



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

**SOLUCIÓN AL PROBLEMA DE  
RUTEO DE VEHÍCULOS CON  
RESTRICCIONES DE CAPACIDAD  
Y REORDENAMIENTO DE CARGA  
EN LOS SITIOS DE DEMANDA.**

**ACHILLE LIMONE MUÑOZ**

Tesis para optar al grado de  
Magister en Ciencias de la Ingeniería

Profesor Supervisor:  
**RICARDO GIESEN ENCINA**

Santiago de Chile, Agosto 2011.

© 2011, Achille Limone Muñoz



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE  
ESCUELA DE INGENIERIA

# **SOLUCIÓN AL PROBLEMA DE RUTEO DE VEHÍCULOS CON RESTRICCIONES DE CAPACIDAD Y REORDENAMIENTO DE CARGA EN LOS SITIOS DE DEMANDA.**

**ACHILLE LIMONE MUÑOZ**

Tesis presentada a la comisión integrada por los profesores:

**RICARDO GIESEN**

**JUAN CARLOS MUÑOZ**

**LUIS CONTESSE**

**FERNANDO ORDOÑEZ**

**JORGE VERA**

Para completar las exigencias del grado de  
Magister en Ciencias de la Ingeniería

Santiago de Chile, Agosto 2011.

A mi familia, por su extraordinario  
cariño y a Carolina Ibaceta por su  
amor, paciencia y apoyo.

## **AGRADECIMIENTOS**

Deseo agradecer al profesor Ricardo Giesen por apoyarme y orientarme en este largo proceso y al profesor Juan Carlos Muñoz, por creer en mí, colaborarme e integrarse en mi trabajo, a pesar de la distancia. Para mi han sido verdaderos ejemplos a seguir por su integridad y profesionalismo.

A mis amigos y colegas de ShiftUC, por asistirme y darme fuerzas en los momentos que más los necesité, especialmente a Diego Jorquera, Rodolfo Cuevas, Mathías Klapp y Nelson Vásquez.

A Robert Fourer y Manuel Iori, que a tantos kilómetros de distancia estuvieron siempre dispuestos a aportar sus conocimientos y recomendaciones respecto a la modelación y programación de problemas complejos como los abordados en este trabajo.

A los profesores, compañeros y amigos que me han brindado ánimo y coraje a lo largo de la carrera.

A todos infinitamente gracias.

# ÍNDICE GENERAL

Pág.

AGRADECIMIENTOS .....	ii
INDICE DE TABLAS .....	v
INDICE DE FIGURAS.....	vi
RESUMEN .....	vii
ABSTRACT.....	ix
1. INTRODUCCIÓN .....	1
2. ANTECEDENTES DEL PROBLEMA Y REVISIÓN BIBLIOGRÁFICA .....	9
2.1. Ruteo de vehículos .....	9
2.1.1. Problema del Vendedor Viajero.....	9
2.1.2. Problema de ruteo de vehículos con capacidad .....	10
2.1.3. Métodos de solución para el CVRP .....	12
2.2. Problemas de <i>Packing</i> .....	13
2.2.1. <i>Strip Packing Problem</i> .....	13
2.2.2. <i>Bin Packing Problem</i> .....	15
2.3. Ruteo y <i>packing</i> .....	17
3. FORMULACIÓN MATEMÁTICA DEL 2LCVRP-H .....	22
3.1. Formulación Matemática: Conjuntos y Parámetros del Modelo .....	24
3.2. Formulación Matemática: Variables de Decisión .....	25
3.3. Formulación Matemática: Función Objetivo y Restricciones .....	28
4. ENFOQUE DE SOLUCIÓN PROPUESTO .....	36
4.1. Etapa I: Heurística de Clarke & Wright Modificada.....	38
4.2. Etapa II: Heurística de <i>Floor &amp; Ceiling (FC)</i> .....	42
4.3. Etapa III: Optimización de la solución .....	45

5. EXPERIMENTO Y ANÁLISIS DE RESULTADOS .....	47
5.1. Diseño del experimento .....	47
5.2. Opciones de optimización .....	50
5.3. Implementación .....	51
5.4. Análisis de resultados .....	51
5.4.1 Optimización conjunta .....	52
5.4.2 Optimización parcialmente separada .....	55
6. CONCLUSIONES, RECOMENDACIONES Y EXTENSIONES .....	62
BIBLIOGRAFÍA .....	64
<b>A N E X O S</b> .....	67
Anexo A: Resultados Optimización parcialmente separada SIN orientación fija de los ítems (permitiendo giros). .....	68
Anexo B: Resultados Optimización parcialmente separada CON orientación fija de los ítems (sin permitir giros). .....	70
Anexo C: Resultados 2LCVRP presentes en la literatura .....	71
Anexo D: Ejemplo de manipulación en cadena. ....	72

## INDICE DE TABLAS

Tabla 1: Características consideradas en cada referencia. ....	21
Tabla 2: Clases usadas para la generación de objetos.....	48
Tabla 3: Resultados optimización conjunta. ....	53
Tabla 4: Resumen de instancias resueltas relajando restricción de orientación de los ítems. .....	59
Tabla 5: Resultados relajando restricción de orientación de los ítems .....	68
Tabla 6: Resultados considerando activa la restricción de orientación de los ítems. .	70
Tabla 7: Resultados en la literatura sin considerar manipulaciones. ....	71

## INDICE DE FIGURAS

Figura 1: <i>Trade off</i> entre ruteo y manipulación. ....	2
Figura 2: Único <i>packing</i> factible para esta instancia, salvo simetrías. ....	6
Figura 3: Ejemplo de empaque infactible al asumir carga ortogonal. ....	7
Figura 4: <i>Packing</i> realizado utilizando FF.....	14
Figura 5: Representación gráfica de variables y parámetros de empaque desde una vista superior .....	23
Figura 6: Distribución de pisos en la superficie del vehículo. ....	37
Figura 7: Vista superior de los primeros dos pisos en la aplicación de Heurística FC, señalando <i>floor</i> y <i>ceiling</i> del piso 1. ....	43
Figura 8: Costo de ruteo vs. " <i>cm</i> ". Al aumentar " <i>cm</i> " el costo de ruteo aumenta, pero la cantidad de manipulaciones es disminuye. ....	49
Figura 9: Crecimiento del número de variables vs número de ítems.....	58
Figura 10: Aumento del tiempo de ejecución vs. cantidad de ítems.....	58
Figura 11: Aumento de instancias resueltas en menos de 2100 segundos al permitir giros y al no permitirlos. ....	61
Figura 12: manipulación en cadena provocada por la salida del objeto <i>i</i> del cliente <i>c</i> .72	

## RESUMEN

La industria del transporte de carga ha crecido enormemente la última década. Los líderes de la industria operan redes de miles de vehículos terrestres y bodegas. Por lo tanto, se vuelve particularmente importante optimizar los procesos de reparto.

En la actualidad se han investigado diversas variantes del problema de ruteo de vehículos capacitados (CVRP), sin embargo, los aspectos relacionados a la manipulación de carga no han sido estudiados a cabalidad.

En la presente tesis se analiza el problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones y manipulación de carga en los sitios de demanda (2LCVRP-H). Éste consiste en un problema operacional de transporte de carga que integra tanto el ámbito del ruteo como el de carga de objetos al interior de vehículos. Los objetos poseen distintas densidades y dimensiones, mientras la flota se considera fija y homogénea.

Se considera conocida la ubicación de los clientes y su respectiva demanda como un conjunto de objetos rectangulares a ser entregados en sólo una visita, respetando las restricciones de capacidad de los vehículos y buscando minimizar los costos asociados al ruteo y a la manipulación de objetos en los sitios de demanda, a diferencia de lo realizado en estudios anteriores. Para resolver el modelo formulado se propone un proceso metaheurístico de tres etapas: ruteo, empaque y optimización, que amplía la gama de instancias que se puede resolver de manera eficiente.

El modelo de programación entera mixta (MIP) presenta un buen desempeño sobre instancias pequeñas, considerando el *trade off* entre ruteo y *packing*, logrando resolver de manera exacta instancias de 6 clientes, 10 ítems y 2 vehículos en un tiempo promedio aproximado de 10 minutos. Mientras que el proceso heurístico propuesto logró resolver instancias de hasta 20 clientes, 60 ítems y 5 vehículos en menos de 30 minutos.

Finalmente, dada la estructura del MIP, se propone adaptar el modelo presentado para considerar *pick-up and deliveries* y ventanas de tiempo.

*Palabras clave:* Problema de ruteo de vehículos capacitados; restricciones de 2 dimensiones; manipulación de carga; CVRP; 2L-CVRP; CVRP-H.

## ABSTRACT

The freight transportation industry has grown enormously the last decade. The industry leaders operate networks of thousands of vehicles and warehouses. Therefore, it becomes particularly important to optimize the distribution processes.

At the present time, diverse variants of the capacitated vehicle routing problem (*CVRP*) has been investigated, nevertheless, the aspects related to freight manipulation have not been studied completely.

The present thesis analyzes the capacitated vehicle routing problem with two dimensional loading constraints and freight handling at service points (*2LCVRP-H*). It consists in an operational problem of freight transportation that integrates the routing ambit and the packing of items into vehicles. The items have different density and dimensions, while the vehicle fleet is considered fixed and homogeneous.

Customers locations is known and their respective demand is considered as a given set of rectangular objects to be delivered in only one visit, always satisfying the loading constraints and seeking to minimize routing cost as well as the handling associated costs, differently of what has been made so far in previous studies.

To solve the formulated model, a metaheuristic process is proposed and separated in three stages: routing, packing and optimization, which extends the range of instances that can be solved efficiently.

The mixed integer programming model (*MIP*) presents a good performance over small instances, considering the tradeoff between routing and packing, accomplishing to exactly solve instances of 6 clients, 10 items and 2 vehicles in an average approximated time of 10 minutes. While the proposed heuristic process managed to solve instances that had up to 20 clients, 60 items and 5 vehicles in less than 30 minutes.

Finally, given the MIP structure, we propose to adapt the presented model for considering pick-up and deliveries or time windows.

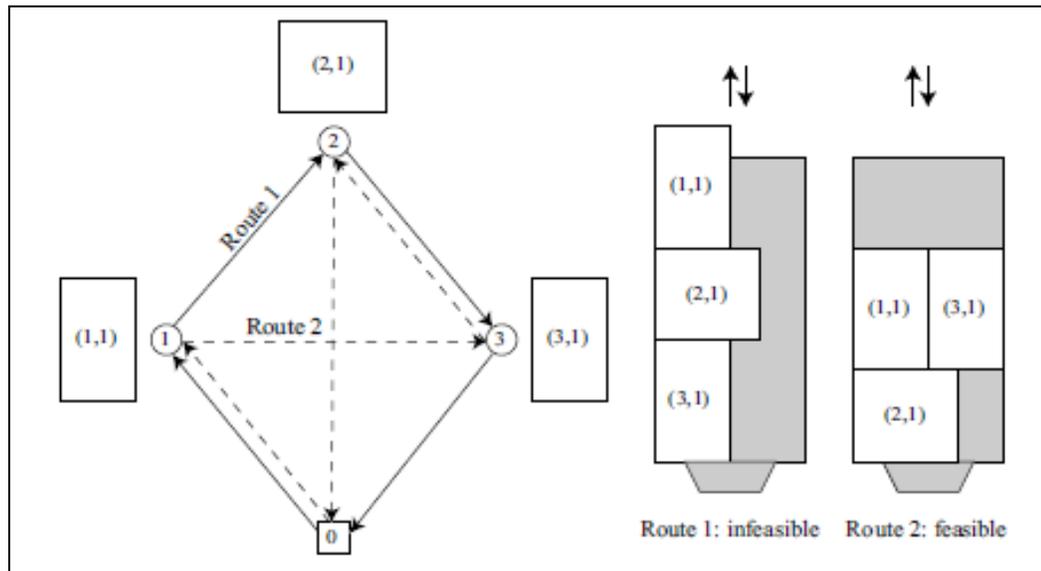
*Keywords:* Capacitated vehicle routing problem; two dimensional loading constraints; cargo handling; CVRP; 2L-CVRP; CVRP-H.

## 1. INTRODUCCIÓN

La industria del transporte de carga se ha expandido globalmente en las últimas décadas. En la actualidad, empresas dedicadas a este rubro, como TNT *Express*, se encuentran presentes en numerosos países. Esta sola empresa maneja más de 26.000 vehículos terrestres, 2.300 bodegas y dan empleo a más de 75.000 personas a nivel global. Sólo en Chile, TNT LitCargo cuenta con 350 vehículos de reparto y 60.000m<sup>2</sup> de bodega. Por lo que se vuelve cada vez más importante determinar si los mecanismos utilizados son en realidad los más apropiados para realizar las tareas de logística y distribución. Por otra parte, debido a que actualmente no existen modelos capaces de abstraer satisfactoriamente el proceso de *packing* y reparto de la empresa, el estado de la práctica aún se basa en la experiencia de los operarios.

Un ejemplo del servicio prestado por este tipo de empresas es la entrega de electrodomésticos, artículos con base fija que por su fragilidad y gran tamaño no pueden ser apilados dentro de los vehículos. En la figura 1 se ilustra el ejemplo (Iori *et al.*, 2005), donde un único vehículo satisface la demanda de tres clientes partiendo de una bodega representada por el nodo "0". El vehículo debe volver a la bodega al terminar su recorrido. Cada cliente demanda sólo un ítem y los tamaños de los ítems y la superficie de carga son proporcionales a los que muestra la figura. En ella, se aprecia que la ruta 1 es más corta, sin embargo, es infactible si se quiere manipular sólo una vez cada producto (*i.e.* al entregarlo), por lo que se vuelve necesario realizar una ruta más larga (*i.e.* ruta 2) si se busca minimizar el número de manipulaciones.

Es importante destacar que la solución manipulando un producto sólo si es el momento de entrega en el ejemplo ilustrado en la figura 1, corresponde a la ruta y *packing* número 2. Sin embargo, el costo de ruteo podría disminuir al permitir la ruta 1 con el *packing* de la ruta 2, descargando el objeto (3,1) al visitar al cliente 2 y volviendo a cargarlo posteriormente.



**Figura 1: Trade off entre ruteo y manipulación.**

Fuente: Iori *et al.* (2005).

Como se aprecia en el ejemplo anterior, los aspectos relacionados a la manipulación de carga no han sido estudiados por completo en los problemas de ruteo de vehículos presentes en la literatura. Por lo tanto, se procederá a estudiar el problema mencionado y los posibles beneficios que se generan a partir del el *trade off* entre ruteo y *packing*.

En la literatura existen estudios del problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones (2LCVRP, del inglés *Vehicle Routing Problem with Two Dimensional Loading Constraints*) es un problema combinatorial que tiene aplicación práctica a nivel operativo en el transporte terrestre. En este problema se manipula ítems cuyas características, tales como su fragilidad, incapacidad de posicionarse sobre una de sus caras, tamaño, etc., impiden apilar la carga verticalmente, por lo tanto, la altura de cada ítem resulta irrelevante (mientras quepa en la superficie de carga) para su localización en el vehículo, quedando sólo el largo, ancho y el peso como parte del *input* relevante asociado a cada ítem. De esta manera, se asume que los objetos tienen una base

fija y sólo pueden ser reposicionados manteniendo dicha base en contacto con la superficie de carga.

En comparación con el 2LCVRP, el problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones y reordenamiento de carga en los sitios de demanda (2LCVRP-H, del inglés *Vehicle Routing Problem with Two Dimensional Loading Constraints and Handling Costs*) considera adicionalmente la posibilidad de realizar una mayor cantidad de manipulaciones de carga al realizar la entrega, con el objetivo de reducir el costo total que está compuesto por el costo de ruteo y el costo en el que se incurre al manipular la carga.

El presente trabajo aborda el denominado problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones y manipulación de carga en los sitios de demanda (2LCVRP-H), que contempla al conocido *Two dimensional Bin Packing Problem (2BPP)* y al *Vehicle Routing Problem (VRP)*, como casos particulares del problema. El 2BPP consiste en ubicar un conjunto dado de rectángulos de diferentes dimensiones en una cantidad finita de cajas rectangulares de capacidad dada, minimizando el total de cajas utilizadas. Por otra parte, el VRP consiste en determinar las rutas de una flota de transporte de capacidad conocida para satisfacer la demanda de un conjunto de clientes cuyas ubicaciones y demandas son conocidas.

El 2LCVRP-H consiste en empacar objetos de tamaño y peso diferentes, todos rectangulares, en un número finito de contenedores de capacidad dada y simultáneamente determinar rutas para visitar un conjunto dado de clientes incurriendo en el menor costo total posible con una flota fija y homogénea de vehículos, los que hacen entrega de los objetos demandados. La ubicación del depósito y de los clientes son conocidas.

La carga y descarga de un vehículo se realiza a través de la misma puerta, ubicada en la parte posterior del mismo. Este proceso de *packing* se realiza físicamente en la bodega, donde todas las rutas comienzan y finalizan. Todos los objetos demandados por un mismo

cliente deberán ser transportados en un mismo vehículo y ser entregados en una única visita, es decir, no se permiten entregas divididas (*i.e. split deliveries*).

El 2LCVRP y el 2LCVRP-H son versiones más complejas del problema de ruteo de vehículos con restricciones de capacidad (CVRP, del inglés *Capacitated Vehicle Routing Problem*), ya que éste último sólo considera el peso de los objetos transportados y no sus dimensiones. El CVRP ha sido ampliamente estudiado debido al interés que generó a partir de fines del siglo pasado en las áreas de transporte, distribución y logística (Golden, 2008). Se han estudiado distintas variantes del problema original, considerando cada vez más características que guarden relación con dificultades que surgen en la práctica, para una revisión reciente se recomienda al lector ver Iori y Martello (2010).

Existen distintos enfoques de solución al problema de ruteo y *packing* (*e.g.* problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones, 2LCVRP, o con tres dimensiones, 3LCVRP, entre otros), como los propuestos por Iori *et al.* (2005), Zachariadis *et al.* (2007) o Fuellerer *et al.* (2009), los que serán analizados más adelante. Sin embargo, en la literatura existente no es posible encontrar estudios enfocados a la optimización de la manipulación de carga en forma conjunta al ruteo y al *packing*.

En estudios anteriores se ha utilizado diferentes heurísticas para resolver el 2LCVRP, todas ellas con distintos niveles de aplicabilidad dadas las condiciones específicas del problema. Gendreau *et al.* (2008) proponen un método de búsqueda tabú para resolver el subproblema de ruteo comprendido en el 2LCVRP, mientras que el *packing* se realiza planteando el conocido *Strip Packing Problem (SPP)*, que consiste en empaquetar un conjunto dado de objetos en una superficie de ancho fijo, minimizando su largo. De esta manera, se permiten empaques infactibles (*i.e.* objetos fuera de la superficie del vehículo) sujetos a penalidades en la función objetivo.

Zachariadis *et al.* (2007) plantean una metodología metaheurística para el mismo problema. A través de una búsqueda tabú "guiada" por un mecanismo que altera la función objetivo, se explora el espacio de soluciones factibles considerando los aspectos del ruteo,

mientras el empaque se aborda utilizando un conjunto de heurísticas de *packing* presentes en la literatura. Esta metodología es extendida posteriormente por Tarantilis *et al.* (2009), para resolver el 3LCVRP considerando también fragilidad de los productos.

Fuellerer *et al.* (2009) abordan el problema a partir de un sistema de optimización de colonias de hormigas (ACO), y para asegurar que el empaque sea factible, recurre a heurísticas de *packing* como "*Bottom Left Fill*", "*Max Touching Perimeter*", entre otras.

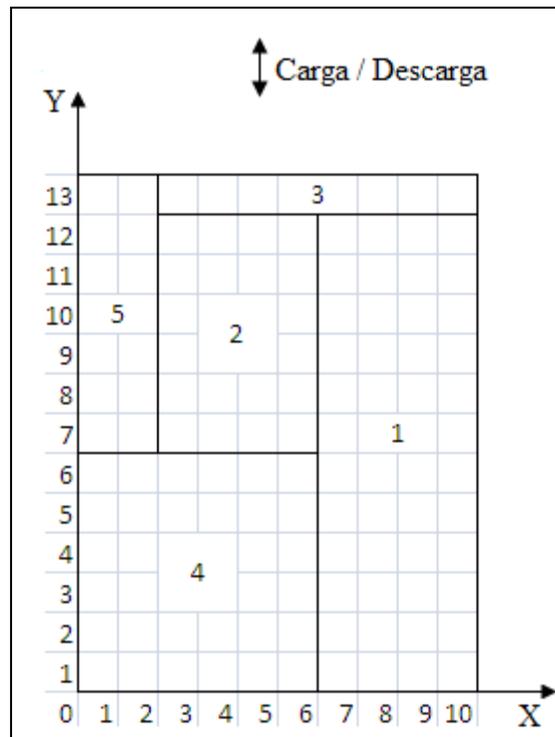
Finalmente, Leung *et al.* (2010) proponen mejorar la búsqueda local guiada de manera que no se pierdan mejores soluciones debido a la penalización que se agrega en la función objetivo en el proceso de búsqueda local (*i.e. local search*).

La diferencia fundamental del análisis que se presenta en este estudio es la posibilidad de realizar arreglos en la ubicación de la carga en los puntos de entrega, minimizando conjuntamente el costo de ruteo y de manipulación de carga, sin caer en la denominada "carga secuencial" (*Sequential Loading*), ni tampoco en "carga irrestricta" (*Unrestricted Loading*), que no considera el costo de este tipo de modificaciones. En la primera condición, cada objeto a ser entregado en una determinada visita debe estar ubicado exactamente a la salida del compartimento de carga, sin que ningún otro objeto obstruya la línea recta que describe dicho objeto al ser extraído del vehículo. En la segunda condición, "carga irrestricta", se podría exigir descargar todos los ítems en cada parada y volver a cargar los ítems correspondientes a los clientes restantes, sin que esto impacte el costo asociado a la solución. Por lo tanto, se vuelve especialmente interesante formular un modelo que aborde este nuevo problema, y desarrollar una metodología que permita resolver instancias complejas, entregando buenos resultados en términos de costos de manipulación y de ruteo.

En trabajos anteriores, usualmente se ha modelado el 2LCVRP utilizando ambas estrategias mencionadas, sin embargo, hay muchos casos en que ninguna de ellas es apropiada. En la figura 2, se muestra un ejemplo donde se asume que la carga/descarga se realiza en la dirección señalada, por lo que para entregar simultáneamente los ítems 3 y 4,

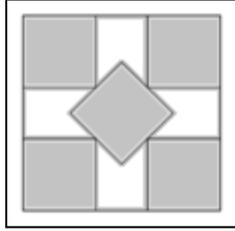
es necesario manipular adicionalmente los ítems 2 y 5, ya que todos ellos obstruyen la salida del ítem 4.

El enfoque de ruteo y *packing* simultáneo, considerando la posibilidad de realizar una manipulación óptima de la carga, no ha sido estudiado aún en la literatura. Aunque en ella sí se reconoce el *trade off* entre ruteo y *packing*.



**Figura 2:** Único *packing* factible para esta instancia, salvo simetrías.

Para efectos de análisis se considera que el proceso de *packing* de los vehículos se realizará con carga ortogonal, es decir, las caras de los objetos deberán estar siempre paralelas a las caras del compartimento de carga, quedando prohibidos empaques como el que se observa en la figura 3, independiente de que ésa sea la configuración óptima y, en este caso, la única factible.



**Figura 3: Ejemplo de empaque infactible al asumir carga ortogonal.**

En la presente tesis se formuló un modelo de programación entera mixta (MIP) para resolver instancias pequeñas del 2LCVRP-H. Para instancias mayores se desarrolló un proceso heurístico que separa parcialmente el ruteo del *packing* y posteriormente optimiza la solución.

El modelo de programación entera mixta formulado en esta tesis es capaz de alcanzar una solución en tiempo razonable sólo para instancias pequeñas, por lo que para instancias de mayor tamaño se aplica secuencialmente un proceso metaheurístico de tres etapas: ruteo, empaque y optimización, que amplía la gama de instancias que se puede resolver de manera eficiente. Éste proceso fija parte de las rutas a ser recorridas en el resultado final, considerando únicamente el costo de ruteo y despreciando inicialmente el *packing*, por lo tanto, las soluciones pueden no ser óptimas con respecto al modelo MIP propuesto.

Esta tesis se ha dividido en seis capítulos. Luego de esta introducción, en el segundo capítulo se realiza una revisión bibliográfica sobre algunos de los trabajos más importantes presentes en la literatura respecto a este problema. En el tercer capítulo se describe el modelo de programación entera mixta (MIP) que permite resolver instancias de menor tamaño. En el cuarto capítulo, se presenta el enfoque metaheurístico para resolver instancias de mayor tamaño del 2LCVRP-H, enfoque que incluye en su etapa final la aplicación del modelo descrito en el segundo capítulo. En la primera sección de éste capítulo se describe detalladamente el algoritmo utilizado para la obtención de rutas iniciales a través de la heurística de Clarke & Wright, en la segunda sección se describe el

proceso inicial de *packing* a través de la heurística de *Floor-Ceiling* y finalmente, en la tercera sección, se hace referencia a la aplicación del MIP detallado en el segundo capítulo para mejorar la solución proveniente de las heurísticas anteriores. En el quinto capítulo se describe el experimento diseñado para resolver instancias pequeñas y otro para instancias de mayor tamaño. Finalmente, se presentan conclusiones, análisis de los aportes de este trabajo y futuros desarrollos.

## 2. ANTECEDENTES DEL PROBLEMA Y REVISIÓN BIBLIOGRÁFICA

En este capítulo se procede a revisar los antecedentes del problema abordado en esta tesis y se ha dividido en 3 secciones. La primera aborda el ruteo de vehículos, la segunda sección estudia los problemas clásicos de *packing* y, finalmente, la tercera sección describe los problemas de ruteo y *packing* simultáneo más atinentes al problema abordado en esta tesis.

### 2.1. Ruteo de vehículos

En esta sección se describirán los problemas de ruteo más importantes en el desarrollo de esta tesis. Introduciendo el "problema del vendedor viajero" (*TSP*, del inglés *Travelling Salesman Problem*), luego el problema de ruteo de vehículos con capacidad y posteriormente los distintos métodos de solución para el CVRP.

#### 2.1.1. Problema del Vendedor Viajero

El problema base en el área de ruteo es el conocido Problema del Vendedor Viajero (*TSP*, del inglés *Traveling Salesman Problem*). Dado un conjunto de clientes y el costo de recorrer la distancia entre cualquier par de ellos, el *TSP* consiste en encontrar un *tour* de costo mínimo que visita todas las ciudades. El problema es *NP-Hard* y su solución es un desafío clásico en la optimización combinatorial. Puede ser modelado a través de un grafo  $G=(V,E)$ , donde  $V=\{0,1,\dots,n\}$  es el conjunto de nodos,  $E=\{(i,j) : i,j \in V\}$  es el conjunto de arcos y  $c_{ij}$  con  $(i,j) \in E$  es el costo de recorrer el arco que une al nodo  $i$  con el nodo  $j$  (desde  $i$  a  $j$  en la versión asimétrica del *TSP* y en cualquiera de los dos sentidos en la versión simétrica). Usualmente se constata la desigualdad triangular, es decir,  $c_{ij} \leq c_{ik} + c_{kj} \forall i,j,k \in V$ . El *TSP* ha sido largamente estudiado durante las últimas décadas y no se

profundizará en la enorme literatura existente respecto a este problema. Se puede encontrar revisiones recientes en Golden (2008), Wang *et al.* (2009) o Iori y Martello (2010).

### 2.1.2. Problema de ruteo de vehículos con capacidad

La extensión natural del TSP fue el problema de ruteo de vehículos (VRP, del inglés *Vehicle Routing Problem*) y posteriormente el problema de ruteo de vehículos con capacidad (CVRP), que también forma parte de los problemas de optimización combinatorial más estudiados debido a su relevancia práctica y a su considerable dificultad. Éste consiste en determinar una serie de rutas para una flota de  $K$  vehículos a partir de un depósito, para un cierto número de clientes o nodos geográficamente dispersos. Todos los vehículos cuentan con la misma capacidad  $Q$  y cada cliente demanda  $q_i$  (con  $0 \leq q_i \leq Q$  para  $i = 1, \dots, n$ ).

El modelo propuesto por Toth y Vigo (2002) para resolver la versión simétrica del CVRP (*i.e.* el costo para ir de  $i$  a  $j$  es el mismo en el que se incurre para ir de  $j$  a  $i$ ) es utilizado más tarde por Iori *et al.* (2005) para resolver el 2L-CVRP. El modelo original se describe a continuación.

$$\min \sum_{e \in E} c_e x_e \quad (1)$$

$$s.t. \sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V \setminus \{0\}, \quad (2)$$

$$\sum_{e \in \delta(0)} x_e = 2K, \quad (3)$$

$$\sum_{e \in \delta(S)} x_e \geq 2\gamma(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (4)$$

$$x_e \in \{0,1,2\} \quad \forall e \in \delta(0), \quad (5)$$

$$x_e \in \{0,1\} \quad \forall e \notin \delta(0). \quad (6)$$

Siendo  $V$  el conjunto de nodos presentes en el problema,  $E$  el conjunto de arcos,  $\delta(S)$  los arcos que tienen uno de sus extremos en  $S$  y el otro en  $V \setminus S$  (sin considerar el sentido de cada arco, ya que se resuelve la versión simétrica del problema). Como se ha realizado en la literatura, se utilizará  $\delta(i)$  en lugar de  $\delta(\{i\})$ .  $x_e$  corresponde a una variable entera que representa la cantidad de veces que es recorrido el arco  $e$  en la solución. Esta variable puede tomar el valor 2 para el caso en que se presente una ruta con un único cliente, ya que se utiliza el mismo arco para retornar a la bodega.

En este modelo, (1) corresponde a la función objetivo, que minimiza el costo total de ruteo. Las restricciones (2) y (3) determinan la cantidad de arcos salientes o entrantes para los clientes y para la bodega, respectivamente. Las restricciones (4), conocidas como las "restricciones de capacidad-corte", implican conectividad en la solución y factibilidad respecto a las restricciones clásicas del BPP. En estas restricciones,  $\gamma(S)$  corresponde al mínimo número de vehículos necesarios para satisfacer a los clientes en  $S$ , sin considerar la secuencia en que son visitados, por lo que podría requerirse de más vehículos que los determinados por  $\gamma(S)$  en caso de lograrse un *packing* no factible debido a la restricción de "carga secuencial". Por lo general,  $\gamma(S)$  se calcula según  $\gamma(S) = \lceil \sum_{i \in S} q_i / Q \rceil$ . Por lo tanto,  $\gamma(S)$  vehículos deben cruzar desde el subconjunto de nodos  $V \setminus S$  a  $S$  para luego retornar utilizando igual cantidad de arcos, es decir, al menos  $2\gamma(S)$  arcos unen ambos subconjuntos de nodos.

Finalmente, las restricciones (5) y (6) implican que cada arco entre dos clientes es recorrido como máximo sólo una ocasión y cada arco incidente a la bodega es recorrido como máximo dos veces. Éste último caso corresponde a un vehículo visitando a sólo un cliente. En este modelo, las restricciones de eliminación de sub-tours se cumplen implícitamente debido a (2) y (4).

Distintas variantes de este problema han sido extensamente investigadas en la literatura, considerando distintas variantes del VRP original (*e.g. loading constraints, pick-up and deliveries*, Malapert *et al.*, 2008), dentro de las cuales son principalmente atingentes a este trabajo el CVRP (Dantzig y Ramster, 1959; Jacobs *et al.*, 1998; Toth y Vigo, 2002 y Lei *et al.*, 2011, entre otros) y el 2LCVRP (Iori *et al.*, 2005; Zachariadis *et al.*, 2007; Gendreau *et al.*, 2008; Fuellerer *et al.*, 2009 y Leung *et al.*, 2010, entre otros).

### **2.1.3. Métodos de solución para el CVRP**

Se han estudiado distintos métodos de solución para el problema de ruteo de vehículos con capacidad desde el trabajo original de Dantzig y Ramster (1959), donde todas las demandas son determinísticas, conocidas de antemano y no pueden ser divididas en más de un vehículo. Éstos son idénticos, de capacidad fija y conocida, todos ellos ubicados inicialmente en una única bodega, a la que tendrán que retornar al terminar sus recorridos.

Toth y Vigo, (2002) señalan que a pesar del progreso logrado por los algoritmos tipo *branch and bound* con respecto a los de generaciones previas (*e.g.* Fisher, 1994), el tamaño de las instancias que podían ser efectivamente resueltas de manera sistemática por los enfoques presentes hasta ese momento no superaba las pocas decenas de clientes. Desde hace ya algunos años se ha propuesto resolver estos problemas a través de metaheurísticas. Ribeiro y Laporte (2011) proponen un trabajo basado en la búsqueda local de soluciones y destacan su aplicación en logística y distribución de emergencia (*e.g.* ayuda solidaria en desastres naturales). Lei *et al.*(2011) afirman que si se trabaja con un proceso de búsqueda local, el elemento clave para el éxito es aceptar movimientos apropiados que empeoran eventualmente la solución para posteriormente alcanzar otra mejor.

La formulación de modelos y el uso de metaheurísticas para resolver el CVRP para problemas reales puede ser limitado y difícil debido a la existencia de características prácticas que no son fáciles de abstraer. Así es como adicionalmente se ha estudiado este

problema con “restricciones de distancia” (DCVRP), donde cada arco tiene asociado un tiempo de recorrido y los vehículos deben recorrer toda su ruta sin exceder un tiempo máximo disponible (*e.g.* la duración de la jornada del conductor) para estar de regreso en la bodega. En el CVRP con ventanas de tiempo (CVRP-TW), cada cliente tiene asociado un intervalo de tiempo en el que debe realizarse su respectiva visita. En el CVRP con *pickup and delivery* (CVRP-PD), además de los clientes a los que se debe entregar una cierta cantidad de productos, existen aquellos clientes de los que se debe recoger otra cantidad de productos, una variante de este problema es el CVRP con *backhauls*, donde todos los clientes que entregan productos son visitados antes que los que demandan (Jacobs y Goetschalckx, 1998). Éstas y otras variantes del CVRP han sido estudiadas a profundidad en estudios previos (Toth y Vigo, 2002; Golden, 2008).

## **2.2. Problemas de *Packing***

En el CVRP la demanda de un cliente queda especificada como el peso total de los ítems que deben ser entregados a dicho cliente, sin embargo, en la práctica se debe asociar cada objeto o producto a una determinada forma. Además, a menudo se debe considerar complicaciones correspondientes a las características de los objetos a transportar, como son la fragilidad o estabilidad de los mismos. A causa de esto es importante el estudio de los problemas de *packing* que a menudo se asocian a los problemas de ruteo. A continuación se describen algunos de los problemas de *packing* clásicos que anteceden a esta tesis.

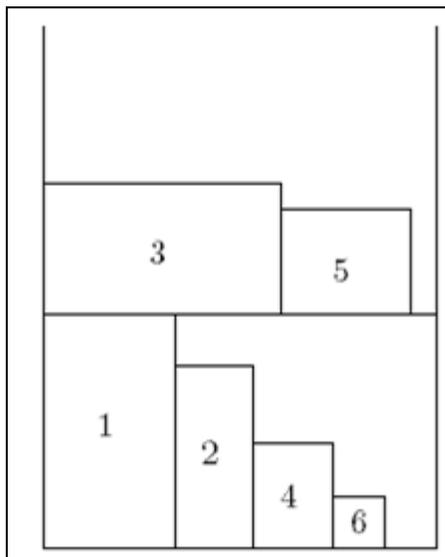
### **2.2.1. *Strip Packing Problem***

El *Strip Packing Problem* consiste en empaquetar un conjunto de objetos de dimensiones distintas y conocidas, en un contenedor de ancho fijo  $W$  y largo variable, buscando minimizar el largo total utilizado para lograrlo.

Para verificar la factibilidad de este problema, el ancho máximo de los objetos a empacar debe ser menor a  $W$ . Por lo general, al realizar el *packing* los objetos no se pueden traslapar ni ser rotados.

Este problema es frecuente en la industria textil donde se cuenta con rollos de material de un ancho fijo y se pretende satisfacer la demanda por rectángulos de distinto ancho y largo minimizando la pérdida de material.

Dentro de las aproximaciones clásicas más conocidas para resolver este problema se encuentran los algoritmos *Bottom-Left* (BL) y *First-Fit* (FF), entre otros (Coffman, 1980). Generalmente, los algoritmos mencionados comienzan generando una lista que incluye todos los ítems a empacar según sus largos. BL ubica los ítems uno a uno lo más abajo posible dentro del contenedor, y luego lo más a la izquierda posible sin traslaparse con otro objeto empacado anteriormente. FF empaca cada ítem en el primer nivel en el que quepa. Si no existe un nivel capaz de contener dicho ítem, entonces un nuevo nivel es creado. Cada ítem es posicionado dentro de un nivel aplicando localmente el algoritmo BL, como se observa en la figura 4.



**Figura 4:** *Packing* realizado utilizando FF

### 2.2.2. Bin Packing Problem

Este problema consiste en empaquetar un conjunto de objetos en varias cajas o contenedores iguales, tal que el peso (1BPP), superficie (2BPP) o el volumen total (3BPP) no exceda la capacidad de las cajas.

Se considera que el conjunto de productos es finito y conocido, es decir, el peso y dimensiones de cada uno de ellos son conocidos. El objetivo de este problema es empaquetar todos los objetos, minimizando el número de cajas utilizadas.

Técnicamente, el 1BPP puede ser descrito de la siguiente forma (Lodi, *et al.*, 2002):

Dado un conjunto finito de elementos  $E=\{e_1, \dots, e_m\}$  con pesos asociados  $P=\{p_1, \dots, p_m\}$  tales que  $0 \leq p_i \leq Q$ , donde  $Q$  corresponde a la capacidad del contenedor. Se procede a dividir el conjunto de elementos  $E$  en  $N$  subconjuntos, de forma que la suma de pesos en cada partición sea a lo sumo  $Q$ , con el objetivo de que  $N$  sea mínimo.

El 2BPP puede ser descrito de manera similar, pero en este caso, cada elemento tiene asociado un ancho  $w_i$  y un largo  $l_i$ , tal que  $0 \leq w_i \leq W$  y  $0 \leq l_i \leq L$ , donde  $W$  y  $L$  corresponden al ancho y largo de cada contenedor.

A diferencia del 1BPP, en el 2BPP deben hacerse ciertas consideraciones respecto a la distribución de los objetos dentro de cada contenedor al realizar el *packing*. A continuación se especifican algunos de los aspectos más importantes que han sido estudiados en la literatura (Wang, *et al.*, 2009), relativos a características de los objetos empaquetados tanto para el BPP como para el CVRP.

#### a) Restricciones clásicas del *Bin Packing Problem* (BPP)

Los productos no pueden estar traslapados unos con otros y deben estar completamente contenidos en la superficie de carga.

b) Restricción de capacidad

La suma del peso de los ítems transportados no debe superar la capacidad de carga de los vehículos.

c) Ortogonalidad

Cada objeto debe ser empacado con sus extremos paralelos a las caras del compartimento de carga.

d) Orientación de los ítems

Los objetos tienen una base y orientación fijas respecto del compartimento de carga, por lo que no podrán ser ladeados ni rotados en 90° respecto de ningún eje.

e) Fragilidad

Los objetos pueden ser divididos en dos grupos: Frágiles y No-Frágiles. Los primeros no podrán cargar ningún tipo de objeto sobre sí mismos.

f) Estabilidad de los ítems

Cuando un objeto es cargado sobre otros, su base debe estar apoyada sobre una superficie mínima de soporte (Eley, 2002).

g) Carga Secuencial (LIFO)

Los objetos a ser entregados deben estar inmediatamente contiguos a las puertas traseras del compartimento de carga y ningún otro objeto debe obstruir su salida, es decir, no se permiten movimientos laterales de carga.

h) Optimización de manipulación.

La manipulación de carga realizada en los sitios de demanda asegura tener asociado un costo mínimo, es decir, si no se puede realizar "carga secuencial" por infactibilidad como la que muestra la figura 2, entonces el *packing* realizado debe considerar la manipulación de carga en visitas previas a su entrega.

Dependiendo de las dimensiones de los productos a considerar (*i.e.* inclusión de 1BPP, 2BPP o 3BPP), la resolución del CVRP asociado tendrá un distinto nivel de dificultad. Particularmente, la presente tesis se centra en un análisis bidimensional de los objetos, por lo que parece interesante estudiar los modelos propuestos actualmente para el problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones (2LCVRP).

### **2.3. Ruteo y *packing***

En la literatura se ha estudiado el problema de ruteo y *packing* simultáneos considerando objetos bidimensionales (*i.e.* despreciando su altura) y tridimensionales. A continuación se describe el problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones, que es más atinente al trabajo desarrollado en esta tesis. Para análisis más profundos respecto a este problema considerando ítems tridimensionales, se recomienda al lector acudir a Iori y Martello (2010).

A continuación se procede a revisar la literatura existente respecto al problema de ruteo de vehículos con restricciones de capacidad de dos dimensiones (2LCVRP).

Como fue mencionado en el capítulo anterior, el 2LCVRP es un problema combinatorial en el que se manipula ítems cuyas características, tales como su fragilidad, incapacidad de posicionarse sobre una de sus caras, tamaño, etc., impiden apilar la carga verticalmente, por lo tanto, la altura de cada ítem resulta irrelevante (mientras quepa en la superficie de carga) para su localización en el vehículo, quedando sólo el largo, ancho y el peso como parte del *input* relevante asociado a cada ítem. De esta manera, se asume que los objetos tienen una base fija y sólo pueden ser reposicionados manteniendo dicha base en contacto con la superficie de carga.

El modelo planteado por Iori *et al.* (2005), encuentra una configuración factible y óptima en gran parte de las instancias consideradas (Toth y Vigo, 2002), optimalidad sujeta a la utilización de la restricción orientación de los ítems y de carga secuencial, es decir, el desarrollo propuesto no es capaz de resolver instancias como la descrita en la figura 2, donde la manipulación adicional de carga es obligatoria. Estos autores proponen restricciones de “*capacity-cut*”, que aseguran la conectividad de las rutas y factibilidad de empaque respecto a las restricciones clásicas del *Bin Packing Problem* mencionadas en la sección anterior. El modelo utilizado es una adaptación al propuesto por Toth y Vigo (2002) para resolver el CVRP y se describe a continuación.

$$\min \sum_{e \in E} c_e x_e \quad (7)$$

$$s.t. \sum_{e \in \delta(i)} x_e = 2 \quad \forall i \in V \setminus \{0\}, \quad (8)$$

$$\sum_{e \in \delta(0)} x_e = 2K, \quad (9)$$

$$\sum_{e \in \delta(S)} x_e \geq 2\gamma(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset, \quad (10)$$

$$\sum_{e \in E(S, \sigma)} x_e \leq |S| - 1 \quad \forall (S, \sigma) / \sigma \notin \Sigma(S), \quad (11)$$

$$x_e \in \{0, 1, 2\} \quad \forall e \in \delta(0), \quad (12)$$

$$x_e \in \{0, 1\} \quad \forall e \notin \delta(0). \quad (13)$$

La función objetivo (7) es idéntica a (1) y las restricciones (8) a (13) son equivalentes a las restricciones (2) a (6), con la diferencia que en este caso se añade las restricciones (11), conocidas como "restricciones de rutas infactibles", que consideran "carga secuencial", eliminando ruteos infactibles asociados al conjunto de clientes  $S$  a través de un algoritmo que comprueba cada ruteo y, de no ser factible, añade la respectiva restricción asociada a (11).

En éste modelo,  $\gamma(S)$  corresponde a la cantidad mínima de vehículos necesarios para satisfacer al subconjunto de clientes  $S$ , respetando las respectivas restricciones de *packing*. Este parámetro es difícil de calcular en el CVRP clásico, debido a que se debía resolver la versión unidimensional del *Bin Packing Problem* (1BPP) asociado a los pesos de la carga del subconjunto de clientes  $S$ . El valor de  $\gamma(S)$  será aún más difícil de calcular en este caso, donde se debe considerar el peso, ancho y largo de cada objeto (2BPP), por lo tanto, Iori *et al.* (2005) lo calculan de manera aproximada para resolver el 2L-CVRP según:

$$\gamma'(S) = \max \left\{ \left\lceil \sum_{i \in S} q_i / Q \right\rceil, \left\lceil \sum_{i \in S} a_i / A \right\rceil \right\} \quad (14)$$

Donde  $q_i$  corresponde al peso de todos los objetos del cliente  $i$ ,  $Q$  corresponde a la capacidad de carga del vehículo,  $a_i$  es la suma de las superficies utilizadas por los objetos del cliente  $i$ , y  $A$  corresponde a la superficie de carga de cada vehículo (*i.e.* se asume flota homogénea).

Hay que destacar que (14) no asegura una cantidad suficiente de vehículos para resolver el problema, ya que no considera el área que se pierde en cada vehículo al realizar el *packing* de los objetos del subconjunto de clientes  $S$ .

Éste modelo incluye las restricciones (10) y (11) que inducen a un crecimiento exponencial en la cantidad de restricciones con el número de clientes e ítems. Para evitar éste problema, se utiliza simultáneamente un proceso heurístico que revisa la existencia de un *packing* factible para las rutas evaluadas, denominado *Check-2L*. Como éste algoritmo puede consumir gran cantidad de tiempo, se limita el número de *backtrackings* a realizar, lo que implica que la solución final del 2L-CVRP puede no ser óptima.

El modelo propuesto por Iori *et al.* (2005) para resolver el 2LCVRP, determina la cantidad de vehículos disponibles como el máximo entre el valor  $K$  que la respectiva instancia trae como input y otro  $K'$  determinado al resolver el 2BPP con todos los objetos de la

instancia. Por lo tanto, es poco probable que se produzca la infactibilidad de la instancia a la hora de realizar el *packing*.

En términos generales, el modelo anterior es bastante completo. El problema es que, como fue mencionado inicialmente, al asumir "carga secuencial" éste modelo no es capaz de resolver ciertas instancias donde las características de los objetos no permitan encontrar un *packing* factible. Por consiguiente, el modelo no es capaz de contemplar manipulación de los objetos en los casos en que es inevitable. El trabajo realizado en esta tesis será capaz de resolver este tipo de instancias.

Zachariadis *et al.* (2009) proponen una metaheurística para resolver el 2L-CVRP, utilizando distintas heurísticas en la construcción de rutas y verificando la factibilidad de las soluciones propuestas. Los aspectos del ruteo del problema son manejados por un método de Búsqueda Tabú, diversificando la exploración y eliminando rasgos de baja calidad de las soluciones finales. Sin embargo, este trabajo tampoco es capaz de resolver instancias como la que muestra la figura 2, ya que en ella se utiliza la totalidad de la superficie de carga con el *packing* propuesto (el único factible), por lo tanto, se requiere obligatoriamente realizar manipulaciones adicionales de carga o contar con un vehículo adicional para realizar su entrega.

El 2L-CVRP es resuelto por Fuellerer *et al.* (2009) abordando el problema planteado por Iori *et al.* (2005) a través de distintas políticas de empaque, las que combinan "Carga Secuencial" (*rear loading*) o "Carga irrestricta" (*unrestricted*), fijando o permitiendo el giro de los objetos. De esta manera se determinan las siguientes clasificaciones.

- a) 2|RO|L: *two-dimensional rear oriented loading*;
- b) 2|UO|L: *two-dimensional unrestricted oriented loading*;
- c) 2|RN|L: *two-dimensional rear non-oriented loading*;
- d) 2|UN|L: *two-dimensional unrestricted non-oriented loading*.

En su desarrollo utiliza algoritmos de optimización tipo *Ant Colony* (ACO), basados en la conocida heurística de ahorros de Clarke y Wright (1964). Este proceso se basa en la

iteración de tres pasos: (1) generación de soluciones de una “población de hormigas” de acuerdo a información base y a la entregada por “feromonas”, (2) aplicación de búsqueda local a las soluciones encontradas, y (3) actualización de la información de las “feromonas”.

Dentro de los resultados de su trabajo, se puede destacar que cerca del 5% del costo total de transporte se puede ahorrar al relajar las restricciones de carga secuencial y orientación de los ítems. Por otro lado, relajando sólo una se puede lograr hasta un 3% de ahorro sobre el costo total. No obstante, en las clasificaciones 2|UO|L y 2|UN|L, al igual que en los demás trabajos revisados, no se consideran las manipulaciones en las que se incurre al descargar los objetos de cada vehículo en la entrega de la carga, omitiéndose el costo asociado a estas operaciones.

En base a los trabajos revisados se ha elaborado la tabla 1, que determina las características descritas en la sección 2.1 que son cubiertas por cada autor.

	Referencia	<i>Iori et al.</i>	<i>Zachariadis et al.</i>	<i>Gendreau et al.</i>	<i>Fuellerer et al.</i>	<i>Leung et al.</i>
	Año	2005	2007	2008	2009	2010
Restriccion						
BPP		Si	Si	Si	Si	Si
Capacidad		Si	Si	Si	Si	Si
Ortogonalidad		Si	Si	Si	Si	Si
Orientación de los ítems		Si	Si	Si	Si, pero también considera permitir rotación en 90º.	Si
Fragilidad		-	-	-	-	-
Estabilidad		-	-	-	-	-
Carga Secuencial		Si	Si	Si	Si	Si
Unrestricted Loading		No	Si	Si	Si	Si
Optimizacion Manipulaciones		No	No	No	No	No

**Tabla 1: Características consideradas en cada referencia.**

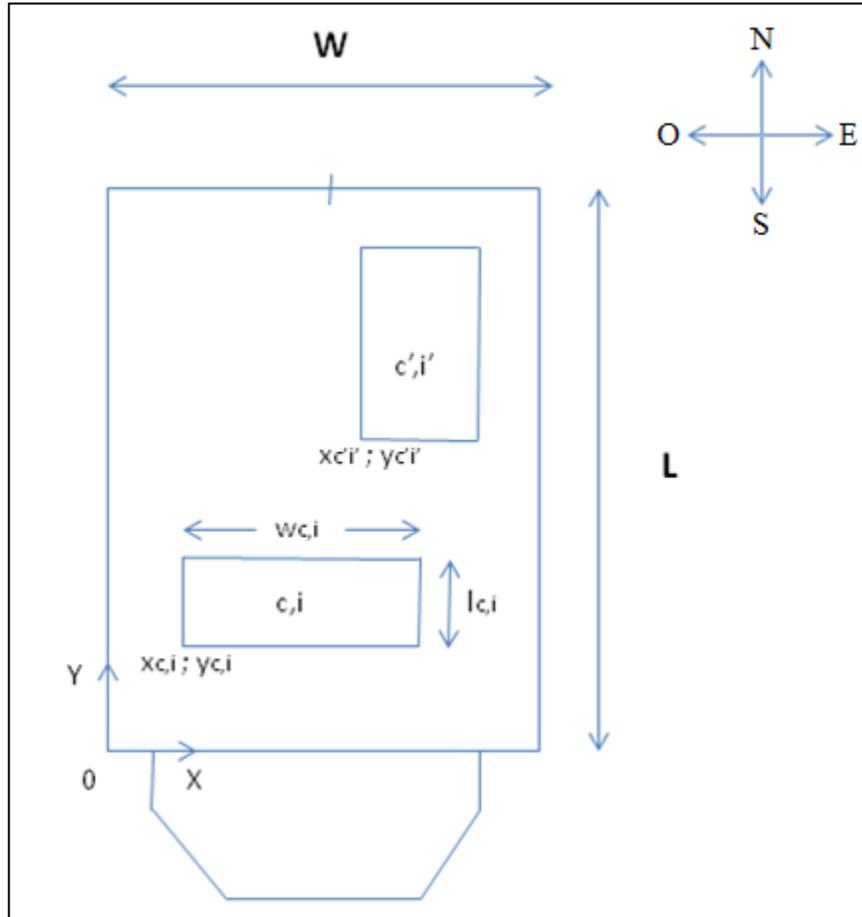
### 3. FORMULACIÓN MATEMÁTICA DEL 2LCVRP-H

El modelo desarrollado a continuación permite resolver eficientemente instancias de tamaño reducido, respetando las restricciones de *packing* a lo largo de la ruta (*e.g.* no traslape de objetos, ítems contenidos en cada vehículo) y minimizando el costo asociado a la distancia por cada vehículo en conjunto al costo de manipulación de carga. Por simplicidad, asumiremos como manipulación el desplazamiento o rotación de un objeto, independiente si éste es entregado al cliente, si sale y retorna al vehículo una vez que otro ítem fue entregado o si es desplazado al interior del vehículo sin salir de éste. El costo de dicha manipulación será independiente de la ubicación del objeto manipulado dentro del vehículo que lo transporta antes y después de la manipulación.

Esta formulación determina la ubicación y orientación de cada uno de los objetos dentro de todos los vehículos al finalizar cada visita que éstos realizan. El modelo asocia a cada vehículo un conjunto  $J$  de potenciales visitas genéricas y a continuación determina cuántas de éstas se usan para cada vehículo, descartando el resto.

El ancho, largo y el peso de cada ítem son conocidos. Para incorporarlos, a cada objeto se le asocia uno de sus lados al ancho y otro al largo (*i.e.* información "referencial"), pero éstos podrán rotarse al ingresar al vehículo y a lo largo de la ruta.

Semejante a la mayor parte de los trabajos presentes en la literatura, el interior del compartimento de carga queda representado por un plano cartesiano cuyo "origen" está ubicado en su esquina inferior izquierda, con la puerta de carga/descarga ubicada en la parte superior de dicho plano (Ver figuras 1 y 2). Sólo se permite el desplazamiento de ítems en el eje  $Y$ , por lo tanto, cualquier objeto ubicado sobre alguna parte del desplazamiento vertical directo de otro objeto se interpreta como una eventual obstrucción. Por ejemplo, en la figura 5 se aprecia que el objeto del cliente  $c'$  obstruye la eventual descarga del objeto del cliente  $c$ .



**Figura 5: Representación gráfica de variables y parámetros de empaque desde una vista superior**

La posición de un objeto queda determinada por las coordenadas de su esquina inferior izquierda, su ancho y largo, tal como se muestra en la figura 4. Esta información permite diferenciar si un objeto obstruye a otro, pues es posible identificar, por ejemplo, si la cara oeste de un objeto está al oeste de la cara este de otro, o si la cara norte de un objeto está al sur de la cara sur de otro.

En el caso de la figura 5, estas observaciones permitirán al modelo determinar que el objeto  $i'$  obstruye al  $i$ .

En el modelo, el depósito es representado en forma dual. Tanto el "cliente 0" como el "cliente  $|C| - 1$ " representarán al origen y el final de cada ruta, respectivamente.

El modelo de programación entera mixta desarrollado se presenta a continuación.

### 3.1. Formulación Matemática: Conjuntos y Parámetros del Modelo

Se utilizan los siguientes conjuntos y parámetros:

#### Conjuntos:

$C$ : Conjunto de clientes que se debe visitar. Este conjunto contemplará además un cliente "0" y un cliente " $|C| - 1$ ", los que representan al depósito en el origen y fin de cada ruta.

$I_c$ : Conjunto de objetos pertenecientes al cliente  $c$ . Se define  $I_0 = I_{|C|-1} = \{i_0\}$ , siendo  $i_0$  un ítem auxiliar (*i.e.* sus dimensiones son todas nulas) tal que la dimensión de éste conjunto sea no nula para todo cliente  $c$ .

$J$ : Conjunto genérico de posibles paradas para cada ruta. Su tamaño corresponde al máximo número de visitas que se podrá observar para una ruta (*i.e.*  $|J| = |C|$ ).

$K$ : Conjunto de vehículos.

#### Parámetros:

$L$ : Largo del compartimento de carga de los vehículos de la flota homogénea.

$W$ :	Ancho del compartimento de carga de los vehículos de la flota homogénea.
$Q$ :	Capacidad de carga de cada vehículo de la flota homogénea en Kilogramos.
$cm_{ci}$ :	Costo de manipular el objeto $i$ del cliente $c$ . Se considera fijo e independiente de la ubicación de los ítems dentro del vehículo.
$width_{ci}$ :	Parámetro entero. Ancho "referencial" del producto $i$ del cliente $c$ .
$length_{ci}$ :	Parámetro entero. Largo "referencial" del producto $i$ del cliente $c$ .
$weigh_{ci}$ :	Parámetro entero. Peso "referencial" del producto $i$ del cliente $c$ .
$costo_{c,c'}$ :	Costo de recorrer el arco $c, c'$ .
$M$ :	Constante de capacidad ( <i>i.e.</i> número suficientemente grande). Éste parámetro se utiliza para mantener la estructura del modelo.

### 3.2. Formulación Matemática: Variables de Decisión

Dados los conjuntos definidos en la sección 2.1, se definen las siguientes variables de decisión.

$$z_{c,c',k} = \begin{cases} 1, & \text{Si el vehículo } k \text{ recorre el arco } (c, c') \\ 0, & \text{si no} \end{cases}$$

$$Entrega_{cij k} = \begin{cases} 1, & \text{Si el objeto } i \text{ del cliente } c \text{ es entregado en la visita } j \text{ de la ruta } k \\ 0, & \text{si no} \end{cases}$$

$$Visita_{ck} = \begin{cases} 1, & \text{Si el vehículo } k \text{ es asignado para visitar al cliente } c \\ 0, & \text{si no} \end{cases}$$

$Presencia_{cij k}$

$$= \begin{cases} 1, & \text{Si el objeto } i \text{ del cliente } c \text{ se encuentra presente en el vehículo } k \\ & \text{en el trayecto del cliente } j \text{ al } j + 1 \\ 0, & \text{si no} \end{cases}$$

$x_{cij} =$  Variable entera. Posición en el eje X de la esquina más cercana al origen del producto  $i$  del cliente  $c$ , en el trayecto del cliente  $j$  al  $j+1$ .

$y_{cij} =$  Variable entera. Posición en el eje Y de la esquina más cercana al origen del producto  $i$  del cliente  $c$ , en el trayecto del cliente  $j$  al  $j+1$ .

$w_{cij} =$  Variable entera. Ancho en el eje X del producto  $i$  del cliente  $c$ , en el trayecto del cliente  $j$  al  $j+1$ .

$l_{cij} =$  Variable entera. Largo en el eje Y del producto  $i$  del cliente  $c$ , en el trayecto del cliente  $j$  al  $j+1$ .

$B1_{cic'i'j}$

$$= \begin{cases} 1, & \text{Si la cara Este del objeto } i \text{ del cliente } c \text{ está al Oeste de la cara Oeste} \\ & \text{del objeto } i' \text{ del cliente } c' \text{ en el trayecto del cliente } j \text{ al } j + 1. \\ 0, & \text{si no} \end{cases}$$

Cabe destacar que si la variable  $B1_{cic'i'j}$  toma el valor 1, quiere decir que ambos objetos no se obstruyen.

$$B2_{cic'i'j} = \begin{cases} 1, & \text{Si la cara Norte del objeto } i \text{ del cliente } c \text{ está al Sur de la cara Sur} \\ & \text{del objeto } i' \text{ del cliente } c' \text{ en el trayecto del cliente } j \text{ al } j + 1. \\ 0, & \text{si no} \end{cases}$$

$$Giro_{c,i,j} = \begin{cases} 1, & \text{Si el objeto } i \text{ del cliente } c \text{ está girado en } 90^\circ \text{ respecto a sus dimensiones} \\ & \text{referenciales, en el trayecto del cliente } j \text{ al } j + 1. \\ 0, & \text{si no} \end{cases}$$

$$NoObstruye_{cic'i'j} = \begin{cases} 1, & \text{el objeto } i \text{ del cliente } c \text{ no es obstruido por el objeto } i' \text{ del} \\ & \text{cliente } c' \text{ en el trayecto del cliente } j \text{ al } j + 1. \\ 0, & \text{si no} \end{cases}$$

$$SeManipula_{cij} = \begin{cases} 1, & \text{Si el objeto } i \text{ del cliente } c \text{ es manipulado durante la visita } j \\ 0, & \text{si no} \end{cases}$$

$Manipulaciones\_x\_Objeto_{ci} =$  Total de manipulaciones del objeto  $i$  del cliente  $c$ .  
Corresponde a  $\sum_{j \in J} SeManipula_{c,i,j}$

Las variables  $x_{cij}$ ,  $y_{cij}$ ,  $w_{cij}$  y  $l_{cij}$  son todas enteras debido a que las dimensiones de los objetos y de la superficie de carga también lo son. Por otra parte, si fueran continuas existiría el riesgo de cometer errores de aproximación con las variables binarias que tienen asociadas en el modelo.

En la figura 5 se observa que según las definiciones presentadas en esta sección, la variable  $B1_{cic'i'j}$  tomaría el valor 0 y la variable  $B2_{cic'i'j}$  tomaría el valor 1, pues el objeto  $i$  no se encuentra a la izquierda del objeto  $i'$  y además se encuentra más abajo que el objeto  $i'$ , por lo que no podría ser descargado sin la necesidad de manipular también al objeto  $i'$ .

Las variables  $B1_{cic'i'j}$  y  $B2_{cic'i'j}$  son absolutas respecto a los ejes coordenados ubicados en el vehículo, por lo tanto, cuando la variable  $Giro_{c,i,j}$  toma el valor 1, las caras Norte,

Sur, Este y Oeste del objeto  $i$  del cliente  $c$  quedan determinadas por su orientación respecto a la superficie de carga, como lo muestra la figura 5.

### 3.3. Formulación Matemática: Función Objetivo y Restricciones

Función Objetivo:

$$\begin{aligned} \min \quad & \sum_{k \in K} \sum_{c \in C} \sum_{\substack{c' \in C: \\ c' \neq c}} \text{costo}_{c,c'} \cdot Z_{c,c',k} \\ & + \sum_{\substack{c \in C: \\ c \neq \{0, |C|-1\}}} \sum_{i \in i_c} cm_{c,i} \cdot \text{Manipulaciones}_x_{Objeto}_{c,i} \end{aligned} \quad (15)$$

La función objetivo considera en su primer término el costo de ruteo y, en el segundo término, el costo por manipulación de carga en todas las paradas de cada ruta. Las manipulaciones en distintas paradas de cada objeto  $i$  de cada cliente  $c$  se agregan en la variable  $\text{Manipulaciones}_x_{Objeto}_{c,i}$ , como será detallado en la condición (42), "Continuidad de manipulaciones".

Restricciones:

Las restricciones a continuación están ordenadas según la dimensión principal del problema que abordan. Inicialmente se presentan restricciones asociadas al ruteo de los vehículos, para luego presentar restricciones que abordan la carga, la orientación de los objetos y finalmente la manipulación de la carga.

Restricción clásica de Ruteo

$$\sum_{\substack{c' \in C: \\ c' \neq c}} Z_{c,c',k} \leq 1 \quad \forall c \in C, k \in K \quad (16)$$

Restricciones clásicas del VRP, aseguran que no más de un arco incida sobre cada nodo.

### Continuidad Entrega-Ruta

$$z_{c,c',k} \geq Entregac_{c,i=1,j-1,k} + Entregac'_{c',i'=1,j,k} - 1 \quad (17)$$
$$\forall c \in C, c' \in C, j \in J, k \in K: c \neq c'$$

Obligan a  $z_{c,c',k}$  a ser 1 si se pasa del cliente  $c$  en la parada  $j-1$  al cliente  $c'$  en la parada  $j$ . Debido a que  $z_{c,c',k}$  es penalizado en la función objetivo, la variable tomará el valor cero si tiene la posibilidad de hacerlo.

### Bodega al comienzo

$$\sum_{k \in K} Entregac_{0,1,1,k} = |K| \quad (18)$$

Para que  $i_c \neq \emptyset$  se genera un ítem ficticio (*i.e.* ancho, alto y peso nulos) para la bodega y su dual. De esta manera se mantiene la estructura del modelo y se evita errores computacionales.

### Bodega al final

$$\sum_{k \in K} \sum_{\substack{c \in C: \\ c \neq |C|-1}} z_{|C|-1,c,k} \leq 0 \quad (19)$$

Evitan que un vehículo viaje desde la bodega dual hacia otro cliente, ya que en la bodega dual los vehículos deben finalizar sus recorridos.

### Entrega Ítems del mismo cliente

$$Entregac_{c,i=1,j,k} = Entregac_{ci'jk} \quad \forall c \in C, i' \in I_c, j \in J, k \in K \quad (20)$$

Estas restricciones implican que al visitar un cliente se le entreguen todos sus ítems.

### Continuidad Entrega-Satisfacción

$$\sum_{j \in J} Entrega_{c,i=1,j,k} = Visita_{c,k} \quad \forall c \in C, k \in K \quad (21)$$

Restricciones estructurales que exigen al vehículo  $k$  a visitar al cliente  $c$ , si es éste quien le hace entrega de sus ítems.

### Satisfacción entrega

$$\sum_{k \in K} \sum_{j \in J} Entrega_{cij,k} = 1 \quad \forall c \in C \setminus \{0, |C| - 1\}, i \in i_c \quad (22)$$

Exigen que cada objeto de cada cliente sea entregado en una única parada de uno de los vehículos.

### Visitas Bodega

$$\sum_{k \in K} \sum_{j \in J} Entrega_{c,i=1,j,k} = |K| \quad \forall c \in \{0, |C| - 1\} \quad (23)$$

Certifican que todos los vehículos visiten la bodega al comienzo y al final de sus recorridos. Estas restricciones son necesarias debido a que las restricciones (18) y (19), "Bodega al comienzo" y "Bodega al final", por la estructura de la formulación no son capaces por sí solas de asegurar que cada vehículo comience y finalice su respectiva ruta en la bodega. Sin estas restricciones se producen irregularidades, como rutas que no finalizan su recorrido en la bodega, ya que la restricción (19) exige que no hayan vehículos que salgan de la bodega dual, pero no asegura que las rutas finalicen en la bodega. A través de estas restricciones y las anteriores se asegura la visita de todos los nodos del problema.

### Rutas Continuas

$$\sum_{c \in C} Entrega_{c,i=1,j-1,k} \geq \sum_{c \in C} Entrega_{c,i=1,j,k} \quad \forall j \in J, k \in K: j > 1 \quad (24)$$

Evitan soluciones con paradas en las que no se entregan objetos. Esto se debe a que la variable *Entrega* no aumenta el costo directamente en la función objetivo, la variable que lo hace es  $z_{c,c',k}$ . Ver ecuación (17), "Continuidad Entrega-Ruta", donde se aprecia que la variable  $z_{c,c',k}$  puede ser nula si alguna de las variables *Entrega* involucradas también lo es.

### Concatenar Presencia

$$Presencia_{c,i,j-1,k} \geq Presencia_{c,i,j,k} \quad \forall c \in C, i \in I_c, j \in J, k \in K: j > 1 \quad (25)$$

Similar a las restricciones anteriores, si un objeto estuvo presente en alguna etapa, entonces también lo estuvo en las anteriores, y si un objeto no está presente en una etapa, tampoco lo estará en las posteriores.

### Un cliente por parada

$$\sum_{c \in C} Entrega_{c,i=1,j,k} \leq 1 \quad \forall j \in J, k \in K \quad (26)$$

Estas restricciones exigen que sólo un cliente pueda ser visitado en cada parada.

### Limite de peso

$$\sum_{\substack{i \in I_c \\ c \in C}} weigh_{c,i} \cdot Presencia_{cijk} \leq Q \quad \forall k \in K, j \in J \quad (27)$$

El peso agregado de los objetos transportados en cada ruta no debe exceder la capacidad del vehículo que los transporta.

### Post Entrega

$$1 - Entrega_{cij k} \geq Presencia_{c,i,j+1,k} \quad (28)$$

$$\forall c \in C, i \in I_c, j \in J, k \in K: j < |C|$$

Obligan a la variable “Presencia” a ser cero justo después de la entrega del objeto.

### Capacidad superficial

$$-M \cdot SeManipula_{c,i,j} \leq x_{c,i,j} \quad \forall c \in C, i \in I_c, j \in J \quad (29)$$

$$W - x_{c,i,j} - w_{c,i,j} \geq M \cdot (Presencia_{cij k} - SeManipula_{c,i,j} - 1) \quad (30)$$

$$\forall c \in C, i \in I_c, j \in J, k \in K$$

$$-M \cdot SeManipula_{c,i,j} \leq y_{c,i,j} \quad \forall c \in C, i \in I_c, j \in J \quad (31)$$

$$L - y_{c,i,j} - l_{c,i,j} \geq M \cdot (Presencia_{cij k} - SeManipula_{c,i,j} - 1) \quad (32)$$

$$\forall c \in C, i \in I_c, j \in J, k \in K$$

Todos los objetos deben estar completamente contenidos en la superficie de carga, a menos que se manipulen, caso en que se considera que se pueden sacar del vehículo.

### Packing Inicial

$$0 \leq x_{c,i,j=1} \quad \forall c \in C, i \in I_c \quad (33)$$

$$W - x_{c,i,j=1} - w_{c,i,j=1} \geq M \cdot (Presencia_{c,i,j=1,k} - 1) \quad \forall c \in C, i \in I_c, k \in K \quad (34)$$

$$0 \leq y_{c,i,j=1} \quad \forall c \in C, i \in I_c \quad (35)$$

$$W - x_{c,i,j=1} - w_{c,i,j=1} \geq M \cdot (Presencia_{c,i,j=1,k} - 1) \quad \forall c \in C, i \in I_c, k \in K \quad (36)$$

Al empacar en bodega, todos los productos deben estar contenidos dentro de las rutas correspondientes.

### Orientación de los ítems

$$w_{c,i,j} = Giro_{c,i,j}length_{c,i} + (1 - Giro_{c,i,j})width_{c,i} \quad \forall c \in C, i \in I_c, j \in J \quad (37)$$

$$l_{c,i,j} = Giro_{c,i,j}width_{c,i} + (1 - Giro_{c,i,j})length_{c,i} \quad \forall c \in C, i \in I_c, j \in J \quad (38)$$

Determinan si el objeto  $i$  del cliente  $c$  se encuentra rotado respecto a sus dimensiones referenciales.

### Manipulación en cadena

$$\begin{aligned} & SeManipula_{c',i',j} - NoObstruye_{c',i',c,i,j-1} \\ & \leq SeManipula_{c,i,j} + (2 - Presencia_{c,i,j,k} - Presencia_{c',i',j,k}) \quad (39) \\ & \forall c \in C, i \in I_c, c' \in C, i' \in I_{c'}, j \in J, k \in K : j > 1 \end{aligned}$$

Si los ítems  $i$  e  $i'$  están presentes en el mismo vehículo y se manipula el objeto  $i'$ , entonces también se manipula el objeto  $i$ , sólo si éste último lo obstruye. Ver ejemplo en Anexo D.

### Manipulación en entrega

$$\begin{aligned} SeManipula_{i,c,j} & \geq \sum_{k \in K} Entrega_{cij,k} \quad (40) \\ & \forall c \in C \setminus \{0, |C| - 1\}, i \in I_c, j \in J \end{aligned}$$

Exigen que los objetos sean manipulados al ser entregados.

### Continuidad Satisface – Presencia

$$\sum_{i \in I_c} \sum_{j \in J} Presencia_{c,i,j,k} \leq M \cdot Visita_{c,k} \quad \forall c \in C, k \in K \quad (41)$$

Estas restricciones exigen que cada ítem pueda estar presente sólo en vehículo que los transporta.

### Continuidad de manipulaciones

$$Manipulaciones\_x\_Objeto_{c,i} \geq \sum_{j \in J} SeManipula_{c,i,j} \quad \forall c \in C \setminus \{0, |C| - 1\}, i \in I_c \quad (42)$$

La variable  $Manipulaciones\_x\_Objeto_{c,i}$  agrega las manipulaciones realizadas en cada parada al ítem  $i$  del cliente  $c$ .

### Manipulación en presencia

$$Visita_{c,k} + SeManipula_{c,i,j} - 1 \leq Presencia_{c,i,j,k} \quad (43)$$
$$\forall c \in C, i \in I_c, j \in J, k \in K$$

Obligan a estar presente a un objeto que es manipulado en el vehículo que lo entrega.

### Manipulación entre etapas

$$M \cdot SeManipula_{c,i,j-1} \geq x_{c,i,j} - x_{c,i,j-1} \quad (44)$$

$$M \cdot SeManipula_{c,i,j-1} \geq x_{c,i,j-1} - x_{c,i,j} \quad (45)$$

$$M \cdot SeManipula_{c,i,j-1} \geq y_{c,i,j} - y_{c,i,j-1} \quad (46)$$

$$M \cdot SeManipula_{c,i,j-1} \geq y_{c,i,j-1} - y_{c,i,j} \quad (47)$$

$$M \cdot SeManipula_{c,i,j-1} \geq Giro_{c,i,j} - Giro_{c,i,j-1} \quad (48)$$

$$M \cdot SeManipula_{c,i,j-1} \geq Giro_{c,i,j-1} - Giro_{c,i,j} \quad (49)$$

$$\forall c \in C, i \in I_c, j \in J$$

Un objeto es manipulado si tiene distinta posición u orientación al terminar una etapa respecto de la anterior. También será manipulado si un objeto al que obstruye fue, a su vez, manipulado. Ver restricciones (39), “Manipulación en cadena”.

### No apilamiento

$$(x_{c,i,j} + w_{c,i,j}) - x_{c',i',j} \leq M(3 - B1_{cic'i'j} - Presencia_{c,i,j,k} - Presencia_{c',i',j,k}) \quad (50)$$

$$(y_{c,i,j} + l_{c,i,j}) - y_{c',i',j} \leq M(3 - B2_{cic'i'j} - Presencia_{c,i,j,k} - Presencia_{c',i',j,k}) \quad (51)$$

$$B1_{cic',i'j} + B1_{c',i'cij} + B2_{cic',i'j} + B2_{c',i'cij} \geq 1 \quad (52)$$

$$\forall c \in C, i \in I_c, c' \in C, i' \in I_{c'}, j \in J, k \in K: (i, c) \neq (i', c')$$

Si los ítems  $i$  e  $i'$  están presentes en el mismo vehículo, se debe cumplir que uno de ellos esté ubicado a la izquierda, o bien, más arriba que el otro. De manera que no se traslapen.

### No obstrucción

$$NoObstruye_{cic'i'j} \leq B1_{cic'i'j} + B1_{c'i'cij} + B2_{c'i'cij} \quad (53)$$

$$\forall c \in C, c' \in C, i \in I_c, i' \in I_{c'}, j \in J: (i, c) \neq (i', c')$$

Estas restricciones implican que el objeto  $i'$  no obstruye al objeto  $i$ , sólo si el primero no se encuentra ubicado interceptando el camino vertical que sigue el objeto  $i$  al ser extraído del vehículo.

Hay que destacar que no se necesitan restricciones para controlar los sub tours ni asegurar la existencia de un arco saliente (entrante) a cada nodo, ya que la variable “*Entrega*” obliga a que se satisfaga la restricción de eliminación de ciclo de forma implícita.

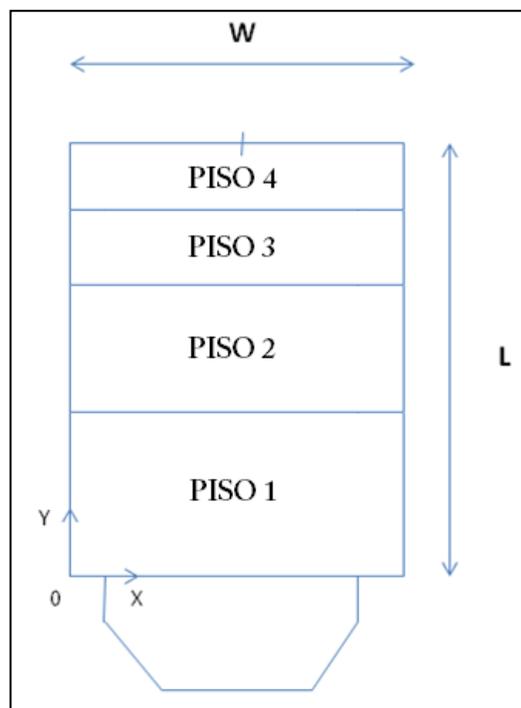
#### 4. ENFOQUE DE SOLUCIÓN PROPUESTO

Para resolver el 2LCVRP-H se formuló el problema de programación entera mixta (MIP) descrito, sin embargo, debido a la complejidad de este modelo, sólo fue posible solucionarlo a optimalidad para instancias pequeñas, cuyo tamaño alcanza los 6 clientes, 12 ítems y 2 vehículos en promedio. Por lo tanto, se desarrolló una herramienta metaheurística que hace uso de este modelo sólo en la última de sus etapas cuando ya se han fijado algunas de las variables, lo que permite resolverlo. Esta herramienta resuelve separada y secuencialmente el ruteo y *packing* de los vehículos. Este proceso no obtiene inmediatamente una solución inicial factible, sino que fija una serie de variables del modelo original y así limita el espacio de soluciones posibles con las que cuenta el *solver* al resolver el MIP. Esto permite que el proceso completo se lleve a cabo en un tiempo razonable para instancias de mayor tamaño. La herramienta metaheurística se descompone en las siguientes tres etapas que se describen en más detalle en las secciones 4.1, 4.2 y 4.3 del documento:

- i) la primera, se basa sólo en la información de los clientes, incluyendo ubicación y demanda por ítems con sus respectivas áreas y pesos, así como la topografía de la red. Con esto se aplica la heurística de Clarke & Wright (1964) modificada para considerar áreas y pesos de los ítems, es decir, esta heurística constructiva de tipo *greedy* se emplea cuidando no sobrepasar la capacidad del vehículo ni la superficie disponible en el compartimento de carga. Este proceso entrega valores iniciales para la mayor parte de las variables  $Entrega_{cijk}$ , asociadas al ruteo de cada vehículo, debido a que, para dar más flexibilidad a la heurística, no todos los clientes son asignados a los vehículos disponibles.

Luego se aplica un algoritmo 2-opt al interior de cada ruta para optimizar el costo de ruteo. Este proceso busca ahorros de costo producto de intercambiar el orden dentro de la ruta en que dos clientes son visitados.

- ii) en la segunda etapa, se aplica la heurística de *Floor-Ceiling* (FC) para obtener el *packing* de los objetos de cada ruta, considerando las rutas entregadas por el algoritmo del punto anterior, pero no así el orden de las visitas. Esta heurística es de tipo constructiva y se aplica a cada ruta independientemente. La heurística comienza generando una lista de todos los objetos de la ruta ordenados de forma descendiente, según el largo del mayor de sus lados. Por lo tanto, inicialmente se tendrá que  $l_{cij} \geq w_{cij}$  para todo cliente  $c$ , ítem  $i$  y parada  $j$ . Posteriormente se encarga de realizar secuencialmente el *packing* de los objetos en subdivisiones rectangulares del compartimento de carga denominadas “pisos”, cuyo ancho  $W$ , es el mismo ancho de la superficie de carga, es decir, el largo de la superficie de carga corresponde a la suma de las alturas de todos los pisos generados por la heurística FC. Ver figura 6.



**Figura 6: Distribución de pisos en la superficie del vehículo.**

Como los objetos se empacan según la lista anteriormente mencionada, el primer objeto ubicado en cada piso será el más largo y determinará el largo  $L'$  del piso. Se

sitúan los ítems uno a uno desde la esquina inferior izquierda del piso respetando las restricciones de no traslape de objetos. Éste también corresponde a un proceso tipo *greedy* y se repite hasta completar la base del piso. Luego se repite el procedimiento de empaque de objetos comenzando desde la esquina superior derecha del mismo piso, lugar donde aún es probable que quepan algunos de los objetos restantes en la lista.

Al completar un piso, se procede a iniciar uno nuevo. La cantidad de pisos que se generan dependerá del largo de cada uno de ellos. La solución obtenida corresponde a un *packing* factible para los objetos posicionados dentro de cada vehículo, sin embargo, este *packing* no es definitivo, ya que aún existen variables  $Entrega_{cijk}$  correspondientes a clientes que aún no han sido asignados a ninguna ruta, es decir, el valor de estas variables no fue determinado en la primera etapa y el *packing* de estos objetos se determina en la etapa que se describe a continuación.

- iii) en la tercera etapa y final, se busca mejorar la solución formada a partir del ruteo realizado en la primera etapa y el *packing* propuesto en la segunda. Este proceso se realiza aplicando el modelo MIP desarrollado en el capítulo 2 para resolver el 2LCVRP-H, pero manteniendo fijas las rutas preliminares establecidas en la primera etapa, de esta manera, el *solver* modificará el *packing* propuesto y determinará el ruteo definitivo, considerando a todos aquellos clientes que no fueron asignados a ninguna ruta en la primera etapa. El modelo determina para estos clientes no sólo el ruteo sino también el *packing* definitivo.

#### **4.1. Etapa I: Heurística de Clarke & Wright Modificada**

Se utiliza una versión modificada de la heurística de Clarke & Wright como paso inicial en la búsqueda de una solución inicial sólo para las variables  $Entrega_{cijk}$  de ruteo. La heurística de Clarke & Wright entrega en promedio soluciones cuyo gap promedio es de 4,87 % para instancias con un promedio de 63 nodos (Faulin *et al.*, 2010), por lo tanto, es

razonable esperar un gap aún menor para instancias de menor tamaño cuyas rutas serán mejoradas con un proceso 2-opt, a pesar de las restricciones adicionales con las que cuenta el 2LCVRP-H.

Para reducir la carga computacional que este procedimiento implica, en esta versión de la heurística de Clarke & Wright sólo se considera la inclusión de nodos adyacentes al depósito, por lo tanto, no se evalúa la inclusión de un arco  $(a,b)$  a una ruta en el caso que  $a$  o  $b$  no sean el primer o último nodo en ser visitado de sus actuales rutas. A causa de este procedimiento, se puede construir rutas con arcos que se cruzan entre sí, no obstante, el algoritmo 2-Opt que se aplica posteriormente corrige este tipo de rutas.

Este proceso comienza asignando una ruta separada para cada cliente, es decir, inicialmente se consideran  $|C|$  rutas. Luego, se ordenan todos los pares de clientes  $a$  y  $b$  en forma descendiente en función de los ahorros asociados a recorrer el arco que los conecta. Para realizar esto se define el ahorro asociado a visitar al cliente  $a$  y al cliente  $b$  en la misma ruta como:

$$Saving_{a,b} = c_{0,a} + c_{0,b} - c_{a,b} \quad (55)$$

Siendo  $c_{a,b}$  el costo de recorrer el arco  $(a,b)$  y considerando que 0 corresponde al depósito ( $c_{a,b} = c_{b,a}$ , es decir, costos simétricos).

Hay que destacar que en esta etapa no se utilizan configuraciones de empaque al momento en que este algoritmo añade un nuevo ítem, ya que al hacer esto se estaría resolviendo conjuntamente ruteo y empaque, que es lo que se quiere evitar. Sin embargo, sí se utiliza el peso y las dimensiones de los mismos para determinar cuándo se sobrepasa la capacidad de carga (ec. 56) o de superficie (ec. 57) del vehículo. Para una ruta  $k$ , ésta condición queda definida como:

$$\sum_{\substack{c \text{ in } \text{CLIENTES} \\ \text{de Ruta } k}} \sum_{i \text{ in } \text{ITEMS}_c} \text{weigh}_{i,c} \leq Q \quad (56)$$

o bien,

$$\sum_{\substack{c \text{ in } \text{CLIENTES} \\ \text{de Ruta } k}} \sum_{i \text{ in } \text{ITEMS}_c} w_{i,c} \cdot l_{i,c} \leq L \cdot W \cdot \delta \quad (57)$$

Siendo  $Q$  la capacidad de cada vehículo (flota homogénea),  $\text{weigh}_{i,c}$ ,  $w_{i,c}$  y  $l_{i,c}$  el peso, ancho y largo del ítem  $i$  del cliente  $c$  respectivamente. En este proceso es necesario considerar una superficie de carga efectiva menor a la disponible, pues por una parte la estiba de los ítems (incluso en la solución óptima) podría exigir perder ciertos espacios en el camión. Por otra parte, dado que el algoritmo consistirá en fijar la asignación de ítems a vehículos, cuánto más holgura se entregue en esta etapa, mayor será la flexibilidad que tendrá la herramienta para reasignar ítems y entregar una mejor solución (por otra parte mayor será el tiempo que tomará en resolverse). Así, en esta etapa se contempla un factor de reserva en la superficie del vehículo,  $\delta$ , que se utiliza para determinar cuánto menor será la superficie de carga con respecto a la disponible.

Se desarrollan todas las rutas simultáneamente, verificando que estos criterios se cumplan al añadir un nuevo cliente a una ruta determinada. Si se comprueba que la ecuación (56) o (57) no se cumplen, no se añade ese cliente a esa ruta y se procede a evaluar la inclusión del siguiente arco que implique el mayor ahorro. Este procedimiento se reitera hasta que ya no sea conveniente incluir algún arco a una ruta distinta a la actual. Esta etapa puede utilizar una cantidad superior a los  $K$  vehículos disponibles, pero sólo las primeras  $K$  rutas se consideran dentro del input de la Etapa III, dando libertad al MIP para definir el ruteo y *packing* definitivo, incluyendo la reasignación de los clientes que no fueron asignados a las primeras  $K$  rutas.

En este proceso se evalúa añadir cada arco  $(a,b)$  a una ruta según su posición en la lista y sólo se añade mientras no se sobrepase la capacidad de carga ni la capacidad superficial multiplicada por el factor de reserva para asegurar un *packing* factible dentro del vehículo.

Se consideran los siguientes tres escenarios para añadir un arco  $(a,b)$  a una ruta:

- i) El cliente  $a$  y el  $b$  no forman parte de una ruta, es decir, se encuentran en las rutas creadas inicialmente, en las que un vehículo sólo los satisface a un cliente individualmente. En este caso, se modifican las rutas  $(0,a,0)$  y  $(0,b,0)$  a cambio de la ruta  $(0,a,b,0)$ .
- ii) Exactamente uno de los clientes ( $a$  o  $b$ ) ha sido incluido en una ruta y es no interno (*i.e.* es adyacente al depósito, ya sea como primer cliente visitado o el último antes de retornar). En éste caso, se agrega el arco  $(a,b)$  al comienzo de la ruta (o respectivamente el arco  $(b,a)$  al final).
- iii) Ambos clientes forman parte de una ruta y son no internos. En este caso se realiza la unificación de ambas rutas.

Posteriormente se aplica el algoritmo 2-opt a las rutas obtenidas, con el objeto de mejorar la solución. Este algoritmo corta dos arcos de la misma ruta y reconecta los 4 puntos involucrados de manera que se obtenga un nuevo ruteo, asegurando que no se construyan 2 *subtours*. Este proceso se repite para todos los pares posibles de la ruta y si el costo total de recorrer la misma es menor que el original, entonces se realiza este cambio. Se itera hasta que no haya cambios posibles de este tipo, que sean capaces de reducir los costos de ruteo en alguna ruta.

Como fue mencionado anteriormente, en caso que la cantidad de rutas determinadas en esta etapa sea mayor al tamaño de la flota de la instancia, sólo se fijan las variables  $Entrega_{cijk}$  para las primeras  $K$  rutas (siendo  $K$  la cantidad de vehículos disponibles),

cuyos ahorros de los arcos que las componen son superiores, de manera que no se exceda el tamaño de la flota. En este caso, los clientes no satisfechos deberán ser asignados a rutas en la etapa III, donde se aplicará el MIP para mejorar la solución definitiva y hacerla factible, en caso que aún no lo sea.

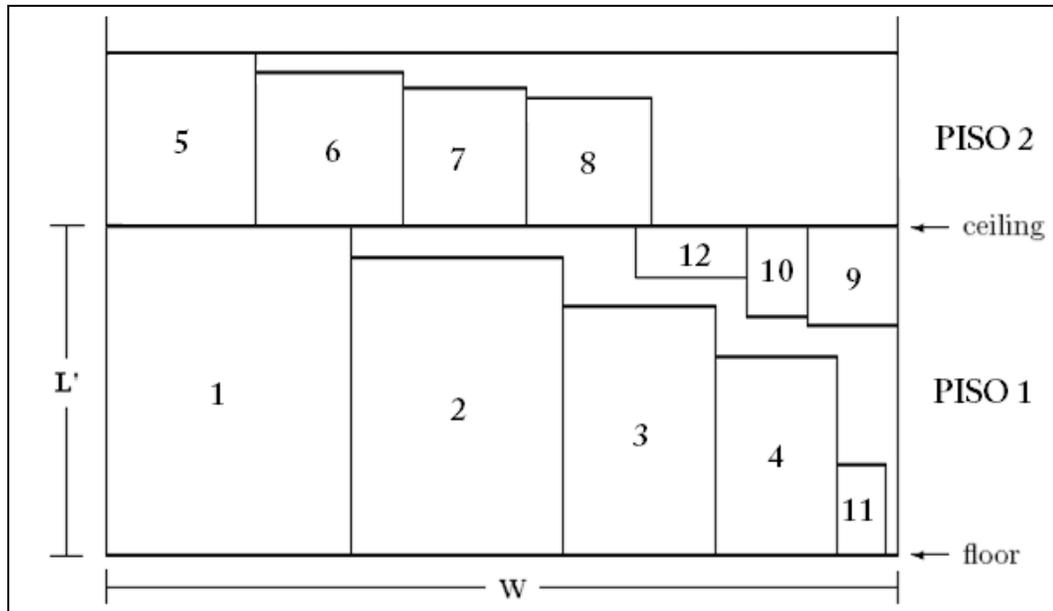
La decisión de operar con más rutas que las definidas en la instancia podría haberse incluido en el modelo como una restricción blanda con una penalidad asociada al uso de vehículos ficticios, pero se asumió el tamaño de la flota como una característica relevante para el modelo, en el sentido que se forzará a ser respetada o la instancia será declarada infactible.

El factor de reserva se fija inicialmente en uno para cada instancia, luego, si el problema se hace infactible debido a que las rutas creadas no tienen *packings* factibles, se procede a reducir el factor de reserva  $\delta$  en un 5%. Se realiza esto hasta que el problema sea factible o se alcance un factor de reserva nulo. Cabe mencionar que mientras menor es el factor de reserva, las rutas presentan menos visitas, por lo que al finalizar el ruteo quedan más clientes sin asignar. Todo esto entrega un mayor grado de libertad al problema de optimización, ya que se reduce el número de variables a las que se fija su valor, aumentando exponencialmente la complejidad del problema. Adicionalmente, se debe considerar que a la dificultad en la resolución de una instancia de CVRP se le debe añadir la complejidad del 2BPP asociado a los objetos presentes en la misma, lo que hace disminuir el dominio de soluciones factibles de la instancia.

#### **4.2. Etapa II: Heurística de *Floor & Ceiling* (FC)**

A partir de las rutas determinadas en la primera etapa, se procede a realizar el *packing* inicial de cada ruta. Con este propósito se ordenan todos los ítems de los clientes visitados por cada ruta en orden descendente, según el largo del mayor de sus lados,  $l_{ci}$ . Así se determina simultáneamente un valor inicial para la variable asociada al giro de cada objeto. Luego, inicialmente se tendrá que  $l_{cij} \geq w_{cij}$  para todo cliente  $c$ , ítem  $i$  y parada  $j$ .

El largo  $L'$  de un piso queda definido por la cara inferior y superior del ítem más largo en dicho piso (en figura 7, objeto 1 en el piso 1 y objeto 5 en el piso 2). Como los objetos están previamente ordenados según largo, este ítem siempre se encuentra posicionado en la esquina inferior izquierda del piso respectivo.



**Figura 7: Vista superior de los primeros dos pisos en la aplicación de Heurística FC, señalando *floor* y *ceiling* del piso 1.**

A continuación se procede a disponer los productos de acuerdo a la lista, comenzando por la esquina inferior izquierda y avanzando hacia la derecha como muestra la figura 7 (objetos 1, 2, 3 y 4). Si el siguiente objeto de la lista no puede ser ubicado a continuación por exceder el ancho del vehículo, se procede a intentar ubicar los siguientes objetos de la lista. En el ejemplo de la figura 7, se prueba con los objetos 5 a 10 sin éxito, hasta que el objeto 11 es posible ubicarlo entre el objeto 4 y la cara lateral derecha del compartimento de carga. Una vez que no se puede ubicar el siguiente producto a la derecha de los anteriores, se procede a ubicar objetos desde la esquina superior derecha, avanzando hacia

la izquierda, sólo si ninguno de éstos se traslapa con los situados previamente, ni se excede la capacidad superficial del piso.

Cuando un objeto a ser posicionado en el “techo” del piso se traslapa con los anteriores, se le aplica un giro en 90°, y en caso de aún ser infactible se procede a posicionar el siguiente producto. En el ejemplo, se ubican de esta manera los objetos 9, 10 y 12, ya que los anteriores (5, 6, 7 y 8) superaban la capacidad superficial del piso 1. Este procedimiento ha sido contrastado con otras metodologías en la literatura (Lodi *et al.* 2002), mostrando que la heurística *FC* se encuentra dentro de las que tienen mejor desempeño y superan ampliamente los enfoques clásicos (*e.g. Next Fit, First fit, Best fit*).

El proceso de creación de pisos se reitera  $n$  veces, de manera que se logre ubicar la mayor parte de los objetos dentro del vehículo. El valor de  $n$  depende de la instancia, ya que si los objetos en el vehículo analizado son de dimensiones más bien pequeñas, se busca trabajar con un valor de  $n$  mayor, de manera que el *solver* no pase demasiado tiempo determinando el valor para todas las variables estructurales del MIP desarrollado en el capítulo 2.

Para cada ruta en el algoritmo de *Floor Ceiling*, la cantidad de pisos, " $n$ ", a construir se determina según:

$$n = \begin{cases} 1, & \text{Max\_Largo\_Item} > L/2 \\ 2, & \text{en otro caso} \end{cases}$$

esta definición de  $n$  fue calibrada en corridas previas y mantiene un buen equilibrio entre tiempo de cómputo para lograr la solución inicial y desempeño general. Esto se puede verificar más adelante, en el capítulo de experimentación, al contrastar con instancias a las que no se entregan valores iniciales de *packing* ni ruteo, provocando un gap cercano al 80% en la solución inicial factible encontrada por el *solver*, mientras que en las instancias resueltas a las que sí se les entrega un *packing* y ruteo inicial se observa gaps iniciales cercanos al 4% en menos de 2 minutos.

Se debe mencionar que la solución inicial provista por las etapas I y II de ruteo y *packing* respectivamente, probablemente no será óptima en términos de manipulaciones, ya que el empaque realizado está condicionado a disponer los productos de los clientes asignados exclusivamente a esa ruta, proceso en el que no se consideraron las manipulaciones que dicho ruteo acarrearía. Aún si los objetos quedaran ubicados linealmente paralelos a la cara izquierda del compartimento de carga, podría ocurrir que el primer producto a ser entregado se encuentre obstruido por los demás. Éste tipo de escenarios sirve como motivación para la aplicación del modelo MIP y la mejora de la solución provista en la etapa III.

### **4.3. Etapa III: Optimización de la solución**

A partir del ruteo obtenido en la Etapa I y el *packing* propuesto en la Etapa II, se procede a resolver el MIP desarrollado inicialmente para el 2LCVRP-H en el segundo capítulo. Para este paso se conservan las rutas parciales obtenidas al terminar la primera etapa, es decir, al *solver* se le impide alterar la secuencia de clientes visitados por cada vehículo (*i.e.* el orden de visitas se mantiene), pero sí puede agregar clientes al final de las rutas definitivas y mejorar el *packing* determinado en la segunda etapa.

Como las rutas obtenidas de la etapa I pueden no satisfacer a todos los clientes debido a que se excede la flota disponible de la instancia, *packing* y ruteo no son separables, ya que para serlo se debería contar con las rutas definitivas y saber de antemano que existe al menos un *packing* factible asociado a dicho ruteo. Considerar que en dicho caso, sólo algunas las variables  $Entrega_{cijk}$  tendrán valores fijos provenientes de la Etapa I y que las variables  $Giro_{c,i,j}$ ,  $x_{cij}$ ,  $y_{cij}$ ,  $w_{cij}$  e incluso  $l_{cij}$  podrían no tener asignados valores iniciales al finalizar la Etapa II. Inicialmente, se dejan algunas de estas variables sin asignar debido a que el *packing* o el ruteo podrían ser infactibles. En caso de que a todas se les diera un valor inicial, sin excepción para todos los subíndices asociados a ellas, se forzaría al *solver* a malgastar recursos en tratar de determinar un valor factible para todas las demás variables del modelo sin asegurar que éstos existan.

Cabe mencionar que debido a que el ruteo y el *packing* se realizan separados en el enfoque de solución propuesto en este trabajo, se podrá obtener buenas soluciones en relación a distancia recorrida, sin embargo, las mismas podrían implicar una mayor cantidad de manipulaciones que las estrictamente necesarias si se hubiese realizado *packing* y ruteo simultáneos.

## 5. EXPERIMENTO Y ANÁLISIS DE RESULTADOS

En la primera sección de éste capítulo se presenta el diseño del experimento realizado para evaluar los métodos de solución propuestos sobre instancias de tamaño reducido y, posteriormente, en la resolución de instancias de mayor tamaño. En la segunda sección se definen las opciones de optimización utilizadas para mejorar el rendimiento del proceso de optimización. En la tercera sección, se describe el *hardware* utilizado para realizar los experimentos. Finalmente, en la cuarta sección de éste capítulos se analizan los resultados tanto para la optimización conjunta como para la optimización parcialmente separada.

### 5.1. Diseño del experimento

La modelación conjunta (*i.e.* aplicación directa del MIP inicial) fue testeada sin aplicar algoritmos de solución inicial ni fijación de rutas en 15 instancias arbitrarias creadas aleatoriamente, según se describe a continuación de tamaño aproximado a 6 clientes, entre 5 y 10 ítems, y 1 ó 2 rutas.

Para probar el desempeño del algoritmo propuesto para optimización parcialmente separada (*i.e.* Heurística de tres etapas), se definió un conjunto de instancias basado en las utilizadas originalmente por Toth y Vigo (2002) para el CVRP y extendido posteriormente por Iori *et. al* (2005) y Gendreau *et. al* (2005) para el 2L-CVRP. Cada una de las instancias incluye la información del grafo (*i.e.* ubicaciones de clientes y bodega), el peso, ancho y largo de los ítems demandados por cada cliente, la cantidad de vehículos disponibles, con su respectiva capacidad y dimensiones (flota homogénea). Los costos de los arcos se obtuvieron a través del cálculo de distancias euclidianas entre cada par de nodos.

Para testear la optimización parcialmente separada, se seleccionaron al azar las instancias con aproximadamente 60 productos, 20 clientes y 5 vehículos, pues se comprobó en corridas previas que el proceso heurístico de 3 etapas no logra resolver instancias de mayor tamaño en tiempo razonable, llegando incluso a transcurrir días sin alcanzar una

solución. Cada instancia puede ser clasificada según la tabla 2, donde se observa que cada objeto generado para una determinada instancia tuvo la misma probabilidad de pertenecer a la categoría “Vertical”, “Homogéneo” u “Horizontal” (categorías en relación a las dimensiones de referencia del objeto). Dentro cada una de estas clasificaciones se determinó aleatoriamente el ancho y alto de los objetos con respecto a las dimensiones del vehículo de carga, de manera similar a como se hace en la literatura para el 2BPP.

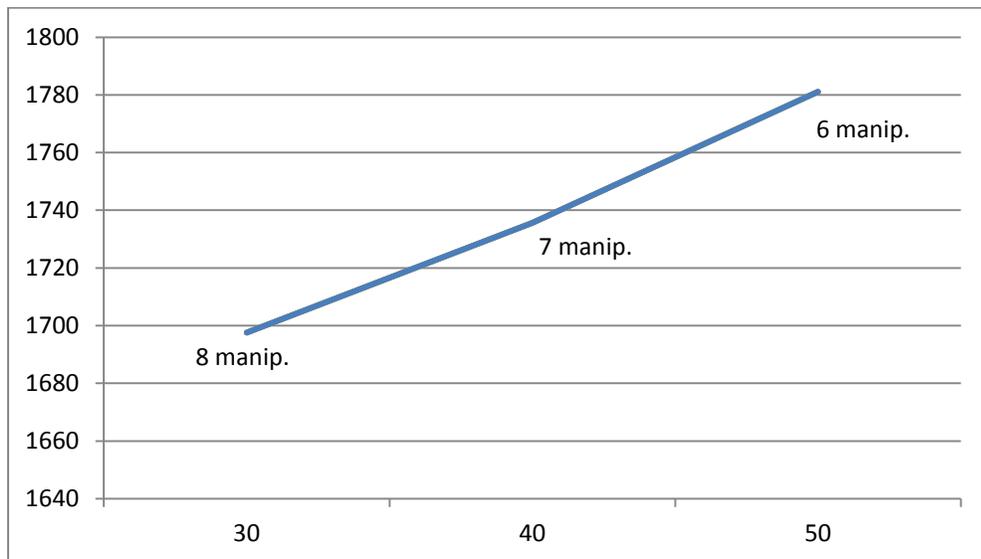
Clase	$m_i$	Verticales		Homogéneos		Horizontales	
		$h_{ci}$	$w_{ci}$	$h_{ci}$	$w_{ci}$	$h_{ci}$	$w_{ci}$
1	1	1	1	1	1	1	1
2	[1,2]	$\left[\frac{4L}{10}, \frac{9L}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{2L}{10}, \frac{5L}{10}\right]$	$\left[\frac{2W}{10}, \frac{5W}{10}\right]$	$\left[\frac{L}{10}, \frac{2L}{10}\right]$	$\left[\frac{4W}{10}, \frac{9W}{10}\right]$
3	[1,3]	$\left[\frac{3L}{10}, \frac{8L}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{2L}{10}, \frac{4L}{10}\right]$	$\left[\frac{2W}{10}, \frac{4W}{10}\right]$	$\left[\frac{L}{10}, \frac{2L}{10}\right]$	$\left[\frac{3W}{10}, \frac{8W}{10}\right]$
4	[1,4]	$\left[\frac{2L}{10}, \frac{7L}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{L}{10}, \frac{4L}{10}\right]$	$\left[\frac{W}{10}, \frac{4W}{10}\right]$	$\left[\frac{L}{10}, \frac{2L}{10}\right]$	$\left[\frac{2W}{10}, \frac{7W}{10}\right]$
5	[1,5]	$\left[\frac{L}{10}, \frac{6L}{10}\right]$	$\left[\frac{W}{10}, \frac{2W}{10}\right]$	$\left[\frac{L}{10}, \frac{3L}{10}\right]$	$\left[\frac{W}{10}, \frac{3W}{10}\right]$	$\left[\frac{L}{10}, \frac{2L}{10}\right]$	$\left[\frac{W}{10}, \frac{6W}{10}\right]$

**Tabla 2: Clases usadas para la generación de objetos.**

Las instancias están ordenadas en 5 clases y se sigue la notación provista para el modelo MIP en el capítulo 3, donde  $h_{ci}$  y  $w_{ci}$  corresponden al largo y ancho de referencia del objeto  $i$  del cliente  $c$ , respectivamente. En la clase " $i$ " la cantidad de ítems demandados por cada cliente,  $m_i$ , varían de 1 hasta  $i$ . Cabe destacar que la clase 1 corresponde a la instancia original del CVRP y sólo considera la posibilidad de generar un ítem con dimensiones unitarias para cada cliente.

Para todas las instancias se consideró el costo de manipulación  $cm$  como unitario para todos los objetos, de manera que se pudiera comparar nuestros resultados de optimización parcialmente separada con los presentados por Iori *et al.* (2005) y Gendreau *et al.* (2005), es decir, cada manipulación se considera equivalente a 1 kilómetro de distancia en la función objetivo.

Hay que destacar que en la medida que aumentaban los valores de  $cm$ , la solución entregada incluía una menor manipulación de carga, asemejándose a lo realizado con "carga secuencial", y en la medida que los valores de  $cm$  disminuían, el ruteo realizado tendía más bien a "carga irrestricta", lo que se puede observar en la figura 8, donde se observa que al incrementar el costo de manipulación " $cm$ ", se trata de minimizar la cantidad de veces que se manipulan los objetos a costa de un mayor costo de ruteo.



**Figura 8: Costo de ruteo vs. " $cm$ ". Al aumentar " $cm$ " el costo de ruteo aumenta, pero la cantidad de manipulaciones es disminuye.**

## 5.2. Opciones de optimización

Debido a la intención de dar utilidad al algoritmo en problemas reales, se impone un máximo de 1800 segundos como tiempo límite de optimización en CPLEX 12 y durante ese tiempo se busca la mejor solución entera posible, es decir, la que tenga el menor *gap* relativo respecto a la mejor cota (*i.e. Best Bound*) o mejor valor objetivo de las relajaciones lineales de los nodos restantes del problema lineal (*i.e. Linear Problem, LP*). Así, el *gap* relativo queda definido según la siguiente ecuación:

$$gap\ relativo = \frac{|Valor\ Obj.entero - Mejor\ Valor\ Obj.LP|}{1.0 + Mejor\ Valor\ Obj.LP} \quad (58)$$

Luego, si el proceso de optimización halla una solución cuyo *gap* relativo es menor a un 5 %, se procede a "pulir" la solución (*i.e. Polishing*) durante 300 segundos. Esto se hace a través de una variación de *branch & cut* a partir de una solución inicial. Esta rutina se encuentra incorporada en las librerías de CPLEX y se encarga de buscar mejores soluciones en situaciones donde buenas soluciones serían, de otra manera, difíciles de alcanzar. De hecho, es necesaria al menos una solución para aplicar este procedimiento, ya que explora la vecindad de soluciones encontradas previamente. Debido al alto costo computacional que acarrea este procedimiento, el sistema sólo lo ejecuta después que el proceso habitual de *branch & cut* de CPLEX ha finalizado. Por lo tanto, no puede comprobar la optimalidad, incluso si la solución óptima ha sido alcanzada.

Con el propósito de agilizar el proceso de búsqueda de soluciones al invertir más recursos en conseguir soluciones factibles de calidad, el énfasis de búsqueda en el árbol de *branch & cut* se ajustó fijando el parámetro *MIPEmphasis* en la opción *Hidden Feasibility*. De otra manera se encuentran soluciones más lejanas al óptimo, respecto a las halladas con el énfasis seleccionado.

Cabe mencionar que cuando la búsqueda de soluciones de CPLEX debía seleccionar el siguiente nodo a revisar, se elegía el nodo con mejor valor objetivo estimado que se

obtendría al resolver la relajación lineal del problema. Esta rutina es interna en las librería de ILOG y se aplica fijando el parámetro *NodeSel* en *Best Estimate*. Al escoger otra estrategia de selección de nodo en corridas previas, se observó un incremento en el tiempo de ejecución del *solver*. No alcanzando siquiera una solución factible dentro del límite de tiempo establecido en algunas instancias.

### **5.3. Implementación**

La implementación fue realizada en C# y se utilizó las librerías de Concert desarrolladas por ILOG para la etapa de optimización en CPLEX 12. El programa fue ejecutado en un procesador Intel® Core™ i7 2.67GHz con 6GB de RAM, utilizando simultáneamente los 8 *threads* (4 núcleos con *Hyper Threading Technology*) de los que se disponía.

### **5.4. Análisis de resultados**

Debido a que el modelo por sí solo es poco práctico en instancias reales de gran tamaño, se decidió experimentar la optimización conjunta en instancias de pocos clientes, los suficientes para referirse principalmente a las cualidades del modelo y a su comportamiento respecto a la cantidad de clientes, ítems y vehículos disponibles.

Posteriormente, en la subsección 5.4.2., se analiza instancias de mayor tamaño, incluyendo la mayor de ellas con 21 clientes, 57 ítems y 5 vehículos. En éste segundo set se analizaron 70 instancias. La mitad de ellas permiten la rotación de los objetos, mientras que sobre la otra mitad se aplica la restricción de orientación de los ítems, es decir, los objetos no se pueden rotar.

### 5.4.1 Optimización conjunta

Se analizó un set de 15 instancias generadas aleatoriamente por el proceso descrito en el capítulo anterior, comprobando que la utilización de la superficie de los vehículos fuera alta (*i.e.* al menos superior al 85%), con el fin de analizar el desempeño del modelo al intentar resolver de manera exacta y óptima el problema. Cada conjunto de 5 instancias denominadas “Test00”, “Test01” y “Test02”, conservan las ubicaciones de los clientes pero se modifica la cantidad de ítems de cada uno de ellos. Estas instancias quedan definidas en la tabla 3, donde "Inst." es el nombre de la instancia, "n" es el número de clientes, "M" es el total de ítems en la instancia, "K" es el número de vehículos disponibles, "z\*" es el valor alcanzado tras la optimización, "z\_SL" corresponde al valor óptimo alcanzado agregando un vehículo adicional a la instancia y utilizando *Sequential Loading*, "diff%" es la diferencia porcentual entre el costo de ruteo logrado utilizando *Sequential Loading* y el costo de ruteo utilizando el modelo propuesto, "#Man." es la cantidad de manipulaciones realizadas en la solución propuesta, "Tpo 1° sol" corresponde a la cantidad de segundos transcurridos desde el inicio de la optimización hasta alcanzar la primera solución factible, "%gap 1° sol" corresponde al gap relativo de la primera solución factible encontrada, "Tpo total" corresponde a la duración total de la corrida, "%gap final" se refiere al gap relativo de la solución final de la corrida, "Bins" corresponde a la cantidad de variables binarias presentes en la instancia, "Ints" es la cantidad de variables no binarias en la instancia y, finalmente, "%Area" corresponde porcentualmente al cociente entre la superficie utilizada por ítems al salir de la bodega y la superficie total disponible en los vehículos.

En la tabla 3 se puede observar que el tiempo de ejecución crece exponencialmente tanto con el número de clientes, como en el de ítems y vehículos. De hecho, las instancias "Test02" con 9 y 10 clientes fueron abortadas a los 2100 segundos ya que, de no hacerlo, se mantenían iterando durante muchas horas.

Cabe destacar que la primera solución factible se alcanzaba rápidamente para la experimentación sin fijar rutas ni otorgar solución inicial al *solver*, pero a partir de ese momento, la cota superior comenzaba a mejorar de forma mucho más lenta. Lo mismo ocurría con el óptimo, si bien se alcanzaba rápidamente, el programa se tomaba la mayor parte del tiempo recorriendo el resto de las soluciones posibles que aún no habían sido revisadas, mejorando la cota dual para asegurar que la solución encontrada sea efectivamente óptima. Es por eso que a pesar que se verifica que el gap para las instancias mencionadas fuera alto, en realidad ya se había encontrado la solución óptima, basta con verificar que la diferencia del valor óptimo con las corridas previas es exactamente el de una unidad, correspondiente a una manipulación extra, debida a la presencia de un producto adicional.

Instancia	n	M	K	z*	z_SL	diff%	# Man.	Tpo 1° sol (seg)	%gap 1° sol	Tpo Total (seg)	%gap Final	Bins	Ints	% Area
<b>Test00</b>	5	6	1	115.02	131.86	16.38%	8	~ 0	74.25%	14.52	0.00%	759	84	100.00%
	5	7	1	115.02	124.93	9.49%	8	~ 0	47.84%	142.47	0.00%	1019	98	99.23%
	5	8	1	116.02	128.03	11.21%	9	~ 0	74.01%	504.33	0.00%	1319	112	99.23%
	5	9	1	116.02	116.02	0.00%	9	~ 0	72.26%	43.61	0.00%	1659	126	98.46%
	5	10	1	117.02	117.02	0.00%	10	~ 0	77.02%	390.84	0.00%	2032	140	97.69%
<b>Test01</b>	6	6	1	128.29	140.23	10.09%	7	~ 0	84.62%	12.66	0.00%	839	96	100.00%
	6	7	1	128.29	128.29	0.00%	7	~ 0	81.98%	102.06	0.00%	1132	112	89.23%
	6	8	1	129.29	129.29	0.00%	8	~ 0	70.37%	269.53	0.00%	1461	128	93.85%
	6	9	1	130.29	130.29	0.00%	9	~ 0	80.86%	670.05	0.00%	1834	144	94.62%
	6	10	1	131.29	131.29	0.00%	10	~ 0	72.89%	1214.9	0.00%	2251	160	93.85%
<b>Test02</b>	6	6	2	142.38	142.38	0.00%	6	~ 0	89.47%	391.25	0.00%	987	96	88.46%
	6	7	2	143.38	143.38	0.00%	7	~ 0	76.62%	1369.3	0.00%	1292	112	92.31%
	6	8	2	144.38	144.38	0.00%	8	883.9	72.74%	1747.6	0.00%	1651	128	92.31%
	6	9	2	145.38	145.38	0.00%	9	267.4	75.90%	2100	26.18%	2048	144	91.92%
	6	10	2	146.38	146.38	0.00%	10	722.4	78.04%	2100	73.15%	2491	160	90.38%

**Tabla 3: Resultados optimización conjunta.**

También se confirmó en las instancias de la tabla 3 y en corridas previas, que problemas con pocos clientes y muchos ítems se resuelven mucho más rápido que las instancias con

muchos clientes y pocos ítems por cliente. Esto tiene sentido ya que el *packing* es prácticamente el mismo, pero disminuye la dificultad del ruteo, por lo que la cantidad de variables presentes en el problema disminuye de manera brusca.

Al aumentar el número de ítems en la instancia a analizar, se aprecia también una tendencia al crecimiento exponencial en el número de variables binarias en el problema. El crecimiento en el número de variables, provoca que en las instancias más complejas no sea buena decisión utilizar únicamente el MIP para resolver el problema. Por lo tanto, se justifica el uso de heurísticas o fijación de valores para alcanzar soluciones razonablemente buenas.

Al observar las 2 primeras instancias en "Test01" o la 3° y 4° instancias en "Test00", se observa un aumento en la cantidad de objetos presentes en el vehículo, sin un aumento en el número de manipulaciones. Esto se debe que aumenta el número de objetos pero sus dimensiones cambian, disminuyendo el área de los objetos y manteniendo un alto porcentaje de utilización de la superficie de carga, por lo que se logra realizar un *packing* que no implica realizar manipulaciones adicionales.

Para las 2 primeras instancias "Test00", la cantidad mínima de manipulaciones involucra una manipulación más que el número de productos en el vehículo. Lo que apunta a que estas instancias serían infactibles utilizando "*Sequential Loading*", condición frecuentemente utilizada en gran parte de la literatura existente al respecto.

En caso de utilizar "*Sequential Loading*" y forzar la obtención de una solución para esas instancias, una medida razonable sería la inclusión de un vehículo adicional. Sin embargo, realizar esto implicaría mayores costos de ruteo por desigualdad triangular al utilizar una ruta adicional a pesar de que el vehículo original por si solo era capaz de transportar la totalidad de la carga, y el costo de operar otro vehículo. Esto se puede comparar al observar la columna "diff%" para aquellas instancias cuya solución óptima incluía más de una manipulación por objeto (*i.e.* #Man > M), donde en promedio se obtuvo rutas un

11,79% más cortas que utilizando *Sequential Loading* con un vehículo adicional para evitar la infactibilidad.

El resultado anterior tiene sentido, ya que al plantear un modelo más flexible al determinar el *packing*, se encontrarán rutas no peores que con *Sequential Loading*, ya que se busca sobre un dominio mayor de soluciones factibles. Esta característica se hace evidente en aquellas instancias que hubiesen sido infactibles utilizando *Sequential Loading* (i.e.  $\#Man > M$ ), donde las rutas finales serán siempre peores que las encontradas utilizando el MIP propuesto en esta tesis.

#### **5.4.2 Optimización parcialmente separada**

Con este enfoque, el tamaño de las instancias que fue posible resolver aumentó dramáticamente. Para un máximo de 2100 segundos, se logró resolver instancias en promedio de 36,7 ítems, con un gap promedio menor al 0,17% y 37,6 manipulaciones promedio, es decir, en gran parte de los casos se logra satisfacer el *sequential loading*, sin necesidad de haberlo impuesto directamente como una restricción (ver tabla 5 en el Anexo A).

El modelo propuesto, al ser más flexible en el *packing*, considerando distribuciones espaciales que en la literatura se descartaban al utilizar *Sequential Loading*, ofrece la posibilidad de hallar mejores rutas (ver tabla 3). Sin embargo, como sólo se está realizando el experimento manteniendo fijas parte de las rutas alcanzadas inicialmente por la heurística de Clarke & Wright, la eventual mejora no se ve reflejada en los resultados de las instancias mencionadas en la tabla 5 (Anexo A).

Una forma de alcanzar mejores resultados sin incurrir en un aumento radical del tiempo de cómputo, sería incluir una parte aleatoria en la determinación de las rutas que se fijan inicialmente, o bien, realizar un proceso heurístico (e.g. búsqueda tabú) en las etapas iniciales del algoritmo propuesto, de manera que se busque rutas más exhaustivamente y

todas ellas se ingresen individualmente como solución inicial al MIP propuesto. Finalmente, se escogería la mejor de las soluciones.

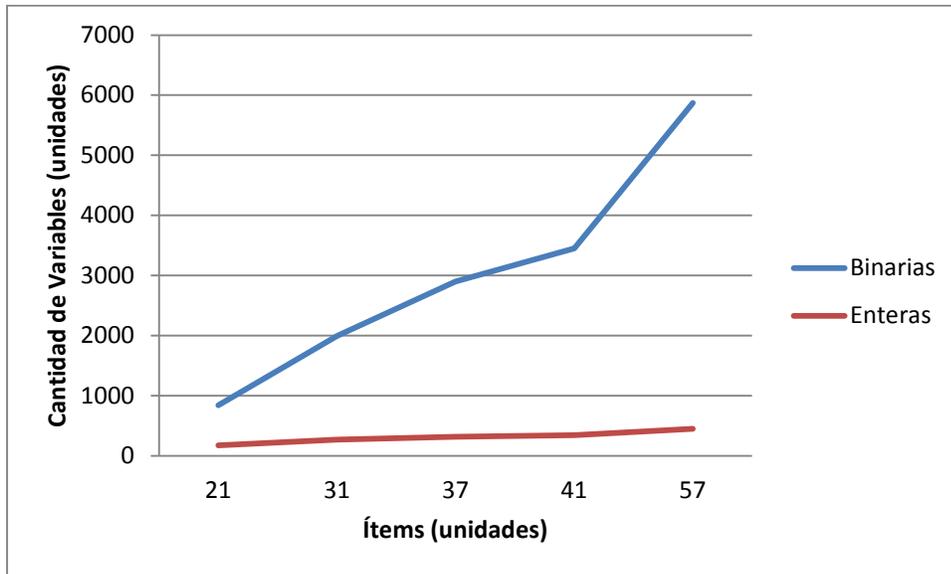
Dentro de los resultados más importantes respecto a este trabajo, se puede mencionar que el enfoque de solución propuesto no es apto para ser aplicado simultáneamente a la restricción de orientación de los ítems, ya que hay altas probabilidades de que las instancias a analizar se vuelvan infactibles (ver tabla 6 en el Anexo B). Al fijar la orientación de los ítems, la cantidad de instancias que pudo ser efectivamente resuelta disminuyó en un 31,8%. La causa de esto es que al fijar las rutas y además fijar la orientación de los ítems, la rigidez de las variables de las que se dispone no permite lograr un *packing* factible. Obligando a bajar el nivel de fijación de rutas a través de modificar el valor del factor de reserva  $\delta$ .

Hay que destacar que la cantidad de variables binarias no es estrictamente decreciente con el factor de reserva  $\delta$ , ya que se puede dar el caso que para un  $\delta$  levemente mayor, se incluya un cliente en una ruta inicial, que de otra manera no se hubiese incluido, impidiendo que otros dos clientes se incluyeran en alguna de las rutas iniciales restantes, aumentando el número de variables "libres" (*i.e.* no fijadas en las rutas iniciales). Por ambos motivos se puede deducir que la curva del número de variables no fijas que llegan al *solver* vs. el factor de reserva  $\delta$  es decreciente, mas no estrictamente decreciente (*e.g.* en la instancia E021-04m, clase 2, con orientación fija de los ítems, el salto se producía al aumentar el  $\delta$  de 0,82 a 0,83; donde el número de variables aumentaba de aproximadamente 6000 variables a cerca de 9000, en lugar de disminuir).

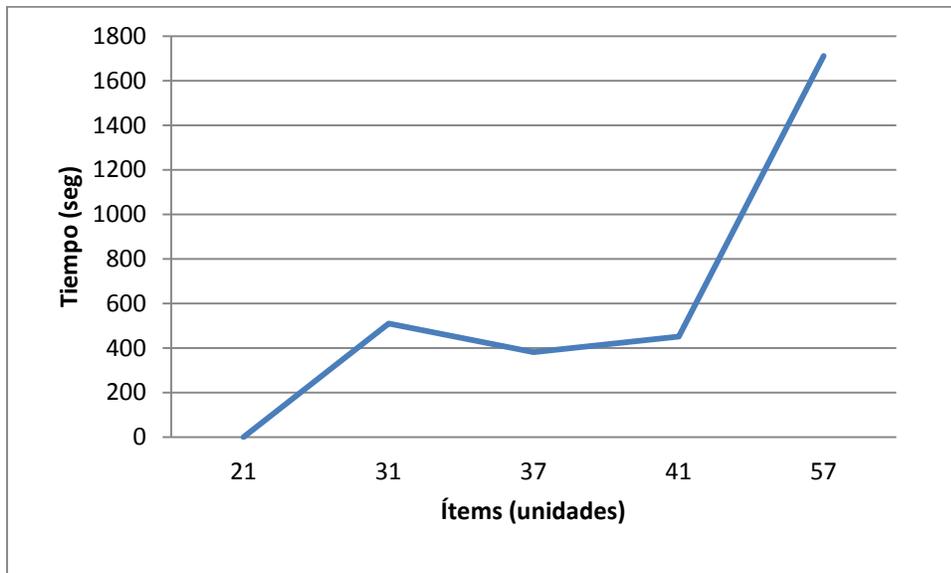
En la tabla 5 (Ver Anexo A), aquellas instancias con gap final distinto de cero y con tiempo total menor que los 1800 segundos impuestos, se deben a que posteriormente se caían debido a la excepción *Out of Memory*, lo que apunta a la incapacidad de manejar la información de los nodos debido a la insuficiencia de memoria virtual. Por ésta razón se decidió abortar el proceso de optimización antes que se lanzara la excepción mencionada.

Por lo general, debido a que fijar las rutas para reducir el número de variables que se entregan al *solver*, implica dejar fuera de análisis gran parte de las soluciones factibles al problema, algunas instancias se vuelven infactibles. Por lo tanto, como fue mencionado anteriormente, se vuelve a resolver el problema con rutas menos fijas, pero con más variables para el *solver*, provocando un brusco aumento en los requisitos de recursos del sistema, por lo que se vuelve muy difícil encontrar soluciones rápidamente. Esto es particularmente cierto para el caso en que se fija la orientación de los productos transportados, prohibiendo su giro dentro de los vehículos. En éste caso es aún más probable que la instancia se vuelva infactible, por lo que se relaja la fijación de rutas del algoritmo de propuesto, provocando que la gran mayoría de estas instancias no pudiera ser resueltas en un tiempo razonable o incluso que el programa no fuera capaz de procesar toda la información asociada a esa enorme cantidad de variables, finalizando por arrojar la excepción "*Out of Memory*".

Las figuras 8 y 9 corresponden a gráficos de una de las instancias más representativas del comportamiento general observado, el número de variables binarias crece de manera mucho más violenta que las enteras. A excepción de la caída en el tiempo de ejecución para 37 ítems, se aprecia un claro crecimiento en el tiempo de ejecución vs. la cantidad de ítems en la instancia.



**Figura 9: Crecimiento del número de variables vs número de ítems.**



**Figura 10: Aumento del tiempo de ejecución vs. cantidad de ítems.**

En todas las instancias resueltas, la heurística de ruteo en conjunto con el algoritmo de *Floor & Ceiling* jamás se tomaron más de 2 minutos en entregar valores iniciales a las variables de ruteo y *packing*.

Clase	Items	# Man.	Tpo 1º sol (seg)	%gap 1º sol	Tpo Total (seg)	%gap Final	Bool	Integer
1	20.14	20.14	0.00	5.86%	0.89	0.00%	2075.4	243.7
2	31.40	32.60	29.33	3.35%	351.88	0.24%	2729.2	307.2
3	41.00	41.75	32.34	6.40%	417.79	0.17%	3228.8	358.0
4	40.67	41.00	45.99	5.83%	197.43	0.00%	2696.0	300.0
5	50.33	52.33	90.58	7.42%	794.52	0.45%	4568.7	372.7

**Tabla 4: Resumen de instancias resueltas relajando restricción de orientación de los ítems.**

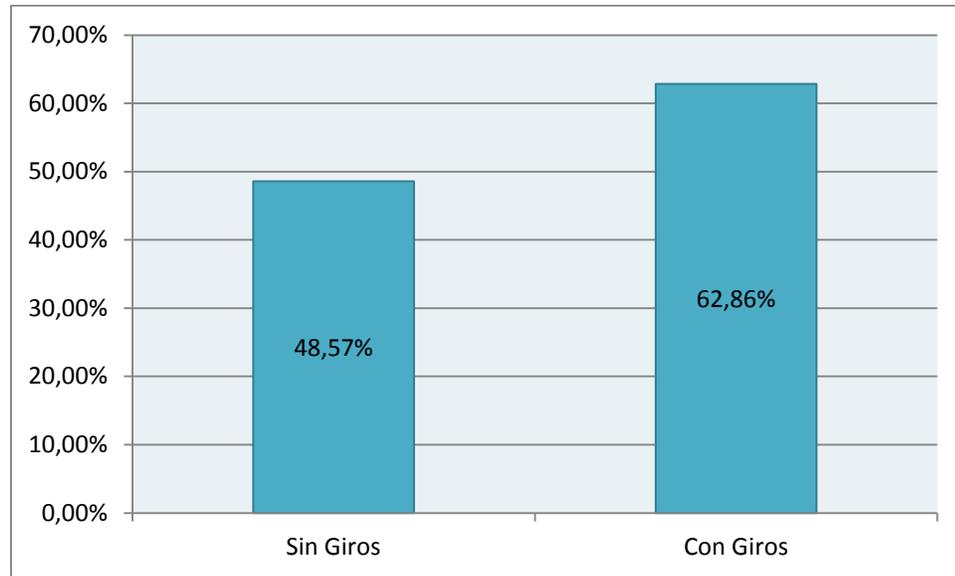
Las dificultades al resolver la instancias E021-06m van de la mano con el hecho de que siempre se utiliza como máximo la cantidad de vehículos que viene en la instancia, a diferencia de lo que se hacía en trabajos anteriores presentes en la literatura (Iori *et al.*, 2005) donde se permitía utilizar un número de vehículos  $K$  igual al máximo entre lo indicado en la instancia original y lo que se obtenía al resolver heurísticamente el 2BPP asociado.

También se puede apreciar que los costos de ruteo determinados por el modelo propuesto son mayores que los presentes en la literatura (Iori *et al.*, 2005 o Gendreau *et al.*, 2005) para el 2LCVRP (Ver Anexo C), esto ocurre debido a que se utilizaron las rutas entregadas por la heurística de Clarke & Wright y una mejora 2-opt, lo que no asegura que se alcancen las rutas óptimas para cada instancia. Además se debe tener en cuenta que hasta el día de hoy, no se había abordado el tema de optimizar la cantidad de manipulaciones, manteniendo la factibilidad de las instancias, a pesar de que la restricción de *Sequential Loading* no lo permitiera.

Para la instancia E016-03m clase 3, CPLEX no logró encontrar una solución inicial factible dentro del límite de tiempo establecido, ni tampoco construir una a base de los valores de ruteo y *packing* que se le entregaron.

Se logró resolver instancias de hasta 21 clientes con 57 ítems y 5 vehículos con gaps cercanos a cero, habiendo fijado previamente las rutas, es decir, el gap no es respecto al óptimo de la instancia, sino al *best bound* del problema con parte de las rutas determinadas sin considerar manipulaciones ni *packing*.

En la figura 11 se muestra el aumento en la cantidad de instancias resueltas al permitir giros, es decir, relajar la restricción de orientación de los ítems. Se debe considerar que los valores observados no son mayores porque el conjunto de instancias presente en la literatura estaba diseñado para el 2LCVRP, que es menos complejo que el problema resuelto en esta tesis (2LCVRP-H). Fue necesario utilizar éste conjunto de instancias (Iori *et al.*, 2005) para poder mostrar la comparación de los resultados encontrados (ver Anexo A, B y C).



**Figura 11: Aumento de instancias resueltas en menos de 2100 segundos al permitir giros y al no permitirlos.**

Dentro de las soluciones comparables entre las instancias a las que se les aplicó la restricción de orientación de los ítems vs. las que no, se logra apreciar una mejora inferior al 1% en el costo total, lo que no está muy lejos de lo comentado en la literatura por Fuellerer *et al.* (2009), quienes sostienen que la solución mejora un 3,38% al relajar la restricción de orientación de los ítems.

## 6. CONCLUSIONES, RECOMENDACIONES Y EXTENSIONES

Se ha desarrollado un modelo que permite determinar una forma (óptima con certeza para instancias pequeñas) de cargar un conjunto de vehículos con objetos heterogéneos en términos de volumen y densidad, y simultáneamente planificar rutas a seguir para satisfacer la demanda de un conjunto dado de clientes. El modelo desarrollado permite resolver instancias actualmente infactibles para los trabajos realizados en la literatura, ya que estos últimos consideran metodologías ineficientes o inapropiadas para una optimización a nivel de manipulación de carga in situ. Ya que, para el caso de carga secuencial, se obliga a que cada vez que se visite un cliente, todos los objetos a ser entregados puedan descargarse simplemente con movimientos paralelos al eje Y del plano de carga, o bien, para el caso de carga irrestricta, no se asegura la minimización de la manipulación de carga durante el trayecto de cada ruta.

Este trabajo corresponde a un avance en lo que concierne a los problemas de ruteo que consideran costos de manipulación. Siendo problema del vendedor viajero con cargas, descargas y manipulación de carga unidimensional (Battarra *et al.*, (2010), uno de los trabajos más avanzados en esta área, previo a la presente tesis.

En el presente trabajo se incorporó el uso en etapas de distintos procesos heurísticos, determinando valores iniciales que posteriormente son ingresados como *input* al MIP desarrollado, lo que permitió resolver en un tiempo razonable instancias mucho mayores.

Dentro de los resultados se aprecia que el enfoque de solución propuesto es capaz de resolver en menos de 30 minutos, instancias con un total de hasta 22 clientes, 57 ítems y 5 vehículos. Cantidad no menor si se considera la complejidad intrínseca del problema abordado (2LCVRP-H), ya que se busca la resolución conjunta de dos problemas que por sí solos son NP-*Hard*, el CVRP y el 2BPP (Iori M, Martello S, 2010).

Dada la estructura que posee el modelo, en que analiza la configuración y el espacio y capacidad de peso en cada parada, el trabajo desarrollado puede ser fácilmente adaptado a problemas con *pick up* y *delivery*, o bien, con ventanas de tiempo.

El algoritmo propuesto podría ser mejorado en futuros desarrollos incluyendo un proceso de búsqueda tabú en la determinación de rutas iniciales, de esta manera se procedería a optimizar el *packing* asociado a distintas rutas, para escoger finalmente la de mejor valor óptimo.

Otro posibles desarrollo sería incorporar procesos más sofisticados para determinar soluciones iniciales aún más cercanas al óptimo. Esto probablemente se logre a través del uso de metaheurísticas de última generación, con las que podría incluso optimizarse simultáneamente el ruteo y el *packing* en tiempo razonable.

## BIBLIOGRAFÍA

Battarra M, Erdogan G, Laporte G, Vigo D (2010) The travelling salesman problem with pickups, deliveries and handling costs. *Transp Sci.* doi:10.1287/trsc.1100.0316

Clarke G, Wright JV (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Journal of the operational research society* 12:568–581

Coffman, E.G. Jr, Garey, M.R., Johnson, D.S., Tarjan, R.E. (1980). Performance bounds for level-oriented two-dimensional packing algorithms. *SIAM Journal on Computing*, 9:808--826.

Dantzig, G., Ramster, J., (1959). The truck dispatching problem. *Management Science*, Vol. 6, No. 1 (Oct., 1959), pp. 80-91

Eley, M. (2002). Solving container loading problems by block arrangement. *European Journal of Operational Research*, 141(2), 393-409.

Faulin, J., Juan, A., Jorba, J., Riera, D., Masip, D., Barrios., B. (2010). On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics. *Journal of the Operational Research Society* (2011) 62, 1085 –1097.

Fisher, M.L. (1994). Optimal solution of vehicle routing problems using minimum k-trees, *Oper. Res.* 42, 626–642.

Fuellerer, G., Doerner, K., Hartl, R., Iori, M. (2009). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3), 751-759.

Gendreau, M., Iori, M., Laporte, G., Martello, S. (2005). A Tabu Search Algorithm for a Routing and Container Loading Problem. *Transportation Science*, 40(3), 342-350.

Gendreau, M., Iori, M., Laporte, G. y Martello, S. (2008) “A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints”, *Networks*, 2008, 51(1): pp. 4-18.

Golden, B., (2008). The vehicle routing problem latest advances and new challenges. *Springer*. DOI: 10.1007/978-0-387-77778-8

Iori M, Martello S (2010) Routing problems with loading constraints. Sociedad de Estadística e Investigación Operativa. doi: 10.1007/s11750-010-0144-x

Iori, M., Salazar, J., Vigo, D. (2005). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2), 253-264.

Jacobs, C., Goetschalckx, M., (1998). Vehicle Routing Problem with Backhauls. Technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia.

Lei, H., Laporte, G., Guo, B., (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research* 38 (2011) 1775–1783.

Leung, S., Zhou, X., Zhang, D., Zheng, J. (2010). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers & Operations Research, In Press, Corrected Proof*.

Lodi, A., Martello, S., Vigo, D. (2002). Recent advances on two dimensional bin packing problems. *Discrete Applied Mathematics* 123 (2002) 379 – 396.

Malapert, A., Gueret, C., Jussien, N., Langevin, A., Rousseau, L. (2008). Two-dimensional Pickup and Delivery Routing Problem with Loading Constraints. *First CPAIOR Workshop on Bin Packing and Placement Constraints (BPPC'08)*.

Ribeiro, G., Laporte, G., (2011). An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *ComputersandOperationsResearch*(2011),doi:10.1016/j.cor.2011.05.005

Tarantilis, C., Zachariadis, E., Kiranoudis, C. (2009). A Hybrid Metaheuristic Algorithm for the Integrated Vehicle Routing and Three-Dimensional Container-Loading Problem. *IEEE Transactions on Intelligent Transportation Systems*, 10(2), 255-271.

Toth, P., Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123, 487-512.

Wang, F., Tao, Y., Shi, N. (2009). A Survey on Vehicle Routing Problem with Loading Constraints. *International Joint Conference on Computational Sciences and Optimization*. 602-606.

Zachariadis, E., Tarantilis, C., Kiranoudis, C. (2007). A Guided Tabu Search for the Vehicle Routing Problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3), 729-743.

## **ANEXOS**

**Anexo A: RESULTADOS OPTIMIZACIÓN PARCIALMENTE SEPARADA  
SIN ORIENTACIÓN FIJA DE LOS ÍTEMS (PERMITIENDO  
GIROS).**

Instancia	Clase	n	M	K	z	Manip.	$\Delta$	$\Delta\%$	Tpo 1° sol (seg)	%gap 1° sol	Tpo Total (seg)	%gap Final	Binarias	Generales
E016-03m	1	15	15	3	313.56	15	25.56	9.36%	0	0.00%	0.09	0.00%	949	162
	2	15	24	3	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	15	31	3	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	15	37	4	335.06	37	10.06	3.49%	29.06	3.73%	73.3	0.00%	2107	250
	5	15	45	4	332.38	45	8.38	3.00%	61.2	5.94%	124.27	0.00%	3669	302
E021-04m	1	20	20	4	411.55	20	40.55	11.55%	0	15.59%	3.91	0.00%	2266	294
	2	20	29	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	20	46	5	453.08	49	17.08	4.41%	34.72	7.79%	333.61	0.66%	2914	336
	4	20	44	5	437.09	44	19.09	5.10%	0	6.62%	68.06	0.00%	2531	306
	5	20	49	5	446.72	55	22.72	6.16%	172.86	7.16%	548.6	1.34%	4168	366
E021-06m	1	20	20	6	607.41	20	164.41	38.87%	0	0.33%	1.11	0.00%	2443	276
	2	20	32	6	623.32	34	155.32	35.79%	57.72	2.81%	473.91	0.63%	6244	470
	3	20	43	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	20	50	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	5	20	62	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
E022-04g	1	21	21	4	409.78	21	21.78	5.93%	0	0.00%	0.06	0.00%	841	170
	2	21	31	4	419.78	31	8.78	2.31%	0	5.83%	510.1	0.00%	1996	268
	3	21	37	4	447.86	37	37.86	10.15%	47.95	7.63%	380.5	0.00%	2901	316
	4	21	41	4	429.78	42	10.78	2.86%	108.91	7.13%	450.92	0.00%	3450	344
	5	21	57	5	445.78	57	-0.22	-0.06%	37.67	9.17%	1710.69	0.00%	5869	450
E022-06m	1	21	21	6	578.27	21	69.27	14.19%	0	21.77%	0.05	0.00%	1243	212
	2	21	33	6	592.27	35	66.27	13.50%	21.23	1.83%	364	0.34%	2418	336
	3	21	40	6	597.27	40	61.27	12.35%	11.75	3.40%	649.72	0.00%	3778	432
	4	21	57	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	5	21	56	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
E023-03g	1	22	22	3	590.55	22	10.55	1.89%	0	2.32%	0.58	0.00%	3393	296
	2	22	32	5	802.15	34	44.15	6.10%	67.7	2.79%	369.74	0.25%	1579	240
	3	22	41	5	768.71	41	29.71	4.26%	34.95	6.79%	307.31	0.00%	3322	348
	4	22	51	5	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	5	22	55	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
E023-05s	1	22	22	5	590.55	22	-88.45	-13.46%	0	1.01%	0.44	0.00%	3393	296
	2	22	29	5	779.58	29	30.58	4.25%	0	3.47%	41.67	0.00%	1409	222
	3	22	42	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	22	48	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	5	22	52	5	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM

**Tabla 5: Resultados relajando restricción de orientación de los ítems**

En esta tabla, cada columna corresponde a los valores descritos para la tabla 3, en el capítulo 5, que contiene los resultados para la optimización conjunta.

$\Delta$  y  $\Delta\%$  corresponden respectivamente a la diferencia (y diferencia porcentual) en el costo de ruteo del valor objetivo alcanzado menos el alcanzado en la tabla 7 por Iori *et al.* Esta diferencia se debe a que en la heurística propuesta en la presente tesis se fija parte de las rutas en su etapa inicial, por lo tanto, los resultados finales establecen un *packing* aceptable, y óptimo en la mayor parte de los casos, pero sólo para la ruta final encontrada

y no para el MIP descrito en el capítulo 3. Por esta razón, por lo general se observa que las soluciones encontradas son peores a las existentes en la literatura.

Las filas con la expresión "OOM" se refieren a "*Out of Memory exception*", que corresponde al error que lanza C# cuando no logra construir el problema en CPLEX debido a la falta de memoria virtual. Este error era recurrente en instancias complejas o con gran número de variables, a pesar de que, para ahorrar memoria, se guardaban fragmentos de información de los nodos para hacer *backtracking* en el proceso de optimización.

En promedio, la solución alcanzada en esta tesis relajando la restricción de orientación de los ítems es un 8,27% superior a la mejor solución lograda en la literatura (Iori *et al.*, 2005 y Gendreau *et al.*, 2005).

**Anexo B: RESULTADOS OPTIMIZACIÓN PARCIALMENTE SEPARADA  
CON ORIENTACIÓN FIJA DE LOS ÍTEMS (SIN PERMITIR  
GIROS).**

Instancia	Clase	n	M	K	z	Manipulaciones	Tpo 1° sol (seg)	%gap 1° sol	Tpo Total (seg)	%gap Final	Binarias	Generales
E016-03m	1	15	15	3	313.56	15	~ 0	0.00%	11	0.00%	834	162
	2	15	24	3	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	15	31	3	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	15	37	4	338.06	40	58.22	5.10%	2099.12	0.89%	1917	250
	5	15	45	4	332.38	45	34.72	8.53%	489.09	0.00%	3508	302
E021-04m	1	20	20	4	411.55	20	~ 0	13.33%	1.63	0.00%	2045	294
	2	20	29	5	553.38	32	54.16	28.51%	2131.98	19.76%	5807	412
	3	20	46	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	20	44	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	5	20	49	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
E021-06m	1	20	20	6	607.41	20	18.55	45.65%	102.28	0.00%	3694	356
	2	20	32	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	20	43	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	20	50	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	5	20	62	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
E022-04g	1	21	21	4	409.78	21	~ 0	0.00%	0.06	0.00%	777	170
	2	21	31	4	419.55	32	~ 0	8.72%	2099.11	0.24%	1642	262
	3	21	37	4	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	21	41	4	430.78	42	191.17	11.35%	2098.42	0.23%	3244	344
	5	21	57	5	445.78	57	29.61	8.23%	153.36	0.00%	5653	450
E022-06m	1	21	21	6	578.27	21	~ 0	21.77%	0.09	0.00%	1023	212
	2	21	33	6	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	21	40	6	598.27	41	7.92	1.65%	2097.95	0.17%	3428	432
	4	21	57	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	5	21	56	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
E023-03g	1	22	22	3	590.55	22	~ 0	0.51%	0.50	0.00%	3267	296
	2	22	32	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	3	22	41	5	769.71	42	34.47	5.76%	721.14	0.13%	3092	348
	4	22	51	5	802.34	75	338.23	5.69%	724.84	2.99%	4053	390
	5	22	55	6	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
E023-05s	1	22	22	5	590.55	22	~ 0	4.37%	0.89	0.00%	3267	296
	2	22	29	5	782.58	32	~ 0	2.01%	102.33	0.25%	1212	219
	3	22	42	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	4	22	48	5	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim	Time Lim
	5	22	52	5	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM

**Tabla 6: Resultados considerando activa la restricción de orientación de los ítems.**

Se destaca que para las instancias resueltas, los resultados alcanzados corresponden exactamente a los valores objetivo encontrados sin considerar activa la restricción de orientación de los ítems (ver tabla 5).

### Anexo C: RESULTADOS 2LCVRP PRESENTES EN LA LITERATURA.

Name	Class	n	M	K	Loading				Routing			Overall		
					Pool	$N_{BB}$	Fails	$T_{load}$	Cuts	$\%_{gap}$	$T_{rout}$	z	$T_{heur}$	$T_{tot}$
E016 – 03m	1	15	15	3	25	9	0	0.02	209	0.00	0.91	273	0.75	0.92
	2	15	24	3	115	69	0	7.05	885	0.00	17.44	285	22.19	24.49
	3	15	31	3	73	29	0	16.59	641	0.00	5.46	280	21.94	22.05
	4	15	37	4	29	9	0	2.64	28	0.00	0.05	288	2.28	2.69
	5	15	45	4	16	5	0	39.00	4	0.00	0.00	279	39.00	39.00
E021 – 04m	1	20	20	4	42	18	0	0.00	352	0.00	4.75	351	4.72	4.75
	2	20	29	5	937	418	0	5.41	1748	0.00	67.23	389	63.53	72.64
	3	20	46	5	49	37	0	3.64	144	0.00	1.41	387	4.97	5.05
	4	20	44	5	83	37	0	59.06	86	0.00	0.30	374	57.74	59.36
	5	20	49	5	26	10	0	0.20	46	0.00	0.05	369	0.23	0.25
E021 – 06m	1	20	20	6	35	9	0	0.00	93	0.00	0.19	423	0.17	0.19
	2	20	32	6	31	7	0	0.00	238	0.00	2.08	434	1.56	2.08
	3	20	43	6	70	14	0	3.38	432	0.00	2.78	432	4.83	6.16
	4	20	50	6	44	6	0	0.77	368	0.00	5.31	438	4.92	6.08
	5	20	62	6	36	9	0	45.95	154	0.00	0.36	423	23.45	46.31
E022 – 04g	1	21	21	4	20	8	0	0.00	41	0.00	0.05	367	0.05	0.05
	2	21	31	4	50	27	0	1.19	126	0.00	1.16	380	1.47	2.34
	3	21	37	4	43	10	0	1.86	52	0.00	0.09	373	1.88	1.95
	4	21	41	4	109	31	0	32.17	383	0.00	3.14	377	35.23	35.31
	5	21	57	5	43	18	0	1865.52	39	0.00	0.14	389	1865.55	1865.66
E022 – 06m	1	21	21	6	45	18	0	0.00	519	0.00	5.73	488	5.39	5.73
	2	21	33	6	75	20	0	0.78	1010	0.00	25.73	491	26.13	26.52
	3	21	40	6	76	39	0	2.50	1081	0.00	32.95	496	27.61	35.45
	4	21	57	6	83	39	0	3.91	491	0.00	3.28	489	7.00	7.19
	5	21	56	6	54	28	0	0.14	509	0.00	4.53	488	3.86	4.67
E023 – 03g	1	22	22	3	3	3	0	0.00	30	0.00	0.02	558	0.02	0.02
	2	22	32	5	317	209	0	10.10	801	0.00	13.09	724	21.89	23.19
	3	22	41	5	39	19	0	3.09	47	0.00	0.16	698	2.31	3.25
	4	22	51	5	90	46	0	1666.85	238	0.00	1.20	714	1667.58	1668.05
	5	22	55	6	64	37	0	782.36	20	0.00	0.06	742	771.78	782.42
E023 – 05s	1	22	22	5	22	10	0	0.00	6	0.00	0.02	657	0.02	0.02
	2	22	29	5	720	486	0	3.58	980	0.00	30.85	720	32.97	34.42
	3	22	42	5	252	84	0	67.60	357	0.00	1.45	730	68.94	69.05
	4	22	48	5	29	10	0	17.08	37	0.00	0.08	701	9.59	17.16
	5	22	52	6	13	6	0	956.05	8	0.00	0.00	721	956.05	956.05

**Tabla 7: Resultados en la literatura sin considerar manipulaciones.**

Fuente: Iori et al. (2005).

Los resultados presentes en la tabla 6 son idénticos, en cuanto a valor óptimo se refiere, a los encontrados por Gendreau *et al.* (2005). Pero ninguno de ellos consideró optimizar la manipulación de objetos para todo tipo de instancias.

## Anexo D: EJEMPLO DE MANIPULACIÓN EN CADENA.

Dada la configuración que muestra la figura 12, donde se desea extraer el objeto  $(c,i)$  del vehículo, el objeto  $(c',i')$  no obstruye directamente el desplazamiento longitudinal del objeto  $(c,i)$ , sin embargo, el objeto  $(c^*,i^*)$  sí lo hace, por lo tanto, es necesario extraerlo del vehículo para permitir la salida del objeto  $(c,i)$ .

Luego, se observa que el desplazamiento longitudinal del objeto  $(c^*,i^*)$  sí es obstruida por  $(c',i')$ , por lo que es necesario sacarlo del vehículo para permitir el desplazamiento de los dos objetos anteriores.

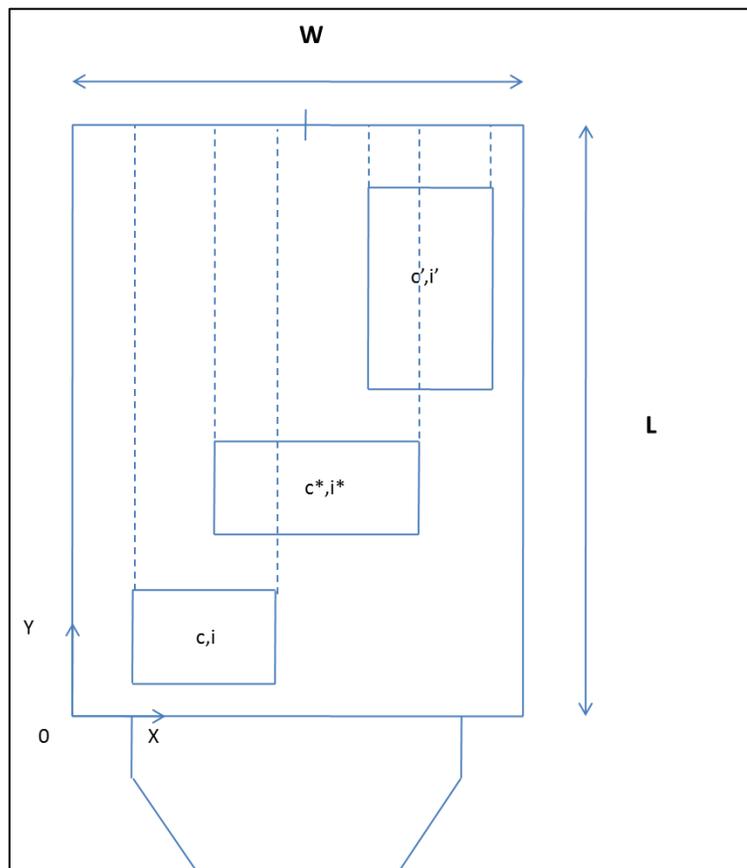


Figura 12: manipulación en cadena provocada por la salida del objeto  $i$  del cliente  $c$ .