vehículos. Sin embargo, si usan 3 ó 4 vehículos, estos tiempos son mucho más elevados. La capacidad influye de una manera similar a la de los casos del CVRP y del VRPTW. Por último, señalar que esta formulación utiliza un mayor número de restricciones, en comparación con las estudiadas anteriormente para las otras variantes del VRP.

## 4.5. Resultados del VRPB

Para el caso del VRPB, se ha implementado en la herramienta matemática GUSEK, la formulación propuesta en 2.5.1, tal como detallamos en 3.5. Para poder analizar dicha formulación a la hora de resolver problemas concretos, se han generado numerosos ejemplos, con el fin de estudiar los resultados obtenidos. Estos ejemplos varían en número de nodos, vehículos y capacidad. Además, debido ha que en el VRPB los clientes se dividen en clientes *Linehaul* y *Backhaul*, también varían las cantidades de cada tipo de cliente en los diferentes ejemplos. Por lo tanto, añadiremos a la siguiente tabla de resultados, 4.5, las dos columnas *Line* y *Back*, las cuales indican el número de clientes de cada tipo que hay en cada ejemplo. Los diferentes ejemplos que hemos creado tienen entre 17 y 23 nodos, utilizan de 2 a 5 vehículos y cuentan con capacidades que van de 40 a 70 unidades de producto que pueden transportar. Las demandas son distintas en cada caso y toman valores comprendidos en el intervalo  $(0, 25]$ . La suma de las demandas de cada tipo de cliente, por separado, siempre es mayor que la capacidad de lo vehículos. Por último, los costes son iguales para todos los ejemplos y son valores que van de 0 a 100.

Nodos	K	Line	Back	C	Rows	Columns	Iter	UB	Opt	Diff	Time	Mem
17	3	8	8	40	1046	915	22597	636	Si	0.0	5.0	4.1
				60	1046	915	7394	593	Si	0.0	1.3	2.6
	4	8	8	40	1389	1220	50760	677	Si	0.0	12.2	7.4
				60	1389	1220	18267	659	Si	0.0	4.5	4.3
19	3	8	10	50	1306	1137	52512	657	Si	0.0	13.4	7.7
		9	9	60	1309	1137	13459	622	Si	0.0	2.8	3.5
		10	8	50	1306	1307	88216	647	Si	0.0	21.0	9.5
		11	7	60	1297	1137	65114	626	Si	0.0	15.7	7.8
21	2	10	10	70	1075	922	35242	643	Si	0.0	6.3	4.6
	3	10	10	50	1602	1383	441373	693	Si	0.0	137.5	40.2
				60	1602	1383	219424	684	Si	0.0	57.1	19.5
	4	10	10	40	2129	1844	3778421	767	Si	0.0	2610.3	340.6
5	10	10	40	2651	2305	3653032	824	Si	0.0	2715.4	294.8	
23	2	11	11	60	1291	1102	433833	675	Si	0.0	92.5	31.6
	3	11	11	60	1925	1653	506426	701	Si	0.0	143.7	43.0
				70	1925	1653	99919	671	Si	0.0	31.0	11.1
	4	11	11	40	2559	2204	4106760	780	No	2.2	+3600.0	512.8
60				2559	2204	412267	738	Si	0.0	153.1	35.6	

Cuadro 4.5: Resultados computacionales del VRPB

En el cuadro 4.5 se observa, una vez más, conforme aumenta el número de nodos y vehículos del ejemplo, los tiempos computacionales se incrementan rápidamente. La capacidad repercute en los tiempos de ejecución de una forma parecida a como lo hace en los anteriores problemas que hemos estudiado. Además, parece influir muy notablemente en estos tiempos, pues vemos que aumentan considerablemente para capacidades algo inferiores. Las distintas instancias de esta variante poseen una cantidad de restricciones algo superior a las del CVRP. Finalmente, señalar que la cantidad de cada tipo de clientes en un ejemplo, repercute de forma destacada en la obtención del resultado.

## Capítulo 5

# Conclusiones

En este trabajo hemos introducido uno de los problemas más importantes y estudiados de la optimización combinatoria. Su importancia reside sobre todo en el gran número de aplicaciones reales que existen tales como el reparto de mercancías o la recogida de basuras. Además, existe un amplio conjunto de variantes y adaptaciones de este problema.

En cuanto al desarrollo de este proyecto, cabe destacar la familiarización con algunas de las variantes más conocidas, y con diferentes formulaciones que permiten resolverlas. Así como, el aprendizaje de un lenguaje de modelado denominado GMPL, que nos permite resolver estos problemas utilizando un software matemático llamado GUSEK. También ha sido necesario familiarizarse con este software matemático y sus distintas características, como puede ser el solucionador GLPSOL.

En lo referente a la experiencia computacional, hemos analizado el comportamiento de una de las formulaciones propuestas de cada variante, a la hora de resolver distintos ejemplos con parámetros concretos. Se ha comprobado que los métodos propuestos encuentran la mejor solución al problema aunque con un costo computacional muy alto, siendo apropiados en problemas pequeños, pero no para problemas de dimensiones mayores. Se ha podido ver como alguna de las variantes entrañan una mayor dificultad a la hora de obtener una solución óptima en una instancia concreta, y los tiempos de ejecución son más elevados frente a otras variantes más sencillas como el VRP sin capacidad. Para cada variante, los tiempos computacionales dependen de los parámetros introducidos en cada ejemplo, pues el número de nodos y vehículos, así como la capacidad elegida, influyen en estos tiempos. Las ventanas de tiempo en el VRPTW y el número de cada tipo de clientes en el VRPB son valores que también repercuten a la hora de hallar la solución del problema. Cabe destacar que con herramientas matemáticas con Xpress o CPLEX, la obtención de los resultados computacionales hubiese resultado algo más cómoda, pues dichas herramientas son más potentes a la hora de resolver problemas muy complejos, pero no hemos podido acceder a ellas dado que ambas son de pago.

Finalmente, comentar que existe un gran número de líneas de investigación relacionadas

con los problemas de rutas de vehículos y que podrían servir para futuros trabajos de fin de grado. Podrían ser propuestas variantes con más de un depósito, con flotas de vehículos heterogéneas o incluso combinar algunas de las variantes propuestas en este trabajo, por ejemplo, el VRP con *recogida y entrega* y *ventanas de tiempo*.

# Bibliografía

- [1] *Modeling Language GNU MathProg*, February 2016. Language Reference for GLPK Version 4.58.
- [2] Golden Bruce, S. Raghavan, and E. Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. 2008.
- [3] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12:568–581, 1964.
- [4] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management science*, 1959.
- [5] G. Gutin and A. P. Punnen. *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media, 2006.
- [6] A. N. Letchford and J.J. Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2-3):251–274, 2006.
- [7] S. N. Parragh, K. Doerner, and R. F. Hartl. A survey on pickup and delivery models part ii: Transportation between pickup and delivery locations. Technical report, Working paper, Chair of Production and Operations Management, University of Vienna, 2006.
- [8] P. Toth and D. Vigo. *The vehicle routing problem*. Philadelphia : SIAM, cop., 2002.
- [9] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*, volume 18. Siam, 2<sup>a</sup> edition, 2014.