



Tesis Doctoral

COOPERACIÓN HORIZONTAL EN PROBLEMAS DE RUTAS DE
VEHÍCULOS CON RETORNOS

Autor: Javier Beloso Ezcurra



Directores: Dr. Javier Faulín y Dr. Ángel A. Juan

Programa Doctoral: Matemáticas y Estadística

Pamplona/Iruña 2017

Javier Belloso Ezcurra. Departamento de Estadística e Investigación Operativa. Universidad Pública de Navarra. Campus de Arrosadía, s/n, 31006 Pamplona, Navarra. Correo electrónico: javier.belloso@unavarra.es

A mis ángeles, Javier, María e Íñigo.

Agradecimientos

En primer lugar, quiero expresar mi más sincero agradecimiento y admiración a los profesores Dr. D. Javier Faulín Fajardo y Dr. D. Ángel Alejandro Juan Pérez, codirectores de esta tesis, por haberme puesto en el camino, por su dirección, su apoyo constante y por introducirme en este mundo apasionante de la investigación. Su ayuda, me ha permitido compaginar este trabajo con el resto de responsabilidades, y superar los “subes y bajas” que tiene todo proceso como este.

A los departamentos de Estadística e Investigación Operativa y el de Ingeniería Matemática e Informática por darme la oportunidad de realizar esta tesis. En particular a Fermín, Cristina, Lola y José Luis.

A mi mujer Idoia y mis tres hijos Javier, María e Iñigo. Hacer este trabajo (y sobretodo acabarlo) nos va a venir muy bien a todos.

A mi madre y mi padre (q.e.d.) de los que sobra decir que les debo todo lo que soy, pero en especial les agradezco que me hayan sabido transmitir su sabiduría y positivismo. A mi hermana Natacha, por su afecto, bondad y generosidad sin límites

A Jose, mi jefe en lo profesional, pero sobretodo guía y amigo. He aprendido muchas cosas en el mundo de la automoción que se han reflejado en este trabajo.

No puedo olvidarme de Viena, Richard, y el Verolog 2015 que me inspiró y me permitió buscar soluciones a problemas que creía imposibles. Aproveché el impulso de una vuelta de vals para subirme a la ola que me ha traído hasta aquí. Aunque Philly y Nueva York también tuvieron algo que ver.

Como el final de esta ruta ha sido ha sido el buscado, incluso los malos momentos se recuerdan con cariño. Este ha sido un camino intenso de vida que abre muchos otros para el futuro. Para concluir voy a utilizar lo que Maurice Herzog predijo después de ser el primer humano con su compañero en ascender un “ochomil”: “Una vida acaba y otra comienza, porque siempre habrá otros Annapurna en la vida de los hombres”.

Resumen

Belloso, J. 2017. Cooperación horizontal en problemas de rutas de vehículos con retornos. Tesis doctoral. Departamento de Estadística e Investigación Operativa. Universidad Pública de Navarra.

En el mundo competitivo y global en que se desarrollan las actividades de las empresas, las prácticas de cooperación entre entidades distintas suponen una gran oportunidad para aumentar la eficiencia en las operaciones. Cuando la cooperación se desarrolla entre participantes que actúan al mismo nivel dentro de la cadena de suministro es llamada cooperación horizontal y para garantizar su validez, debe ser analizada en términos de costes y eficiencia.

El propósito principal de esta tesis es conseguir las herramientas necesarias para poder evaluar el impacto y los beneficios potenciales de la puesta en marcha de actividades de cooperación horizontal entre empresas del ámbito de la logística y el transporte cuando realizan labores de distribución y recogida dentro de sus cadenas de reparto y abastecimiento respectivamente.

En primer lugar, se hace una revisión de la literatura de las prácticas reales de cooperación horizontal en actividades de logística y transporte. Partiendo de procesos de negocio, se realiza su modelado en forma de problemas de optimización combinatoria. En particular se estudia su relación con el conocido problema de las rutas de vehículos (*Vehicle Routing Problem* o VRP). En este problema, se considera la optimización de la distribución de materiales entre clientes y depósito central con la creación de modelos de optimización combinatoria. Como resultado de este análisis, se obtienen tres modelos para analizar. Las rutas de vehículos con retornos, las rutas de vehículos con recogidas y entregas, y la consideración de la flota heterogénea.

A continuación, con los modelos seleccionados se hace una revisión de la literatura existente como paso anterior a la creación de metodologías que los solucionen. Los tres problemas tratados son considerados como NP-complejos, por tanto, se plantea el diseño de meta-heurísticos que aborden estos problemas con las características más reales posibles. Se diseñan distintas metodologías para cada modelo basadas en la aleatoriedad sesgada para solucionar estos problemas de la forma más eficiente, simple y flexible posible.

Finalmente, se contrastan los resultados obtenidos por la metodología propuesta, en comparación con las instancias utilizadas por otros algoritmos de la literatura que solucionan los mismos problemas. Se realizan experimentos exhaustivos y los resultados se comparan utilizando herramientas estadísticas. La conclusión es que la metodología propuesta consigue muy buenos resultados ya que mejora las mejores soluciones conocidas de 21 instancias reportadas en la literatura.

En resumen, esta tesis doctoral muestra que se han conseguido resultados de alta calidad utilizando algoritmos sencillos basados en la aleatoriedad sesgada que están basados en heurísticos clásicos. Estos algoritmos obtienen resultados competitivos en coste y tiempo y, por tanto, se pueden utilizar para evaluar las economías de escala alcanzadas por las empresas en sus prácticas de cooperación horizontal.

Palabras claves: *Problemas de Rutas de Vehículos con Retornos, Heurísticos aleatorios sesgados, Metaheurísticas, Optimización, Logística, Problemas combinatorios, Cooperación Horizontal.*

Abstract

Belloso, J. 2017. Horizontal Cooperation in Vehicle Routing Problems with Backhauls. PhD Thesis. Statistics and Operational Research Department. Public University of Navarra.

In the competitive and global world in which business activities take place, business-to-business cooperation practices present a challenge to increase efficiency in activities at the same level within the supply chain. This cooperation is called horizontal cooperation and must be analyzed in terms of costs and efficiency inside the vehicle routing problems with backhauls/pickup and delivery constraints/ VRP with heterogeneous fleet.

The main objective of this work is to obtain the necessary tools to be able to evaluate the potential impact and benefits of the implementation of horizontal cooperation activities between companies in the field of logistics and transport.

First, a review of the literature on available practices of horizontal cooperation in logistics and transport activities is carried out. Starting from real processes, its modeling is done in the form of combinatorial optimization problems. In particular, their relationship with the known problem of VRP vehicle routes is studied. In this problem, the optimization of the shipments between clients and a central warehouse is considered. As a result of this analysis, three models are obtained for analysis. Vehicle Routing Problem with Clustered Backhauls (VRPCB), Vehicle Routing Problem with Mixed Backhauls (VRPMB) and Heterogeneous fleet Vehicle Routing Problems with Backhaul (HVRPB).

Then, with the selected models, a review of the existing literature is done as a step prior to the creation of methodologies that solve them. The three problems treated are considered as NP-hard, therefore, it is proposed the design of metaheuristics that address these problems with the most real features possible. Different methodologies for each model are designed based on biased randomization techniques to solve these problems in the most efficient, simple and flexible way possible.

Finally, for some well-known benchmarks, we contrast the results obtained by the proposed methodology with those provided by other algorithms in the literature. Comprehensive experiments are performed and the results are compared using statistical tools. The conclusion is that the proposed methodology achieves is quite competitive since it improves the results of 21 instances reported in the literature.

In summary, the outcomes of this doctoral thesis show that high quality results have been obtained using simple algorithms that rely upon biased randomization of classical heuristics. These algorithms obtain competitive results in cost and time and, therefore, can be used to evaluate the economies of scale reached by companies in their cooperation activities.

Keywords: *Vehicle Routing Problem with Backhauls; Biased Randomization of Heuristics; Metaheuristics. Optimization, Logistics, Combinatorial Optimization, Horizontal Cooperation.*

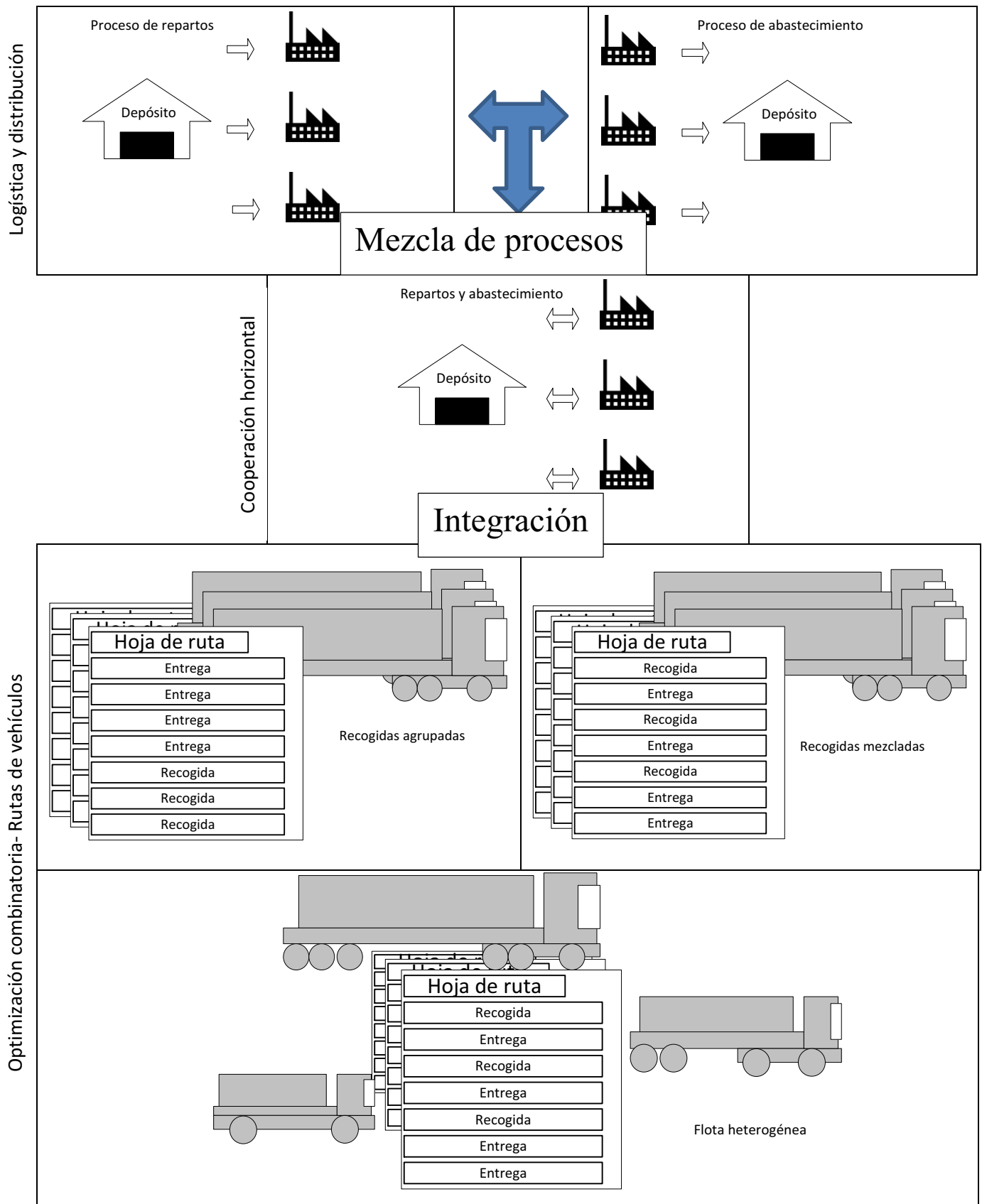


Figura 0.1: Resumen de los temas principales de esta tesis.

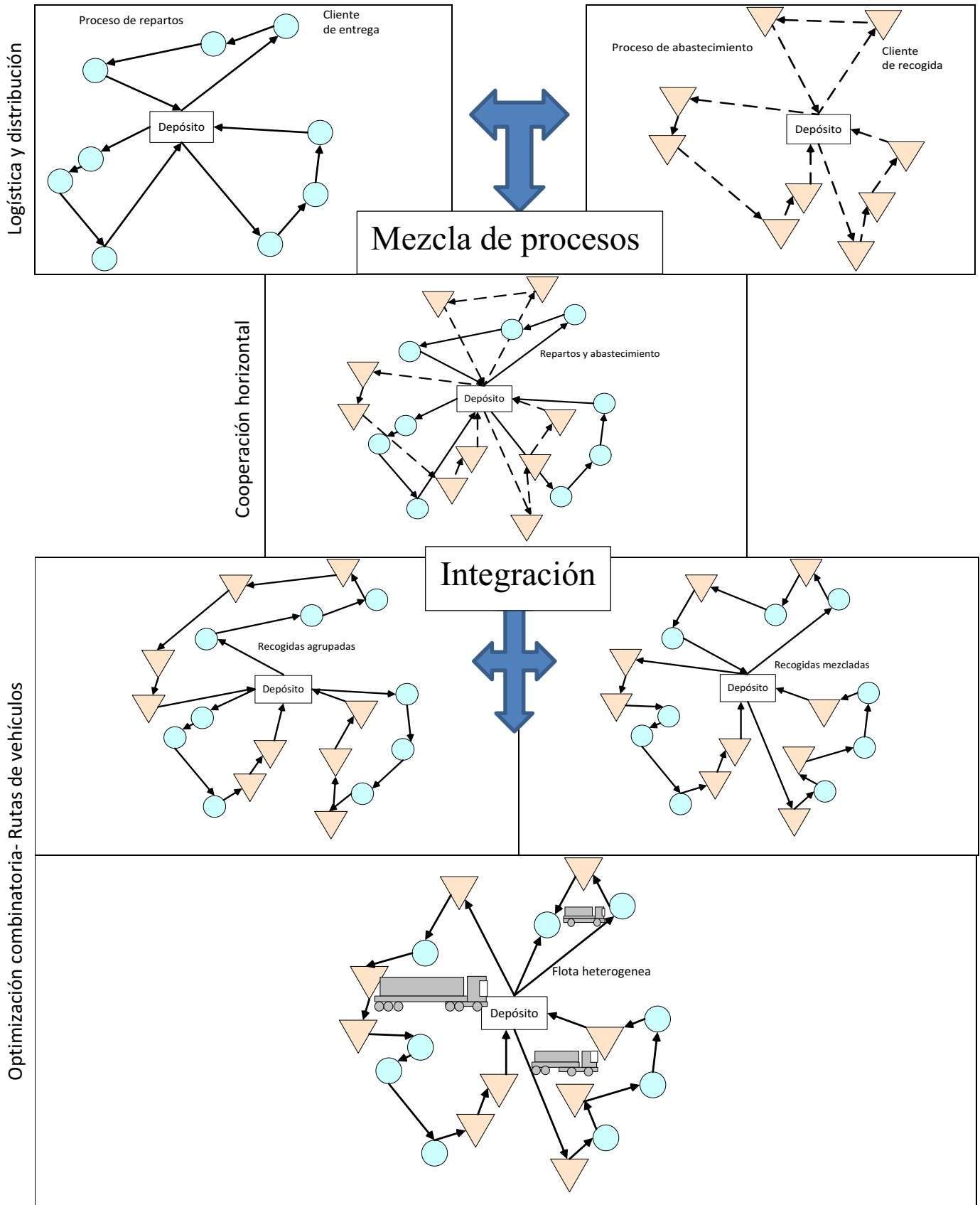


Figura 0.2: Resumen de los temas principales de esta tesis en forma de COPs.

Acrónimos

Acrónimo	Descripción <i>Inglés/Castellano</i>
<i>APGP</i>	<i>Automatic Playlist Generation Problem</i> Generación de listas de reproducción
<i>B&B</i>	<i>Branch and Bound</i> Ramificación y acotación
<i>B&C</i>	<i>Branch and Cut</i> Ramificación y corte
<i>B&P</i>	<i>Branch and Price</i> Ramificación y precio
<i>BILP</i>	<i>Binary Integer Linear Optimization Problem</i> Programación entera binaria
<i>BKS</i>	<i>Best Known Solution</i> Mejor solución conocida
<i>CVRP</i>	<i>Capacitated Vehicle Routing Problem</i> Problema de rutas de vehículos con capacidad limitada
<i>CMI</i>	<i>Clay Mathematics Institute</i> Instituto Clay de matemáticas
<i>COP</i>	<i>Combinatorial Optimization Problem</i> Problema de optimización combinatoria
<i>Dc&C</i>	<i>Decrease and Conquer</i> Decrece y Vencerás
<i>D&C</i>	<i>Divide and Conquer</i> Divide y Vencerás
<i>DARP</i>	<i>Dial a Ride Problem</i> Problema de trazar un camino
<i>DP</i>	<i>Dynamic Programming</i> Programación dinámica

Tabla 0.1: Resumen de acrónimos utilizados en la tesis.

Acrónimo	Descripción <i>Inglés/Castellano</i>
<i>DSS</i>	<i>Decision Support System</i> Sistema de apoyo a decisiones
<i>FSMVRPB</i>	<i>Fleet Size and Mix Vehicle Routing Problem with Backhauls</i> Problema del tamaño de la flota y composición de rutas de vehículos con retornos
<i>GRASP</i>	<i>Greedy Randomized Adaptive Search Procedure</i> Procedimiento de búsqueda adaptativa aleatoria voraz
<i>HC</i>	<i>Horizontal Cooperation</i>
CH	Cooperación Horizontal
<i>HBSS</i>	<i>Heuristic-Biased Stochastic Sampling</i> Heurístico de casos de aleatoriedad sesgada
<i>HVRPB</i>	<i>Heterogeneous fleet Vehicle Routing Problem with Backhauls</i> Problema de rutas de vehículos con retornos y flota heterogénea
<i>IHPO</i>	<i>Iterative Homogeneous Problems Operator</i> Operador generador de problemas homogéneos
<i>ILP</i>	<i>Integer Linear Optimization Problem</i> Problema de programación lineal entera
<i>ILS</i>	<i>Iterated Local Search</i> Búsqueda local iterada
<i>IP</i>	<i>Integer Programming</i> Programación entera
<i>LNS</i>	<i>Large Neighborhood Search</i> Búsqueda por vecindarios amplios
<i>LOP</i>	<i>Linear Optimization Problem</i> Problema de optimización lineal
<i>LP</i>	<i>Linear Programming</i> Programación lineal
LTL	<i>Less than Truck Load</i> Camión de carga fraccionada

Tabla 0.2: Resumen de acrónimos utilizados en la tesis (continuación).

Acrónimo	Descripción <i>Inglés/Castellano</i>
<i>MCS</i>	<i>Monte Carlo Simulation</i> Simulación de monte carlo
<i>MDHVRPB</i>	<i>Multi-Depot Vehicle Routing Problem with Backhauls</i> Problema de rutas de vehículos con retornos y depósitos múltiples
<i>MILP</i>	<i>Mixed Integer Linear optimization Problem</i> Problema de optimización lineal entera mixta
<i>MIRHA</i>	<i>Multi-start biased Randomization of classical Heuristics with Adaptive local search framework</i> Aleatorización de heurísticos con búsqueda local adaptativa
<i>MS</i>	<i>Multi-start</i> Multi-arranque
<i>NP</i>	<i>Nondeterministic Polynomial</i> Polinomial no determinista
<i>NPC</i>	<i>NP-Complete</i> NP-completo
<i>NPH</i>	<i>NP-Hard</i> NP-complejo
<i>OBS</i>	<i>Our Best Solution</i> Nuestra mejor solución
<i>OR</i>	<i>Operations Research</i>
<i>IO</i>	Investigación operativa
<i>P</i>	<i>Polynomial</i> Polinomial
<i>P&DP</i>	<i>General Pick and Delivery Vehicle Routing Problem</i> Problemas generales de recogida y entrega
<i>RCO</i>	<i>Recursive Corrective Operator</i> Operador corrector recursivo
<i>SA</i>	<i>Simulated Annealing</i> Cristalización simulada

Tabla 0.3: Resumen de acrónimos utilizados en la tesis (continuación).

Acrónimo	Descripción <i>Inglés/Castellano</i>
<i>SAM</i>	<i>Successive Approximation Method</i> Método de aproximaciones sucesivas
<i>SCOP</i>	<i>Stochastic Combinatorial Optimization Problem</i> Problema estocástico de optimización combinatoria
<i>TSP</i>	<i>Travelling Salesman Problem</i> Problema del agente viajero
<i>VMI</i>	<i>Vendor Managed Inventory</i> Inventario gestionado por el proveedor
<i>VNS</i>	<i>Variable Neighborhood Search</i> Búsqueda por vecindarios variables
<i>VRP</i>	<i>Vehicle Routing Problem</i> Problema de rutas de vehículos
<i>VRPB</i>	<i>Vehicle Routing Problem with Backhauls</i> Problema de rutas de vehículos con retornos
<i>VRPCB</i>	<i>Vehicle Routing Problem with Clustered Backhauls</i> Problemas de rutas de vehículos con retornos agrupados
<i>VRPDDP</i>	<i>Vehicle Routing Problem with Divisible Deliveries and Pick-ups</i> Problemas de rutas de vehículos con entregas y recogidas divisibles
<i>VRPMB</i>	<i>Vehicle Routing Problem with Mixed Backhauls</i> Problemas de rutas de vehículos con retornos mezclados
<i>VRPSD</i>	<i>Vehicle Routing Problem with Stochastic Demand</i> Problema de las rutas de vehículos con demandas estocásticas
<i>VRPSDP</i>	<i>Vehicle Routing Problem with Simultaneous Deliveries and Pick-ups</i> Problemas de rutas de vehículos con entregas y recogidas simultaneas

Tabla 0.4: Resumen de acrónimos utilizados en la tesis (continuación).

Índice General

Índice	xix
Índice de figuras	xxiii
Índice de tablas	xxv
Índice de algoritmos	xxvi
1. Introducción general.	1
1.1. Introducción.....	1
1.2. Logística y distribución.....	4
1.3. La cooperación entre empresas.....	6
1.4. Los problemas de optimización combinatoria.....	7
1.4.1. Formulación matemática.....	9
1.4.2. Algoritmia y complejidad computacional.....	11
1.5. Los problemas de rutas de vehículos.....	17
1.5.1. El problema general de rutas con recogida y entrega.....	23
1.5.2. Rutas de vehículos con retornos agrupados (VRPCB).....	25
1.5.3. Rutas de vehículos con retornos mezclados (VRPMB).....	26
1.5.4. Rutas de vehículos con flotas heterogéneas.....	27
1.6. Métodos de resolución.....	28
1.6.1. Exactos.....	28
1.6.2. Heurísticos.....	30
1.6.3. Metaheurísticos.....	31

1.6.4. Matheurísticos.....	33
1.6.5. Simheurísticos.....	34
1.6.6. Aleatorización de heurísticos.....	36
1.7. Marco de la tesis.....	41
1.7.1. Proyectos de investigación.....	41
1.7.2. Grupo DECYL de la Universidad Pública de Navarra.....	42
1.7.3. Grupo DPCS-ICSO de la Universitat Oberta de Catalunya.....	43
1.8. Organización de la tesis.....	45
1.9. Objetivos de investigación.....	48
1.10. Contribuciones originales.....	50
2. Estado del arte.	53
2.1. Introducción.....	53
2.2. Cooperación Horizontal.....	53
2.2.1. Relaciones de cooperación.....	55
2.2.2. Logística.....	58
2.2.3. Determinación y distribución de ganancias.....	60
2.2.4. La cooperación horizontal en problemas de rutas de vehículos.....	61
2.2.4.1. Recogida y entrega.....	62
2.2.4.2. Multi-depósitos.....	63
2.2.4.3. Flota heterogénea.....	64
2.2.4.4. Retornos (Backhauling).....	65
2.2.5. Criterios medioambientales.....	67
2.3. Problemas de rutas de vehículos con retornos.....	68
2.4. Flotas heterogéneas.....	72

2.5. Conclusiones.....	73
3. El problema de las rutas de vehículos con retornos agrupados (VRPCB).	75
3.1. Introducción.....	75
3.2. Formulación del problema.....	78
3.3. Metodología.....	82
3.4. Pseudocódigo del algoritmo utilizado.....	87
3.5. Experimentos computacionales.....	92
3.6. Discusión y comparativa.....	96
3.6.1. Resultados obtenidos.....	96
3.6.2. Tiempos computacionales.....	98
3.7. Conclusiones y relación del VRPCB con el grado de cooperación horizontal.....	99
4. El problema de las rutas de vehículos con retornos mezclados (VRPMB).	103
4.1. Introducción.....	103
4.2. Formulación del problema.....	104
4.3. Metodología.....	104
4.4. Pseudocódigo del algoritmo utilizado.....	110
4.5. Experimentos computacionales.....	115
4.6. Conclusiones y relación del VRPMB con el grado de cooperación horizontal.....	116
5. El problema de las rutas de vehículos con retornos y flota heterogénea (HVRPB)	119
5.1. Introducción.....	119
5.2. Descripción del problema.....	120
5.3. Metodología.....	120
5.4. Pseudocódigo del algoritmo utilizado.....	123
5.5. Experimentos computacionales.....	124

5.6. Conclusiones del HVRPB y su relación con el grado de cooperación horizontal.	126
6. Conclusiones finales.	129
6.1. Conclusiones.....	129
6.2. Trabajos futuros.....	133
6.3. Artículos.....	134
6.4. Congresos.....	138
6.4.1. Congresos Internacionales.....	138
6.4.2. Congresos Nacionales.....	141
Bibliografía.	143
Apéndices.	165
A1. Artículos aceptados en revistas indexadas en ISI JCR y notificación de aceptación.	
A2. Artículo publicado en revista internacional.	

Índice de figuras

0.1. Resumen de los temas principales de esta tesis.....	12
0.2. Resumen de los temas principales de esta tesis en forma de COPS.....	13
1.1. Formulación matemática en formato MILP (Hoffman, 2013).....	10
1.2. Reducción del problema A al problema B.....	14
1.3. El espacio NP de los problemas de decisión asumiendo que $P \neq NP$	15
1.4. Relación entre problemas NP-completos y problemas NP-complejos.....	17
1.5 Reducción del problema TSP al problema VRP.....	18
1.6. Clasificación de los problemas generales de recogida y entrega.....	24
1.7. Esquema de un algoritmo matheurístico.....	33
1.8. Esquema de un algoritmo simheurístico.....	35
1.9. Organización de la tesis.....	46
2.1. Representación gráfica del modelo de recogidas y entregas.....	63
2.2. Representación gráfica del modelo de depósitos múltiples.....	64
2.3. Representación gráfica del modelo de flota heterogénea.....	65
2.4. Representación gráfica del modelo de retornos.....	66
3.1. Combinación de los procesos de reparto y abastecimiento en VRPCB y VRPMB	76
3.2. Una representación visual del VRP con retornos agrupados.....	81
3.3. Esquema de la metodología empleada para la resolución del VRPCB.....	83
3.4. Utilización de un arco interfaz para conectar rutas de entrega y recogida.....	85
3.5. Comparación de la calidad de las soluciones para el conjunto de instancias GJB.....	97
3.6. Comparación de la calidad de las soluciones para el conjunto de instancias TV.....	97
3.7. Comparación de los tiempos computacionales entre distintos algoritmos.....	99
3.8. Grado de cooperación horizontal utilizando los modelos de VRP.....	101
3.9. No cooperativo (arriba) vs. cooperativo (abajo) utilizando el VRPCB.....	102
4.1. Diagrama de la metodología propuesta para la resolución del VRPMB.....	106
4.2. Procedimiento de divide y vencerás utilizando rectas y ángulos.....	109
4.3. No cooperativo (arriba) vs. Cooperativo (abajo) utilizando el VRPMB.....	118
5.1. Diagrama de la metodología propuesta para la resolución del VRPMB	122
5.2. Comparación de nuestros resultados con BKS y los valores óptimos (si existen o LB (Lower Bound o cota inferior).....	126

5.3: No cooperativo (arriba) vs. Cooperativo (abajo) utilizando el HVRPB..... 128

Índice de tablas

0.1. Resumen de acrónimos utilizados en la tesis.....	xv
0.2. Resumen de acrónimos utilizados en la tesis (continuación).....	xvi
0.3. Resumen de acrónimos utilizados en la tesis (continuación).....	xvii
0.4. Resumen de acrónimos utilizados en la tesis (continuación).....	xviii
1.1. Variantes del VRP según los actores.....	20
1.2. Variantes del VRP según las restricciones.....	22
1.3. Variantes del VRP según los objetivos.....	23
3.1 Resultados para las instancias GJB.....	95
3.2. Resultados para las instancias TV.....	96
4.1. Resultados comparativos para las instancias de Salhi y Nagy (1999)	116
5.1. Resultados computacionales para las instancias del FSMVRPB.....	126
6.1. Problemas clásicos del VRP que se pueden resolver con el planteamiento de esta tesis doctoral.....	126

Índice de algoritmos

1.1. Pseudocódigo del algoritmo GRASP.....	37
3.1. Pseudocódigo del procedimiento principal del ILS-BRCW.....	89
3.2. Creación de la lista de ahorros aleatoria con penalización.....	90
3.3. Operador recursivo correctivo.....	92
4.1. Pseudocódigo del procedimiento principal del MS-BR.....	110
4.2. Pseudocódigo del procedimiento de divide y vencerás utilizando rectas y ángulos...	113
4.3. Asignación de rutas a áreas utilizando los ángulos de las rectas.....	114
5.1. Pseudocódigo del algoritmo principal.....	123

Capítulo 1.

Introducción general.

“...recorrimos los suburbios,
anduvimos juntos entre la maleza,
dormimos en los cobertizos.
Lo que el poeta dice,
lo que dice el poeta a la adivina,
al bisabuelo judío que dormía en la comuna
y aún vaga con su barba blanca por ahí
proclamando su consigna a las abejas:

Las estrellas para quien las trabaja.”

-Juan Carlos Mestre

1.1. Introducción.

La matemática clásica, considera los problemas combinatorios como básicos o triviales, ya que asume como válida la técnica de la enumeración general, esto es, la revisión exhaustiva de todas las posibles soluciones. Aparentemente es sencillo elegir el mejor valor dentro de un número finito de posibilidades. Pero desde el punto de vista de la matemática moderna que intenta dar soluciones a problemas actuales de la economía, ingeniería, física, etc., los problemas combinatorios no son en absoluto triviales, sino que, algunos de ellos, suponen grandes retos.

Si un aficionado al fútbol se planteara utilizar sus vacaciones para visitar los estadios de los equipos de fútbol de 1ª división de “La Liga”, intentaría encontrar en primer lugar la

secuencia de visitas que minimizara los kilómetros recorridos y por tanto la optimización del tiempo disponible en sus días libres. Teniendo en cuenta que actualmente el número de equipos de “La Liga” es de 20 y que no hay dos equipos que utilicen el mismo estadio, el número de ordenaciones posibles de las visitas a los estadios es de $(20-1)!/2=6,08*10^{16}$.

Si un ordenador pudiese revisar 20 billones de soluciones en un segundo, podría recorrer exhaustivamente todas las soluciones y evaluar la más óptima en poco menos de una hora. Pero si el número de equipos de “La Liga” se amplía a 26 equipos, al mismo ordenador no le bastaría con 12 siglos para calcular la ruta óptima dentro de las $25!/2=7,76*10^{24}$ posibles. Es decir, un aumento en el número de equipos del 30% (aumento en el tamaño de la instancia que pasa de 20 a 26 ciudades), supone una explosión en el número de soluciones factibles que se incrementa en un 12.751.199.900%.

Seguramente, el aficionado, al darse cuenta de esta circunstancia intentaría simplificar el problema haciendo grupos de ciudades próximas y resolviendo el problema para los grupos de tal forma que la enumeración de todas las posibles soluciones fuera manejable en un espacio de tiempo razonable. Aunque él posiblemente no lo sepa, está intentando solucionar un problema muy clásico y conocido que es el problema del agente viajero (*Travelling Salesman Problem* o TSP) utilizando un método aproximado.

En caso de que finalmente utilizando el procedimiento elegido haya encontrado una secuencia válida de visitas a estadios, imaginemos que quiere preparar una lista de reproducción de música automática (*Automatic Playlist Generation Problem* o APGP) que le distraiga durante los distintos trayectos. Su aplicación tratará de generar una lista de reproducción que satisfaga sus preferencias (restricciones). El recorrido exhaustivo evaluando cada canción entre las más de 20 millones de canciones que tenga etiquetadas con respecto a sus gustos para elegir el orden adecuado sería una tarea larguísima que requeriría un tiempo que la aplicación no dispone para evaluar las $(20.000.000-1)!/2$ posibilidades. Para obtener una lista de reproducción válida, la aplicación podría utilizar alguno de los algoritmos disponibles, como por ejemplo el explicado en [Hsu y Lai \(2014\)](#).

Estas son formas de resolver problemas de forma no exacta que proporcionan resultados suficientemente válidos, aunque es posible que no óptimos. Este tipo de planteamientos es el origen de los métodos de solución aproximados, que no realizan un recorrido por todo el espacio

de posibles soluciones sino sólo por aquellas zonas más prometedoras aprovechando la particular estructura combinatoria del dominio de factibilidad. La evaluación de “qué zonas son más prometedoras” constituye la clave del éxito de estos métodos y representa uno de los objetivos principales que se derivan de esta tesis doctoral.

Ambos ejemplos pertenecen al grupo de problemas conocidos como problemas de optimización combinatoria (*Combinatorial Optimization Problems* o COPs). Desde una perspectiva teórica, los COPs tienen una definición bien estructurada que consiste en una función objetivo que necesita ser minimizada o maximizada, y una serie de restricciones que deben ser respetadas. Estos problemas tienen un interés por sí mismos debido a las matemáticas involucradas en su modelado, análisis y solución. Por ejemplo, los procesos de toma de decisiones en campos como la logística, el transporte y la producción contienen muchos problemas complejos que pueden expresarse como COPs (Faulin et al., 2012; Montoya et al., 2011). En consecuencia, los investigadores de diferentes áreas, e.g. matemáticas aplicadas, informática e inteligencia artificial, han dirigido sus esfuerzos a concebir técnicas para modelar, analizar y resolver COPs. Encontrar la solución consiste en encontrar un objeto u objetos óptimos dentro de un conjunto finito de los mismos. Como hemos visto en los ejemplos del inicio de esta sección, un primer acercamiento a este tipo de problemas nos llevaría a intentar recorrer el conjunto de objetos exhaustivamente y obtener el mejor de ellos según la función adecuada que nos permita evaluarlos. Pero como hemos visto este procedimiento es en ocasiones difícil o incluso imposible de seguir debido a la dimensión del problema. Estos problemas de optimización son llamados NP-complejos. Para este tipo de problemas, uno de los mecanismos más utilizados en su resolución es el conocido como metaheurístico. Éste es una clase de algoritmo que obtiene soluciones aproximadas de buena calidad en poco tiempo de cálculo para problemas de dimensión elevada.

Muchos de los metaheurísticos que se pueden encontrar en la literatura son de naturaleza no deterministas ya que incluyen la utilización de variables aleatorias dentro de sus procesos de decisión. En la mayoría de los casos, esta aleatoriedad se introduce incluyendo una distribución de probabilidad uniforme, sin embargo, como se muestra en los resultados de esta tesis doctoral, el uso de una aleatorización sesgada (no uniforme) puede llegar a considerarse como una estrategia incluso más eficiente para resolver problemas de optimización (Juan et al, 2011c).

El presente trabajo de investigación se circunscribe dentro del ámbito de la logística y distribución y más concretamente dentro del transporte de mercancías que se realiza por carretera. En este entorno, la definición de algoritmos que utilizan la aleatorización en sus distintos formatos representa la base de sobre la cual se solucionan los problemas planteados en esta tesis.

1.2. Logística y Distribución (L&D).

Uno de los fundamentos de la logística y en particular de la distribución, consiste en trasladar un producto o servicio de un lugar a otro, y para ello requiere la utilización de una serie de recursos: energía, equipos de transporte, tiempo de trabajo, etc. La eficiencia de un transporte se mide por el número de recursos cualificados que emplee en su realización. El uso de estos recursos se traduce, a su vez, en unos costes económicos y medio ambientales. Por tanto, su optimización forma parte de las actividades de mejora continua que las organizaciones deben llegar a cabo para cumplir sus objetivos y asegurar su supervivencia. Son muchas las variables que afectan al coste total de las actividades logística y que son por tanto susceptibles de mejorar. Según el Ministerio de Fomento, en España, el sector del “transporte terrestre (carretera y ferrocarril) y por tubería”, supuso en el año 2012 en torno al 2,15% del Valor Añadido Bruto (VAB). Según el Instituto Nacional de Estadística (INE), este mismo sector, durante el año 2013 tuvo un volumen de negocio de 31.610 millones de euros, generado por unas 108.180 empresas, que ocuparon a 307.047 personas de media en ese mismo periodo. Estos datos muestran una facturación promedio de unos 292.198 euros por empresa.

El transporte de mercancías por carretera aportó 13.548,63 millones de euros, en impuestos y tasas durante el año 2012 (Vassallo et al., 2014). Ese mismo año, el gasto en conservación de la red de carreteras interurbana fue de 1.445,71 millones de euros. Estos datos contrastan con los del transporte de mercancías por ferrocarril, que aportó ese mismo año 98,11 millones, mientras que el gasto de mantenimiento de la red ferroviaria para este mismo año fue de 1.593,74 millones de euros. Por lo tanto, queda claro que el transporte de mercancías por carretera es capaz de cubrir los gastos de mantenimiento de la red de carreteras interurbanas. Una conclusión similar fue alcanzada por el estudio realizado por el Instituto Alemán de Investigación Económica (DIW), en 2009, donde el transporte de mercancías por carretera cubría el 99% de los costes de infraestructura de la red de carreteras, mientras que el tráfico

ferroviario de mercancías cubría tan solo el 11% de sus costes de infraestructura. Como puede verse, los procesos de transporte y logística tienen un gran impacto económico en la mayoría de los países y sus costes se han incrementado considerablemente durante los últimos años. Esto está forzando a las empresas a buscar la forma de disminuir sus costes de la forma más eficiente posible y una de las formas de hacerlo es la cooperación entre entidades que desarrollan actividades semejantes. De los procesos de transporte y logística se estima que el 49% está absorbido por el transporte, el 23% son costes de almacenamiento, 22% manejo de inventario, 4% servicio al cliente y 2% gastos administrativos. Según el MEMO/11/197 de la Comisión Europea, representa entre el 10% y el 15% del coste de un producto acabado para las empresas europeas. Kotler (1991) considera que se pueden conseguir importantes ahorros en las actividades de distribución de materiales. Esta gestión de distribución física necesita resolver una gran variedad de problemas de decisión del ámbito estratégico, táctico y operativo. Entre los primeros se encuentran los problemas de localización de plantas, almacenes o asignación de proveedores. Entre los problemas tácticos están los del dimensionamiento de la flota, composición, alquiler, etc. Los problemas del ámbito operacional son entre otros, el establecimiento de las rutas, así como su secuenciación y composición.

Por tanto, la planificación de las operaciones de distribución física requiere un elevado número de pequeñas decisiones que interactúan entre sí. Además, la complejidad de las instancias crece exponencialmente con la dimensión del problema. Siguiendo los ejemplos incluidos en la introducción de este capítulo, encontrar la mejor secuencia de visitas a 5 ciudades requiere revisar $4!/2=12$ posibilidades mientras que si son 10 las ciudades a visitar, el número de posibilidades se multiplica por $(9-4)!=15.120$, hasta un total de 181.440 posibles ordenaciones de las visitas a las 10 ciudades. Esta complejidad suele llevar a los decisores a intentar simplificar los problemas y resolverlos basándose en la experiencia de errores anteriores. Por tanto, hay un potencial de mejora importante para las empresas en este ámbito en base a utilizar técnicas eficientes de optimización de rutas. La explosión combinatoria del espacio de soluciones impide que la optimización sea en muchos casos reales manejable utilizando técnicas de resolución exactas. Para cubrir esta carencia se dispone de métodos alternativos para obtener soluciones que, aunque no se garantice su optimalidad si al menos permiten obtener soluciones eficientes de calidad suficiente.

De esta forma los problemas de logística y distribución se han convertido en una de las partes importantes de los problemas combinatorias. Esto es debido a que, al tratarse de operaciones muy repetitivas, si se consigue recortar una pequeña parte de los costes, se pueden conseguir importantes ahorros económicos, así como reducir los impactos medioambientales ocasionados por la polución y el ruido. En paralelo se puede mejorar de forma significativa la satisfacción de los clientes al mejorar la calidad de las operaciones.

1.3. La cooperación entre empresas.

En el mercado actual, la supervivencia y sostenibilidad de las empresas está condicionada a su capacidad de mejorar constantemente para adaptarse a los cambios en los mercados y la progresión de sus competidores. Una empresa puede mejorar internamente para hacer más eficientes sus operaciones y también puede hacerlo cooperando con otras empresas que comparten alguno o algunos de sus procesos.

Una posible estrategia que las empresas pueden seguir para mejorar su competitividad es la de colaborar con otras empresas similares (Cooperación Horizontal) aprovechándose así de las economías de escala ([Sanchez-Rodrigues et al., 2015](#)). En particular, la optimización de la distribución de materiales en forma de optimización de las distancias, los vehículos y la utilización de los mismos representan algunas de las principales variables susceptibles de ser incluidas dentro de la cooperación entre empresas ([Perez-Bernabeu et al., 2015](#)).

La Unión Europea ([European Union, 2001](#)) define la cooperación horizontal como el conjunto de actividades concertadas por las compañías que operan al mismo nivel en el mercado. En un mercado tan competitivo como el actual, las reducciones de costes gracias a las economías de escala alcanzadas utilizando estas prácticas constituyen un elemento diferenciador para las empresas. La cooperación con otras compañías representa una importante estrategia para poder ser más eficientes. De una forma más formal, [Bahinipati et al. \(2009\)](#) definieron la cooperación horizontal como un acuerdo entre dos o más compañías al mismo nivel en la cadena de suministro para facilitar el trabajo y la cooperación con el objetivo de alcanzar un objetivo común.

Además de la optimización de la carga y las rutas, uno de las contribuciones de la cooperación horizontal es la reducción de los viajes vacíos de vuelta al depósito o *deadheading*.

Según la Comisión Europea ([European Commission, 2011](#)), el 25% de las actividades de transporte por carretera se realizan con la carga vacía. Este objetivo se puede alcanzar introduciendo prácticas de *backhauling* (incluyendo clientes de recogida) en la generación y posterior gestión de las rutas ([Juan et al., 2014](#); [Adensio-Díaz et al., 2014](#)).

Asimismo, un beneficio que aporta la cooperación horizontal está relacionado con la reducción del impacto medioambiental de las actividades de distribución. Nuestra revisión de la literatura mostrará cómo la práctica del *backhauling* se considera una de la más básicas, eficientes y fáciles de utilizar para disminuir tanto los costes económicos como los medioambientales ([Ubeda et al., 2011](#)). Esta práctica será estudiada en los capítulos siguientes en sus dos principales versiones como eje central de este trabajo de investigación.

Otro de los factores que contribuyen a la eficiencia de las actividades de distribución es la adaptación de la flota a la carga a transportar ([Dominguez et al., 2016](#)). El objetivo sería disponer de la flota de vehículos que en la que se utilice al máximo su capacidad respetando al mismo tiempo las limitaciones impuestas por la reglamentación. De esta forma, parece coherente considerar un cierto grado de heterogeneidad en el dimensionamiento de la flota. Disponer de una flota heterogénea es muy habitual en la mayoría de situaciones reales. De hecho, cuando se consideran los costes fijos y variables, incluso dos vehículos con la misma capacidad pueden ser considerados como diferentes debido a factores como la depreciación, seguros, mantenimiento, conductor, etc.

1.4. La optimización combinatoria.

Optimizar consiste en obtener los valores óptimos, máximos o mínimos de una función definida dentro de un dominio o región de factibilidad. En el mundo empresarial se suelen presentar problemas en los que hay que decidir la asignación óptima de una serie de recursos limitados (humanos, materiales, temporales, etc.) con una serie de requisitos a cumplir (demandas, producciones, tiempos de descanso, etc.) que limitan la elección de la solución adecuada desde un punto de vista estratégico, táctico u operativo. Además, estas asignaciones con estas restricciones suelen tener objetivos a cumplir que suelen ser del tipo de minimizar costes, maximizar beneficios, maximizar el nivel de servicio, minimizar impactos medioambientales, etc. Estos objetivos pueden buscarse por separado o de forma conjunta y en

ocasiones incluso contradictoria. El número de variables a utilizar para su resolución puede ser muy alto en los problemas reales lo que hace que su resolución de forma exacta sea inviable. En estos casos, se suelen utilizar métodos aproximados para obtener soluciones eficientes (puede que no sean óptimas) que sean satisfactoria. Un tipo concreto de problemas de optimización son los denominados problemas de optimización combinatoria. El modelado y resolución de estos problemas es de gran utilidad para la toma de decisiones en problemas de logística y distribución.

Papadimitriou y Steigliz (1982) definen la *optimización* como el proceso de búsqueda del máximo o mínimo de una función objetivo sobre la región de soluciones factibles, también conocida como región de factibilidad. Por otro lado, **Lawler (2001)** define la *combinatoria* como la rama de la matemática que trata de ordenar objetos usualmente finitos en número y sujetos a varias restricciones. Los problemas de optimización en los que las variables de decisión son enteras reciben el nombre de problemas de optimización combinatoria (*Combinatory Optimization Problem* o COP). Se trata de hallar el mejor valor de entre un número de soluciones viables finito o reducible a finito. Tenemos muchos ejemplos de COPs en nuestra vida diaria, el más común es el que se nos presenta cuando tenemos que tomar una decisión entre un conjunto de dos o más posibilidades.

De una forma más rigurosa **Papadimitriou y Steigliz (1982)**, definieron un problema de optimización combinatoria $P=(S,f)$ formado por un conjunto finito de objetos S y por una función objetivo $f: S \rightarrow \mathbb{R}^+$ que asigna un coste positivo a cada objeto $s \in S$. Resolver el problema es encontrar el objeto que tenga coste mínimo, es decir se trata de hallar:

$$s' \in S | f(s') \leq f(s), \forall s \in S \text{ y } s \neq s'$$

Minimizar una función f es lo mismo que maximizar su función opuesta $-f$. De esta forma los problemas de optimización combinatoria pueden escribirse tanto en su vertiente de maximización como de minimización.

1.4.1. Formulación matemática del problema.

Para nuestros propósitos, un problema será una cuestión general a ser respondida, que por lo general posee varios parámetros o variables, cuyos valores no se especifican. Para describir un problema se necesita obtener:

- Una descripción general de todos sus parámetros.
- La descripción de las propiedades que debe satisfacer la respuesta, o solución.

Una *instancia* del problema se obtiene especificando los valores particulares para todos los parámetros que lo definen. Se conoce como problema de optimización entera, aquel cuyos recursos pueden dividirse en partes enteras. En esos casos, las restricciones de divisibilidad de estos recursos, que pueden ser personas, máquinas, trayectos u otras entradas discretas, puede restringir las posibles alternativas a un conjunto finito. Sin embargo, por lo general las alternativas para hacer una enumeración completa son demasiadas para que la búsqueda exhaustiva sea una alternativa viable.

La forma habitual de modelar estos problemas es a través de su formulación matemática siguiendo la nomenclatura conocida como problemas de optimización lineal entera mixta (*Mixed Integer Linear optimization Problem* o MILP). Su fórmula matemática general se puede ver en la [Figura 1.1](#).

En términos matemáticos, optimizar un proceso consiste en configurar la función que lo define y hallar su máximo o mínimo dentro de su dominio de definición que es el espacio común a todas las restricciones. Dicho de otra forma, se trata de evaluar la función objetivo en todos los puntos del espacio común de las restricciones y proporcionar el punto cuya evaluación resulta ser máxima o mínima.

$$\text{Maximizar } \sum_{j \in B} c_j x_j + \sum_{j \in I} c_j x_j + \sum_{j \in C} c_j x_j \quad (1)$$

$$\text{Sujeto a } \sum_{j \in B} a_{ij} x_j + \sum_{j \in I} a_{ij} x_j + \sum_{j \in C} a_{ij} x_j + b_i \leq 0 \quad i = 1, \dots, m \quad (2)$$

$$l_j \leq x_j \leq u_j \quad \forall j \in N = B \cup I \cup C \quad (3)$$

$$x_j \in \{0,1\} \quad \forall j \in B \quad (4)$$

$$x_j \in \mathbb{Z} \quad \forall j \in I \quad (5)$$

$$x_j \in \mathbb{R} \quad \forall j \in C \quad (6)$$

Figura 1.1: Formulación matemática en formato MILP (Hoffman, 2013).

La solución al problema cuya formulación se ve en la Figura 1.1 de (1) a (6) es un conjunto de valores asignados a las variables $x_j, j \in N$. El objetivo es maximizar la función (1) que suma los productos de las variables con sus pesos respectivos. Los pesos son los coeficientes que definen la instancia $c_j; j \in N$. l_j y u_j son los valores de los límites inferior y superior, respectivamente, para la variable $x_j, j \in N$.

Se pueden encontrar tres tipos de variables según su naturaleza y consideramos tres conjuntos que contienen a los distintos índices de las variables.

- B es el conjunto de índices de las variables binarias (*binary*).
- I es el conjunto de índices de las variables enteras (*integer*).
- C es el conjunto de índices de las variables continuas (*continuous*).

La ecuación (2) representa las restricciones que pueden ser desigualdades (“ \leq ”, “ \geq ”) o igualdades (“ $=$ ”). l_j y u_j son las cotas superiores e inferiores de cada variable.

Según el número de variables de cada tipo, los problemas se clasifican (Hoffman, 2013):

- Cuando $B=I=\emptyset$, se conoce como problema de optimización lineal (*Linear Optimization Problem* o LOP).
- Cuando $C=I=\emptyset$, se conoce como problema de optimización lineal binario (*Binary Integer Linear optimization Problem* o BILP).

- Cuando $C=\emptyset$, se conoce como problema de optimización lineal entero (*Integer Linear optimization Problem* o ILP).
- En caso contrario, el problema es conocido como problema mixto de optimización lineal entero (*Mixed Integer Linear optimization Problem* o MILP).

La formulación de los COPs se realiza en la mayoría de las ocasiones siguiendo la formulación BILP.

1.4.2. Algoritmia y complejidad computacional.

Los *algoritmos* son una secuencia de actividades, que paso a paso, se utilizan para dar solución a un problema determinado. La búsqueda de algoritmos efectivos para resolver problemas se viene realizando desde hace más de 2.000 años. Los matemáticos griegos, pusieron gran énfasis en las técnicas constructivas (ahora llamadas algoritmia), como se comprueba en los trabajos de Euclides y Pitágoras.

En geometría se plantearon algunos problemas no resueltos y que han constituido materia de investigación durante mucho tiempo, como el problema de la cuadratura del círculo: “Dado un círculo, construir un cuadrado con la misma área utilizando regla y compás”. Un algoritmo que lo solucione será aquel que obtenga la secuencia de actividades que partiendo de un círculo obtenga un cuadrado que respete la restricción del área y sólo utilice una regla y un compás. Pero este algoritmo se sabe que no existe hoy en día, ya que no existe un método de construcción apropiado que lo resuelva.

En **Smullyan (1991)** se puede leer la definición de algoritmo que proporciona el científico Gödel como una secuencia de reglas para formar funciones matemáticas complicadas a partir de funciones matemáticas más simples. En el mismo libro se explican los teoremas de Gödel que explican que existen problemas para los cuales no puede hallarse algoritmos que los resuelvan. **Turing (1950)** combinó los conceptos de algoritmo y computación con una definición matemática del primero. Utilizó un dispositivo lógico llamado máquina de computación lógica (conocida como máquina de Turing), y definió un algoritmo como cualquier conjunto de instrucciones para dicha máquina. La máquina de Turing define su estado

siguiente según el estado actual y el símbolo que va a procesar. En el caso de que para cada par (estado, símbolo) posible exista a lo sumo un estado siguiente, se dirá que es una máquina de Turing determinista, mientras que si existe más de una posible combinación de estados siguientes se dirá que se trata de una máquina de Turing no determinista.

De una forma simplificada, podemos pensar en un algoritmo simplemente como un programa informático. Se dice que un algoritmo resuelve un problema si ese algoritmo se puede aplicar a cualquier instancia del problema garantizando siempre la obtención de una solución para esa instancia. El término "solución" se entiende aquí estrictamente en el sentido de resolver un problema, no solamente una instancia particular del mismo. Un algoritmo no resuelve un problema a menos que lo haga para todas las instancias que pueden considerarse. La algoritmia es la ciencia que busca los algoritmos más eficientes teniendo en cuenta el mejor uso de los recursos computacionales utilizados, que normalmente están medidos en forma de la velocidad de ejecución y obtención de los resultados.

Algoritmia y computación están íntimamente ligados y por tanto la identificación de los problemas para los que se pueden encontrar algoritmos que los resuelvan y por tanto sean computables y los que no lo son tiene un considerable interés pues establece los límites de la computación o, en cierta medida, los límites teóricos de los ordenadores. Una medida de la "eficiencia" de un algoritmo en la resolución de un problema, se obtiene asignándole una función f cuyo dominio es el espacio de los números naturales. Esta función se conoce como *complejidad del algoritmo*:

- $f(n)$ es el máximo de la cantidad de pasos que precisa el algoritmo para resolver instancias del problema de tamaño $\leq n$.

Esta función es muy difícil de evaluar porque depende de muchos factores externos y su principal objetivo es la comparación entre distintos algoritmos. Para ello se utilizan funciones sencillas que tengan el mismo tipo de crecimiento como $n \log(n)$, las polinómicas tipo n^3 , o exponenciales tipo 2^n . De esta forma se dice que la complejidad de un algoritmo es de orden $g(n)$, y se representa por $O(g(n))$. Un algoritmo se dice *polinomial* cuando su complejidad es $O(g(n))$ con g acotado superiormente mediante un polinomio en n . En caso contrario se dice que el algoritmo es *exponencial*. Por tanto, la complejidad temporal estará directamente ligada

al tamaño del problema y al tiempo de ejecución. La distinción entre estos dos tipos de algoritmos tiene especial importancia al considerar problemas de dimensión elevada.

Por otro lado, la *complejidad de un problema* es equivalente a la complejidad del mejor algoritmo que lo resuelve. Como regla general, se dice que un problema no está solucionado adecuadamente hasta que se ha encontrado un algoritmo que lo resuelve en tiempo polinomial. En consecuencia, se dice que un problema es difícil (*intractable*) si no existe un algoritmo de complejidad polinomial que pueda resolverlo y fácil (*tractable*) en caso contrario. La distinción entre algoritmos cuya complejidad es polinómica (eficiente) y exponencial (ineficiente) admite muchas excepciones cuando las instancias a comparar tienen distintos tamaños. Es decir, un problema considerado difícil, puede ser resuelto de forma eficiente por un algoritmo de complejidad exponencial en instancias de dimensión reducida.

La teoría de la complejidad de problemas está basada en los problemas de decisión. Un problema de decisión es aquel cuya solución es sí o no. Un problema de optimización siempre se puede reducir a un problema de decisión haciendo una transformación de una instancia del primer problema en una instancia equivalente del segundo problema. De forma equivalente, esta transformación proporciona la forma de convertir o reducir cualquier algoritmo que resuelva el segundo problema en el algoritmo correspondiente que resuelve el primer problema. La reducción de unos problemas en otros se puede ver de forma esquemática en la **Figura 1.2**. El proceso de reducir un algoritmo que soluciona el problema A en un algoritmo que soluciona el problema B consiste en lo siguiente. En primer lugar, se hace el mapeo de la instancia del problema A en la instancia del problema B. Este mapeo debe realizarse en tiempo polinomial. En este momento ya tenemos la instancia A transformada en formato de la instancia B que por tanto puede ser la entrada del algoritmo B. A continuación, el algoritmo B soluciona el problema para esa instancia y por tanto produce la salida en forma de solución B. Es aquí cuando se hace la transformación de la solución B en la solución A utilizando el mapeo que debe realizarse en tiempo polinomial. De esta forma se cierra el círculo habiendo obtenido una solución al problema A utilizando el algoritmo que soluciona problemas de tipo B. Como los mapeos $A \gg B$ y $B \gg A$ se han realizado en tiempo polinomial, podemos concluir que la complejidad del algoritmo A es la misma que la del algoritmo B.

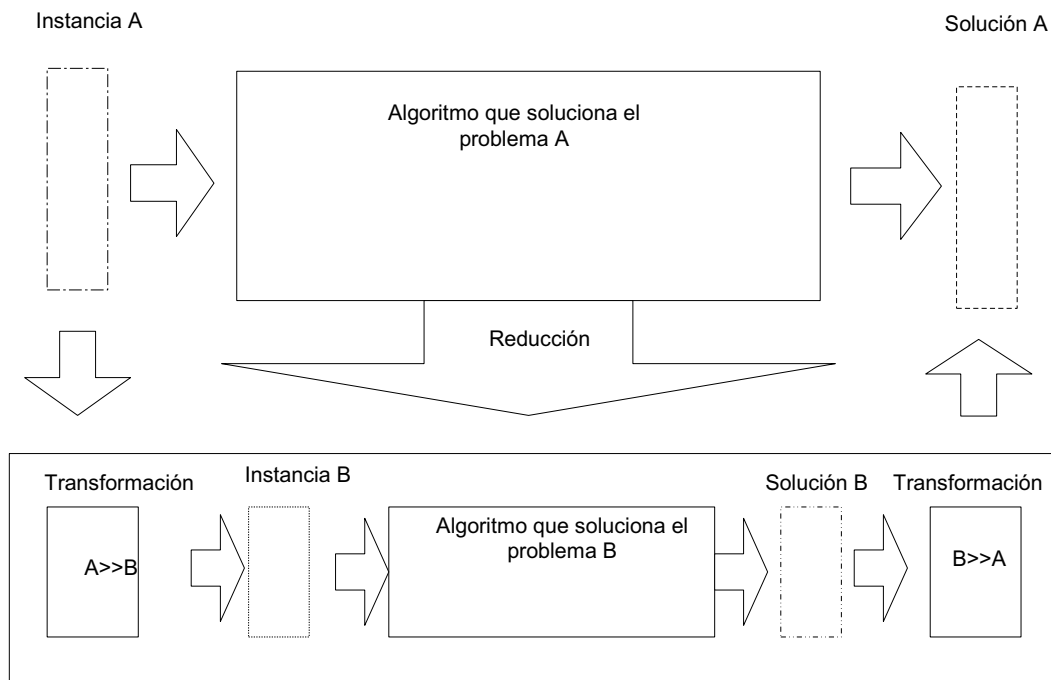


Figura 1.2: Reducción del problema A al problema B.

Un aspecto importante de la teoría de la complejidad es la clasificación de los problemas en clases de complejidad. Las clases representan al conjunto de problemas que se pueden solucionar utilizando una determinada cantidad de recursos. Se consideran dos clases de problemas principales:

- P , polinomial (*Polynomial*), aquellos cuya solución se puede encontrar en tiempos que crecen de forma polinómica cuando su dimensión se incrementa de forma lineal. que contiene. Estos problemas se pueden resolver en una máquina de Turing determinística y secuencial en tiempo polinomial.
- NP , polinomial no determinístico (*Nondeterministic Polynomial*), los que se solucionan en tiempos que crecen de forma no polinomial cuando su dimensión se incrementa de forma lineal. Los problemas se resuelven en tiempo polinomial en una máquina de Turing no determinística. Lo que equivale a decir que este algoritmo en tiempo polinomial no se ha encontrado todavía.

Por definición $P \subseteq NP$ ya que, si un problema se resuelve en tiempo polinomial en una máquina de Turing no determinística, también se resolverá en un tiempo equivalente en una máquina determinística. Una clasificación de los conjuntos se puede ver en la [Figura 1.3](#).

Solamente los problemas del primer tipo P , se pueden resolver eficientemente a través de algoritmos polinomiales o al menos se han encontrado algoritmos polinomiales que lo hacen.

Hasta el día de hoy todos los algoritmos propuestos para resolver problemas de la clase NP mediante una máquina de Turing determinística requieren un tiempo exponencial, es decir, no se puede encontrar la solución en un tiempo razonable. La pregunta que no se ha respondido todavía se refiere a la relación entre estas dos clases: ¿Puede hallarse un algoritmo polinomial para resolver cualquier problema?, o lo que es lo mismo:

- ¿Es la clase P igual a la clase NP ?

Si alguien encuentra un algoritmo que resuelva un problema de la clase NP en tiempo polinomial mediante una máquina de Turing determinística entonces demostraría que todos los problemas de la clase NP también se pueden resolver en tiempo polinomial.

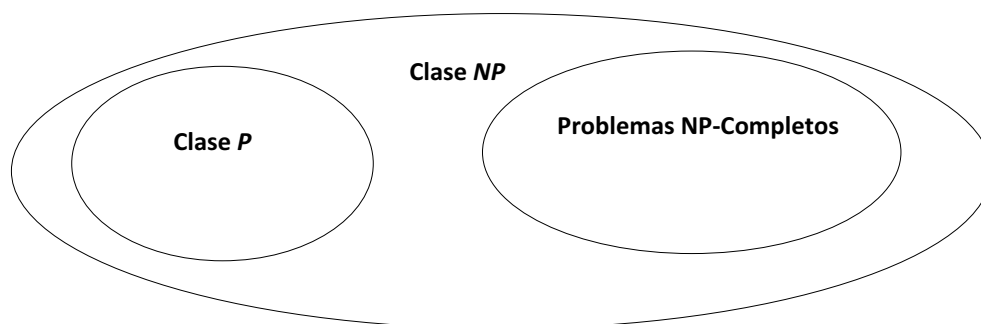


Figura 1.3: El espacio NP de los problemas de decisión asumiendo que $P \neq NP$ (Talbi, 2009).

Después de más de 50 años nadie ha sido capaz de dar una respuesta a dicha cuestión. El Instituto de Matemáticas Clay ([Clay Mathematics Institute, 2000](#)) dedicado a incrementar y diseminar el conocimiento matemático anunció en el año 2000 un premio de un millón de dólares a la primera persona que logre dar una demostración de alguno de los siete problemas del milenio. Su antecedente sucedió 100 años antes, en 1900, cuando David Hilbert en un discurso frente al congreso internacional de matemáticos, propuso una lista de 23 problemas. El instituto Clay fue fundado en 1998 por Landon T. Clay, y por el matemático Arthur Jaffe de la Universidad de Harvard. El Instituto de Matemáticas Clay es una fundación sin ánimo de lucro localizada en Cambridge, Massachusetts, dedicada a fomentar el conocimiento

matemático. El primer problema incluido en la lista consiste en demostrar si $P=NP$ (P versus NP). Estos son los siete problemas matemáticos del milenio:

1. P versus NP .
2. La conjetura de Hodge.
3. La conjetura de Poincaré.
4. La hipótesis de Riemann.
5. Existencia de Yang-Mills y del salto de masa.
6. Las ecuaciones de Navier-Stokes.
7. La conjetura de Birch y Swinnerton-Dyer.

La **Figura 1.3** muestra las relaciones entre las clases P , NP y los problemas NP-completos. Un problema de decisión se dice que es NP-completo (*NP-Complete* o NPC) si cualquier problema en NP puede reducirse a aquel en tiempo polinomial. De todas formas, para evidenciar que un problema es imposible de resolver eficientemente con un algoritmo (en tiempo polinomial) no hace falta demostrar que pertenece al conjunto NP-completo. Dado un problema Ψ que no se sabe si pertenece a NPC, si existe un problema catalogado como NPC que puede reducirse a Ψ (con una transformación de instancias y soluciones que se realiza en tiempo polinomial) entonces podemos concluir que Ψ no puede resolverse en tiempo polinomial (salvo que se demuestre que $P=NP$). Como se ve en la **Figura 1.4**, a los problemas de optimización cuyo problema de decisión equivalente es NP-completo se les conoce como NP-complejos (*NP-Hard* o NPH). Como hemos visto al principio de esta sección, esta reducción se puede hacer para cualquier problema de optimización.

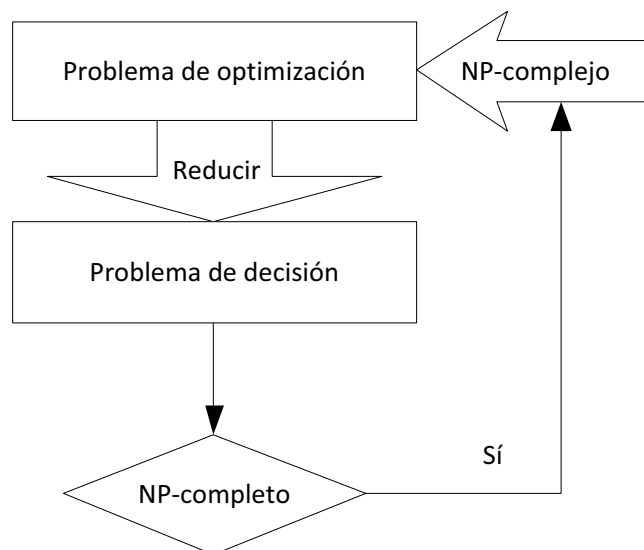


Figura 1.4: Relación entre problemas NP-completos y problemas NP-complejos.

Se puede leer una revisión más completa de la complejidad computacional en [Garey y Johnson \(1979\)](#), [Papadimitriou y Steiglitz \(1982\)](#), [Schrijver \(1986\)](#) y [Talbi \(2009\)](#).

1.5. Problemas de rutas de vehículos.

Uno de los problemas clásicos que se pueden formular como COP es el conocido como problema de rutas de vehículos (*Vehicle Routing Problem* o VRP). En esta sección se introducen algunas variantes del problema que sirven para familiarizarse con las características de los problemas que se discuten en las siguientes secciones.

El proceso de transporte de materiales está presente en todas las etapas de los sistemas de producción y distribución y representa un componente relevante (generalmente del 10% al 20%) del coste final de los productos. En esta tesis, consideramos los problemas relativos a la distribución de productos entre los almacenes o depósitos y cliente finales. El VRP es un problema combinatorio que debido a la complejidad de su resolución pertenece al conjunto denominado NP-complejo ([Lenstra y Rinnooy Kan, 1981](#)). Su reducción se puede ver de forma esquemática partiendo el problema del agente viajero (*Travelling Salesman Problem* o TSP) que es un problema conocido como NP-Complejo. El objetivo que se busca en el TSP es encontrar el camino más corto posible para visitar a un grupo de ciudades determinadas donde cada ciudad se visita exactamente una vez y se vuelve a la ciudad de origen.

Según la **Figura 1.5**, podemos reducir el problema TSP al problema VRP y por tanto ya que aquel es NP-complejo, no se conoce un algoritmo polinomial que resuelva este y por tanto pertenece a la clase NP-complejo.

La distribución de productos se refiere al envío, en un período de tiempo determinado, de un conjunto de clientes por un conjunto de vehículos, que están ubicados en uno o más depósitos, son operados por un conjunto de conductores y realizan sus desplazamientos utilizando una red de carreteras apropiada. En particular, decimos que la solución de un VRP está determinada por la selección de un conjunto de rutas, cada una realizada por un único vehículo que se inicia y termina en un depósito, de tal forma que se cumplen todos los requisitos de los clientes, se cumplen todas las restricciones operativas, y se minimiza el coste global del transporte.

Con este propósito general, necesitamos modelar el problema considerando todos los elementos que lo constituyen. En primer lugar, los *objetos* o actores que serán: rutas, clientes, depósitos, conductores, vehículos y productos (**Tabla 1.1**). A continuación, la condiciones o *restricciones* que pueden imponerse en la configuración de la solución (**Tabla 1.2**) y finalmente los posibles *objetivos* a alcanzar en el proceso de optimización (**Tabla 1.3**).

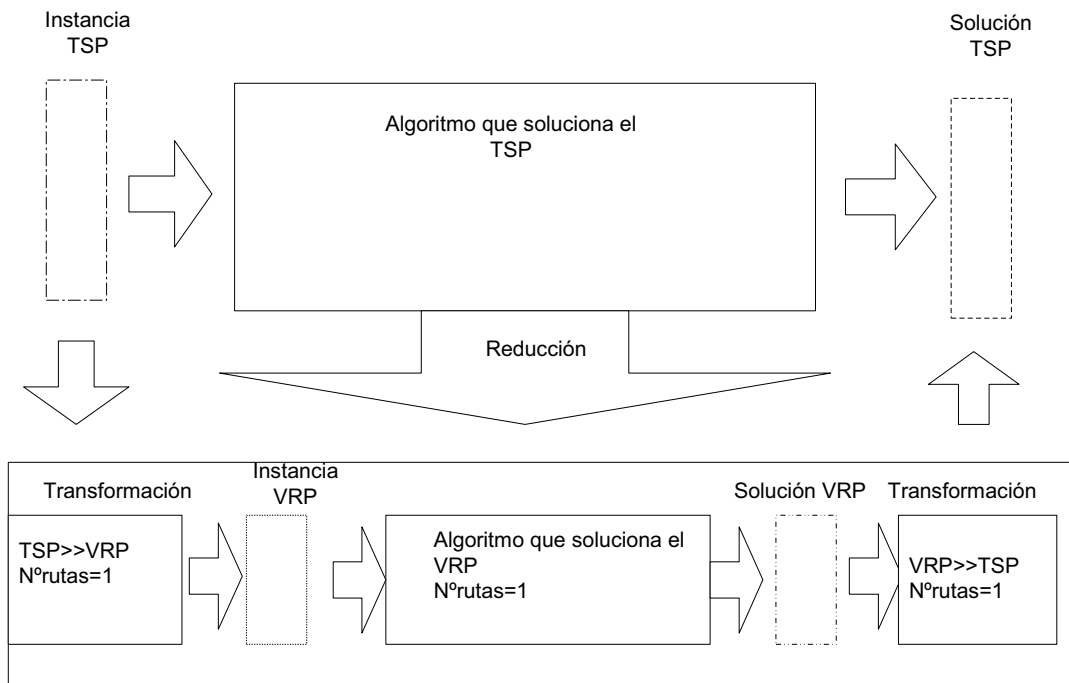


Figura 1.5: Reducción del problema TSP al problema VRP.

La *ruta* utilizada para el transporte de mercancías, se representa con un grafo, cuyos arcos representan los caminos a recorrer y los vértices corresponden a las ubicaciones de depósitos y clientes. Los *depósitos* o almacenes pueden ser el origen y final de cada ruta, en caso contrario la mercancía se sirve directamente de unos clientes a otros. Los arcos pueden ser direccionados o no, dependiendo de que tengamos alguna restricción de orientación o los costes no sean simétricos. Cada arco tiene asociado un coste, que normalmente representa su longitud, y un tiempo de recorrido, que dependerá del tipo de vehículo y/o del período durante el cual se atraviesa el arco. Los clientes pueden tener distintas características según:

- Su ubicación espacial.
- La cantidad de mercancía (demanda), que deben recibir o entregar.
- Los períodos del día (ventanas de tiempo) durante las cuales el cliente puede ser servido (por ejemplo, debido a períodos específicos durante los cuales el cliente está abierto o la ubicación puede ser alcanzada, debido a limitaciones de tráfico).
- Los tiempos necesarios para entregar o recoger las mercancías en el lugar del cliente (tiempos de descarga o de carga, respectivamente), posiblemente dependiendo del tipo de vehículo.
- Un subconjunto de los vehículos disponibles que pueden utilizarse para servir al cliente (por ejemplo, debido a posibles limitaciones de acceso o requisitos de carga y descarga).
- Pueden requerir la entrega o la recogida de mercancías, o ambas posibilidades al mismo tiempo.
- Pueden ser servidos sólo dentro de los períodos de trabajo de los conductores asociados con los vehículos que visitan.

Tipo	Valores	Descripción
Multi-almacén	Sí/no	Hay más de un depósito de donde los vehículos salen y llegan.
Ventanas de tiempo almacén	Sí/no	El depósito está abierto durante un período de tiempo.

Múltiples productos	Sí/no	Algunos vehículos pueden transportar varios tipos de productos.
Incompatibilidad de visitas	Sí/no/casos	Algunos vehículos no pueden visitar algunos nodos.
Demanda estocástica/dinámica	Fija/variable	Las demandas de los clientes pueden cambiar durante el proceso o ser estocásticas.
Ventanas de tiempo	Sí/no/múltiples	Cada cliente tiene una ventana de tiempo particular.
Capacidad del cliente	fija/variable	Los clientes pueden tener una capacidad máxima.
Capacidad en varias dimensiones	1D/2D/3D	La capacidad de los vehículos se considera en 2 Dimensiones o en 3 Dimensiones.
Capacidad limitada	Sí/no	La capacidad de los vehículos es limitada.
Flota homogénea	Sí/no	Todos los vehículos tienen la misma capacidad.
Flota ilimitada	Sí/no	El número de vehículos considerados es ilimitado.
Coste fijo/variable por vehículo	Sí/no	El uso de un vehículo implica un coste adicional.
Múltiples viajes	Sí/no/algunos	Todos o algunos vehículos pueden ejecutar más de un viaje.
Incompatibilidad de trayectos	Sí/no/casos	Algunos vehículos no pueden pasar por algunos trayectos.
Regulación conductor	Sí/no	El conductor tiene que respetar algunos límites horarios para descansos.

Tabla 1.1: Variantes del VRP según los valores de los objetos o actores.

Se pueden imponer restricciones de precedencia en el orden en que se visitan los clientes en una ruta. Un tipo de restricción de precedencia requiere que un cliente dado sea servido en la misma ruta que sirve a un subconjunto dado de otros clientes y que el cliente debe ser visitado antes o después de los clientes que pertenecen al subconjunto asociado. Este es el caso, por ejemplo, de los denominados problemas de recogida y entrega, en los que en la misma ruta pueden realizar operaciones de recogida y entrega de mercancías, y las mercancías recogidas de los clientes de recogida pueden ser llevadas a los clientes de entrega correspondientes por el mismo vehículo si no existe un depósito central. En el caso de existir el depósito, otro tipo de restricción de precedencia impone que, si los clientes de diferentes tipos son servidos en la misma ruta, el orden en el que se visitan los clientes es fijo. Esta situación se presenta, por ejemplo, en el llamado problema de rutas de vehículos con retornos (*Vehicle Routing Problema with Backhauls* o VRPB), en el que las rutas pueden realizar tanto la recogida como la entrega de mercancías, pero las limitaciones asociadas con la carga y la dificultad de reordenar la carga del vehículo a lo largo de la ruta, imponen que todas las entregas deben realizarse antes de las

recogidas. Este es el problema que se aborda en el capítulo 3 de esta tesis. Cuando no se considera esta restricción de precedencia, se pueden visitar a los clientes en cualquier secuencia de la ruta. Este es el problema que se aborda en el capítulo 4 de esta tesis.

Considerando las características de la *flota de vehículos* su composición y tamaño pueden ser fijos o configurables, y su coste y capacidad puede ser homogéneo o heterogéneo. Además, los vehículos pueden tener algunas restricciones operativas que habrá que considerar:

- El depósito de origen y el depósito final pueden ser distintos.
- La capacidad del vehículo puede expresarse en términos de peso máximo, o de volumen, o número de contenedores.
- Se puede subdividir en compartimentos según las necesidades de los clientes o de los productos a transportar.
- Algunos vehículos pueden necesitar algunos dispositivos especiales para las operaciones de carga y descarga.
- Algunos vehículos pueden no tener permiso para atravesar algunas rutas o arcos de rutas.
- Los costes asociados con la utilización del vehículo pueden ser de varios tipos, por unidad de distancia, por unidad de tiempo, por ruta, etc.

Tipo	Valores	Descripción
Duración de la ruta	Variable/fija	La duración de cada ruta no puede exceder un valor máximo.
Carga de rutas balanceadas	Sí/no	La carga de rutas o vehículos debe estar equilibrada.
Costes simétricos	Sí/no	La matriz de coste tiene una naturaleza simétrica.
Recarga dentro de la ruta	Sí/no	Los vehículos deben ser recargados en algún punto de las rutas.
Tiempos de rutas variables	Sí/no	Los tiempos de viaje pueden variar durante un día. La ubicación / distancia de los clientes cambia.
Recogida y entrega	Sí/no	La construcción de rutas debe considerar la recogida de productos en algunos clientes y la entrega a otros.
Recogida y entrega	Agrupados/mezclados	
Recogida y entrega	Visita única/múltiple	

Multi-periodo	Sí/no	La optimización se realiza a lo largo de periodo de tiempo mayor de un día
Control de inventario	Sí/no	Los costes de los niveles de stock se consideran con los costes de enrutamiento.
Rutas abiertas	Sí/no	Las rutas empiezan en el depósito y acaban en el último cliente.
Inicios y fin de rutas diferentes	Sí/no.	Los vehículos empiezan y terminan en diferentes depósitos.
Inicios y fin de rutas diferentes	Inicio/Fin	Los vehículos empiezan en diferentes localizaciones.
Precedencias de visitas	Sí/no	El orden de visitas de algunos clientes debe ser respetado
Restricciones de cargas/vehículos	Sí/no	Algunos vehículos no pueden cargar/descargar en algunos clientes/depósitos

Tabla 1.2: Variantes del VRP según las restricciones.

En cuanto a los *conductores* que manejan los vehículos deben cumplir con varias restricciones legales en cuanto a, por ejemplo, los períodos de trabajo durante el día, el número y la duración de las pausas durante el servicio, la duración máxima de los períodos de conducción, las horas extraordinarias. Las *rutas* deben satisfacer varias limitaciones operacionales que dependen de la naturaleza de los bienes transportados, de la calidad del nivel de servicio y de las características de los clientes y de los vehículos. Otras restricciones operacionales son por ejemplo que la carga actual del vehículo no puede exceder la capacidad del vehículo. Si consideramos los *objetivos*, empezaremos por hacer la evaluación del coste de las rutas respetando las restricciones impuestas. Si utilizamos el grafo que hemos definido al principio de esta sección, para cada par de vértices i y j se define un arco (i, j) cuyo coste c_{ij} viene dado por el coste del trayecto más corto partiendo del vértice i y llegando al vértice j . Se pueden considerar simultáneamente varios objetivos, a menudo contradictorios. Algunos de los objetivos que suelen utilizarse son:

- Minimizar el coste global del transporte, dependiendo de la distancia recorrida y de los costes fijos asociados con los vehículos usados.
- Minimizar el número de vehículos (o conductores) requeridos para servir a todos los clientes.
- Equilibrar las rutas, para el tiempo de viaje y la carga del vehículo.

- Minimizar las penalizaciones asociadas con el servicio no completo de los clientes.
- Cualquier combinación de los anteriores.

Tipo	Valores	Descripción
Objetivos múltiples	Sí/no	Los objetivos pueden ser múltiples y contradictorios.
Objetivo simple	Maximizar	
Objetivo simple	Minimizar	
Objetivos múltiples	Maximizar/Minimizar	

Tabla 1.3: Variantes del VRP según los objetivos.

Han transcurrido más de 50 años desde que **Dantzig y Ramser (1959)** introdujeron el VRP. En su trabajo, los autores describieron una aplicación del mundo real (relativa a la entrega de gasolina a gasolineras) y propusieron la primera formulación de la programación matemática y el enfoque algorítmico para la solución del problema. Desde entonces se han escrito muchos artículos sobre este asunto proponiendo distintos tipos de soluciones a los modelos planteados. Esta tesis doctoral propondrá diferentes soluciones a los modelos que se plantean para tres variantes distintas del problema general VRP.

1.5.1. El Problema General de las Rutas de Vehículos con Recogida y Entrega.

Tanto el problema de las rutas con retornos como el de las entregas y recogidas se incluyen dentro del conocido como problemas de rutas de vehículos con recogida y entrega (*General Pick and Delivery Vehicle Routing Problem* o GPDP) que incluye algunos modelos adicionales como pueden verse en la **Figura 1.6**. Esta clasificación detallada puede encontrarse en **Parragh et al. (2008)**.

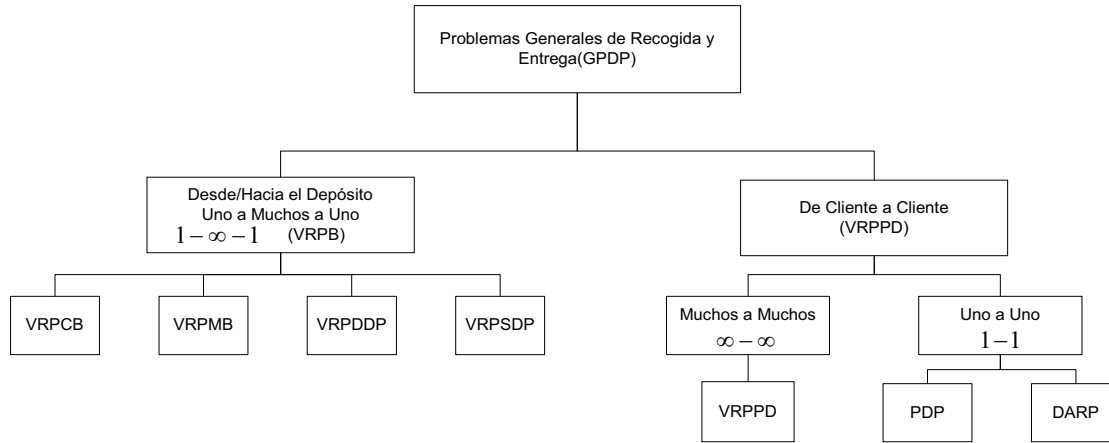


Figura 1.6: Clasificación de los problemas generales de recogida y entrega (Parragh et al., 2008).

Como puede verse en la Figura 1.6, los problemas donde tenemos clientes de recogida de material y clientes de entrega de material se nombran como GPDP. Estos a su vez pueden dividirse en dos tipos según que incluyan un depósito central o los envíos se hagan directamente de un cliente a otro. Cuando se considera un depósito central como origen y fin de las rutas recibe el nombre de problema de rutas de vehículos con retornos (*Vehicle Routing Problem with Backhauls* o VRPB). El VRPB contiene cuatro problemas distintos. El primero es el conocido como Problemas de Rutas de Vehículos con retornos agrupados (*Vehicle Routing Problem with Clustered Backhauls* o VRPCB) donde los clientes de recogida son visitados dentro de una ruta cuando se han completados todas las entregas. El segundo problema es conocido como problemas de rutas de vehículos con retornos mezclados (*Vehicle Routing Problem with Mixed Backhauls* o VRPMB) donde las vistas a clientes de entrega y recogida se pueden realizar en cualquier secuencia. Ambos problemas son abordados en esta tesis.

El tercero se conoce como problema de rutas de vehículos con entregas y recogidas divisibles (*Vehicle Routing Problem with Divisible Deliveries and Pick-ups* o VRPDDP) se produce la relajación de la restricción que considera que cada cliente sólo sea visitado una vez. Por tanto, las recogidas y las entregas pueden dividirse de tal forma que un cliente pueda ser visitado en más de una ocasión. En el cuarto caso, se trata del problema de rutas de vehículos con entregas y recogidas simultaneas (*Vehicle Routing Problem with Simultaneous Deliveries and Pick-ups* o VRPSDP) donde un cliente puede ser al mismo tiempo de entrega y de recogida.

Estos son los modelos cuyas rutas empiezan y terminan en el depósito central. Cuando las rutas empiezan y acaban en los clientes, tenemos los problemas considerados como problemas de rutas de vehículos con recogidas y entregas. En este caso, las mercancías que se entregan a los clientes deben haber sido recogidas previamente en otros clientes. Por tanto, los materiales se transportan directamente entre clientes de recogida y clientes de entrega sin que haya ningún almacén o depósito intermedio. Aquí podemos diferenciar dos tipos según la naturaleza de las relaciones entre clientes que entregan material y los que lo reciben. Si la relación es de muchos a muchos tenemos el problema llamado de recogida y entrega. En cambio, si la relación es de uno a uno, tenemos dos tipos. El primero de ellos es conocido como recogidas y entregas donde cada cliente de recogida sólo tiene asociado un cliente de entrega y viceversa. El segundo es conocido como problema de trazar un camino (*Dial a Ride Problem* o *DARP*) que específicamente aborda el problema del transporte de pasajeros.

1.5.2. Rutas de vehículos con retornos agrupados (VRPCB).

El objetivo del VRPCB es el diseño de un conjunto de rutas de tal forma que:

- Las rutas comienzan y terminan en el depósito central.
- Se minimice el coste de las rutas generadas.
- El número de rutas es igual a K , el número de vehículos a utilizar se da por adelantado y no se permiten rutas con más ni con menos vehículos.
- Si una ruta contiene clientes de recogida y clientes de entrega, los clientes de recogida son visitados cuando todos los clientes de entrega han sido servidos.
- Cada ruta contiene al menos un cliente de entrega.
- Cada cliente es visitado únicamente una vez.
- No se excede la capacidad del vehículo.

Nuestro interés en el VRPB está motivado por su importancia tanto práctica como teórica. En la perspectiva práctica, el VRPB se encuentra frecuentemente en las compañías que tienen que transportar materiales desde sus plantas de producción a sus clientes (entrega). Al mismo

tiempo, las plantas de producción son servidas desde los proveedores (recogida). Un ejemplo común es el trasiego de contenedores (llenos) a los clientes, y la recogida de estos (vacíos) para volver a ser llenados. Un ejemplo en la industria de la alimentación puede verse en el artículo (Ubeda et al., 2011).

Es ampliamente reconocido que se alcanzan significantes ahorros en costes de transporte en estos contextos donde se mezcla recogidas y entregas (Juan et al., 2014b; Ubeda et al., 2011). Desde un punto de vista teórico, el VRPCB aumenta la complejidad del básico VRP ya que añade un nuevo tipo de clientes. Por tanto, es también un problema NP-complejo dentro de los problemas combinatorios.

1.5.3. Rutas de vehículos con retornos mezclados (VRPMB).

El segundo problema abordado por esta tesis, no contiene la restricción de precedencia y las entregas y recogidas pueden hacerse en cualquier momento de la ruta con la única limitación de no superar la capacidad del vehículo. Dentro de la definición mostrada en la sección anterior, hay que excluir:

- Si una ruta contiene clientes de recogida y clientes de entrega, los clientes de recogida son visitados cuando todos los clientes de entrega han sido servidos.
- El número de rutas es igual a K , el número de vehículos a utilizar se da por adelantado y no se permiten rutas con más ni con menos vehículos.

Ya que ahora las visitas a ambos tipos de clientes se pueden alternar con la única restricción de respetar la capacidad del camión. Además, el tamaño de la flota deja de ser una restricción. Esta característica no pertenece al modelo de forma intrínseca, sino que ha sido asumida por todos los investigadores a la hora de comparar sus resultados. Además, las instancias utilizadas por la mayoría de los investigadores para este problema, incluye la introducción de la distancia máxima de una ruta determinada. Esta variable ha sido también incluida dentro de la metodología propuesta.

1.5.4. Rutas de vehículos con flotas heterogéneas (HVRPB).

En el contexto de las compañías que cooperan y comparten clientes y vehículos, es muy posible que las flotas no sean homogéneas. Por tanto, habrá que incluir dentro de las decisiones de rutas, cuál es la mejor utilización de la flota heterogénea que va a dar servicio a la nueva relación cooperativa. Esto lo hacemos resolviendo el problema de las rutas de vehículos con retornos y flota heterogénea (*Heterogeneous fleet Vehicle Routing Problem with Backhauls* o HVRPB) en un entorno que hace posible que se resuelvan ambos problemas de retornos enumerados anteriormente, con agrupación y sin ella. Consideramos la flota heterogénea cuya configuración puede ser:

- Fija.
- Configurable.

Nuestro algoritmo es capaz de solucionar ambos problemas. Por defecto se soluciona el de la flota heterogénea fija ya que en su proceso iterativo se selecciona el número de rutas (que corresponde con el número de vehículos de ese tipo existente) de forma secuenciada. Para el caso en que la flota es configurable, se añade una nueva variable aleatoria que obtiene un valor entre 1 y el número de rutas obtenidas en el problema particular. En el caso de que el número de vehículos de cada tipo sea configurable, cuando se selecciona el último tipo de vehículos, el total de las rutas obtenidas para el sub-problema pasarán a formar parte de la solución final. En este caso el problema recibe la denominación de tamaño de la flota y composición en problemas de rutas de vehículos con retornos (*Fleet Size and Mix in Vehicle Routing Problem with Backhauls* o FSMVRPB).

El objetivo del VRPB con flota heterogénea tiene que ver con la mejor utilización de la flota heterogénea considerando que los costes de los vehículos son de naturaleza fija y variable. Los costes fijos se asignan directamente cuando un determinado tipo de vehículo es utilizado (independientemente de las características de la ruta), mientras que los costes variables normalmente dependen del tipo de vehículo y de la longitud de la ruta. El objetivo es la minimización de ambos costes necesarios para entregar y recoger los productos a los clientes. Como se ha comentado antes, esta es una situación habitual en la mayoría de los casos prácticos.

Haciendo el razonamiento en pasiva, es muy difícil (o quizá imposible) encontrar flotas cuyos costes fijos y variables sean los mismos para todos sus vehículos.

1.6. Métodos de resolución.

Los métodos de resolución son por un lado los algoritmos exactos que garantizan obtener una solución óptima al problema y por el otro los algoritmos aproximados, que obtienen soluciones casi-óptimas en tiempos de computación cortos o por lo menos aceptables.

1.6.1. Exactos.

Los métodos de solución exactos aplicados a problemas de rutas de vehículos utilizan métodos y paradigmas bien conocidos en la optimización combinatoria, como por ejemplo los algoritmos de ramificación y acotación (*Branch and Bound* o B&B) (Toth y Vigo, 1997), ramificación y corte (*Branch and Cut* o B&C) (Ropke et al., 2007), y ramificación y precio (*Branch and Price* o B&P) (Dell'Amico et al., 2006). En algoritmos de ramificación y acotación, primero se resuelve la relajación correspondiente de la programación lineal (*Linear Programming* o LP) al problema inicial formulado como un problema de programación entera (*Integer Programming* o IP). La solución de la relajación LP proporcionará un límite inferior o superior para la solución del IP original según el problema sea de minimización o maximización. Si la solución del LP es un número entero, se ha encontrado la solución óptima al IP original. De lo contrario, se comienza con la ramificación generando dos nodos secundarios a partir de la raíz. En cada nodo hijo, se resuelve un nuevo LP con una restricción adicional, estableciendo un límite superior o un límite inferior en una de las variables que se supone que son enteros pero que están asociados con un valor no entero (por tanto, inválido) en la solución actual. En las iteraciones posteriores cada nodo secundario sirve como nodo padre para dos nuevos nodos secundarios en el árbol. Los límites óptimos se obtienen mediante los valores óptimos de solución para el LP en los nodos del árbol de búsqueda. Si el límite inferior en algún nodo del árbol de búsqueda es mayor que el límite superior obtenido en otro nodo, el nodo anterior puede excluirse de la búsqueda.

La mayoría de los problemas de optimización se pueden simplificar reduciendo o relajando las restricciones complicadas. Existen distintas técnicas de relajación, la más conocida es la

relajación lagrangiana (Held y Karp, 1970, 1971) que consiste en relajar las restricciones complicadas en base a penalizaciones en la función objetivo por su no cumplimiento.

Otro tipo de relajación es la descomposición de Dantzig-Wolfe (Dantzig y Wolfe, 1960; 1961) que se basa en descomponer el problema con estructura diagonal en bloques de subproblemas más sencillos. Se dice que un problema tiene estructura diagonal en bloque si existe un pequeño grupo de restricciones en que están incluidas todas las variables de decisión y el resto de restricciones se pueden dividir en subgrupos que contienen subconjuntos disjuntos de variables.

Un método alternativo es el denominado algoritmo del plano de corte. En el caso de que la solución obtenida al problema relajado (LP para un IP) no sea entera, se genera un plano de corte que separa la solución óptima al problema relajado del conjunto factible real. La elección de este plano de corte permite que sólo se considere un subconjunto de las restricciones originales en la relajación LP. De esta forma, se obtiene una sucesión de relajaciones de mejor calidad en cada paso obteniendo cotas inferiores los más ajustadas a la solución óptima (Gomory 1958).

Esta idea es la utilizada en los algoritmos llamados de Ramificación y Corte. Los llamados algoritmos de separación son los encargados de determinar los planos de corte y verificar la solución de la relajación LP en el nodo actual del árbol de búsqueda para las restricciones que se han omitido. En caso de que se cumplan todas las restricciones omitidas, se ha encontrado una solución óptima al LP. De lo contrario, la restricción que no se cumple se agrega al LP y se resuelve de nuevo. Esto se repite hasta que los procedimientos de separación no detecten restricciones incumplidas.

El algoritmo de ramificación y precio se puede clasificar dentro de los algoritmos de ramificación y acotación Utiliza el método de generación de columnas teniendo como principio básico que no es necesario considerar todas las columnas a la vez. La idea es buscar de manera eficiente columnas que pueden ser omitidas (y por tanto haciendo la búsqueda más sencilla), teniendo en cuenta la información del problema dual asociado. El término columna se considera en lo referente a columnas de la matriz tecnológica de las restricciones del problema y en particular a rutas dentro de los problemas de rutas de vehículos.

Por último, la programación dinámica (*Dynamic Programming* o DP) es un método algorítmico muy poderoso en el que un problema se resuelve identificando una colección de subproblemas y tratándolos uno por uno. Los problemas más pequeños se tratan en primer lugar y se utilizando sus resultados para resolver los más grandes, hasta que el problema se ha resuelto (Xumari, 1967; Brimberg y Love, 1998).

1.6.2. Heurísticos.

Los métodos heurísticos en el campo de las rutas de vehículos pueden dividirse en heurísticos y metaheurísticos clásicos. Los métodos heurísticos clásicos comprenden:

- Heurísticos de construcción.
- Heurísticos de dos fases.
- Heurísticos de mejora.

Los *heurísticos de construcción* construyen una solución factible tratando de mantener el valor de la función objetivo lo más óptima posible (máxima o mínima). Se va construyendo la solución insertando cada cliente de forma secuencial o paralela. Los heurísticos secuenciales sólo consideran una ruta a la vez, mientras que los heurísticos paralelos consideran múltiples rutas a la vez. Un criterio de inserción utilizado es el de la inserción más barata. El elemento que se inserta es el cliente con el menor aumento de coste con respecto a la solución parcial construida hasta ese momento.

Los *heurísticos de dos fases* describen pueden ser de dos tipos, primero-agrupamiento-después-ruteo (*cluster-first-route-second*) y primero-ruteo-después-agrupamiento (*route-first-cluster-second*). Tienen dos fases, una en la que se realizan las agrupaciones, y la otra en la que se realizan las rutas. Las diferencias están determinadas por el orden en que se hacen ambas.

Los *heurísticos de mejora* pueden considerar cada ruta de vehículo por separado (intra-ruta) o varias al mismo tiempo (inter-ruta). Entre los primeros, lo más populares utilizan el mecanismo λ -opt de Lin (1965) (especialmente 2-opt y 3-opt). Esto se consigue reubicando secuencias de 3, 2 y 1 vértices de la forma más óptima posible. Los heurísticos del segundo

tipo realizan cambios extra-ruta entre dos o más rutas distintas como los propuestos por [Xu y Kelly \(1996\)](#) y [Gendreau et al. \(1992\)](#).

Una mención aparte merece los mecanismos heurísticos que utilizan métodos aleatorios dentro de su proceso de búsqueda. Parten de un heurístico perteneciente a alguno de los grupos explicados en esta sección y asignan aleatoriedad a alguna parte de su funcionamiento, convirtiéndose en no deterministas. La aplicación de la aleatoriedad necesita del encapsulamiento del heurístico dentro de un esquema de búsqueda como se aplica en los resultados obtenidos en esta tesis doctoral o en cualquier otro. Esta aleatoriedad puede ser de dos tipos, uniforme o no uniforme. En los capítulos 4, 5 y 6 de esta tesis se utilizarán algoritmos de este tipo y serán presentados específicamente en la última sección del capítulo actual.

1.6.3. Metaheurísticos.

Una de las primeras ocasiones en que se utilizó el término metaheurístico fue en la introducción de un heurístico de búsqueda tabú en el trabajo de [Glover \(1986\)](#). [Osman y Kelly \(1996\)](#) definen un algoritmo metaheurístico en los siguientes términos: “Los algoritmos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria en los que los heurísticos clásicos no son efectivos ni eficientes. Los metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y la mecánica estadística”. Todos ellos tratan de combinar adecuadamente dos fases:

- Intensificación en la búsqueda para seleccionar estados próximos al actual que mejoren la solución obtenida.
- Diversificación en la búsqueda para orientar la búsqueda hacia regiones que permitan escapar de mínimos locales.

Un metaheurístico va un paso más allá de un heurístico clásico y contiene alguna estructura meta que permite dirigir la búsqueda y escapar de óptimos locales. Durante el proceso se permiten soluciones intermedias infactibles o que empeoren soluciones alcanzadas

anteriormente. Esta característica permite al algoritmo la navegación por el espacio de estados para transformar soluciones infactibles en factibles que en otras circunstancias no hubieran sido consideradas. Esta característica de lanzamiento y evaluación de heurísticos hace que tiendan a necesitar tiempos de cálculo más largos.

Un esquema elemental de algoritmo metaheurístico es el de la estructura de búsqueda llamada de multi-arranque (Boese et al. 1994). Consiste en configurar un esquema de búsqueda (normalmente con un heurístico clásico) al que se ha añadido aleatoriedad, de tal forma que cada ejecución-arranque obtenga soluciones distintas. El resultado estará formado por la mejor solución obtenida entre las distintas ejecuciones-arranques lanzados. Este esquema será mostrado en el capítulo 4 para dirigir el lanzamiento/análisis del resto de elementos que se utilizan. Existen enfoques metaheurísticos basados en la generación de soluciones, como los algoritmos genéticos (Goldberg, 1989; Reeves, 2010) o las colonias de hormigas (Dorigo, 1992; Denebourg et al., 1983). y, por otro lado, existen métodos que se basan en diferentes formas de búsqueda local, como búsqueda tabú (Glover 1989, 1990), búsqueda por vecindarios (Ropke y Pisinger, 2006), cristalización simulada (Kirkpatrick, 1983; Tovey 1988) o búsqueda local iterada (Lourenço et al. 2010). Este último enfoque se utilizará en el capítulo 3 de esta tesis.

Dos soluciones son vecinas si se puede obtener una a partir de la aplicación de un determinado operador de búsqueda local a la otra. En algoritmos genéticos, cada nueva población de soluciones se obtiene a partir de la anterior utilizando operadores como la recombinación y mutación. De este proceso de búsqueda local se mantienen las mejores soluciones. En las colonias de hormigas, se utiliza un heurístico constructivo para obtener nuevas soluciones utilizando la información obtenida en iteraciones anteriores. En la búsqueda tabú se pone énfasis en las operaciones que transforman una solución en otra. En caso de que una operación haya sido satisfactoria convirtiendo una solución en otra mejor, su operación inversa es considerada como tabú durante un número de iteraciones determinado. De esta forma, se prohíbe su utilización ya que es la inversa de una operación exitosa. En el esquema de búsqueda tabú, la nueva solución puede ser peor que la solución anterior. Este concepto permite a la búsqueda escapar de óptimos locales. En la búsqueda de vecindario variable se consideran varios vecindarios de diferente tamaño de tal forma que cada uno contiene al anterior. Puede considerarse como una descomposición del problema, pero en este caso cada sub-problema contiene al anterior. Un vecindario puede estar formado por todos los posibles

movimientos de un determinado vértice. El vecindario inmediatamente superior puede consistir en todos los movimientos factibles de dos vértices y así sucesivamente. En cada iteración se construye una solución en el vecindario actual de forma aleatoria que se intenta optimizar con algún heurístico de mejora. Si es así, se convierte en la solución actual y se continua con el vecindario inmediatamente inferior. Si no, se considera el siguiente vecindario más grande y se construye otra solución aleatoria. En el cristalizado simulado en cada iteración se construye una solución aleatoria en un vecindario determinado. En caso de que esta solución sea mejor que la actual, se convierte en la nueva solución establecida. De lo contrario, se acepta con cierta probabilidad. Esta probabilidad depende del valor de solución de la nueva solución y de un parámetro llamado temperatura que intenta asemejar el proceso al de una cristalización. Este parámetro de temperatura disminuye durante la búsqueda de acuerdo con un cierto esquema. Cuanto más fresco se obtiene, menor es la probabilidad de que una nueva solución deteriorada sea aceptada como la nueva solución.

1.6.4. Matheurísticos.

En los últimos años, se ha dedicado mucha atención a la integración, o hibridación, de metaheurísticos con métodos exactos. Se puede decir que los matheurísticos son algoritmos heurísticos compuestos por la interacción de metaheurísticos y técnicas de programación matemática como se ve en la [Figura 1.7](#).

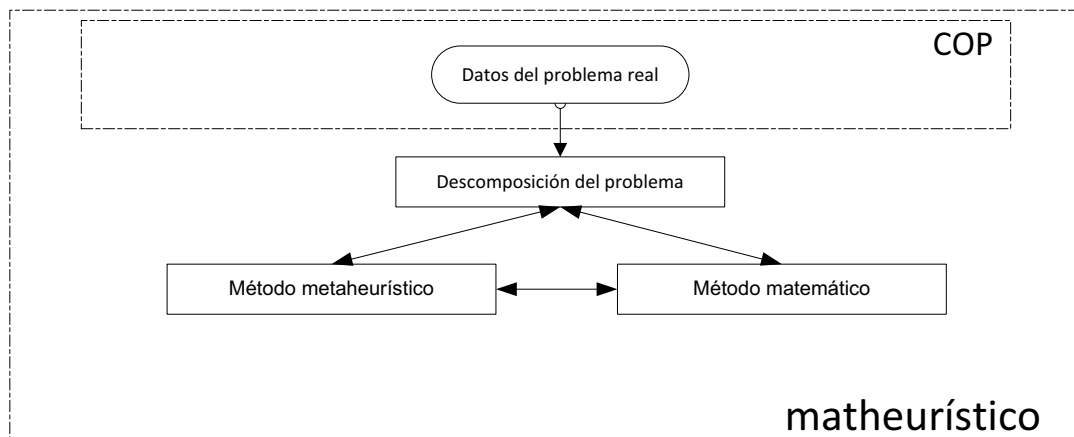


Figura 1.7: Esquema de un algoritmo matheurístico.

Podemos clasificar los matheurísticos en tres clases:

- Descomposición En general, en un enfoque de descomposición el problema se divide en subproblemas más pequeños y más simples y se aplica un método de solución específico a cada sub-problema. Algunos o todos estos subproblemas se resuelven a través de modelos de programación matemática.
- Mejora. En este caso se utilizan modelos de programación matemática para mejorar una solución obtenida con un heurístico. Son muy comunes ya que pueden aplicarse cualquier heurístico para obtener una solución que el modelo de programación matemática pretende mejorar.
- Exactos. Están basado en los métodos de ramificación y precio. Los algoritmos de ramificación y precio se han utilizado ampliamente y con éxito para solucionar problemas de rutas de vehículos. En los matheurísticos basados en estos algoritmos, el método exacto se modifica para acelerar la convergencia, perdiendo así la garantía de optimalidad.

Se recomienda la interesante revisión de la literatura sobre matheurísticos del artículo de [Archetti y Speranza \(2014\)](#).

1.6.5. Simheurísticos.

Existe un interés creciente en la introducción de la aleatoriedad en problemas de optimización combinatoria como una manera de modelar nuevos problemas reales, en los que parte de la información y los datos pueden no ser conocidas de antemano. Estos problemas representan la porción estocástica de los COPs definidos hasta ahora donde se incluyen variables no deterministas. Este grupo de problemas es conocido como problemas estocásticos de optimización combinatoria (*Stochastic Combinatorial Optimization Problem* o SCOP).

El enfoque simheurístico ([Figura 1.8](#)) es un caso particular de simulación de optimización que combina un algoritmo heurístico o metaheurístico con metodologías de simulación con el fin de tratar eficientemente las dos vertientes de un problema tipo SCOP:

- Optimización.
- Estocástica.

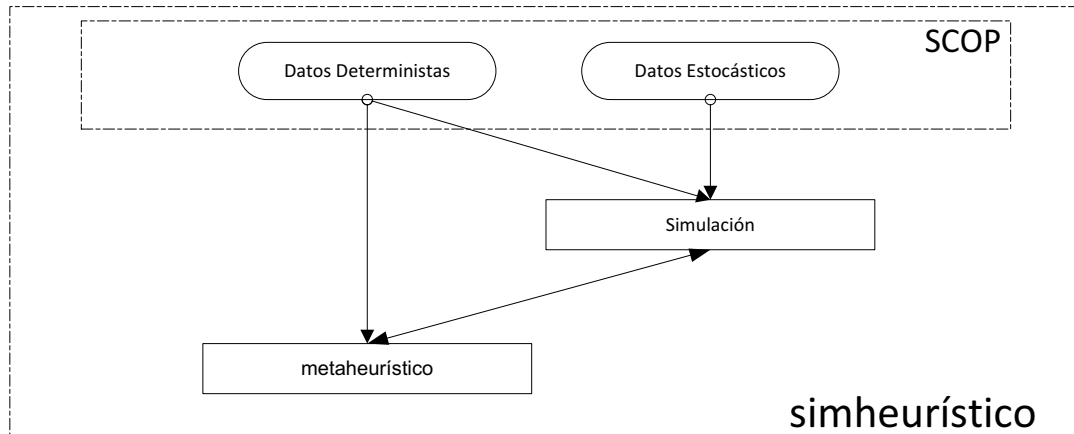


Figura 1.8: Esquema de un algoritmo simheurístico.

Normalmente, dado un modelo SCOP, se ejecuta un algoritmo metaheurístico para realizar una búsqueda orientada dentro del espacio de soluciones. Este proceso iterativo tiene como objetivo encontrar soluciones factibles con el mejor valor posible. Durante el proceso de búsqueda iterativo, el algoritmo debe tratar con la naturaleza estocástica de la instancia SCOP. Una manera natural de hacerlo es aprovechar las posibilidades que ofrecen los métodos de simulación para manejar la aleatoriedad. También se pueden usar otros enfoques en lugar de simulación, e.g. programación dinámica, lógica difusa, etc. Sin embargo, bajo la presencia de datos históricos sobre el comportamiento estocástico, la simulación permite el desarrollo de modelos precisos y flexibles. Específicamente, la aleatoriedad puede ser modelada a lo largo de una distribución de probabilidad sin ninguna suposición o restricción adicional. Por lo tanto, la simulación puede estar integrada con el enfoque metaheurístico y proporcionar retroalimentación dinámica al proceso de búsqueda para mejorar el resultado final. En cierto sentido, la simulación permite extender metaheurísticos ya existentes y eficientes, inicialmente diseñadas para hacer frente a problemas deterministas, para que también puedan ser empleados para resolver problemas estocásticos.

En la práctica, el objetivo de la simulación se orienta principalmente hacia:

- Estimar el valor de coste esperado de una nueva solución siempre que la función objetivo contenga algún elemento estocástico.
- Comprobar que una solución satisface algunas limitaciones probabilísticas.

Un inconveniente de este enfoque es que los resultados no se espera que sean óptimos, ya que este enfoque combina dos metodologías aproximadas. Esto es algo que también sucede en los problemas de la vida real que no suelen ser sencillos y normalmente NP-complejos incluso en sus versiones determinísticas. Los algoritmos simheurísticos constituyen una alternativa para la mayoría de los casos prácticos, ya que establecen métodos relativamente sencillos y flexibles que son capaces de proporcionar soluciones casi óptimas a problemas complejos de la vida real en tiempos de computación competitivos.

Ejemplos de aplicaciones de algoritmos simheurísticos en diferentes campos se pueden encontrar en la literatura de optimización-simulación. Por ejemplo, [Juan et al. \(2011b\)](#) combinan simulación de Monte Carlo (*Monte Carlo Simulation* o MCS) con metaheurísticos para resolver el problema de las rutas de vehículos con demandas estocásticas (*Vehicle Routing Problem with Stochastic Demand* o VRPSD). [Caceres et al. \(2012\)](#) combinan también MCS con un metaheurístico para resolver el problema de las rutas de inventario con roturas de stock y demandas estocásticas.

1.6.6. Aleatorización de heurísticos.

En este capítulo, presentamos el núcleo básico utilizado los algoritmos propuestos en esta tesis. Este elemento común incluido en los métodos para abordar los diferentes modelos, consiste en utilizar la aleatoriedad sesgada en algún momento del proceso de búsqueda. En este capítulo se discute cómo asignar esta aleatoriedad a heurísticos clásicos para transformar su comportamiento determinista en probabilístico. Esta aleatorización se puede realizar usando una distribución de probabilidad uniforme o, utilizando una distribución no simétrica.

El objetivo principal de esta sección es explicar un esquema híbrido que combine un heurístico clásico con procesos de aleatoriedad. Como se verá más adelante, este esquema híbrido representa una manera eficiente, relativamente simple y flexible de tratar con los COPs planteados en este trabajo de investigación. Utilizaremos el término método de búsqueda

aleatoria o simplemente algoritmo aleatorio para referirnos a cualquier algoritmo que haga uso de números pseudoaleatorios para realizar elecciones aleatorias durante la exploración del espacio de solución.

En el contexto de la optimización combinatoria, los algoritmos aleatorios hacen uso de números pseudoaleatorios durante las fases de construcción o búsqueda local. Además de los datos de entrada del problema, un algoritmo aleatorio utiliza números aleatorios para hacer elecciones durante su ejecución. Dada la estructura aleatoria de algún elemento del proceso de búsqueda podemos decir que el camino seguido no es determinista, y por tanto puede producir salidas diferentes en diferentes ejecuciones. Dentro de estos tipos de algoritmos podemos incluir los clasificados como metaheurísticos de la sección anterior de este capítulo: algoritmos genéticos (Reeves, 2010), Procedimiento de búsqueda adaptativa aleatoria voraz (*Greedy Randomized Adaptive Search Procedure* o GRASP) (Resende and Ribeiro, 2010), Cristalizado simulado (Nikolaev y Jacobson, 2010), búsquedas en vecindarios amplios (Ropke y Pisinger, 2006), búsqueda local iterada (Lourenço et al., 2010), colonias de hormigas (Gajpal y Abad, 2009; Dorigo y Stutzle, 2010).

Uno de los métodos más populares de búsqueda aleatoria es GRASP (Festa y Resende, 2009a, 2009b; Resende y Ribeiro, 2010). GRASP es un proceso iterativo que utiliza una lista restringida de candidatos (los n mejores) y elige uno de ellos dependiendo de su aleatorización uniforme (Algoritmo 1.1). En cada iteración se ejecutan dos fases:

- Fase constructiva, que genera una solución convirtiendo un heurístico conocido en aleatorio.
- Fase de búsqueda local, con el propósito de mejorar la solución anterior.

Al final de este proceso constructivo se almacena la mejor solución obtenida.

Algoritmo 1.1 Pseudocódigo del algoritmo GRASP.

Procedure GRASP(datos)

```
01   while {no-se-cumple criterio_de_parada} do
02       solucion<-ConstruirSolucionGreedyAleatorizada(datos)
03       solucion<-AplicarBusquedaLocal(solucion)
```

```

04     MejorSolucion<-ActualizarMejorSolucion(solucion)
05     end while
06     return MejorSolucion
end procedure

```

Este esquema sirve de forma equivalente para nuestra propuesta de aleatorización sesgada. La mayoría de los trabajos publicados sobre algoritmos aleatorios están basados en el uso de números aleatorios uniformes, i.e., se genera el número aleatorio a través de una distribución de probabilidad uniforme simétrica. Pero esta aleatoriedad también puede asignarse a través de cualquier otro tipo de distribución, en particular la sesgada. Por lo que sabemos, el primer planteamiento de este tipo utilizando una aleatorización sesgada es el presentado por **Bresina (1996)** con su metodología heurístico de casos de aleatoriedad sesgada (*Heuristic-Biased Stochastic Sampling* o HBSS). Más recientemente, **Juan et al. (2011a)** propusieron el uso de distribuciones de probabilidad no simétricas para convertir un heurístico clásico en uno aleatorio con un planteamiento llamado aleatorización sesgada de heurísticos clásicos con esquema de búsqueda local adaptativa (*Multi-start biased Randomization of classical Heuristics with Adaptive local search framework* o MIRHA).

Los aspectos comunes de este planteamiento y GRASP son la construcción de una solución inicial utilizando aleatorización y a continuación la aplicación de un proceso de búsqueda local. Pero hay algunas diferencias importantes:

- MIRHA no utiliza una lista reducida de candidatos, que es una de las principales características de GRASP.
- MIRHA utiliza una distribución no simétrica para seleccionar el siguiente elemento de la solución mientras que la mayoría de las implementaciones de GRASP consideran distribuciones uniformes.

En general, el proceso general de aleatorización de un heurístico consiste en primer lugar en seleccionar un heurístico determinista y constructivo con las siguientes características:

- Debe ser capaz de ejecutarse muy rápido incluso para problemas largos. Esto es crítico ya que los algoritmos probabilísticos deben ejecutarse repetidas veces para encontrar soluciones adecuadas.

- En su diseño, debe obtener algún estado que pueda ser aleatorio, e.g., una lista priorizada a través de alguna función voraz (*greedy*).
- Debe obtener buenas soluciones que no estén muy lejos de las obtenidas con otros heurísticos más complejos y que requieren más tiempo de ejecución, e.g., gap alrededor 5-10%.

Una vez que tenemos seleccionado el heurístico al que se va a dotar de aleatoriedad, el algoritmo puede seguir algún tipo de proceso multi-arranque. En cada paso de este proceso se debe generar una solución completa. Para la construcción de esta solución, se convierte el heurístico base en aleatorio utilizando una determinada distribución de probabilidad. La distribución que se utilice dependerá del problema particular, e.g. geométrica, triangular descendente. Adicionalmente se puede añadir un proceso de búsqueda local en cada paso del proceso de muti-arranque (Boese et al. 1994). De esta forma podemos generar de forma rápida muchas soluciones posibles con diferentes características. Por tanto, el que tiene que tomar la decisión puede hacerlo considerando varias soluciones y eligiendo la más adecuada según sus necesidades. Asimismo, el proceso de ajuste de parámetros es muy sencillo ya que las distribuciones de probabilidad utilizada contienen sólo uno (e.g geométrica) o cero parámetros (e.g. triangular descendente).

La principal motivación detrás del diseño de un heurístico de aleatoriedad sesgada es satisfacer muchas de las características deseables según lo descrito por Cordeau et al. (2002), es decir: precisión, velocidad, sencillez y flexibilidad. La mayoría de los metaheurísticos de la literatura se miden en función de la precisión (el grado de proximidad entre la solución obtenida y el valor óptimo) y de la velocidad (el tiempo de cálculo). Sin embargo, hay otros dos atributos importantes que deben considerarse en cualquier método de optimización: simplicidad y flexibilidad. La simplicidad está relacionada con el número de parámetros que se deben establecer y la facilidad de implementación. Esta es una característica importante ya que el método puede aplicarse a instancias diferentes de las iniciales sin perder calidad o rendimiento y sin necesidad de realizar adaptaciones complejas. La flexibilidad está relacionada con la posibilidad de incluir nuevas restricciones y también con la adaptación a otros problemas similares. Teniendo en cuenta todos esto, estos son los principales beneficios de los heurísticos con aleatoriedad sesgada:

- Permiten una simplificación del proceso de ajuste de parámetros, ya que normalmente las distribuciones de probabilidad empleadas requieren solo uno (por ejemplo, el geométrico) o cero parámetros (por ejemplo, la triangular descendente). Esto no es común en la mayoría de los enfoques metaheurísticos actuales, que usualmente emplean muchos parámetros y, por lo tanto, requieren de procesos de ajuste difíciles y costosos en el tiempo para ajustar sus parámetros.
- Basándose en heurísticos clásicos bien probados, son métodos relativamente sencillos y fáciles de implementar, que pueden adaptarse para añadir nuevas restricciones (flexibilidad). Por lo tanto, cuando se enfrenta a un problema de optimización combinatoria con heurísticos ya existentes, algunos de los heurísticos más eficientes pueden ser seleccionados y mejorados a través de la aleatoriedad sesgada.
- Usando distribuciones no uniformes (sesgadas) en vez de uniformes, ofrecen una manera más natural y eficiente de seleccionar el siguiente movimiento de la lista de prioridad, ya que la asignación de aleatoriedad sesgada permite mantener la lógica detrás del heurístico asignando más probabilidades de ser seleccionado para aquellos movimientos que mejor cumplen el heurístico. Contrariamente una aleatorización uniforme no respeta el sentido común del heurístico, ya que asigna iguales probabilidades para todos los movimientos potenciales.
- Al combinar la aleatorización con un proceso de arranque múltiple, promueven la diversificación durante la exploración del espacio de soluciones, es decir, la búsqueda no se limita a un número reducido de regiones (Talbi, 2009). Observe que estas dos características aleatoriedad y enfoque multi-arranque ayudan al algoritmo a escapar de los mínimos locales y también aumentan la diversificación durante la exploración del espacio de soluciones.

Del mismo modo, la combinación de la aleatorización con un proceso de arranque múltiple permite la ejecución en paralelo de manera fácil y natural, ya que, en cada arranque del proceso, la solución obtenida es independiente de la anterior. Hay que tener en cuenta que las probabilidades de encontrar buenas soluciones aumentan a medida que se ejecutan más ejecuciones paralelas del algoritmo.

1.7. Marco de la tesis.

1.7.1. Proyectos de investigación.

La tesis doctoral que se presenta se engloba dentro del marco de los proyectos:

- [TRA2015-71883-REDT](#)

Transporte, logística y producción inteligente.

Financiado por el Plan Nacional de I+D+i. Ministerio de Economía y Competitividad (Madrid, España).

Fecha de inicio: 01-ene-2016.

Fecha de fin: 31-dic-2017.

- [CAN2015-70473](#)

COOP-Transporte Sostenible-2: Logística Inteligente en la gestión sostenible del transporte por carretera.

Financiado por la Fundación CAN (Pamplona, España).

Fecha de inicio: 01-ene-2015.

Fecha de fin: 30-jun-2016.

- [CYTED2014-P514RT0013](#)

Red Iberoamericana de logística inteligente en la gestión sostenible del transporte en núcleos urbanos.

Financiado por CYTED (Madrid, España).

Fecha de inicio: 01-ene-2015.

Fecha de fin: 30-dic-2018.

- [TRA2013-48180-C3-1-P](#)

Cooperación horizontal y costes medioambientales en la gestión sostenible del transporte de mercancías.

Financiado por el Plan Nacional de I+D+i. Ministerio de Economía y Competitividad (Madrid, España).

Fecha de inicio: 01-ene-2014.

Fecha de fin: 31-dic-2017.

La sostenibilidad se presenta como uno de los principales problemas de la gestión del transporte terrestre. Esta situación es significativa, principalmente en el estudio del transporte por carretera y la implantación por parte de la Unión Europea de las técnicas de pago por impacto medioambiental del transporte. Todo ello se quiere estudiar en estos proyectos. Los principales entregables de estos proyectos, están directamente ligados a sus objetivos:

- Desarrollo de nuevos modelos matemáticos, métodos híbridos y software en abierto que promuevan la cooperación horizontal sostenible entre las empresas de transporte y logística.
- Desarrollo de métodos y algoritmos híbridos con el objetivo de hallar soluciones a problemas de enrutamiento que se puedan aplicar en la gestión sostenible de flotas de transporte reales.
- Integración de tecnologías y herramientas. Los desarrollos anteriormente citados se integrarán con las tecnologías disponibles en la actualidad en una plataforma inteligente de apoyo a la toma de decisiones (*Decision Support System* o DSS) con el objetivo de facilitar la gestión de las flotas de transporte bajo criterios de eficiencia económica y sostenibilidad medioambiental.

La tesis doctoral que se presenta está enfocada a conseguir los dos primeros objetivos.

1.7.2. Grupo DECYL de la Universidad Pública de Navarra.

El Grupo DECYL (Datos, Estadística, Calidad y Logística), se creó en el año 1994 dentro del departamento de Estadística e Investigación Operativa de la Universidad Pública de Navarra, España. Su labor investigadora se desarrolla a través de las siguientes líneas de investigación:

- Logística y transporte.
- Modelos de simulación y optimización.
- Minería de datos.
- Probabilidad y procesos estocásticos.
- Fiabilidad y control estadístico de la calidad.
- Diseño y análisis de encuestas.
- Optimización de rutas de vehículos de distribución en logística y transporte. problemas medioambientales.
- Probabilidad y procesos estocásticos. aplicaciones. Diseño óptimo de experimentos.
- Métodos estadísticos multivariantes aplicados a estudios sociales, económicos y de marketing.
- Simulación y optimización basada en la simulación. Aplicación en salud, industria y servicios.
- Métodos cuantitativos para la toma de decisiones en sistemas de salud, de energía y de producción.
- Estadística algebraica.

En los últimos años, el grupo ha publicado más de 100 artículos internacionales, 50 contribuciones en conferencias internacionales y sus miembros son responsables o colaboran en varios proyectos de investigación financiados por el Gobierno de España. Además, el grupo ha colaborado, mediante contratos de investigación, con importantes empresas nacionales y multinacionales.

1.7.3. Grupo DPCS-ICSO de la Universitat Oberta de Catalunya.

El equipo de investigación Internet Computing & Systems Optimization (DPCS-ICSO, <http://dpsc.uoc.edu>) en IN3-UOC es reconocido como un grupo de investigación consolidado por el gobierno catalán (2014 SGR 337). En cuanto a las personas que trabajan a tiempo

completo, el DPCS-ICSO está integrado por 5 profesores, 3 investigadores post doctorales y 6 estudiantes de doctorado.

El grupo centra su investigación en las siguientes dos líneas:

- Optimización de sistemas para un mundo más inteligente: desarrollando algoritmos inteligentes y soluciones software para apoyar procesos de toma de decisiones complejas en la logística y el transporte, producción, posicionamiento en tiempo real y ciudades inteligentes.
- Sistemas de colaboración y computación en Internet: desarrollando, analizando y optimizando sistemas no dedicados, colaborativos de dimensión elevada a través de Internet.

El grupo ha publicado más de 50 artículos en revistas indexadas en durante el periodo 2012-2016, incluyendo revistas de prestigio como: European Journal of Operational Research, Applied Soft Computing, Expert Systems with Applications, Transportation Research Part C, IEEE Systems Journal, Computers and Industrial Engineering, ACM Computing Surveys, etc. Asimismo, en el mismo periodo 2012-2016, los profesores del grupo han supervisado 9 tesis doctorales ya finalizadas y está supervisando actualmente otras 9.

En los últimos 5 años, el grupo ha coordinado varios proyectos de investigación, incluyendo dos del plan nacional español de I+D+i (TRA2010-21644-C03-02 y TRA2013-48180-C3-3-P), uno de la red española de excelencia (TRA2015-71883-REDT), dos de la red de investigación latino-americana (CYTED2010-511RT0419 y CYTED2014-P514RT0013), y dos de la red de investigación Española-Francesa (2009-CTP-00007 y 2014-CTP-00001). Igualmente, el grupo coordina un consorcio Erasmus inter-universitario que promociona la movilidad de los estudiantes de master o doctorado desde España a otros países europeos. Además, el grupo ha colaborado en la organización de prestigiosas conferencias internacionales como la conferencia Winter Simulation, la conferencia internacional de metaheurísticos, la conferencia de la OR-Society, o la conferencia de simulación ASIM.

1.8. Organización de la tesis.

El mapa de capítulos de la tesis doctoral se puede ver en la [Figura 1.9](#). En el *Capítulo 1* se realiza una introducción general para situar el trabajo de investigación dentro del ámbito funcional al que pertenece. Este es el de la optimización combinatoria dentro del campo de la Investigación Operativa. Como capítulo introductorio, se explica el ámbito de las operaciones de logística y distribución como marco en que se realizan las aportaciones de esta tesis doctoral. Se introduce asimismo la cooperación entre empresas trabajando al mismo nivel dentro de la cadena de suministro como una vertiente a explorar en la búsqueda de eficiencia en las operaciones empresariales. A continuación, se introducen los conceptos básicos que definen los problemas de optimización combinatoria. Se hace una definición de aspectos algorítmicos, complejidad computacional y métodos de resolución. En particular se definen y detallan los problemas combinatorios conocidos como problemas de rutas de vehículos y se hace una clasificación de las soluciones propuestas para su resolución tanto exacta como aproximada. Los modelos presentados y las metodologías para solucionarlos serán la base de los resultados de investigación obtenidos en este trabajo. A continuación, se exponen los elementos motivadores en forma de proyectos de investigación que originan la necesidad de responder a las preguntas abordadas en este trabajo. Finalmente, se hace una descripción de los grupos de investigación en cuyo entorno se ha desarrollado el presente trabajo.

En el *Capítulo 2*, se hace una revisión del estado del arte relacionado con los elementos presentados en la sección anterior. En primer lugar, se considera el problema general de la cooperación horizontal con las variantes presentadas en la literatura para su aplicación. En segundo lugar, se revisa la literatura en relación al primer problema abordado en este trabajo, el de las rutas con retornos, tanto en su versión agrupada como no agrupada. En tercer lugar, se aborda la revisión de las publicaciones relativas a los problemas que consideran flotas heterogéneas en su versión estratégica, es decir, el tamaño de la flota y su composición son considerados como variables de decisión. Finalmente se realiza un análisis combinado de los puntos anteriores que sirve como justificación del contenido de esta tesis doctoral.

En el *Capítulo 3* se propone un algoritmo eficiente, denominado ILS-BRCW, con tres parámetros, para resolver el problema de rutas de vehículos con retornos agrupados (*Vehicle Routing Problem with Clustered Backhauls* o VRPCB). Este modelo representa la forma más

básica de mezclar los procesos de repartos y abastecimiento. Tras una introducción al tema, se presenta una formulación matemática del problema a resolver. A continuación, se realiza primero una descripción general del algoritmo y posteriormente se describe el pseudocódigo de los programas utilizados. Finalmente, se muestran los resultados obtenidos en los ensayos realizados, sobre las instancias de referencia que se encuentran en la literatura.

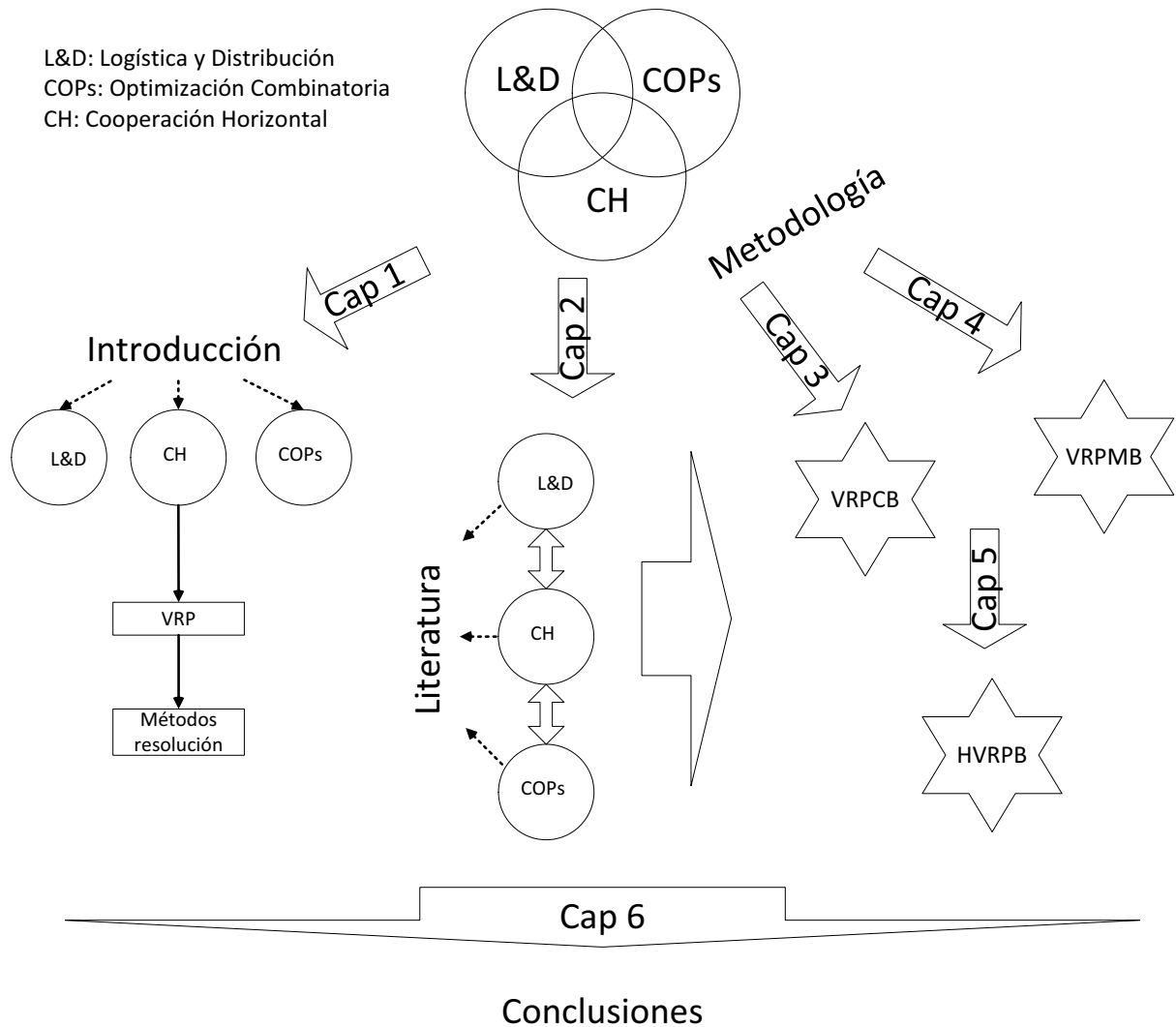


Figura 1.9: Organización de la tesis.

En el *Capítulo 4* se elimina la restricción de precedencia relativa al agrupamiento de los clientes de entrega por un lado y los de recogida por otro. De esta forma, se generaliza el problema anterior y por tanto se enriquece. En esta ocasión, sin embargo, se presenta un metaheurístico híbrido denominado MS-BRCW para tratar la cuestión planteada denominada

problema de rutas de vehículos con retornos mezclados (*Vehicle Routing Problem with Mixed Backhauls* o VRPMB). Este modelo representa el siguiente paso al mostrado en el *Capítulo 3* de mezclar los procesos de repartos y abastecimiento. Con este nuevo proceso se permite que las entregas y las recogidas se realicen en cualquier secuencia. El principal objetivo de plantear este nuevo esquema algorítmico es la simplificación al máximo de la metodología para permitir resolver el problema con un solo parámetro, el que determina la naturaleza aleatoria del heurístico de ahorros embebido en el metaheurístico. Este algoritmo es descrito en detalle y los resultados obtenidos son contrastadas con los mejores algoritmos conocidos.

En el *Capítulo 5* se aborda la extensión de los problemas anteriores considerando una flota de vehículos heterogénea. Podemos decir que la metodología planteada no sólo generaliza los problemas anteriores sino cualquier otro que considere las flotas heterogéneas. El metaheurístico presentado se denomina IHPO. Como se explicará en el capítulo, la razón de esta versatilidad es que nuestro esquema de aproximaciones sucesivas es válido para cualquier heurístico interno que se quiera utilizar. Se ha experimentado con nuestro IHPO utilizando como método de resolución interna el presentado en el *Capítulo 3* pero cualquier otra podría emplearse. Tras una introducción al tema, se hace una descripción del problema a resolver. A continuación, se detalla el algoritmo y el pseudocódigo que lo sustenta. Finalmente, se muestran los resultados obtenidos en los ensayos realizados, sobre las instancias de referencia que se encuentran en la literatura científica. Tras la obtención de estos resultados se puede destacar el excelente comportamiento del algoritmo ya que mejora 20 de las mejores soluciones reportadas hasta la fecha en la literatura. Al final de los *Capítulos 3, 4 y 5* se realizan las conclusiones ensalzando la relación de cada problema con el grado de cooperación horizontal que se alcanza en cada caso.

Finalmente, en el *Capítulo 6*, se presentan las principales conclusiones alcanzadas durante la investigación. A continuación, se incluyen las publicaciones en forma de artículos y congresos que dan soporte a los resultados obtenidos en esta tesis. Finalmente, se describen las líneas de investigación abiertas como consecuencia de los resultados aquí presentados. Algunas de ellas se plantean para futuras investigaciones y otras constituyen ya el presente como continuación del proceso que comienza con la redacción de esta tesis doctoral.

1.9. Objetivos de investigación.

El objetivo de esta tesis doctoral es el desarrollo e implementación de algoritmos eficientes que contribuyan al conocimiento y evaluación de las mejores prácticas de cooperación horizontal en el campo de la logística y la distribución utilizando los modelos del problema de rutas de vehículos (*Vehicle Routing Problem* o VRP). Tras la revisión inicial de la literatura existente, se ha elegido el modelo de las rutas de vehículos con retornos y flota heterogénea.

La razón por la que se han elegido ambos modelos es porque nos sirven para evaluar las economías de escala alcanzadas al integrar los procesos de negocio que van a servir para cooperar entre las empresas. Los procesos elegidos son los de reparto o entrega y recogida o abastecimiento. Y la práctica de cooperación elegida ha sido la de integrar ambos procesos en uno solo. Este modelo representa el caso más básico de cooperación entre dos o más empresas en el ámbito de la logística y la distribución. Los participantes que cooperan de esta forma comparten clientes y flota en su proceso de reparto (clientes de entrega) y abastecimiento (clientes de recogida). El objetivo de investigación inicial consiste en obtener algoritmos eficientes para este modelo particular que incluye varias variantes. Para el caso de las *rutas con retornos*, se consideran dos variantes:

- Los clientes de recogida se visitan cuando se han visitado todos los clientes de entrega en el problema conocido como problema de rutas de vehículos con retornos agrupados (*Vehicle Routing Problem with Clustered Backhauls* o VRPCB).
- Los clientes de recogida y entrega se pueden visitar en cualquier secuencia en el problema conocido como problema de rutas de vehículos con retornos mezclados (*Vehicle Routing Problem with Mixed Backhauls* o VRPMB).

En la primera sólo se permite un grupo de cada tipo de clientes en cada ruta mientras que en la segunda puede haber tantos grupos de cada tipo como sea necesario para la mejor optimización de la ruta.

Para el caso de la *flota heterogénea*, se considera la variante en la que el tamaño de la flota es una variable de decisión, así como su composición. Se partirá de un grupo determinado de tipos de vehículos asumiendo un número ilimitado de cada uno de ellos. El resultado estará formado por la composición de la flota que minimice los objetivos de costes. El problema general considerando la flota heterogénea es conocido como problema de rutas de vehículos con retornos y flota heterogénea (*Heterogeneous Vehicle Routing Problem with Backhauls* o HVRPB). En particular, si el número de vehículos es ilimitado, se conoce como tamaño y composición de la flota en problemas de rutas de vehículos con retornos (*Fleet Size and Mix Vehicle Routing Problem with Backhauls* o FSMVRPB).

Preguntas de investigación:

(P1) ¿Qué conceptos/procesos logísticos se pueden utilizar para permitir la optimización de procesos según las prácticas de cooperación horizontal?

(P2) ¿Qué modelos de rutas de vehículos son de aplicación para facilitar la cooperación horizontal?

(P3) ¿Es posible aplicar algoritmos heurísticos de aleatoriedad sesgada para resolver los problemas descritos?

(P4) ¿Cuál es el grado de eficiencia de estos algoritmos?

Estos algoritmos estarán basados en la aleatoriedad sesgada. Para cada uno de los problemas planteados, se dedica un capítulo que contiene la definición formal, la metodología propuesta y la explicación del pseudocódigo de alto nivel del algoritmo utilizado. Este código de alto nivel se implementará como una aplicación y se medirá su comportamiento sobre un conjunto de instancias conocidas para cada problema. Los resultados serán analizados con herramientas estadísticas y comparados con el comportamiento de los algoritmos reportados en la literatura que obtienen los mejores resultados aplicados sobre las mismas instancias. De esta forma seremos capaces de analizar la calidad de los algoritmos desarrollados.

1.10. Contribuciones originales.

Durante la búsqueda de los objetivos anteriores hemos generado una serie de contribuciones originales. Entre ellas, estas son las más relevantes:

1. **Revisión del estado del arte de las prácticas de cooperación horizontal.** Se ha realizado una rigurosa revisión de la literatura respecto de la cooperación horizontal con el objetivo de responder a la primera pregunta de investigación.
2. **El heurístico de ahorros para el VRPCB:** Hemos definido un heurístico de ahorros basado en Clarke and Wright (CWS) para el CVRP. Este heurístico combina ideas de CWS con la penalización propuesta por Deif and Bodin (1964) y es de ejecución muy rápida.
3. **El algoritmo ILS-BRCW para el VRPCB:** Proponemos un algoritmo con aleatoriedad sesgada en un esquema de búsqueda local iterada (ILS). El algoritmo combina el heurístico del punto dos con dos procesos de mejora local, el de la memoria cache y un procedimiento del tipo decrece (reduce) y vencerás.
4. **El algoritmo MS-BRCW para el VRPMB:** Proponemos un algoritmo con aleatoriedad sesgada de tipo Multi-Arranque. El algoritmo combina la adaptación del heurístico del punto al modelo no agrupado, con dos procesos de mejora local, el de la memoria cache y un procedimiento de descomposición del tipo divide y vencerás.
5. **El algoritmo IHPO para el FSMVRPB:** Proponemos un esquema de acercamientos sucesivos para solucionar el VRPB con flota heterogénea. El algoritmo obtiene una solución con flota heterogénea combinando una estructura recursiva con la invocación de un heurístico interno que soluciona problemas homogéneos.
6. **El operador correctivo recursivo (*Recursive Corrective Operator* o RCO),** evaluador/corrector de rutas para el algoritmo del punto 2. La naturaleza del heurístico de ahorros no permite por su propia construcción la fijación del número de vehículos a utilizar

en la solución del problema. Esto es debido a que el proceso de unión de rutas continua hasta que ya no puedan hacerse más uniones. Gracias a la naturaleza aleatoria del algoritmo propuesto, se propone un operador que evalúa las rutas creadas y permite aceptar o desecharlas según su proximidad con la utilización de la capacidad requerida.

Capítulo 2.

Estado del Arte.

2.1. Introducción.

En esta sección se hace una revisión de la literatura existente en los tres ámbitos principales de esta tesis. Por ello la hemos separado en tres secciones distintas. En la primera se considera el problema general de la cooperación horizontal con el objetivo de encontrar qué variantes del VRP se adaptan en mejor medida para modelar sus mejores prácticas. En la segunda sección, se considera el problema de las rutas con retornos. Se ha elegido esta variante como consecuencia de la revisión de la primera sección y la comprobación de su adecuación a la cooperación horizontal. En la tercera sección se aborda el problema de las flotas heterogéneas por la misma razón explicada para las rutas con retornos. Finalmente se hace un análisis de las razones que han llevado a elegir los contenidos de esta tesis como consecuencia de los tres apartados anteriores.

2.2. Cooperación Horizontal.

En el mercado actual, la supervivencia de las empresas está determinada por su capacidad de mejorar constantemente para adaptarse a los cambios en los mercados y la progresión de sus competidores. Una empresa puede mejorar internamente para hacer más eficientes sus operaciones y también puede hacerlo cooperando con otras empresas que comparten alguna de sus actividades y/o se complementan en otras.

En particular para los procesos de L&D, trasladar un producto o servicio de un lugar a otro, requiere la utilización de una serie de recursos: energía, equipos de transporte, tiempo de trabajo, etc. La eficiencia de un transporte se mide por el número de recursos cualificados que emplee en su realización. El uso de estos recursos se traduce, a su vez, en unos costes económicos y medio ambientales. Por tanto, su optimización forma parte de las actividades de mejora continua que las empresas deben llegar a cabo para cumplir sus objetivos y asegurar su supervivencia.

Son muchas las variables que afectan al coste total de las actividades de L&D y que son por tanto susceptibles de mejorar (Crujisen, 2016). Una posible estrategia que las empresas pueden seguir para mejorar su competitividad es la de colaborar con otras empresas similares aprovechándose así de las economías de escala generadas por esta actividad (Sanchez-Rodrigues et al., 2015). En particular, la optimización de las distancias, los vehículos y la utilización de los mismos representan algunas de las principales variables susceptibles de ser incluidas dentro de las actividades de cooperación entre empresas (Perez-Bernabeu et al., 2015).

La Unión Europea (European Union, 2001) define la Cooperación Horizontal (CH) como el conjunto de actividades concertadas por las compañías que operan al mismo nivel en el mercado. En un mercado tan competitivo como el actual, las reducciones de costes gracias a las economías de escala alcanzadas utilizando estas prácticas constituyen un elemento diferenciador para las empresas. La cooperación con otras compañías representa una importante estrategia para poder ser más eficientes. De una forma más formal, Bahinipati et al. (2009) definieron el término como un acuerdo entre dos o más compañías al mismo nivel en la cadena de suministro para facilitar el trabajo y la cooperación con el objetivo de alcanzar un objetivo común. Además de la optimización de la carga y las rutas, uno de los objetivos de la CH es su contribución a reducir los viajes vacíos de vuelta al depósito en lo que se ha llamado en inglés *deadheading*. Según la Comisión Europea (European Commission, 2011), el 25% de las actividades de transporte por carretera se realizan con la carga vacía. Este objetivo se puede alcanzar introduciendo prácticas de recogidas o retornos (*backhauling*) en la generación y posterior gestión de las rutas (Juan et al., 2014b; Adenso-Diaz et al., 2012, 2014). El *backhauling* consiste en aprovechar los trayectos de los vehículos en su vuelta al depósito para realizar recogidas de materiales a los clientes que lo requieren.

Aunque uno de los principales objetivos de la CH es la reducción de los costes de transporte y aumento en la velocidad de entrega, otro beneficio importante está relacionado con la reducción del impacto medioambiental de las actividades de distribución. La práctica del *backhauling* se considera una de la más básicas, eficientes y fáciles de utilizar para cumplir ambos objetivos (Ubeda et al., 2011). Asimismo, un factor que contribuye a que la carga del vehículo sea eficiente es la adaptación de la flota a la carga a transportar (Dominguez et al., 2016). Esta adaptación consiste en utilizar la capacidad de la flota al máximo respetando al mismo tiempo el resto de restricciones. De esta forma, parece coherente considerar un cierto grado de heterogeneidad en el dimensionamiento de la flota. Hay que tener en cuenta que disponer de una flota heterogénea es muy habitual en la mayoría de situaciones reales. De hecho, cuando se consideran los costes fijos y variables, incluso dos vehículos con la misma capacidad pueden ser considerados como diferentes (heterogéneos) debido a factores como la depreciación, seguros, mantenimiento, conductor, etc.

2.2.1. Relaciones de cooperación.

En esta sección hacemos una revisión de la literatura académica sobre el tema de la CH en general y particularizaremos en el área de Logística y Distribución (L&D).

En el entorno industrial, la cooperación entre empresas puede realizarse de formas muy diversas. Simatupang y Sridharan (2002), hacen una clasificación de los tipos de cooperación según el nivel dentro de la cadena de suministro al que pertenecen los participantes:

- Vertical.
- Lateral.
- Horizontal.

Las relaciones verticales son aquellas que se realizan dentro de la cadena de suministro entre actores de diferentes niveles, con el objetivo de integrar de manera eficiente a proveedores, productores, almacenes, centros de distribución y clientes (Barahmi 2002). La integración debe incluir también coordinación y sincronización de tal forma que el producto es producido y distribuido en la cantidad necesaria, a los lugares que lo requieren y en el momento preciso para minimizar los costes manteniendo los niveles de servicio acordados con los

clientes. Para hacer esto, la gestión de la cadena de suministro busca crear relaciones satisfactorias entre los involucrados que actúan en diferentes niveles de la cadena de suministro para optimizar los costes logísticos y evitar ineficiencias. Esto se consigue fundamentalmente con la buena gestión de la información que es necesaria para una buena gestión de los inventarios y de los transportes. El ejemplo de proceso más conocido de cooperación vertical es el del inventario gestionado por el proveedor (*Vendor Managed Inventory* o VMI). En esta práctica, el proveedor decide los niveles de inventario que debe mantener en las instalaciones del cliente para cumplir con los niveles de servicio acordados con él. El cliente proporciona acceso a la información actualizada sobre los niveles de stock y los consumos producidos para que el proveedor pueda organizar sus envíos de forma sincronizada y óptima. La cooperación lateral es una combinación de actividades que busca sincronizar las prácticas que se realizan horizontal y verticalmente. Esta coordinación se puede realizar a través de una persona u organización que centralice las actividades, [Tavasszy et al. \(2003\)](#).

Las relaciones horizontales se definen como el conjunto de prácticas acordadas entre compañías que operan al mismo nivel dentro de la cadena de suministro. Algunos ejemplos son los centros de consolidación de producción, los grupos de compras, la planificación conjunta de rutas, la compartición de servicios centrales como centralitas, etc.

Considerando las relaciones horizontales, tradicionalmente las interacciones entre empresas del mismo sector y con la misma actividad han estado basadas en la competición ([Bengstoon y Kock, 1999](#)). Pero los conocidos casos de éxito a través de alianzas en diferentes ámbitos han hecho cambiar estos planteamientos competitivos por otros basados en la colaboración y aprovechamiento de las economías de escala. El camino que hay que recorrer para transformar la competición en colaboración es largo y para entender los estados intermedios entre ambas conviene identificarlos y ver sus interacciones. En este sentido, [Bengtsson y Kock \(1999, 2000\)](#) identificaron cuatro tipos de relaciones horizontales:

- Coexistencia donde no hay intercambio económico.
- Cooperación, donde hay una relación estrecha con objetivos comunes.
- Competición, se produce un comportamiento de acción reacción ya que los competidores se siguen entre sí.

- Co-opetición, es una mezcla de las dos anteriores. Los participantes pueden cooperar en algunos aspectos y al mismo tiempo competir en otros.

Una compañía puede jugar uno a varios de estos roles según con qué otra compañía esté interactuando y en qué momento se produzca. Por tanto, el contenido de una relación puede cambiar de competición a cooperación, o coexistencia, o co-opetición según las posiciones relativas de los participantes. Los autores identifican como la más útil para una compañía la de la co-opetición, porque por un lado recibe ayuda en su vertiente cooperativa y por otro lado se ve forzada a desarrollarse en su vertiente competitiva. Por un lado, los actores tienen que competir porque en caso contrario la relación no será efectiva. Por otro lado, es necesaria la cooperación ya que los actores necesitan crear vínculos entre sí para que la relación sea duradera en el tiempo. Los participantes tienen que conocerse para saber hasta dónde puede llegar su asociación. Más información sobre las interrelaciones entre los tipos de relaciones puede encontrarse en [Chiambaretto et al. \(2016\)](#), [Bengtsson y Raza-Ullah \(2016\)](#).

Las razones por las que una empresa puede decidir abordar prácticas de CH pueden ser muy variadas. Según [Crujisen et al. \(2007a\)](#) estos son algunos de los objetivos que intentan alcanzar las empresas y los ejemplos que muestran que mediante la cooperación/co-opetición se pueden alcanzar:

- Mejorar la satisfacción del cliente por el servicio recibido. Por ejemplo, podemos considerar la consolidación de las órdenes del mismo cliente en un solo envío de tal forma que el cliente sólo sea visitado con menos frecuencia.
- Ofrecer un servicio más innovador. Por ejemplo, los camiones pueden ir equipados con sistemas de radiofrecuencia que permitan la comunicación on-line de cualquier problema para su más rápida solución. De esta forma el servicio es más fiable y se pueden realizar acciones de mitigación en caso de problemas tan pronto como se produzcan.
- Acceso a nuevos mercados. Por ejemplo, se pueden expandir geográficamente los servicios con la cooperación con empresas que operan en otros países/continentes.
- Mejorar la comunicación electrónica con los clientes. Por ejemplo, se pueden instalar conexiones utilizando el protocolo EDI (*Electronic Data Interchange*) a través de aplicaciones en modo compartido en cuanto a inversiones.

- Reducir los recursos y los costes. La implementación de un sistema más optimizado de gestión de rutas puede hacer construir rutas más eficientes de tal forma se necesite una flota más reducida para hacer los mismos envíos.
- Reducir el impacto medioambiental. La optimización de las rutas hará que estas sean más verdes y eficientes reduciendo el impacto medioambiental.

En el campo de los negocios en general, la cooperación no es nueva y ha venido desarrollándose en el pasado con el nombre de alianzas (Dogson, 1992). Williamson (1991) define las alianzas como las entidades en que los participantes mantienen la autonomía, pero dependen bilateralmente en un grado no trivial. En este contexto, se refieren a acuerdos de no igualdad entre empresas independientes. Algunos estudios muestran evidencias de mejoras debido a las alianzas (Hagedoorn, 1994), pero en general, la literatura existente no ha mostrado evidencias significativas sobre sus beneficios.

Fuera de las actividades de L&D, la literatura sobre CH es escasa, pero tenemos algún ejemplo como el abordado por Dai y Chen (2011). Estos autores utilizaron el campo de las subastas y formularon explícitamente este entorno teniendo en cuenta las dos visiones y sus interrelaciones, la del subastador y la del postor. Teniendo en cuenta ambas y utilizando simulación evaluaron un entorno colaborativo en que ambas partes se benefician.

2.2.2. Logística.

En un mercado global como el actual, se está produciendo un incremento en la competencia entre las empresas y un cambio constante de los mercados (Leitner et al., 2011). Además, la velocidad del progreso tecnológico y la naturaleza más dinámica de los mercados hace que las empresas están más conducidas por la competitividad que nunca. Las empresas que quieren sobrevivir, están forzadas a reorganizar sus procesos para poder reaccionar de forma rápida y efectiva en costes a las necesidades de sus clientes. Teniendo en cuenta que los costes logísticos suponen una parte importante de los costes de las empresas, la logística está recibiendo mucha atención ya que su optimización permite a las empresas mejorar en eficiencia y flexibilidad.

Pero las empresas individualmente tienen un impacto limitado en la mejora de esta eficiencia. En particular, en el caso de L&D, debido a los volúmenes limitados, no pueden acceder a redes de transporte más eficientes en costes y más productivas. Para ellas, la

posibilidad de acceso a estos servicios pasa por las estrategias de interrelación con otras empresas y en particular por la cooperación. Diferenciaremos las operaciones de transporte logístico en sus variantes marina, aérea y terrestre que detallamos a continuación.

En el transporte marítimo se utiliza el término conferencia para identificar dos o más compañías que cooperan en ese ámbito (Álvarez-SanJaime et al., 2013a, b). Una conferencia es una alianza de varias empresas que ofrecen sus servicios en una línea de transporte específica con una tarifa determinada y colectiva, y los mismos niveles de servicio (Eekhout, 2001). Aunque no es un objetivo explícito, consiguen evitar las guerras de precios entre competidores, y ofrecen ventajas como economías de escala en caso de requerir altos volúmenes de envíos (Shepperd y Seidman, 2001). Los competidores se oponen ya que la posibilidad de competir en estas condiciones es escasa para los que no pertenecen a la conferencia.

La CH juega también un papel determinante en la aviación. Las tres principales alianzas en este terreno son, Skyteam (13 líneas aéreas), Star Alliance (veintisiete líneas aéreas), y OneWorld (12 líneas aéreas). Económicamente las compañías aéreas tienen fuertes incentivos para operar en amplias redes internacionales. El crecimiento a través de uniones y adquisiciones puede proporcionar importantes expansiones a una red. Sin embargo, la concesión de derechos de tráfico internacional se limita en gran medida con vehículos específicos de propiedad de los países individuales (Fan et al., 2001). Además de la calidad del servicio ofrecido a los clientes, las alianzas en aviación proporcionan mayores factores de carga en los aviones, así como una organización más eficiente, Oum et al. (2000).

En relación con la CH en la logística terrestre, Caputo y Mininno (1996) examinan varias políticas como el uso de documentos informáticos estandarizados, contenedores y cartones estandarizados, almacenes compartidos por varios proveedores y una planificación de rutas coordinada en la industria alimentaria de Italia. Los autores plantean que los proveedores compartan un mismo transporte como práctica de CH. Erdmann (1999) construye un modelo para cuantifica las sinergias potenciales de diferentes formas de cooperación en la industria de consumo alemana. Asimismo, desarrolla una serie de consejos y recomendaciones para la asignación de costes y diseño de prácticas de CH.

En la misma medida que las conferencias y alianzas vistas hasta ahora, las empresas que acuerdan una alianza de CH en el ámbito de la logística terrestre esperan una significativa reducción en los costes y una buena interrelación entre las variables.

- Costes.
- Nivel de servicio.
- Sostenibilidad.

2.2.3. Determinación y distribución de las ganancias.

Cuando dos o más empresas deciden cooperar, uno de sus principales objetivos es obtener economías de escala que se suelen traducir en ahorros de costes que deberán ser asignados de la forma más justa posible entre los participantes. Éste es uno de los principales impedimentos para cooperar horizontalmente que se enumeran en [Crujissen et al. \(2007a\)](#), i.e., determinar y dividir las ganancias de la coalición. Según [Varnovermeire et al. \(2014\)](#), el coste total de distribución de una coalición logística es generalmente entre un 9% y un 30% más bajo que la suma de costes de cada participante por separado.

Aunque hay algunos artículos que tratan este problema como el de [Adenso-Díaz et al \(2014\)](#), no existe mucha literatura sobre la distribución de costes y ahorros en el campo de la CH. Para llenar esta laguna, en su lugar, la teoría de juegos propone una dirección para su planteamiento ([Lozano et al., 2013](#)). La teoría de juegos es un campo de investigación que propone un soporte teórico para tomar decisiones en aquellas situaciones en la que los actores participantes actúan por su propio interés, pero dependen del comportamiento de otros jugadores para alcanzar sus objetivos. Más en concreto, los conceptos de la teoría de juegos cooperativa determinan una justa asignación de costes/beneficios. Esta parte de la teoría de juegos se aplica en los casos en que los participantes pueden alcanzar un beneficio mayor actuando juntos que si lo hacen por separado ([Myerson, 1992](#)). Por lo tanto, en lugar de centrarse sólo en la maximización de su propio beneficio, están también preocupados por las ganancias totales de la coalición. De esta forma, la teoría de juegos cooperativa se centra en dos aspectos:

- Cómo calcular los beneficios potenciales para la coalición.

- Cómo compartir estos beneficios entre todos los actores.

Este problema tiene muchas características en común con otros como el del agente viajero, (Engevall et al., 1998), juegos de transporte en general (Sanchez-Soriano et al., 2001), el problema del cartero chino (Hamers et al., 1999), el problema de rutas de vehículos (Engevall et al., 2004), la industria de la automoción (Cachon y Lariviere, 1999), las telecomunicaciones (van den Nouweland et al., 1996), la aviación (Adler, 2001), y los servicios de salud (Ford et al., 2004). Todos estos artículos utilizan la teoría de juegos que tiene en cuenta el impacto de cada jugador dentro del grupo produciendo asignaciones que distribuyen los beneficios de la cooperación en base a reglas justas. El proyecto CO³ (Barbarino, 2011) promueve el valor Shapley como marco para prácticas de CH. Lozano et al. (2013), por un lado, plantean un problema lineal para poder cuantificar las sinergias de los distintos candidatos de la cooperación e identifica las oportunidades más beneficiosas. Por el otro, plantea el problema de la asignación de ahorros utilizando la teoría de juegos cooperativa. Vanovermeire y Sörensen (2014), dan un paso más y sugieren que para ser justos en la asignación de beneficios entre los participantes hay que añadir un factor adicional que es de naturaleza asimétrica. Este factor es la flexibilidad y es una alternativa a considerar y valorar entre las empresas que colaboran. Para más información se recomienda leer la revisión de Guajardo y Rönnqvist (2016).

2.2.4. La cooperación horizontal en problemas de rutas de vehículos (VRP).

Como hemos comentado anteriormente, existe mucha literatura en el campo de la logística marina y la aérea pero no tanto en la terrestre L&D. Un buen ejemplo es el caso de los camiones de carga fraccionada (*Less than Truck Load* o LTL) que operan de forma cooperativa (Liu et al., 2010; Wang y Kopfer, 2014). Sus objetivos son alcanzar sinergias en las siguientes áreas:

- Cobertura territorial.
- Optimización de los tiempos de viaje.
- Aumento de los factores de carga de los vehículos.
- Optimización de los recursos.
- Calidad de servicio.

Lyons et al., 2012 detallan en su artículo la complejidad del problema que empieza en la recogida de las materias primas y acaba en el cliente final. En el medio se producen muchas transacciones logísticas, los primeros receptores de las materias primas son las plantas de producción. Una vez se ha transformado la materia prima, se transporta al almacén, a continuación, al centro de distribución para llegar al cliente final o a otra planta de producción si necesita alguna otra transformación. En este segundo caso, se inicia de nuevo la transferencia al almacén en un bucle que acaba cuando el producto ha sido totalmente transformado y puede ser entregado al cliente final. El flujo completo es lo que se conoce como la cadena de suministro y algunos artículos la abordan en su totalidad mientras que la mayoría se centran en solucionar alguno de los subproblemas asociados. Centrándonos en el movimiento de materiales que se considera en el VRP, Dahl y Derigs (2011) estudian la efectividad de sistema para el soporte en decisiones que permite la cooperación entre empresas para incrementar la utilización de los vehículos. Adenso-Diaz et al. (2012) desarrollan un algoritmo aleatorio GRASP para resolver el problema del diseño conjunto de rutas en el transporte por carretera con el objetivo de mejorar la utilización de los vehículos. Éste es uno de los puntos clave de la CH, la mejora del porcentaje de carga de los vehículos y los modelos de las rutas de vehículos que colaboran en este sentido son los que se muestran a continuación en su relación con la literatura existente sobre CH. Vemos a continuación algunos de los modelos de rutas de vehículos que son de más ayuda en las prácticas de CH.

2.2.4.1. Recogidas y entregas.

En el problema de recogidas y entregas tenemos tres tipos de clientes, los que reciben material (entregas), los que envían material (recogidas) y los que pueden hacer ambas actividades simultáneamente. Este modelo es de muy directa aplicación en casos de CH ya que permite cumplir uno de sus objetivos de evitar los trayectos vacíos y aumentar el grado de carga de vehículos al permitir no sólo entregas sino también recogidas. Krajewska et al. (2008) analizan los márgenes obtenidos como resultado de la CH entre empresas de distribución. Su trabajo combina características de ruteo y programación con teoría de juegos para estimar la asignación de ganancias. Los clientes son de ambos tipos, recogida y entrega. El artículo considera la asignación de beneficios de forma justa para los participantes. Asimismo, se presenta resultados numéricos para casos reales y casos artificialmente creados, mostrando que la colaboración puede proporcionar decrementos considerables de los costes. Finalmente

propone una asignación de los beneficios obtenidos a través de la teoría cooperativa de juegos. **Rieck y Zimmermann (2010)** estudian el modelo en su versión simultánea, i.e., un cliente puede ser de recogida y de entrega al mismo tiempo. **Wang y Kopfer (2015)** utilizan simulación en un entorno de recogidas y entregas para concluir que los resultados obtenidos en un entorno colaborativo son considerablemente más óptimos que en uno no colaborativo. Los ahorros en porcentaje se mantienen prácticamente constantes independientemente de los parámetros utilizados en las pruebas. En la **Figura 2.1** se puede ver una representación gráfica del proceso.

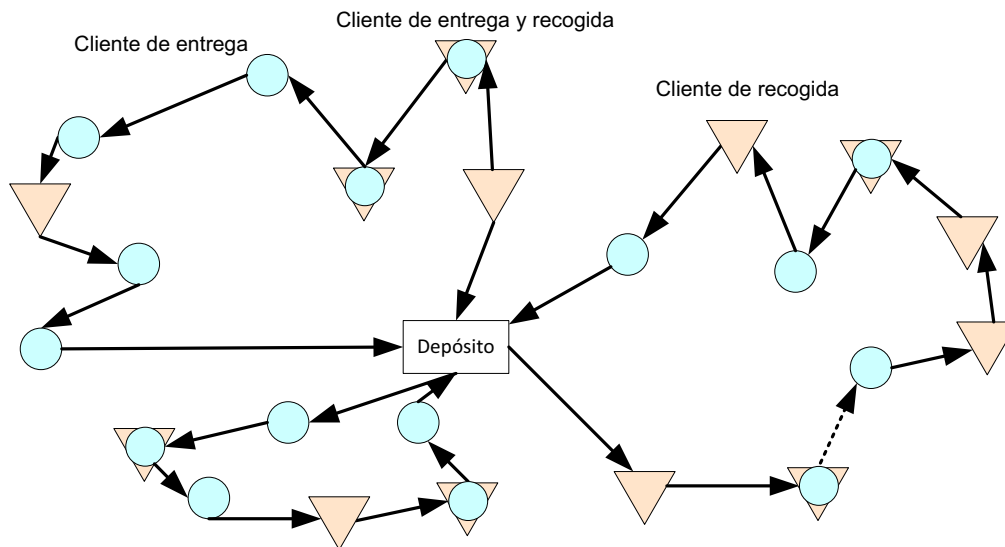


Figura 2.1: Representación gráfica del modelo de recogidas y entregas.

2.2.4.2. Depósitos múltiples.

En el problema de los multi depósitos en su versión de CH, un grupo de clientes cooperan compartiendo sus depósitos para ponerlos a disposición del resto. De esta forma, los clientes pueden ser servidos por el depósito que se encuentre más cerca geográficamente independientemente de que el receptor sea el propietario del depósito o no. **Nadarajah (2008)** introduce en su trabajo el término Problema Colaborativo de Rutas de Vehículos con Ventanas de Tiempo, donde combina los múltiples depósitos con las recogidas y entregas. El problema se formula como un problema de rutas de vehículos multi-depósitos con demandas estocásticas. **Krajewska et al. (2008)** también abordan este problema utilizando la teoría de juegos para la distribución de beneficios. **Pérez-Bernabeu et al. (2015)** analizan el problema de los depósitos múltiples en el entorno de la CH. En sus experimentos se muestra que se puede producir una

reducción en costes entre el 5%-90% dependiendo de la distribución geográfica de los clientes. Asimismo, [Quintero et al. \(2016\)](#) en su artículo utilizan multi-depósitos y comparan un escenario no cooperativo con su equivalente cooperativo. En la [Figura 2.2](#) se puede ver una representación gráfica del modelo de multi depósitos.

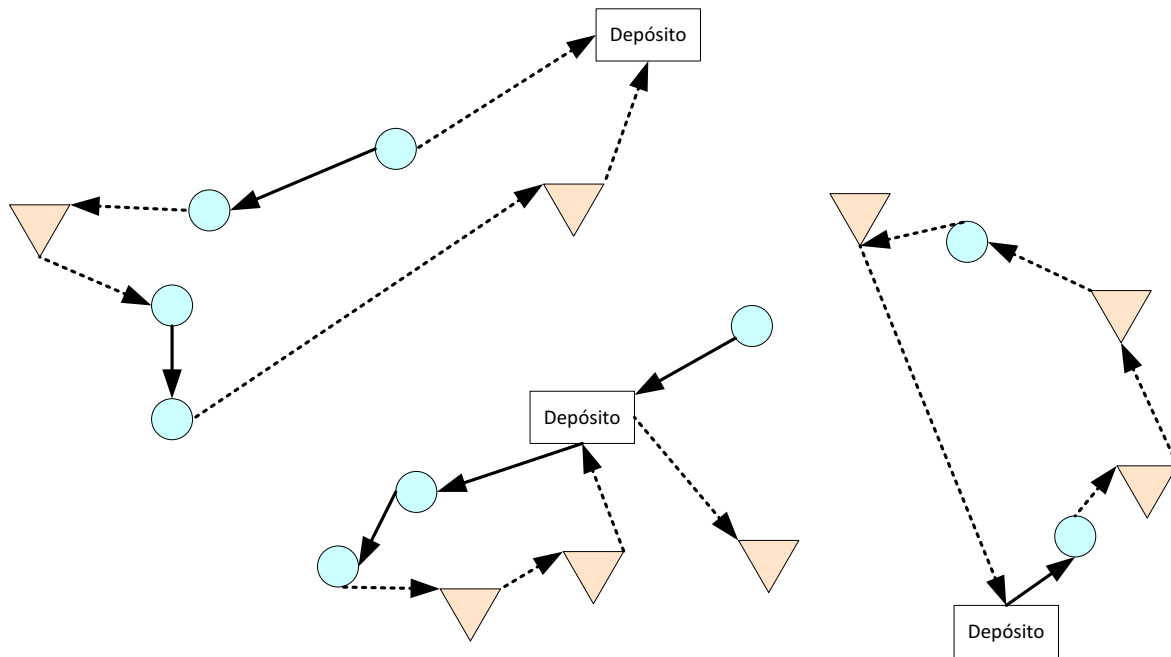


Figura 2.2: Representación gráfica del modelo de depósitos múltiples.

2.2.4.3. Flota Heterogénea.

Cuando dos o más empresas acuerdan cooperar en su proceso de distribución de mercancías, conviene acordar la flota a utilizar de tal forma que se adapte a la carga a transportar y esté formada por los vehículos disponibles en las empresas afectadas. Es difícil que las flotas de los distintos clientes a considerar estén formadas por el mismo tipo de vehículos, por tanto, es casi seguro que la flota a compartir sea de tipo heterogénea ([Dominguez et al. 2016](#)). En caso de que la flota se construya desde cero para el escenario de colaboración, uno de los factores que se utilizan para mejorar la utilización de los vehículos es el uso de las flotas heterogéneas.

La literatura sobre CH con flotas heterogéneas es escasa a pesar del potencial que tiene. Esto se ha traducido en artículos de CH con utilización de este tipo de flotas como en [Irnich \(2000\)](#), que también utiliza la variante de depósitos múltiples o [Engevall et al. \(2004\)](#). En su artículo utilizan la teoría de juegos para estudiar el problema de rutas de vehículos con flota

heterogénea. [Chu \(2005\)](#), plantea el problema de la cooperación en términos de subcontratación de un proveedor y lo aborda con una flota heterogénea de vehículos con capacidad limitada. Asimismo, [Wang et al. \(2014\)](#), introducen la subcontratación, de tal forma que extiende su convencional problema de rutas a un plan de transporte integrado utilizando su flota propia además de los vehículos de sus subcontratados. Muestra datos para contrastar cómo de esta forma el problema de ruteo se soluciona a un coste más reducido. En la [Figura 2.3](#) se puede ver una representación gráfica del proceso.

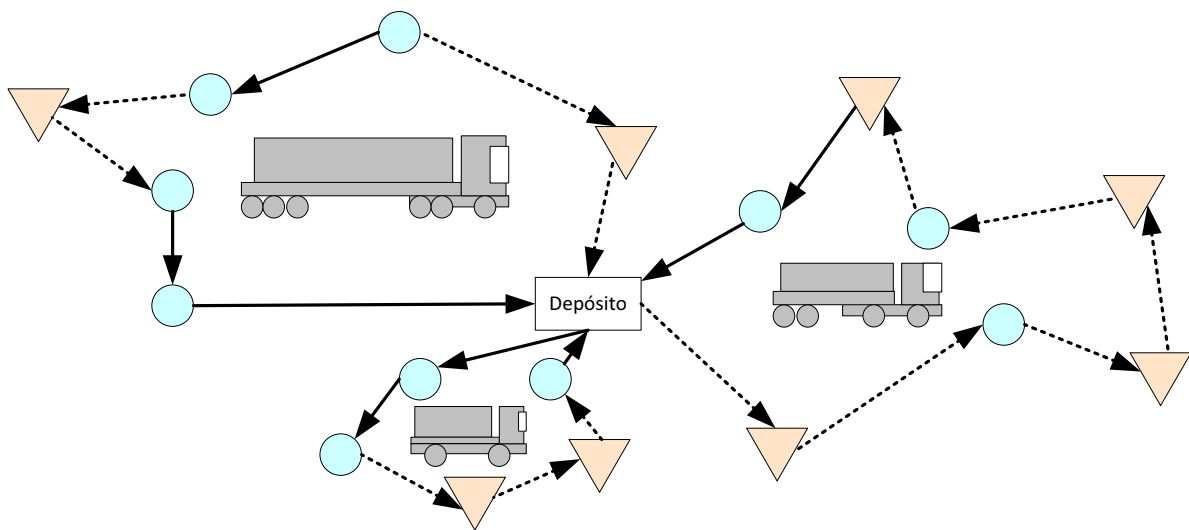


Figura 2.3: Representación gráfica del modelo de flota heterogénea.

2.2.4.4. Retornos (Backhauling).

Uno de los objetivos que busca la CH es evitar los trayectos vacíos de carga y este objetivo se logra utilizando el modelo de rutas con retornos. En esta variante, se aprovechan los trayectos de vuelta al depósito para realizar recogidas de mercancías en clientes que tienen que enviar materiales al depósito central. La literatura es extensa para este tipo de problemas. Hay muchas contribuciones que muestran modelos de optimización para este aspecto de la CH que consiste en utilizar de la mejor forma posible la capacidad de los vehículos en su trayecto de vuelta al depósito.

Palander y Väätäinen (2005) en su artículo muestran cuatro casos de estudio donde se integran los sistemas de transporte de maderas en un esquema de CH. Los autores concluyen que gracias a esta práctica obtienen un 20% de reducción en costes y un 2% adicional si utilizan asimismo las prácticas de *backhauling* (retornos). Frisk et al (2010) presentan un estudio en el mismo ámbito del transporte de madera en bosques. Concluyen que una mejora en la planificación supondría unos ahorros del 5%. Si a ello se añade la colaboración con prácticas de *backhauling*, el ahorro sería entre el 9% y el 14% adicionales. Apfelstädt et al. (2016), presentan una descripción formal cooperativa para las rutas de transporte con el objetivo de prevenir los trayectos vacíos de carga. En la misma línea, Crujisen et al. (2007b) afirman que la CH incrementa la productividad y la reducción de los trayectos vacíos de carga. Asimismo, Bailey et al. (2011) desarrollan y resuelven dos modelos utilizando dos metodologías. En sus resultados, obtienen el porcentaje de ahorros en costes evitando las rutas vacías en los trayectos de vuelta al depósito pueden alcanzar el 27%. Ubeda et al. (2011) muestran un caso de estudio real de una compañía del sector de la alimentación. En su estudio muestra mejoras del 15% en distancias al introducir prácticas de *backhauling* y CH entre los proveedores. Juan et al. (2014b) y Leon et al. (2015) presentan la cooperación horizontal basada en la utilización de prácticas de *backhauling* como una práctica útil para reducir los costes de las operaciones de transporte, así como las emisiones de CO₂. En la Figura 2.4 se puede ver una representación gráfica del proceso.

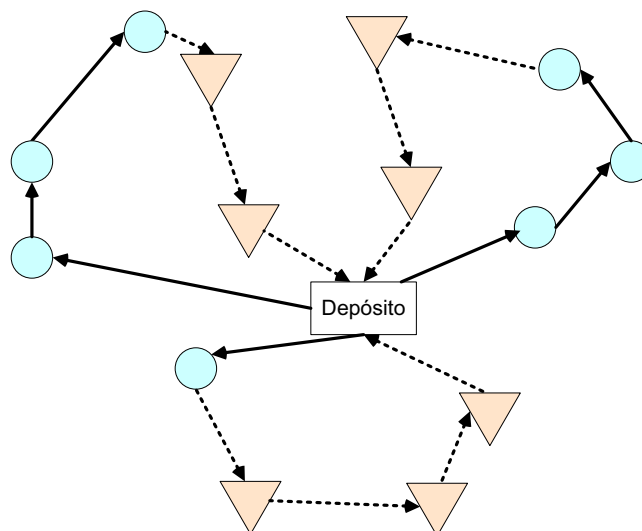


Figura 2.4: Representación gráfica del modelo de retornos.

2.2.5. Criterios Medioambientales.

Uno de los retos principales de la CH en general y de la gestión de la cadena de suministro en particular es la reducción del impacto medioambiental producido por la actividad de transportar mercancías. Además, este objetivo de ser compatible con la eficiencia en los costes y el cumplimiento de los niveles de servicio. La CH ha mostrado los incrementos en eficiencia entre el 10% y el 30%. Además, los beneficios medioambientales de la CH pueden incluso superar las reducciones de costes ya que los largos volúmenes de transporte que se pueden alcanzar cooperando pueden suponer un cambio en los modos de transporte principalmente en la frecuencia de envíos/recepciones. Se espera que se generalice la adopción de prácticas de CH para mejorar la eficiencia de la cadena de suministro. La creación de un marco legal estándar europeo y la publicación de casos de éxito pueden ser un buen facilitador.

Capgemini and Global Commerce Initiative (2008) estima que, para responder al reto de reducir el impacto medioambiental de los medios de transporte, la cadena de suministro del futuro será cooperativa. Por tanto, una importante contribución de la CH en la gestión logística de la distribución de flotas será la reducción del impacto medioambiental causado por el transporte, principalmente relacionado con la emisión de gases de efecto invernadero que es muy difícil de estimar (**Lera-Lopez et al., 2012**). La principal razón para esta reducción es el decremento del tamaño de las flotas y una mejor utilización de estas (heterogeneidad) y los recursos disponibles. Esto implica que una de las políticas más efectivas en la logística verde es la cooperación y en particular la CH (**Dekker et al., 2012; Jabali et al., 2012**).

La optimización de las rutas considerando criterios medioambientales ha dado lugar al problema de la polución en las rutas (*Pollution-Routing Problem*) (**Bektas and Laporte, 2011**) y el problema de rutas verdes de vehículos (**Erdogan y Miller-Hooks, 2012**). Ambos juegan un papel importante en el control de las emisiones de gases de efecto invernadero asociado con las actividades logísticas de distribución. Por ejemplo, **Ubeda et al. (2011)**, como se ha comentado en la sección de retornos, calculan también el impacto medioambiental resultado de la integración de las actividades de entrega y recogida gracias a la cooperación entre sus proveedores, mostrando mejoras en torno al 15% en distancias. **Pulleman (2011)** muestra la creación de una red de transportistas de artículos congelados a través de una alianza de transportistas en España. Su análisis predice una reducción del 26% en emisiones de CO₂.

Valender (2011) muestra el envío de Madera desde el Benelux hacia Europa del Este y vuelta con Nestle, Bacardi y Kraft como promotores. Su alianza muestra una reducción en gases de efecto invernadero sin pérdida de niveles de servicio. **Menedeme et al. (2011)** muestran la colaboración entre dos compañías farmacéuticas UCB y Baxter, para sus envíos a Rumanía. El resultado es una reducción de gases de efecto invernadero mayor del 50%. **Tillemans (2011)** mejora los niveles de servicio y un 50% de reducción de CO₂ en su modelo de CH entre las empresas Heinz y Friesland Campina y cooperación vertical con los empaquetadores.

2.3. Problemas de rutas de vehículos con retornos.

En esta sección se revisa la literatura publicada en relación a los problemas del VRPB. La literatura ha tratado principalmente el VRPB como una extensión del VRP. El principal punto de inspiración ha sido la descomposición natural del problema en dos subproblemas, uno para los clientes de entrega y otro para los clientes de recogida. Por tanto, la forma tradicional de resolverlo ha sido:

1. Resolver el problema considerando sólo los clientes de entrega.
2. Resolver el problema considerando sólo los clientes de recogida.
3. Combinar ambos de la forma más inteligente posible.

Podemos considerar que 1 y 2 son el mismo problema clásico del VRP, mientras que 3 es específico del VRPB. El primer enfoque heurístico para resolver el VRPB fue propuesto por **Deif y Bodin (1984)**. En este trabajo, los autores desarrollan una extensión del heurístico clásico basado en ahorros de **Clarke y Wright (1964)**, lo que les permitía resolver instancias de hasta 300 clientes. El algoritmo se adapta al caso del VRPB teniendo en cuenta que la restricción de precedencia (los clientes de recogida son visitados cuando se han completado todas las visitas a los clientes de entrega) reduce el número de soluciones factibles. Es decir, la solución contiene solo un arco que conecta los dos tipos de clientes (arco interfaz). **Vigo (1996)** presentó una adaptación de este procedimiento al caso en que los ahorros de los arcos son asimétricos. Asimismo, **Golden et al. (1985)** propusieron un método basado en el principio de inserción. En su modelo, inicialmente se generan las rutas de entrega y después se insertan los clientes de recogida de acuerdo a un criterio de inserción. La restricción de precedencia se asegura con la

introducción de un factor de penalización que hace que los clientes de recogida se inserten al final de la ruta. Posiblemente el principal factor que determina el éxito en los algoritmos de inserción es la correcta selección del tramo interfaz.

En el trabajo de [Goetschalckx y Jacobs-Blecha \(1989\)](#), se desarrolla un heurístico de dos fases. En su método, tanto la fase de agrupamiento como la fase de creación de rutas se resuelven haciendo uso de un enfoque de curvas de llenado del espacio (*Space-Filling Curves*). Con este procedimiento se mapean los puntos del espacio euclidiano a puntos de una línea unitaria. Este mapeo conserva las secuencias de tal forma que se creaban dos agrupaciones diferentes para clientes de recogida y clientes de entrega. Las dos secuencias se gestionan por separado para formar rutas factibles y después se unen según sus distancias euclídeas respectivas. De forma semejante, [Goetschalckx y Jacobs-Blecha \(1993\)](#) propusieron un algoritmo del tipo primero-agrupación segundo-ruteo (*cluster-first route-second*). En su planteamiento se consideran las agrupaciones como un problema de asignación y el problema del ruteo como un problema del agente viajero con la restricción de que sólo se permite un tramo interfaz conectando los clientes de recogida con los de entrega. [Anily \(1996\)](#) introdujo otro algoritmo del mismo tipo primero-agrupación segundo-ruteo. La fase de agrupación se completa con un heurístico modificado de partición circular, seguido de la construcción de rutas de agente viajero a través de las distintas agrupaciones. Entonces, las agrupaciones de entregas se fusionan a las de recogidas siguiendo el algoritmo de asignación propuesto por [Kuhn \(1955\)](#). El proceso concluye con una fase de generación de rutas para determinar la conexión óptima entre el depósito, las entregas y las recogidas. Otro algoritmo del mismo grupo es el propuesto por [Toth y Vigo \(1996\)](#). [Toth y Vigo \(1999\)](#) introdujeron la idea de costes asimétricos entre clientes y propusieron otro algoritmo del tipo primero-agrupación segundo-ruteo para el VRPB. En la versión simétrica, la distancia entre cada par de clientes es la misma, mientras que en la versión asimétrica no existe esta asunción.

De acuerdo con [Toth y Vigo \(2014\)](#) el VRPB está incluido dentro de los problemas generales de recogida y entrega que a su vez está dividido en dos clases. La primera clase codificada como VRPB hace referencia a los problemas de rutas que empiezan y terminan en uno o más depósitos centrales. En la segunda clase, las mercancías se transportan entre clientes de recogida y clientes de entrega directamente en el problema conocido como VRP con recogidas y entregas.

Halse (1992) propuso también un heurístico con la secuencia primero-agrupación segundo-ruteo. Salhi y Nagy (1999) resolvieron primero el problema de los clientes de entrega y después insertando los clientes de recogida uno a uno. Wade y Salhi (2001) propusieron un sistema de colonias de hormigas. Dethloff (2002) desarrolló un heurístico de inserción basado en el concepto de capacidad residual del vehículo que fue utilizado inicialmente para el problema simultaneo (un cliente puede ser de recogida y entrega simultáneamente). El criterio de inserción se basa no sólo en la distancia entre clientes sino también en la capacidad residual de la ruta. Wade y Salhi (2002) introdujeron un nuevo mecanismo para determinar la posición dentro de la ruta desde la cual se puede visitar el primer cliente de recogida. Esto se hacía en función de varios factores:

- La experiencia del usuario.
- La capacidad del vehículo.
- El tipo de producto.
- El tipo de vehículo.

Otro heurístico para el VRPB es el propuesto por Nagy y Salhi (2005). Todos estos métodos, aunque no estaban pensados específicamente para el VRPB, en cambio se pueden adaptar fácilmente a la restricción de precedencia que se incluye en este problema. Algo semejante ocurre con los heurísticos del tipo mejora/intercambio. En este caso, la única adaptación necesaria para VRPB es la de no mezclar los clientes de recogida y entrega dentro de los procesos de mejora. Este es el caso de los procedimientos 2-opt, 3-opt (Lin, 1965), y 4-opt (Boctor y Renaud, 1993) que deben aplicarse por separado a cada tipo de cliente dentro de cada ruta. De la misma forma el algoritmo de barrido extendido de Gillett y Miller (1974), puede adaptarse fácilmente al VRPB permitiendo las agrupaciones de tal forma que respeten la restricción de precedencia.

Los métodos metaheurísticos dominan la literatura desde el año 2000. Es el caso de Osman y Wassam (2002), que fueron los primeros en desarrollar un algoritmo de búsqueda tabú para el VRPB. Los autores propusieron dos métodos de construcción basados en ahorros, inserción y asignación. En su planteamiento, la solución que se obtiene inicialmente se introduce dentro de su método reactivo de búsqueda tabú. Brandao (2006), propuso también un algoritmo de

búsqueda tabu y Wassan (2007), mejoró su búsqueda tabú con un programa adaptativo de memoria. Ropke y Pisinger (2006), unificaron los problemas del tipo VRPB como una extensión de lo que llamaron problemas enriquecidos de recogidas y entregas con ventanas de tiempo. Su propuesta incluye la transformación de todas las variantes del problema considerando la restricción de tiempo y proponen un heurístico de búsqueda local. Wassan et al. (2009) y Gajpal y Abad (2009) utilizan un sistema de colonias de hormigas para resolver el problema. En la obra de Reimann y Ulrich (2006) se puede ver una comparación de este tipo de estrategias utilizando hormigas. Un planteamiento distinto es el que propusieron Yazgi Tütüncü et al. (2009), a través de un sistema para soporte de decisiones basado en la aleatoriedad del proceso de búsqueda. Los autores resuelven el VRPB con restricción de precedencia y sin ella en un entorno interactivo. García-Najera (2012) generalizó el VRPB como un problema multi-objetivo y lo resuelve con una versión de algoritmos evolutivos. Zachariadis y Kiranoudis (2012) propusieron una solución basada en un metaheurístico que explora el espacio de soluciones con el intercambio de secuencias de clientes de longitud variable. La solución inicial se obtiene a través de un algoritmo de ahorros.

Vidal et al. (2014) introdujeron una solución para VRP con atributos múltiples. Los atributos son características o restricciones que describen las especificaciones de los VRPs de la vida real. Los autores proponen un algoritmo de propósito general llamado búsqueda genética híbrida unificada (*Unified Hybrid Genetic Search* o UHGS), que utiliza operadores adaptativos de asignación, secuenciación y evaluadores de rutas para resolver 29 variantes distintas del VRP. Su algoritmo iguala o supera las mejores soluciones conocidas de cada problema. Cuervo et al. (2014) propusieron un esquema de búsqueda local iterada donde el componente más novedoso es un heurístico de búsqueda oscilante que realiza transiciones constantes entre soluciones factibles e infactibles. Una de las características principales de su propuesta es el análisis estadístico de los componentes de su algoritmo para comprobar su influencia en los resultados. Wassan et al. (2016) introdujeron una nueva variante llamada VRP con viajes múltiples y resolvieron el problema con un algoritmo de dos niveles de búsqueda de vecindario variable.

Desde un punto de vista práctico, el VRPB es considerado asimismo en el terreno de la logística inversa y la logística verde. Se pueden alcanzar importantes reducciones de costes combinando clientes de recogida y entrega. Asimismo, se pueden alcanzar importantes

reducciones de emisiones de gases como se ha visto en la sección anterior (Berbeglia et al., 2007; Parragh et al., 2008; Sheu y Talley, 2011; Ubeda et al., 2011).

2.4. Flotas heterogéneas.

El problema del tamaño de la flota y composición de rutas con retornos considerando costes fijos y variables fue introducido por Salhi et al. (2013). Antes de él, Tütüncü (2010) analizó una versión simplificada del problema, pero sin considerar el tamaño de la flota y su composición. En su propuesta, el valor de ambas variables es conocido antes de solucionar el problema. Dada la escasa literatura que aborda los dos problemas juntos (retornos y flota heterogénea), haremos una revisión del problema considerando sólo los clientes de tipo de entrega, en el conocido problema del Tamaño de la Flota y Composición del Problema de Rutas de Vehículos (*Fleet Size and Mix Vehicle Routing Problem* o FSMVRP). En el FSMVRP, todos los clientes reciben material del depósito y tienes varias variantes dependiendo de las asunciones que se hacen con el tamaño de la flota (limitada o ilimitada) y la consideración o no de los costes fijos y variables.

Golden et al. (1984), introdujeron el problema considerando los costes variables. Estos costes se computan como la distancia recorrida y se considera que el tamaño de la flota es ilimitado. Utilizaron un heurístico constructivo basado en el algoritmo de ahorros. Con una configuración de costes similar, ha habido varios autores que han propuesto algoritmos de búsqueda tabú para solucionar el FSMVRP (Gendreau, 1999; Osman y Wassan 2002).

Taillard (1999) utilizó el método de generación de columnas para obtener rutas homogéneas para todos los tipos de vehículos. Entonces, se obtiene la solución final como solución del problema de partición de conjuntos asegurando que cada cliente sólo es visitado en una ocasión. Renaud y Boctor (2002) utilizaron un algoritmo similar al que aplicaron algunas mejoras. Choi y Tcha (2007) utilizaron un método de generación de columnas para resolver el problema. Liu y Shen (1999) propusieron un algoritmo genético y Subramanian y Penna (2012) implementaron un algoritmo híbrido que combina una búsqueda local iterada con un mecanismo de partición de conjuntos. Salhi et al. (2013) y Taillard (1999) propusieron una estructura de costes similar considerando costes variables. Renaud, y Boctor (2002) adaptaron el conocido algoritmo del barrido (*sweep*).

En el problema de configurar la flota heterogénea, el objetivo es optimizar la flota a utilizar determinando su tamaño y composición para garantizar la solución más óptima en costes. En cambio, cuando el tamaño y composición de la flota es fijado de antemano, el objetivo es conseguir la mejor utilización de la flota existente, un problema de naturaleza más táctica que el anterior. [Taillard \(1999\)](#) adaptó su algoritmo a la versión de la flota limitada. Para la misma versión del problema [Tarantilis et al. \(2003\)](#) utilizaron un algoritmo de recocido simulado. Obtenían la solución inicial con un heurístico constructivo y alcanzaban mejoras a esta solución a través de movimientos aleatorios que eran evaluados siguiendo distintos criterios.

Para una revisión y comparativa de distintos heurísticos y metaheurísticos aplicadas al problema heterogéneo, es posible acudir a las excelentes revisiones de [Baldacci y Battarra \(2008\)](#) y [Hoff et al. \(2010\)](#).

2.5. Conclusiones.

Cuando dos o más empresas se plantean cooperar en el transporte de mercancías por carretera, su objetivo principal es el de optimizar sus rutas actuales de tal forma que las entregas a sus clientes y las recogidas se hagan de la forma más eficiente en tiempo y calidad de servicio. Esto redundará en beneficios para sus clientes y por tanto en hacer que el negocio de la empresa sea más próspero en el presente y más sostenible en el futuro. La parte más básica en la optimización de las rutas es la optimización de la carga, y dentro de ésta, la eliminación de los trayectos en los que los vehículos circulan vacíos. La forma más elemental de abordar este problema es la de la consideración de que los trayectos de vuelta al depósito se pueden utilizar para visitar a otro tipo de clientes, cuya demanda consiste en enviar materiales al depósito central y son considerados clientes de recogida. Esta práctica es abordada en el primer artículo que compone la presente tesis doctoral donde el modelo considera los dos tipos de clientes, unos de recogida y otros de entrega. En este escenario, la distribución de clientes puede pertenecer a varias empresas diferentes. La medición de los resultados de una potencial cooperación entre varias empresas pasa por el estudio de este escenario y la comparación de los costes cooperando y no cooperando. Para poder hacer esta comparación, necesitamos un algoritmo óptimo que resuelva este problema y esto es lo que hemos obtenido solucionando el problema VRPCB incluido en el capítulo 3 de esta tesis.

Partiendo de este resultado, hemos buscado una mayor flexibilidad en el proceso y para ello hemos considerado el escenario en que las visitas a los clientes de recogida y entrega puede hacerse en cualquier orden. Este caso es una generalización del anterior y no hace falta esperar a que todos los clientes de entrega hayan sido visitados para poder visitar a los clientes de recogida. Las visitas a los clientes se realizan en cualquier secuencia, es decir, el problema es menos restringido y por tanto el número de soluciones posibles es mayor. Si queremos mejorar la solución de un modelo VRPCB, lo primero que tenemos que hacer es relajar la restricción de precedencia para obtener un modelo VRPMB. Esto es lo que hemos conseguido solucionando el modelo VRPMB del capítulo 4 de esta tesis.

Además de la optimización de las rutas, otro de los objetivos de la cooperación es la optimización de las flotas. Como consecuencia de la unión de dos o más empresas, sus rutas son optimizadas y por tanto la flota resultante de la unión también será optimizada. Es muy difícil pensar que los vehículos de las empresas cooperadoras van a tener la misma naturaleza en cuanto a capacidad y a costes. Por tanto, es casi seguro que el problema que haya que solucionar considere la heterogeneidad de las flotas. Este es el problema que abordamos en el capítulo 5 contenido en esta tesis. El problema de la heterogeneidad de las flotas puede abordarse desde dos puntos de vista:

- Mejor utilización de la flota actual (heterogénea).
- Obtener la mejor composición de flota para el problema actual.

El primero es de naturaleza táctica y estática ya que intenta solucionar un problema de rutas de vehículos donde el número de rutas está fijado de antemano y se trata de equilibrar las cargas de la forma más óptima posible. El segundo en cambio es un problema dinámico y selecciona la mejor composición de flotas para el problema de rutas que tiene que resolver. Su objetivo es responder a la pregunta que surge cuando dos o más empresas se plantean colaborar:

- ¿Cuál es la mejor composición de flota para servir a los clientes resultantes de la unión de dos empresas?

Esta es la primera pregunta que se hacen. Según la respuesta, decidirán las inversiones que tengan que realizar dependiendo de sus flotas actuales.

Capítulo 3.

El problema de las rutas de vehículos con retornos agrupados (VRPCB).

Partes de este capítulo se han extraído de la publicación aceptada en la revista Networks:

Belloso, J., Juan, A., Faulin, J. y Martínez, E. 2017. A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls. Networks, edición especial de los mejores artículos del congreso Verolog 2015.

Editores:

Bruce Gloden

Richard Hartl

3.1. Introducción.

El problema de rutas de vehículos con retornos agrupados (*Vehicle Routing Problem with Clustered Backhauls* o VRPCB), es una extensión realista del problema clásico de rutas de vehículos donde existen demandas, tanto de entrega como de recogida. En esta versión agrupada todos los clientes del grupo de entregas de la misma ruta tienen que ser servidos antes de que el primer cliente del grupo de recogidas pueda ser visitado. Los problemas en que se presenta esta restricción de carga, suelen ser aquellos en que los vehículos son cargados y descargados por el lateral trasero, y puede ser complicado o muy costoso recolocar la carga en

los emplazamientos de los clientes, debido al peso y tamaño de los artículos. Además, las áreas dedicadas en cada vehículo a las operaciones de entrega y las operaciones de recogida suelen ser las mismas y por esta razón se espera hasta acabar las entregas para empezar con las recogidas. El objetivo es combinar el proceso de reparto de materiales con el de abastecimiento como se puede ver en la **Figura 3.1**. En la parte superior se pueden ver dos procesos de distribución, el de la izquierda es de entrega de materiales y el de la derecha de abastecimiento. En la parte inferior se pueden ver las dos formas de mezclar ambos procesos. En la imagen de la izquierda se realiza la recogida cuando la entrega ha acabado (VRPCB) mientras que en la imagen de la derecha se realizan tareas de entrega y recogida indistintamente (VRPMB).

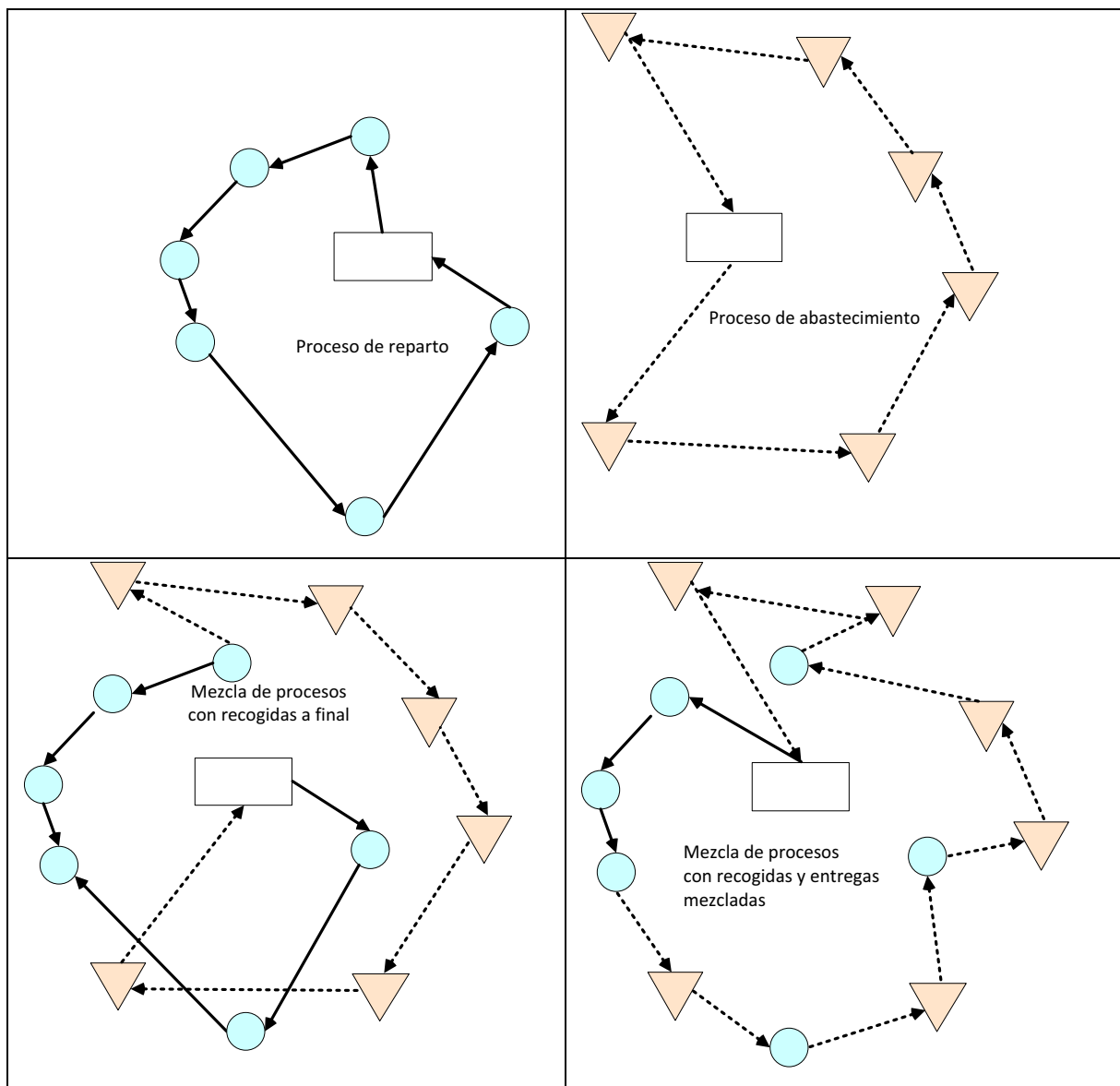


Figura 3.1: Combinación de los procesos de reparto y abastecimiento en VRPCB y VRPMB.

Algunos ejemplos de aplicación de este modelo se pueden encontrar en:

- Clientes que deben devolver contenedores vacíos a sus orígenes.
- En la industria de la alimentación: los supermercados y tiendas son los clientes de entrega y sus proveedores son los clientes de recogida.
- En la industria textil: empresas de tamaño grande tienen tiendas que son servidas desde los almacenes y al mismo tiempo, los almacenes deben recibir materiales desde los proveedores de la misma región.
- Aplicaciones de la logística inversa y logística verde:
 - Devolución de botellas vacías, contenedores, baterías usadas, etc.
 - Entrega de nuevos dispositivos acompañada de la recogida de los antiguos, entrega de productos defectuosos u obsoletos, etc.

En este capítulo se presenta un algoritmo metaheurístico relativamente simple de implementar, pero eficiente que integra una versión del heurístico de ahorros clásico en un marco metaheurístico. Se utiliza una distribución de probabilidad geométrica para inducir un efecto de ordenación aleatoria sesgada (orientada) en la lista de ahorros, y se asigna un coste de penalización a aquellos arcos que conectan a clientes de entrega con clientes de recogida (arcos interfaz) para preservar el orden secuencial de entregas y recogidas. En el enfoque propuesto, la restricción de tamaño de la flota está implícitamente relajada. Se aprovecha esta característica para que el proceso de búsqueda pueda explorar regiones no factibles pero prometedoras en el espacio de soluciones. Cuando el número de rutas no coincide con el número de vehículos disponibles, se emplea un "operador correctivo recursivo" para convertir soluciones infactibles en factibles. Este operador utiliza conceptualmente la recursividad ya que es invocado para resolver problemas reducidos del problema original, aquellos sub-problemas que contienen más rutas que las necesarias.

Esta metodología se muestra eficiente, simple y flexible para solucionar los problemas incluidos en esta tesis. Se proporciona una visión detallada del algoritmo propuesto y se discuten algunas de sus principales características, los parámetros que emplea y la forma concreta en la que se combinan los diferentes heurísticos y procesos de búsqueda local.

Para comprobar la eficiencia de la solución propuesta se compara con las mejores soluciones de la literatura utilizando instancias clásicas de referencia para el VRPCB obteniendo muy buenos resultados en costes y tiempos de ejecución. Para una de las instancias considerada se consigue mejorar su mejor resultado conocido obteniendo una nueva BKS.

3.2. Formulación del problema.

El problema VRPB es una extensión del clásico problema de rutas de vehículos con capacidad limitada (*Capacitated Vehicle Routing Problem* o CVRP). El CVRP considera una flota de vehículos homogénea situados inicialmente en un depósito o almacén central. Estos vehículos deben distribuir un conjunto de artículos, con el fin de satisfacer la demanda de los clientes. De esta forma, el objetivo principal es encontrar una planificación de rutas factible, que minimice los costes de distribución, respetando las restricciones del problema que son:

- Cada cliente recibe la visita de un solo vehículo, que satisface su demanda total.
- Cada vehículo comienza y termina su recorrido en el depósito (rutas cerradas).
- La demanda total cubierta por un vehículo no supera su capacidad máxima, en peso y dimensiones de carga.

La literatura alrededor de este problema es muy extensa y es posible encontrar diferentes modelos matemáticos para este problema, véase por ejemplo [Toth y Vigo \(2001\)](#) y [Golden et al. \(2008\)](#). Modelando el problema en formato ILP el modelo VRPCB se puede definir como un grafo completo $G = (V, A)$ donde:

- $V = \{v_0\} \cup L \cup B$ es el conjunto de nodos. v_0 se denomina depósito (*depot*).
- A es el conjunto de arcos.
- $L = \{v_1, v_2, \dots, v_l\}$ representa el conjunto de clientes de entrega.
- $B = \{v_{l+1}, v_{l+2}, \dots, v_{l+b}\}$ indica el conjunto de clientes de recogida.
- K es el número de vehículos con capacidad Q están disponibles en el vértice v_0 que corresponde al depósito central.

- Cada cliente de entrega (*linehaul*) $v_i \in L$ tiene una cantidad de demanda de entrega asociada, d_i , que debe ser transportada desde el depósito al cliente.
- Cada cliente de recogida (*backhaul*) $v_j \in B$ tiene una cantidad de recogida (*pick-up*) asociada, p_j , que debe ser enviada al depósito.
- Cada arco $(v_i, v_j) \in A$ tiene un coste fijo no negativo correspondiente c_{ij} que es el coste que implica viajar desde la posición v_i a v_j . Consideramos el caso simétrico donde $c_{ij} = c_{ji}$.
- Cada ruta de la solución contiene a lo sumo un arco $(v_i, v_j) \in A$ que conecta un cliente de entrega con uno de recogida.
- Hay al menos un arco $(v_i, v_j) \in A$ en cada ruta de la solución que contiene un cliente de entrega.

El objetivo del VRPCB es diseñar un conjunto de rutas tales que:

- Se minimice el coste total del conjunto de rutas generadas.
- El número de rutas generadas es igual a K (el número de vehículos a utilizar se indica con antelación y no se aceptan rutas con menos o más vehículos).
- Si una ruta contiene clientes de entrega y clientes de recogida, éstos sólo pueden ser atendidos cuando hayan sido visitados todos los clientes de entrega.
- Cada ruta contiene al menos un cliente de entrega.
- Cada cliente es visitado una vez.
- No se sobrepasa la capacidad Q del vehículo.

De esta forma, el problema de rutas de vehículos con retornos se puede expresar como (Lee et al., 2008):

L = número de clientes entrega, $1, \dots, L$ (depósitos el 0).

B = número de clientes recogida, $L + 1, \dots, n$.

n = número total de clientes ($L + B$).

K = tamaño de la flota

Q = capacidad del vehículo

c_{ij} = costo (distancia) del trayecto entre el cliente i y el cliente j

d_i = demanda del cliente i , ($i = 1, \dots, n$).

y_{ij} = carga del vehículo en el arco entre el cliente i y el cliente j .

$$x_{ij} = \begin{cases} 1, & \text{si } \text{arc}(i, j) \text{ es recorrido por algún vehículo} \\ 0, & \text{en caso contrario} \end{cases}$$

$$(1) \text{ minimizar } \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}$$

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

$$(3) \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n$$

$$(4) \sum_{i=1}^n x_{ip} = \sum_{i=1}^n x_{pj} \quad p = 0, \dots, n$$

$$(5) \sum_{i=0}^L y_{ij} - \sum_{i=0}^n y_{jl} = d_j \quad j = 1, \dots, L$$

$$(6) \sum_{l=L+1}^n y_{jl} + y_{j0} = d_j + \sum_{i=1}^n y_{ij} \quad j = L + 1, \dots, n$$

$$(7) y_{ij} = 0 \quad i = 0, \dots, L, \quad j = 0 \text{ y } j = L + 1, \dots, n$$

$$(8) y_{ii} = 0 \quad i = 1, \dots, n$$

$$(9) \sum_{i=L+1}^n y_{i0} = \sum_{i=L+1}^n d_j$$

$$(10) \sum_{j=1}^L y_{0j} = \sum_{j=1}^L d_j$$

$$(11) x_{0j} = 0 \quad j = L + 1, \dots, n, \quad v = 1, \dots, K$$

$$(12) x_{ij} = 0 \quad i = L + 1, \dots, n, \quad j = 1, \dots, L, \quad v = 1, \dots, K$$

$$(13) y_{ij} \leq \sum_{j=L}^n x_{ij} Q \quad i \neq j = 0, \dots, n$$

$$(14) \sum_{i=1}^n x_{i0} = \sum_{j=1}^n x_{0j}$$

$$(15) x_{ij} \in \{0, 1\} \quad i = L + 1, \dots, n, \quad j = 1, \dots, L, \quad v = 1, \dots, K$$

$$(16) y_{ij} \geq 0 \quad i = 0, \dots, n, \quad j = 0, \dots, n$$

En esta formulación la ecuación (1) es la función objetivo de minimización del coste. (2) y (3) aseguran que cada cliente sólo recibe una visita por un vehículo. Para asegurar que el número de rutas coincide con el número de vehículos está la ecuación (4). Las ecuaciones (5) y (6) aseguran la precedencia de los clientes de entrega y los de recogida de todos los clientes de este tipo. La ecuación (7) garantiza que en los arcos de transición entre clientes de entrega y clientes de recogida (sólo un tramo en cada ruta) el camión viaja vacío. La ecuación (8) se utiliza para confirmar que el tramo que va de un cliente a él mismo no tiene ninguna carga. La ecuación (9) impone que la carga del camión en su último tramo coincide con la suma total de demandas de los clientes de recogida en esa ruta. De manera análoga (10) sirve para garantizar lo mismo en el caso de la salida del camión del depósito como suma de las demandas de los clientes de entrega en esa ruta. (11) y (12) aseguran la restricción de que los clientes de recogida van siempre detrás de los clientes de entrega. La ecuación (13) impone la máxima carga del camión es igual a su capacidad y la (14) establece que el número de rutas de la solución es exactamente K . Finalmente las ecuaciones (15) y (16) definen la naturaleza binaria de la variable x , y la naturaleza continua de la variable y .

La estructura del VRPCB, se puede ver de forma gráfica en la [Figura 3.2](#).

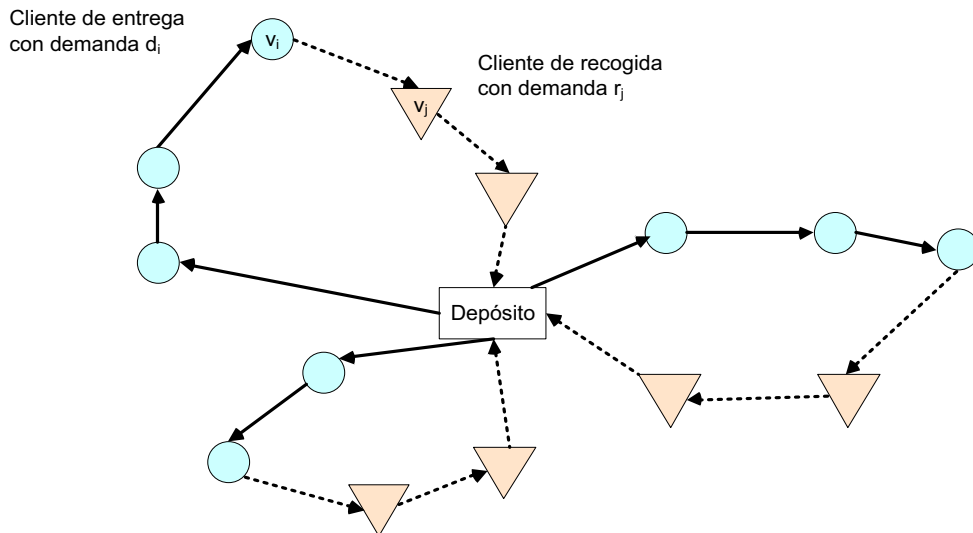


Figura 3.2: Una representación visual del VRP con retornos agrupados.

3.3. Metodología.

El algoritmo híbrido propuesto integra la metodología de la aleatoriedad sesgada aplicada sobre el heurístico clásico de generación de rutas basado en los ahorros, dentro de un marco ILS (Lourenco et al., 2003). ILS presenta una estructura simple, y ha demostrado ser un metaheurístico muy eficiente para resolver COPs. Podemos decir que este esquema es una estructura de segundo nivel cuyo objetivo es reducir la búsqueda de candidatos a soluciones óptimas locales devueltos por algún algoritmo de primer nivel o embebido. Su fundamento es combinar de forma iterativa una etapa de perturbación (destrucción y re-creación) con una etapa de búsqueda local. La etapa de destrucción tiene como objetivo *diversificar* la búsqueda de una solución mejor dentro del espacio de soluciones, mientras que la búsqueda local tiene como objetivo *intensificar* la búsqueda de una solución mejor dentro de un espacio de soluciones próximo al de la solución base actual. Con la diversificación se intenta evitar que el algoritmo quede atrapado en mínimos locales, y con la intensificación se intenta alcanzar el valor óptimo dentro de un entorno cercano. Burke et al. (2010) evaluaron este esquema entre otros aplicados a la resolución de tres problemas clásicos de optimización combinatoria, como son:

- El problema de secuenciación.
- El empaquetado unidimensional.
- La asignación/planificación de personal.

Burke et al. (2010) concluyen que el esquema ILS mejora el resto de los estudiados en su trabajo. Según los autores, dos factores justifican su eficiencia:

- El equilibrio entre la intensificación y la diversificación combinando la perturbación seguida de la búsqueda local.
- El reducido número de parámetros que utiliza.

La Figura 3.3 muestra el diagrama de flujo que ofrece una visión general del algoritmo propuesto. Se trata de un algoritmo del tipo de búsqueda local iterada con aleatoriedad sesgada, denominado ILS-BRCW (*Iterated Local Search-Biased Randomization Clarke Wright*).

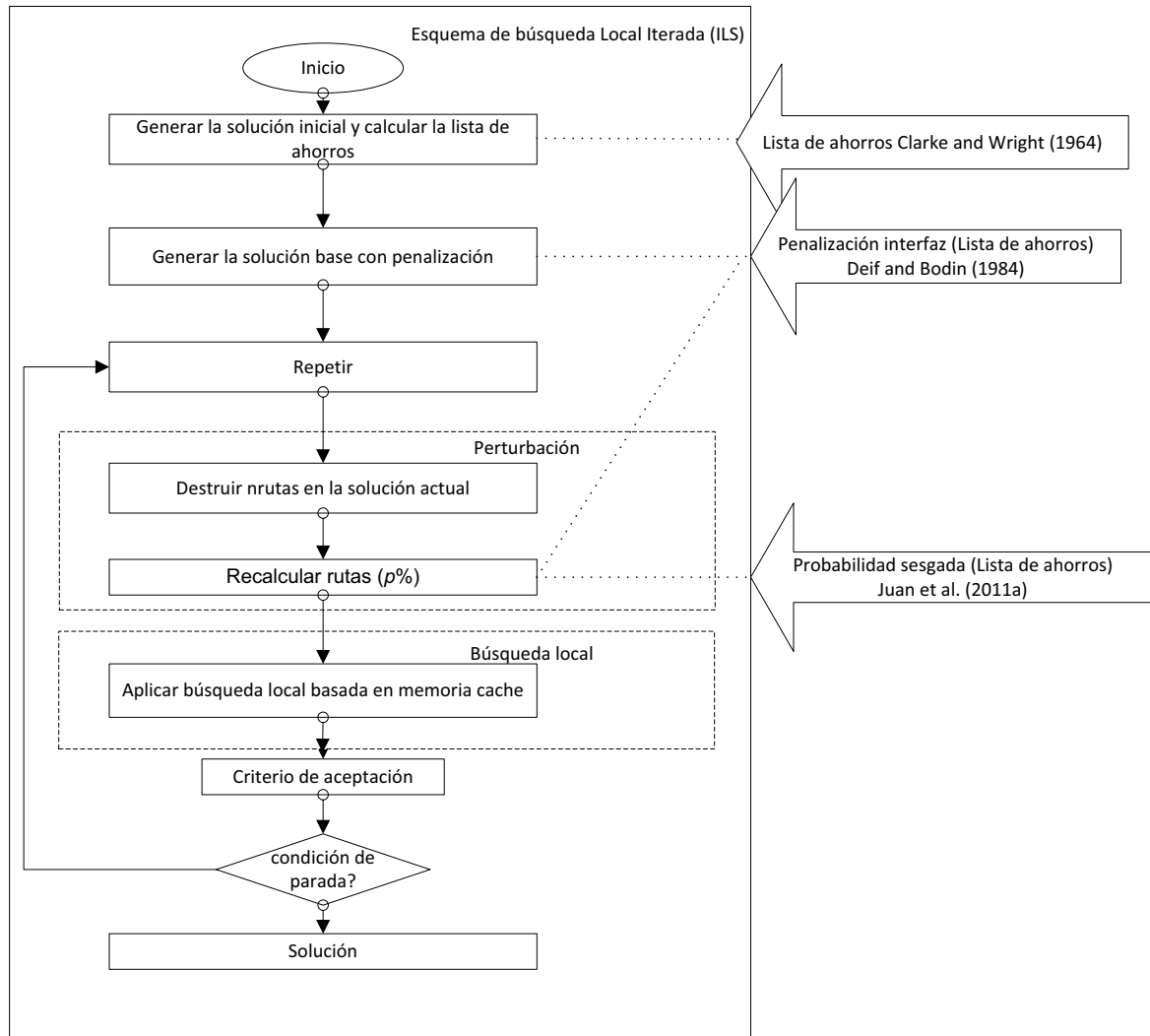


Figura 3.3: Esquema de la metodología empleada para la resolución del problema VRPCB.

El algoritmo comienza calculando una solución trivial. Esta solución inicial asigna una ruta individual de ida y vuelta desde el depósito hasta el cliente considerado, por lo que habrá tantas rutas como clientes. Al mismo tiempo, el algoritmo también calcula los ahorros asociados a cada arco que conecta cada cliente considerado con el resto, según la lógica del heurístico clásico de **Clarke y Wright (1964)** (CWS). Estos arcos se incluyen entonces en una lista ordenada de mayor a menor ahorro. Siguiendo la lógica de CWS, el ahorro asociado a un arco es el obtenido en el caso de incluir ese arco como unión de las dos rutas que contienen a sus nodos. Los costes de la ruta resultante son la suma de los anteriores menos la distancia que va del depósito al nodo inicial, menos la distancia que va desde el depósito al nodo final (ya no hay que recorrerlos) más la distancia que va entre los dos nodos. En forma de ecuación:

$$s_{ij} = d_{i0} + d_{j0} - d_{ij}$$

es el ahorro asociado al tramo (i,j) . Considerando que nodo 0 es el depósito, d_{i0} es la distancia euclídea del nodo i al depósito, d_{j0} es la distancia euclídea del nodo j al depósito y d_{ij} es la distancia entre los nodos i y j . De esta forma, tenemos calculada la lista de ahorros y en este momento se produce la primera integración del heurístico con el método propuesto por **Deif and Bodin (1984)** para la adaptación del algoritmo de ahorros al caso de los problemas con entregas y recogidas. **Deif and Bodin (1984)** propusieron penalizar los arcos que unen un nodo de recogida con un nodo de entrega (arco interfaz) con la siguiente función definida a trozos:

$$s'_{ij} = \begin{cases} s_{ij} - \lambda * S, & \text{si } (i,j) \text{ es interfaz} \\ s_{ij}, & \text{otro caso} \end{cases}$$

Donde s'_{ij} representa el nuevo ahorro asociado al arco (i,j) . La idea que está detrás de esta adaptación es el retraso de la elegibilidad de los arcos interfaz. El parámetro λ es un coeficiente de penalización entre 0 y 1, y S es una estimación del máximo valor del ahorro de un arco que se puede obtener para el problema en particular. Según **Deif and Bodin (1984)**, los mejores resultados se obtienen con λ en el rango entre 0,05 y 0,20. De esta forma se recalculan los ahorros de los nodos interfaz y así se altera el orden obtenido inicialmente. Los arcos que conectan nodos de recogida con nodos de entrega se elegirán más tarde que lo que les correspondería si sólo utilizáramos la ordenación de CWS. Con los nuevos ahorros calculados, se ordena la lista de mayor a menor valor y se inicia el proceso de fusión de rutas siguiendo el nuevo orden. Entonces, se extrae el arco que encabeza en ese momento la lista previamente ordenada y se borra de la lista para no volver a ser utilizado. Este arco conecta dos rutas diferentes. Si, y solo si, estas dos rutas se pueden combinar sin violar ninguna de las restricciones del problema, entonces se lleva a cabo la fusión de ambas rutas. Durante el proceso de construcción y antes de la introducción de los arcos de tipo interfaz, nuestra metodología resuelve por separado los dos sub-problemas que resultan de la descomposición natural: la creación de las rutas de entrega y la creación de las rutas de recogida. Para preservar la restricción de precedencia, las rutas con clientes de recogida (*backhaul*) se agregan al final de las rutas con los clientes de entrega (*linehaul*) a través del uso de los arcos interfaz (**Figura 3.4**).

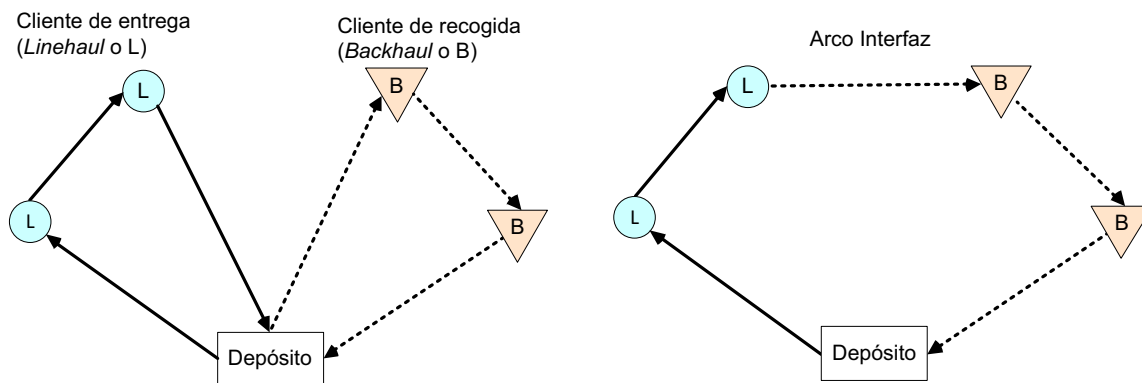


Figura 3.4: Utilización de un arco interfaz para conectar rutas de entrega y recogida.

La fusión de rutas continúa hasta que no se pueden realizar más fusiones entre rutas preservando las restricciones del problema. Entonces ya disponemos de la solución base inicial y se inicia un proceso iterativo de mejora que forma parte del esquema comentado de ILS. Se trata de aplicar una perturbación a la solución base con el objetivo de generar una nueva. Esta operación se implementa a través de la técnica de división del problema en subproblemas. Esta estrategia sigue la máxima de *decrease and conquer* (Dc&C), y realiza secuencialmente los siguientes pasos:

1. A partir de la nueva solución obtenida, se reduce el problema original con el objetivo de disminuir su dimensión. Para seleccionar el conjunto de clientes y vehículos que formaran parte del problema reducido, se elige un conjunto de rutas de acuerdo con algunas reglas aleatorias que seleccionan un número determinado de rutas. Para este subproblema, se disuelven las rutas asociadas y se extraen los clientes y vehículos asociados, para construir los datos de entrada del nuevo subproblema.
2. Comienza un proceso iterativo en el que se utiliza la metodología escogida que puede ser la misma que la del problema original, durante un periodo de tiempo que puede ser controlado por tiempo o por número de iteraciones. El objetivo es encontrar una solución “local” mejorada del subproblema correspondiente.
3. Finalmente, se adapta la solución “local” para obtener una nueva solución “global” mejorada.

Este esquema de descomposición puede verse igualmente como un proceso de perturbación en dos pasos. El primero consiste en destruir algunas de las rutas que forman la solución base,

y el segundo consiste en reconstruirlas utilizando un algoritmo eficiente equivalente al que se ha utilizado anteriormente sobre los nodos incluidos en las rutas destruidas. El número de rutas que son destruidas/reconstruidas está determinado por el porcentaje definido en el parámetro p . En el esquema de búsqueda local iterada sólo se modifican las rutas a destruir, el resto permanecen intactas. A diferencia del algoritmo utilizado para encontrar la solución base, en este momento la lista de ahorros de partida es reordenada aleatoriamente mediante una distribución de probabilidad determinada. En este caso, se emplea una distribución de probabilidad geométrica para inducir un comportamiento aleatorio sesgado. Esta distribución utiliza un solo parámetro, β ($0 < \beta < 1$). [Juan et al. \(2011c\)](#), propusieron esta metodología mediante la realización de una asignación de probabilidad sesgada de la lista de arcos. Con esta reordenación, se seleccionan los arcos en un orden diferente en cada ocasión en que el proceso es invocado, mientras que, al mismo tiempo, la lógica en la que se basa el heurístico original se mantiene, es decir, los arcos con un mayor ahorro tienen más probabilidades de ser seleccionados que aquellos en los que el ahorro es menor. Esta característica ayuda a la *intensificación* de la búsqueda ya que permite insistir en la búsqueda de soluciones próximas a la solución actual.

La solución obtenida en este momento, puede ser mejorada mediante el uso de las técnicas de memoria cache (búsqueda local) y división del problema original en subproblemas (perturbación), según se describe en [Juan et al. \(2011c\)](#). La técnica de la memoria cache es un proceso de búsqueda local rápida que compara cada una de las rutas de la solución actual, con las mejores rutas obtenidas anteriormente, y sustituye las rutas actuales por otras que contienen los mismos clientes, pero con un coste menor. En este momento se chequea la solución para determinar si cumple con las dos condiciones:

- El coste de la solución mejora el coste de la solución obtenida hasta este momento.
- El número de rutas de la solución corresponde con el tamaño de la flota disponible.

Esta segunda condición es de obligado cumplimiento para este problema y todos los autores que han resuelto el VRPCB la han considerado al solucionar las instancias del problema. En la metodología propuesta, la restricción del tamaño de la flota está relajada de forma implícita durante el proceso de búsqueda. Por tanto, esta condición tiene que ser chequeada a posteriori porque la solución propuesta puede requerir un número de rutas distinto al número de vehículos

disponibles para servir a todos los clientes. El método CWS finaliza cuando ya no quedan fusiones por realizar independientemente del número de rutas que se hayan formado (Wassan 2007).

Aprovechamos esta característica para relajar implícitamente el problema y poder permitir al proceso de búsqueda visitar nuevas regiones del espacio de soluciones y obtener soluciones no válidas. Si esto sucede sabemos que la solución no es factible en su conjunto, pero es posible que podamos seleccionar alguna de sus partes para componer una solución válida. La idea de considerar soluciones no factibles no es nueva y ha sido utilizada por varios autores (Cuervo et al, 2014, Vidal et al, 2014). Como se explica en este último artículo, una de las características que influencia el funcionamiento de un metaheurístico tiene que ver con su capacidad de evaluar las rutas creadas durante el proceso de búsqueda. Para realizar esta corrección, utilizamos un operador llamado Operador Recursivo Correctivo (*Recursive Corrective Operator* o RCO) cuyo principal objetivo es restablecer la restricción del tamaño de la flota. Hemos utilizado esta idea para desarrollar nuestro propio operador. Una vez que nuestro heurístico crea una solución, RCO evalúa cada ruta componente dependiendo de la demanda que cubre. Esta evaluación determinará qué rutas de la solución son válidas y cuáles no. Las primeras pasarán a formar parte de la solución mientras que las segundas serán destruidas y sus nodos serán objeto de posterior enrutamiento al marcarse de nuevo como no ruteados. Este proceso también se puede considerar como una perturbación. A diferencia de la perturbación de descomposición anterior, esta perturbación divide el conjunto de rutas en válidas e inválidas según la demanda total que contienen. Las inválidas se destruyen y construyen de nuevo y las válidas se mantienen.

3.4. Pseudocódigo del algoritmo utilizado.

Esta sección presenta una descripción más detallada de los procedimientos que implementan la metodología propuesta, incluyendo su pseudocódigo. El Algoritmo 3.1 muestra el pseudocódigo asociado con el marco de búsqueda local iterada, que es el componente principal de nuestro algoritmo. El procedimiento requiere sólo tres parámetros:

- el coeficiente de penalización para los enlaces de interfaz, λ .
- la que define la distribución geométrica utilizada en el proceso de aleatoriedad parcial, β .

- el porcentaje de rutas a destruir durante la etapa de perturbación, p .

Además del valor de estos parámetros, las variables de entrada (input) contienen la información relacionada con la instancia específica: demandas, capacidad del vehículo, número de vehículos, coordenadas euclidianas para todos los clientes y el depósito, número de clientes de entrega y semilla para el generador de números aleatorios. Una vez que se cargan los parámetros dentro del programa, se invoca al heurístico de CWS (línea 02) después de haber calculado la solución trivial y la lista de ahorros que se obtiene después de considerar el factor de penalización (línea 01). En este punto, comenzamos el proceso iterativo para generar nuevas soluciones que mejoren la solución base actual (línea 04). A continuación, el algoritmo comienza un esquema ILS (líneas 04-22), que tiene por objeto reducir el coste de la mejor solución obtenida mediante la combinación de una etapa de destrucción-construcción (perturbación) de la solución base con una etapa de búsqueda local (líneas 05 a 10).

En cada iteración, se realiza una perturbación de la solución actual seleccionando un porcentaje de las rutas de la solución base determinado por el valor de la variable p (línea 05). Considerando los nodos asociados a las rutas seleccionadas, se construye una nueva solución utilizando nuevamente el heurístico de CWS con aleatoriedad sesgada (línea 08). Es decir, para el problema reducido se utiliza el mismo heurístico que para el problema general. Esta aleatoriedad permite seleccionar los arcos en un orden diferente cada vez que se invoca el procedimiento. La construcción de la lista de ahorros y el mecanismo para seleccionar los arcos conservan la lógica del heurístico de ahorros. El orden sesgado de los arcos es inducido por una distribución geométrica que usa un solo parámetro β . Hemos establecido este parámetro como uniformemente aleatorio en el intervalo (0,06, 0,23). Como se ha explicado al definir la metodología, después de generar una solución, se emplea un procedimiento de memoria caché (memoria) que almacena las mejores rutas obtenidas en iteraciones anteriores (línea 09). Esto puede ser visto como un mecanismo de aprendizaje que se implementa a través de una estructura de datos clásica llamada diccionario o *hash map*. Esta estructura de datos es común en todos los lenguajes de programación modernos y generaliza el concepto de matriz dinámica utilizando claves en lugar de índices numéricos para referenciar a valores u objetos almacenados. En la práctica, esto permite la búsqueda rápida de cualquier valor / objeto a través de su clave (generalmente un código hash) evitando tener que utilizar estructuras de

almacenamiento menos eficientes. Esta memoria caché siempre mantiene el orden óptimo para visitar un conjunto de clientes obtenido en el proceso de búsqueda hasta ese momento.

Bajo determinadas circunstancias, la solución base se actualiza con la nueva solución, incluso si esta última muestra un coste más elevado (líneas 11-21). Esta degradación de la solución base se lleva a cabo de vez en cuando, con el fin de disminuir la probabilidad de que el algoritmo quede atrapado en un mínimo local. Concretamente, para aceptar esta pérdida en el resultado de la solución base se utiliza el siguiente criterio basado:

- La degradación en coste tiene que ser menor que mejora obtenida en la última actualización (representado en el pseudocódigo por la variable *credit*).
- No se permiten dos degradaciones consecutivas de la solución base. Para ello se establece la variable *credit* a 0 cada vez que se degrada la solución base.

El proceso iterativo concluye cuando se alcanza la condición final, que puede expresarse en tiempo o número de iteraciones. Como resultado, al final de cada proceso constructivo, se obtiene una solución factible (línea 23). La calidad de la solución aleatoria obtenida depende del modo en el cual han sido seleccionados los arcos, es decir, del tipo de criterio y probabilidad utilizados. Aquí es donde las distribuciones de probabilidad asimétricas, como por ejemplo la distribución geométrica, pueden ser muy útiles. En otras palabras, si se empleara una distribución de probabilidad uniforme en lugar de una distribución de probabilidad geométrica, los resultados obtenidos no serían competitivos en absoluto, ya que la lógica que sustenta el heurístico original se perdería.

Algoritmo 3.1 Pseudocódigo del procedimiento principal del ILS-BRCW.

```

Procedure ILS-BRCW(inputs,  $\lambda$ ,  $\beta$ ,  $p$ )
01  savingsList <- CreateRandomPenSavList(inputs,  $\lambda$ ,  $\beta$ )
02  baseSol <- CWS(savingsList, inputs) // aleatoriedad sesgada
03  delta <- 0 % variable auxiliar
04  while {condición final no alcanzada} do // tiempo o número de iteraciones
05      subSol <- extractRoutesAtPercentage(baseSo, p)
           // fase de destrucción
06      subInptus <- getSubProblem(subSol)
07      savingsList <- CreateRandomPenSavList(subInputs,  $\lambda$ ,  $\beta$ )
           // aleatoriedad sesgada

```

```

08     newSubSol <- CWS(savingsList, inputs)//algoritmo de ahorros CWS
09     newSubSol <- improveRoutesUsingHashTable(newSubSol)
      // cache
10     newSol <- merge(baseSol, newSubSol)
11     delta <- cost(newSol) - cost(baseSol)
12     if {delta < 0} then // mejora de baseSol
13         credit <- (-delta)
14         baseSol <- newSol
15         if {cost(newSol) < cost(bestSol)} then
16             bestSol <- newSol
17         end if
18     else if {0 < delta < credit} then // criterio de aceptación
19         credit <- 0
20         baseSol <- newSol
21     end if
22 end while
23 return bestSol
end procedure

```

El **Algoritmo 3.2** muestra el pseudocódigo para la creación de la lista de ahorros aleatoria con la penalización de los arcos interfaz. Para poder aplicar la penalización propuesta por **Deif y Bodin (1984)** en primer lugar, se calcula el valor máximo de ahorro (maxsavings). Posteriormente, este valor se multiplica por el valor del parámetro de penalización λ . El valor original de ahorro para los arcos interfaz disminuye linealmente con esta cantidad. Hemos establecido este parámetro como uniformemente aleatorio en el intervalo (0,05, 0,20). Una vez que se obtiene la lista de ahorros, se aplica la aleatorización (línea 08) y se completa el procedimiento.

Algoritmo 3.2 Creación de la lista de ahorros aleatoria con penalización.

```

procedure CreateRandomPenSavList(inputs,  $\lambda$ ,  $\beta$ )
01     savingsList <- createSavingsList // lista de ahorros CWS
02     maxSaving <- obtainMaxSaving(savingsList)
03     for each edge in savingsList do
04         if {edge is interface} then // arco que conecta los %nodos de
      recogida con los de entrega
05             currentSaving(edge) <- currentSaving(edge)*maxSaving* $\lambda$ 
06         end if
07     end for
08     savingsList <- OrderDesc(savingsList) // ordenación
09     for each edge in savingsList do //aleatorización de los ahorros
10         index <- obtainBiasedRandom(inputs, seed,  $\beta$ , #savingsList)

```

```
09         randomPenalizedSavingsList <- addEdge(removeEdge(savingsList,  
edge), index))  
10     end for  
11     return randomPenalizedSavingsList  
end
```

El **Algoritmo 3.3** muestra el pseudocódigo del RCO que se utiliza para transformar soluciones no factibles en soluciones factibles. Este operador se activa siempre que el número de rutas en una solución no coincida con el número de vehículos disponibles. El heurístico que se utiliza para guiar el proceso constructivo de la solución es el CWS, lo que implica que el número de rutas en la solución final sólo depende del proceso de fusión de rutas. Es decir, el número de rutas no es controlable en nuestra propuesta y por lo tanto, la restricción de tamaño de la flota se encuentra relajada. Como se muestra en el **Algoritmo 3.3**, inicialmente el operador calcula las rutas en la solución infactible actual que pueden ser parte de una solución factible (línea 01). Esta selección se realiza mediante un procedimiento de valoración de rutas. En el caso de que la solución contenga un número de rutas inferior al número disponible de vehículos, primero clasificaremos las rutas en orden decreciente de demanda. La condición para que cualquier ruta llegue a ser parte de la solución final es que la demanda asignada hasta ahora sea menor o igual a la demanda total que puede ser servida por los vehículos disponibles. Si este es el caso, la ruta se agrega a la solución final. Si la solución proporcionada por el algoritmo tiene más rutas que los vehículos disponibles, las rutas se ordenan en orden ascendente de demanda. En este orden, se van recorriendo todas las rutas. La condición para que una ruta sea considerada como parte de la solución final es que la demanda pendiente (considerando todas las rutas excepto las que han sido procesadas hasta el momento) pueda ser servida por los vehículos restantes (considerando todos los vehículos excepto los que han sido procesados hasta el momento). Una vez completado este proceso de selección, tenemos una solución parcial compuesta por las rutas válidas según este criterio. Para todas las rutas restantes, sus nodos están marcados como pendientes de ser incluidos en alguna ruta (línea 02) y el algoritmo es invocado de nuevo para generar una nueva solución que tiene que contener un determinado tamaño de flota: el inicial menos el número de rutas que ya forman parte de la solución (línea 05). Para ello se actualiza el número de vehículos (línea 03) y se vuelve a calcular la lista de ahorros para los nodos pendiente de incluir en una ruta (línea 04). Si la solución es factible en

términos de tamaño de la flota (línea 06), se devuelve la solución y el proceso termina. De lo contrario, el procedimiento es invocado recursivamente.

Algoritmo 3.3 Operador Recursivo Correctivo.

```

procedure RCO(solution, K,  $\lambda$ ,  $\beta$ )
01   parcSolution <- SelectRoutes(solution, K)
02   pendingNodes <- ExtractNodes(solution, parcSolution)
03   numVehicles <- numVehicles - #Routes(parcSolution)
04   savingsList <- CreateRandomPenalizedSavingsList(inputs(pendingNodes),
 $\lambda$ ,  $\beta$ )
05   newSol <- CWS(savingsList, inputs(pendingNodes))
06   if (#Routes(newSol) = numVehicles) then
07       return (newSol)
08   else
09       return (parcSolution & RCO(newSol, K,  $\lambda$ ,  $\beta$ ))
end

```

3.5. Experimentos computacionales.

El algoritmo descrito en esta sección se ha implementado como una aplicación Java. Se ha generado la aleatoriedad utilizando la biblioteca SSJ que se obtiene a través de [L'Ecuyer et al \(2002\)](#). Se utilizó un ordenador portátil con una CPU Intel Core i7 a 2,79 GHz y 3,42 GB de RAM para ejecutar todos los experimentos. El programa compilado se ejecutó bajo Netbeans IDE sobre MS Windows 7. Un algoritmo implementado en Java se ejecuta algo más lento que el mismo algoritmo implementado en C/C++ optimizado. Según algunos estudios ([Hundt, 2011](#)), el código Java podría ser aproximadamente 6 veces más lento que el código C/C++. Sin embargo, Java ofrece otras ventajas para el desarrollo de algoritmos, tales como una implementación rápida y una mayor reproducibilidad de los resultados (el rendimiento de un código implementado en C/C++ suele depender más de las habilidades del programador, así como de la configuración específica de la versión compilada a ejecutarse). Hemos utilizado el conjunto de instancias GJB de 62 casos introducidos por [Goetschalckx y Jacobs-Blecha \(1989\)](#) y el conjunto TV de 33 casos introducidos por [Toth y Vigo \(1999\)](#) para probar la efectividad del método propuesto. Para las instancias del primer conjunto, el depósito está situado en coordenadas (x, y) donde $x=12000$ e $y=16000$). La ubicación de los clientes se genera aleatoriamente, usando una distribución de probabilidad uniforme, con el valor de la

coordenada x variando entre 0 y 24.000, y el valor de la coordenada y variando entre 0 y 32.000. Las demandas se generan aleatoriamente a partir de una distribución normal con una media de 500 y una desviación estándar de 200. Se consideran 14 valores para el número total de clientes (entre 25 y 150), siendo el porcentaje de clientes de linehaul igual a 50%, 66% y 80% del total. La capacidad del vehículo se asigna de tal manera que entre 3 y 12 vehículos puedan atender todas las demandas. Las instancias están disponibles en el siguiente sitio web:

<http://www2.isye.gatech.edu/~mgoetsch/lineback.html>. (último acceso 7 de Febrero 2017).

En cuanto a las instancias del segundo conjunto, TV, la fuente original son los datos para VRP inicialmente propuestos por **Christofides y Eilon (1969)**. Siguiendo las convenciones usadas por los autores, se generan tres nuevas instancias usando una mezcla entrega (*linehaul* o LH) / recogida (*backhaul* o BH) (LH / BH) de 50/50, 67/33 y 80/20, respectivamente (es decir, el primer cliente cada dos, tres, o cinco clientes se considera un cliente *backhaul*). Las distancias entre cada par de clientes se han calculado utilizando notación de doble precisión. Además, para las instancias TV, cada distancia se ha redondeado al número entero más cercano. Esta ha sido una convención utilizada por la mayoría de los algoritmos para VRPB en la literatura (**Cuervo et al., 2014**).

A continuación, discutiremos los valores que se han asignado a cada uno de los parámetros de nuestro algoritmo. En el caso del parámetro β , que define la distribución geométrica específica que se va a usar durante el proceso de aleatoriedad sesgada, lo hemos establecido como uniformemente aleatorio en el intervalo (0,06; 0,23), como sugiere **Juan et al. (2010)**. Con respecto al parámetro λ , hemos establecido este parámetro como uniformemente aleatorio en el intervalo (0,05; 0,20), como sugieren **Deif y Bodin (1984)**. En el caso del porcentaje de destrucción durante la etapa de perturbación, p , hemos establecido este parámetro como uniformemente aleatorio en el intervalo (0,10; 0,50). Hemos seleccionado valores muy generales para estos parámetros, sin necesidad de realizar procesos complejos y costosos en tiempo de ajuste fino. Finalmente, el tiempo de terminación se ha fijado en 25 segundos para instancias de hasta 50 nodos, hasta 75 segundos para instancias de hasta 100 nodos y de 300 segundos para instancias con más de 100 nodos.

La **Tabla 3.1** muestra los resultados obtenidos para las instancias GJB, mientras que la **Tabla 3.2** muestra los resultados obtenidos para las instancias TV. La información básica

contenida en cada tabla es: nombre de la instancia, número de clientes de entrega, número de clientes de recogida, capacidad del vehículo (Q), número de rutas en la solución (K), el coste de nuestra solución mejor y media después de 10 ejecuciones (Best10 y Avg10), y la distancia (en porcentaje) entre nuestra mejor solución (*Our Best Solution* o OBS) y la mejor solución conocida (*Best Known Solution* o BKS) en la literatura. Además, para ilustrar la contribución del proceso de aleatoriedad sesgada a la calidad de las soluciones obtenidas, se han incluido dos columnas con título NOBR (No Bias Randomization o No aleatoriedad sesgada) en la [Tabla 3.1](#) y en la [Tabla 3.2](#). La primera columna contiene la diferencia porcentual entre nuestra solución y la mejor solución conocida obtenida sin usar aleatoriedad sesgada. En otras palabras, en este caso la lista de ahorros es ordenada por el valor de ahorro y siguiendo un estricto criterio de mayor a menor. La segunda columna representa la diferencia correspondiente cuando se utiliza el proceso de aleatoriedad sesgada descrito en este documento. Obsérvese que cuando no se emplea el proceso aleatorio, entonces:

- La distancia promedio con respecto a las BKS se incrementa en casi el 6%.
- Hay 24 instancias no se resuelven adecuadamente (en términos del número de vehículos requeridos), 20 en las instancias GJB y 4 en las instancias TV.

Instance	n-lh	n-bh	Q	K	NoBR	Gap NoBR - Best10	Best10		Avg10		Gap:(Best10-BKS)/BKS
							cost	time (s)	cost	time (s)	
A1	20	5	1.550	8	230.730	0,37%	229.886	1	229.886	1	0,00%
A2	20	5	2.550	5	185.228	2,76%	180.119	1	180.119	1	0,00%
A3	20	5	4.050	4	-	-	163.405	3	163.405	3	0,00%
A4	20	5	4.050	3	169.723	8,21%	155.796	4	155.796	4	0,00%
B1	20	10	1.600	7	257.426	7,13%	239.080	1	239.080	1	0,00%
B2	20	10	2.600	5	211.496	6,36%	198.048	2	198.048	2	0,00%
B3	20	10	4.000	3	178.343	5,03%	169.372	2	169.372	2	0,00%
C1	20	20	1.800	7	266.187	5,87%	250.557	2	250.557	2	0,00%
C2	20	20	2.600	5	216.996	0,91%	215.020	2	215.020	2	0,00%
C3	20	20	4.150	5	-	-	199.346	3	199.346	3	0,00%
C4	20	20	4.150	4	203.987	4,23%	195.367	1	195.367	1	0,00%
D1	30	8	1.700	12	-	-	322.530	1	322.530	1	0,00%
D2	30	8	1.700	11	336.598	5,91%	316.709	1	316.709	1	0,00%
D3	30	8	2.750	7	243.846	1,79%	239.479	4	239.479	4	0,00%
D4	30	8	4.075	5	208.685	1,37%	205.832	2	205.832	2	0,00%
E1	30	15	2.650	7	244.487	2,29%	238.880	2	238.880	2	0,00%
E2	30	15	4.300	4	218.597	2,90%	212.263	1	212.263	1	0,00%
E3	30	15	5.225	4	216.307	4,46%	206.659	1	206.659	1	0,00%
F1	30	30	3.000	6	292.802	10,12%	263.174	3	263.274	3	0,00%
F2	30	30	3.000	7	-	-	265.214	4	265.655	4	0,00%
F3	30	30	4.400	5	258.238	6,63%	241.121	5	241.121	5	0,00%
F4	30	30	5.500	4	245.621	4,79%	233.862	1	234.606	1	0,00%
G1	45	12	2.700	10	-	-	306.305	9	306.421	16	0,00%
G2	45	12	4.300	6	250.104	1,86%	245.441	1	245.441	1	0,00%
G3	45	12	5.300	5	241.383	4,92%	229.507	1	230.736	5	0,00%
G4	45	12	6.400	6	-	-	232.521	7	232.642	16	0,00%
G5	45	12	6.400	5	-	-	221.730	1	221.730	4	0,00%

3.- El problema de las rutas de vehículos con retornos agrupados (VRPCB).

G6	45	12	8.000	4	233.665	8,65%	213.457	17	215.241	6	0,00%
H1	45	23	4.000	6	285.618	5,84%	268.933	2	268.933	12	0,00%
H2	45	23	5.100	5	271.493	6,68%	253.365	5	253.462	27	0,00%
H3	45	23	6.100	4	269.395	8,15%	247.449	6	247.593	25	0,00%
H4	45	23	6.100	5	-	-	250.221	3	251.984	9	0,00%
H5	45	23	7.100	4	263.617	6,64%	246.121	9	246.169	24	0,00%
H6	45	23	7.100	5	-	-	249.135	4	250.178	19	0,00%
I1	45	45	3.000	10	-	-	350.245	17	351.245	14	0,00%
I2	45	45	4.000	7	320.635	3,33%	309.944	14	310.468	45	0,00%
I3	45	45	5.700	5	-	-	294.507	23	296.785	42	0,00%
I4	45	45	5.700	6	309.547	4,38%	295.988	2	296.920	13	0,00%
I5	45	45	5.700	7	-	-	301.236	39	302.649	19	0,00%
J1	75	19	4.400	10	345.697	3,09%	335.007	39	336.009	30	0,00%
J2	75	19	5.600	8	326.193	4,84%	310.417	47	310.871	23	0,00%
J3	75	19	8.200	6	294.206	5,09%	279.219	62	282.613	53	0,00%
J4	75	19	6.600	7	312.163	4,87%	296.947	32	297.088	40	0,14%
K1	75	38	4.100	10	423.361	6,92%	394.071	61	395.041	63	0,00%
K2	75	38	5.200	8	396.466	8,66%	362.130	105	364.830	99	0,00%
K3	75	38	5.200	9	-	-	365.694	112	372.452	90	0,00%
K4	75	38	6.200	7	378.999	7,93%	348.949	38	351.035	64	0,00%
L1	75	75	4.400	10	-	-	418.309	143	422.131	57	0,10%
L2	75	75	5.000	8	443.219	9,26%	402.155	85	408.817	176	0,23%
L3	75	75	5.000	9	-	-	405.083	89	407.730	128	0,60%
L4	75	75	6.000	7	412.740	6,81%	384.636	173	387.857	188	0,00%
L5	75	75	6.000	8	-	-	387.565	96	391.740	67	0,00%
M1	100	25	5.200	11	-	-	399.628	122	405.337	40	0,26%
M2	100	25	5.200	10	436.978	9,06%	397.407	91	401.905	128	0,12%
M3	100	25	6.200	9	409.948	8,24%	375.864	119	378.430	24	0,04%
M4	100	25	8.000	7	387.304	10,11%	348.140	71	351.850	100	0,00%
N1	100	50	5.700	11	-	-	408.101	111	416.326	264	0,00%
N2	100	50	5.700	10	465.112	12,27%	408.065	166	414.607	167	0,00%
N3	100	50	6.600	9	437.896	9,95%	394.338	109	402.158	120	0,00%
N4	100	50	6.600	10	-	-	394.788	331	402.362	267	0,00%
N5	100	50	8.500	7	411.866	8,75%	375.835	261	378.328	150	0,63%
N6	100	50	8.500	8	-	-	375.803	174	382.981	156	0,55%
Average	53	28	4.847	7	-	5,89%	290.742	46	292.486	46	0,04%

Tabla 3.1: Resultados para las instancias GJB.

Instance	n-lh	n-bh	Q	K	NoBR	Gap NoBR - Best10	Best10		Avg10		Gap:(Best10-BKS)/BKS
							cost	time (s)	cost	time (s)	
eil22 50	11	10	6.000	3	378	1,85%	371	1	371	1	0,00%
eil22 66	14	7	6.000	3	413	11,38%	366	3	366	3	0,00%
eil22 80	17	4	6.000	3	383	2,09%	375	2	375	2	0,00%
eil23 50	11	11	4.500	2	728	6,32%	682	5	682	5	0,00%
eil23 66	15	7	4.500	2	704	7,81%	649	1	649	1	0,00%
eil23 80	18	4	4.500	2	670	7,01%	623	1	623	1	0,00%
eil30 50	15	14	4.500	2	539	7,05%	501	1	501	1	0,00%
eil30 66	20	9	4.500	3	566	5,12%	537	2	539	2	0,00%
eil30 80	24	5	4.500	3	557	7,72%	514	2	514	2	0,00%
eil33 50	16	16	8.000	3	776	4,90%	738	3	739	3	0,00%
eil33 66	22	10	8.000	3	811	7,52%	750	2	750	2	0,00%
eil33 80	26	6	8.000	3	742	0,81%	736	4	736	4	0,00%
eil51 50	25	25	160	3	591	5,41%	559	9	561	9	0,00%
eil51 66	34	16	160	4	576	4,86%	548	5	548	5	0,00%
eil51 80	40	10	160	4	579	2,42%	565	4	565	4	0,00%
eilA76 50	37	38	140	6	791	6,57%	739	2	740	19	0,00%
eilA76 66	50	25	140	7	800	4,00%	768	16	768	35	0,00%
eilA76 80	60	15	140	8	862	9,40%	781	80	794	41	0,00%
eilB76 50	37	38	100	8	875	8,46%	801	4	802	21	0,00%
eilB76 66	50	25	100	10	935	6,63%	873	13	875	14	0,00%
eilB76 80	60	15	100	12	967	4,96%	919	20	930	11	0,00%
eilC76 50	37	38	180	5	751	5,06%	713	6	715	13	0,00%
eilC76 66	50	25	180	6	777	5,53%	734	1	736	8	0,00%
eilC76 80	60	15	180	7	770	4,81%	733	14	738	17	0,00%
eilD76 50	37	38	220	4	736	6,25%	690	6	690	31	0,00%

eilD76 66	50	25	220	5	762	6,17%	715	2	717	10	0,00%
eilD76 80	60	15	220	6	722	3,88%	694	6	702	19	0,00%
eilA101 50	50	50	200	4	-	-	831	46	846	85	0,00%
eilA101 66	67	33	200	6	911	7,14%	846	44	860	89	0,00%
eilA101 80	80	20	200	6	-	-	856	41	872	85	0,00%
eilB101 50	50	50	112	7	-	-	923	23	935	34	0,00%
eilB101 66	67	33	112	9	-	-	982	27	996	30	-0,10%
eilB101 80	80	20	112	11	1069	5,71%	1008	99	1022	65	0,00%
<i>Average</i>	39	20	103	5		5,75%	700.6	15	704.8	20	0,00%

Tabla 3.2: Resultados para las instancias TV.

3.6. Discusión de resultados y comparación con otros planteamientos.

3.6.1. Resultados obtenidos.

Nuestro algoritmo ha mejorado la BKS de la instancia EILA101_66 encontrado una nueva mejor solución. Además, es capaz de obtener soluciones muy competitivas en tiempos de computación muy cortos. Para las instancias GJB la distancia promedio con respecto a las BKS y los tiempos de cálculo son 0,04% y 46 segundos, respectivamente. También se han obtenido buenos resultados para las instancias TV, con una distancia promedio de 0,00% y 15 segundos. En este punto, es digno de recordar que las BKS son proporcionadas por diferentes algoritmos en la literatura, lo que significa que nuestro enfoque parece ser bastante competitivo en comparación con cada algoritmo individual. Para confirmar esta afirmación, hemos comparado directamente el rendimiento de nuestro algoritmo con el rendimiento de los siguientes metaheurísticos de última generación: la búsqueda tabú (BTS) de [Brandão \(2006\)](#), la búsqueda de amplios vecindarios (LS) de [Ropke y Pisinger \(2006\)](#), la búsqueda tabú reactiva con adaptación reactiva de memoria (RTS-AMP) de [Wassan \(2007\)](#), sistema de colonias de hormigas (MACS) de [Gajpal y Abad \(2009\)](#), el algoritmo de rutas prometedoras (RPA) de [Zachariadis y Kiranoudis \(2012\)](#), el algoritmo multi-atributo de propósito general (UHGS) de [Vidal et al. \(2014\)](#), la búsqueda local iterada (ILS-1000) de [Cuervo et al. \(2014\)](#) y el algoritmo de búsqueda de vecindarios variables de dos niveles de [Wassan et al. \(2016\)](#). Como muestran la [Figura 3.5](#) y la [Figura 3.6](#), todos estos metaheurísticos proporcionan resultados bastante similares y todos ellos están muy cerca de los valores de las BKS. Parece que los algoritmos RTS-AMP, BTS y LNS funcionan peor que el resto para el conjunto GJB y los mejores son los algoritmos RPA, ILS y UHGS. En el caso del conjunto TV, nuestro algoritmo ILS-BRCW (BR) es el que proporciona los mejores resultados, junto con el ILS y el VNS de dos niveles.

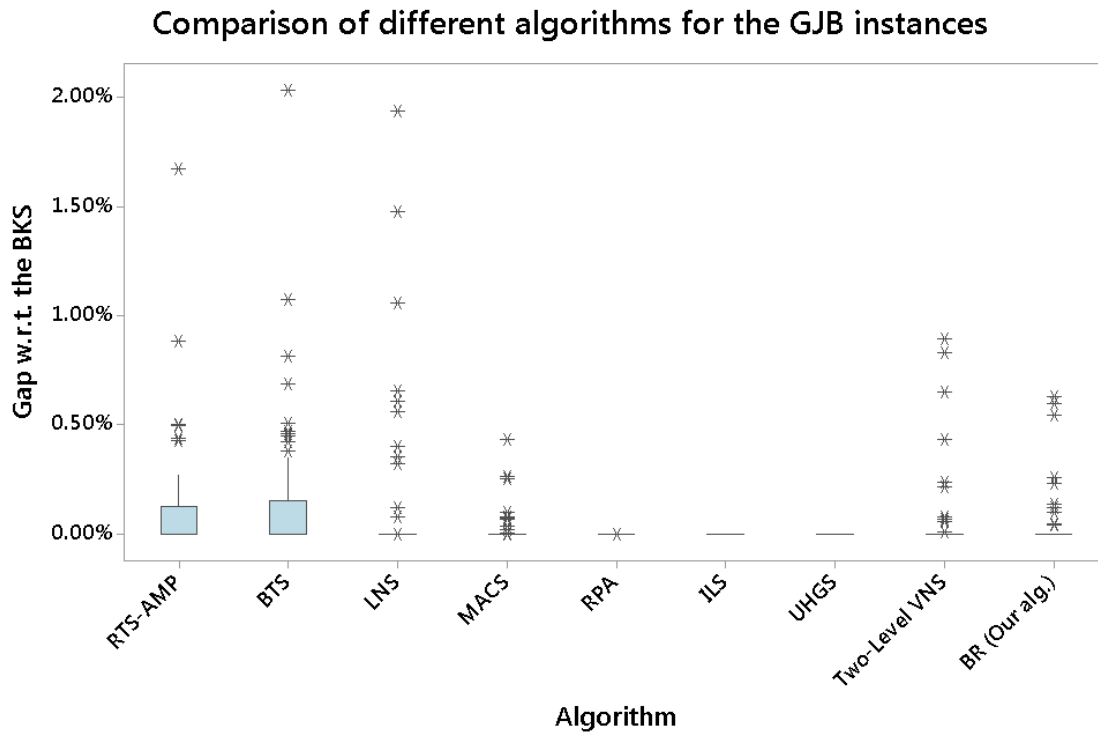


Figura 3.5: Comparación de la calidad de las soluciones para el conjunto de instancias GJB.

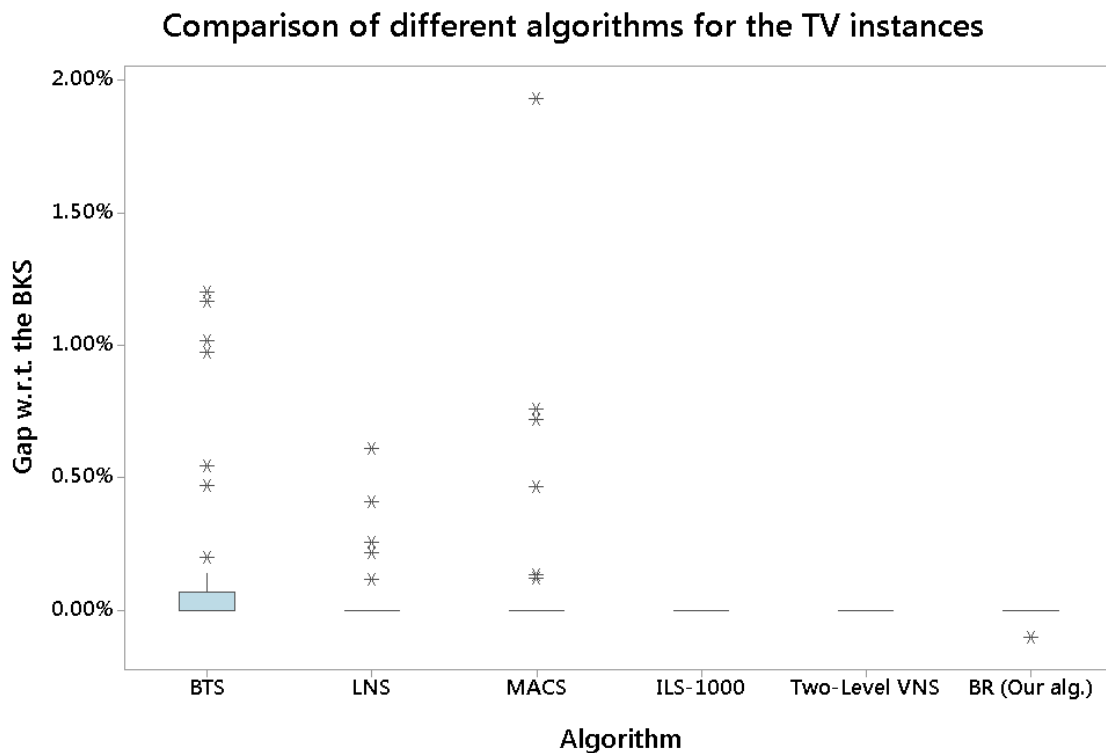


Figura 3.6: Comparación de la calidad de las soluciones para el conjunto de instancias TV.

3.6.2. Tiempos computacionales.

Para hacer una comparación equitativa de los tiempos computacionales empleados por cada algoritmo, hay varios factores objetivos que deben considerarse, por ejemplo:

- Procesador (CPU).
- Memoria.
- Sistema operativo.
- Lenguaje de programación.
- Habilidad del programador.

La única información que se ha reportado por todos los autores en sus trabajos es el nombre del procesador y el lenguaje de programación. Con esta información se puede hacer una comparación de rendimiento relativamente justa utilizando el sistema de referencias propuesto en <https://www.cpubenchmark.net/>. Uno de los objetivos de esta aplicación es comparar el rendimiento de un ordenador con otros que utilizan un procesador diferente. La web antes mencionada asigna una marca promedio a cada clase de CPU que se calcula mediante los informes de miles de pruebas de rendimiento reportadas por los usuarios de la aplicación. En nuestro caso, hemos ejecutado el programa en nuestro ordenador y la máxima marca de CPU obtenida ha sido de 2.703 unidades. Siempre que esté disponible, para el resto de los procesadores de los autores con los que nos comparamos, utilizamos la marca de la CPU que aparece en el sitio web antes mencionado. Para aquellos ordenadores no incluidos en la web, se emplearon los mismos factores de corrección que en [Cuervo et al. \(2014\)](#). Finalmente, siguiendo las conclusiones de [Hundt \(2011\)](#) sobre la velocidad de Java frente a C / C ++, aplicamos un factor de corrección (muy conservador) de 0,5 a los tiempos obtenidos con nuestro código Java. La [Figura 3.8](#) resume los resultados obtenidos. Se observa que los tiempos de cálculo mostrados por nuestro algoritmo ILS-BRCW (BR) son bastante competitivos. En este punto, es importante destacar que una de sus principales fortalezas es su relativa simplicidad, incluyendo el hecho de que sólo se necesitan tres parámetros para obtener resultados competitivos en tiempos de computación muy bajos.

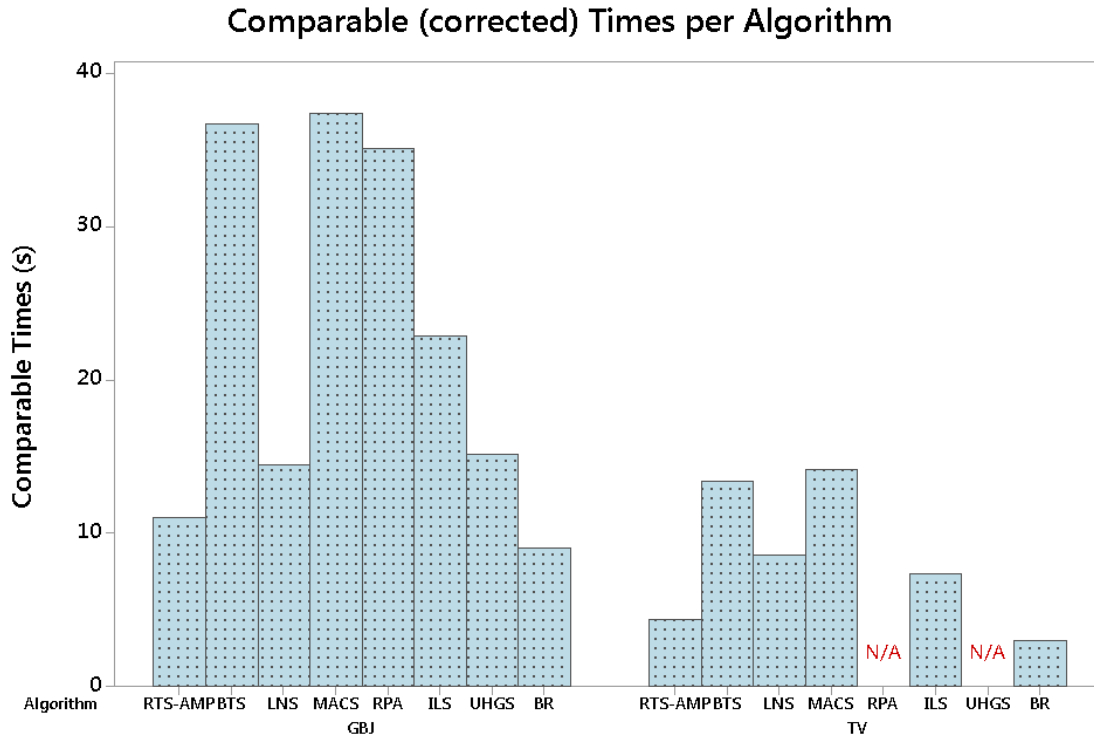


Figura 3.7: Comparación de los tiempos computacionales entre distintos algoritmos.

3.7. Conclusiones y relación del VRPCB con el grado de Cooperación Horizontal

Este capítulo se centra en el problema de rutas de vehículos con retornos agrupados, que se encuentra en aplicaciones prácticas de muchas actividades de transporte del mundo real. La metodología que se presenta combina un marco metaheurístico con una versión aleatoria (orientada) del conocido heurístico de ahorros. Este proceso de aleatorización permite la generación rápida de diferentes soluciones prometedoras cada vez que se ejecuta. También se introduce un operador que corrige el número de rutas de la solución obtenida, si no cumple con el tamaño fijo de la flota.

Se ha llevado a cabo una serie extensa de experimentos con los dos conjuntos de instancias de referencia disponibles en la literatura. De acuerdo con estas pruebas, nuestro algoritmo es capaz de igualar las BKS en la mayoría de los casos en tiempos de ejecución razonablemente bajos. Además, se ha mejorado una BKS para una de las instancias instancia del conjunto TV. El algoritmo presentado aquí es relativamente fácil de implementar, con sólo tres parámetros, y es fácilmente paralelizable. Estas propiedades computacionales hacen de este procedimiento

una alternativa interesante a otros enfoques que son más complejos, requieren más procesos de ajuste y son más difíciles de implementar en la práctica. De esta forma, hemos visto el potencial de la utilización de la aleatoriedad aplicada a un heurístico clásico.

En su vertiente cooperativa, como hemos visto en la revisión del estado del arte, las prácticas de backhauling suponen un primer paso básico para el establecimiento de la cooperación entre empresas. En este caso, se integran los dos procesos principales de la distribución de materiales:

- Reparto.
- Abastecimiento.

Para el modelo presentado en este capítulo, la integración se produce de la forma más básica ya que, aunque la misma ruta puede contener visitas a clientes de ambas naturalezas, estas visitas se realizan de forma agrupada. El abastecimiento en caso de realizarse en una ruta se realiza sólo cuando el reparto ha concluido.

La **Figura 3.8** representa una estructura por capas según el grado de cooperación que se quiere alcanzar. La capa inicial es la más básica donde tenemos el CVRP que optimiza la carga de los camiones para conseguir beneficios entre los participantes. En la siguiente capa tenemos el modelo que hemos solucionado en el presente capítulo. El VRPCB representa el grado de cooperación en que los participantes aprovechan los trayectos vacíos de los vehículos en su retorno al depósito para recoger mercancías de clientes de este tipo. Dos o más entidades cooperan según este modelo compartiendo clientes de entrega y recogida con la restricción de que las recogidas se hagan al final de la ruta. La capa que envuelve a la anterior representa el siguiente paso de la mezcla de clientes de entrega y clientes de recogida. En este caso, las visitas a clientes de entrega y recogida se pueden visitar en cualquier secuencia, este modelo es el VRPMB que será presentado en el siguiente capítulo. Estos dos modelos representan el tipo de cooperación en que se comparten los clientes. En la siguiente capa, adicionalmente se comparte la flota que es altamente probable que será heterogénea. El modelo es el conocido como HVRPB y será incluido en el capítulo 5. Finalmente, el modelo de las rutas de vehículos con retornos y depósitos múltiples (*Multi-Depot Vehicle Routing Problem with Backhauls* o MDHVRPB), supone el grado máximo de cooperación en que los participantes comparten clientes, flota y también los depósitos.

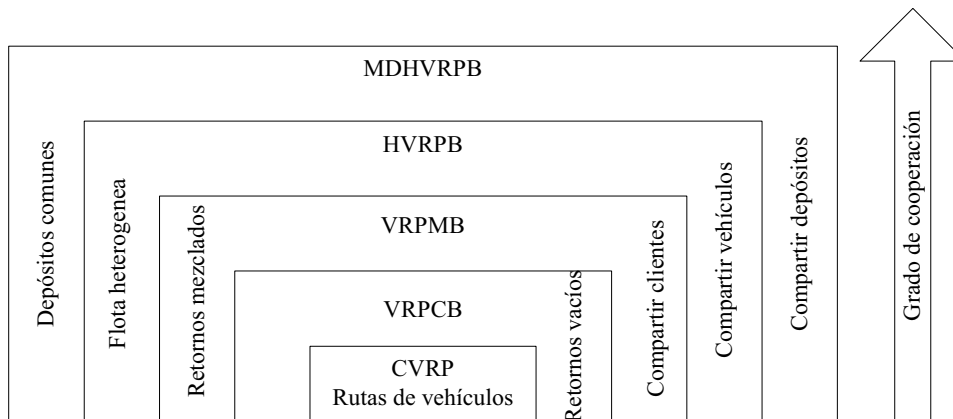


Figura 3.8: Grado de cooperación horizontal utilizando los modelos de VRP.

En la [Figura 3.9](#) se puede ver la comparación de un escenario de no cooperación con uno en el que se coopera al nivel más básico, utilizando los trayectos vacíos de vuelta al depósito para recoger material en clientes de este tipo. Como se puede ver en la [Figura 3.9](#), en un entorno no cooperativo, los clientes de distinta naturaleza se visitan en rutas distintas y los procesos de generación de rutas y operación están separados. En un entorno cooperativo con la misma distribución de clientes y considerando el mismo depósito para ambos (esta es la forma más básica), el número de rutas y por tanto vehículos se reduce ya que se utiliza el retorno de los vehículos vacíos al depósito para visitar a los clientes de recogida y utilizar la capacidad sobrante. El caso actual en que los clientes están agrupados se presenta cuando la topología de los clientes lo permite. Necesitamos solucionar el problema que se presenta en este capítulo para abordar el caso más básico. En caso de que los clientes se puedan visitar en cualquier orden habrá que ajustar la metodología presentada de acuerdo a las nuevas restricciones. Esto lo veremos en el capítulo que viene a continuación que nos permitirá hacer un análisis del tipo qué-pasa-si (*what-if*) donde podremos evaluar la solución del problema agrupado versus el problema mezclado.

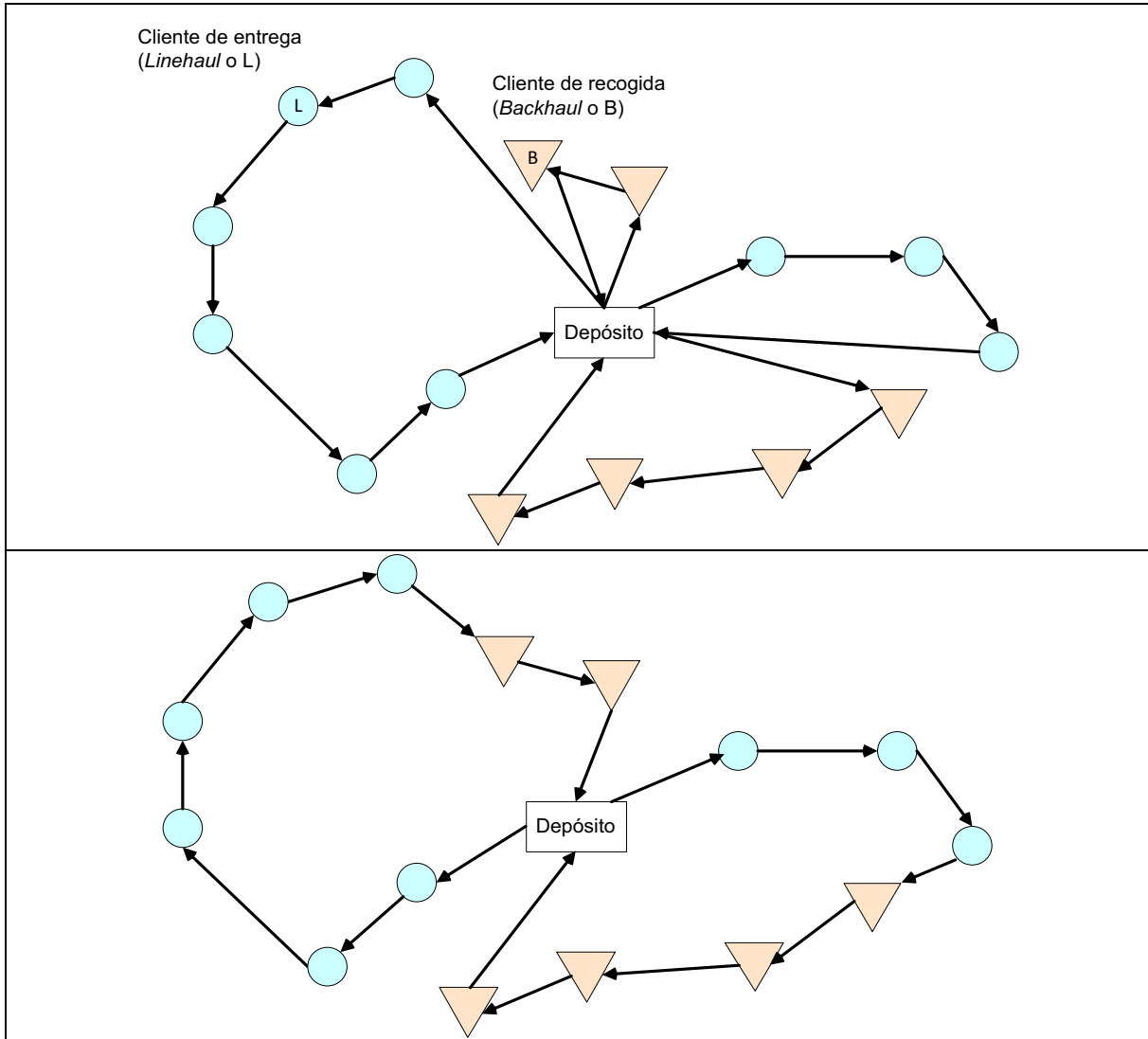


Figura 3.9: No cooperativo (arriba) vs. cooperativo (abajo) utilizando el VRPCB.

Hay muchos casos prácticos reales en empresas. En la sección 3.2.5 se detallan algunos de estos casos prácticos que utilizan este modelo en sus prácticas de CH.

Capítulo 4.

El Problema de la Rutas de Vehículos con Retornos Mezclados (VRPMB).

Partes de este capítulo se han extraído de la publicación aceptada en la revista Networks:

Belloso, J., Juan, A., Faulin, J. y Martínez, E. 2017. A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls. Networks, edición especial de los mejores artículos del congreso Verolog 2015.

Editores:

Bruce Golden

Richard Hartl

4.1. Introducción.

El problema de rutas de vehículos con retornos mezclados (*Vehicle Routing Problem with Mixed Backhauls* o VRPMB), es una extensión realista del problema clásico de rutas de vehículos donde existen demandas, tanto de entrega como de recogida. Continuando con el problema de rutas de vehículos con retornos, en este capítulo se aborda el caso complementario al del capítulo anterior en que las visitas a los clientes estaban agrupadas. En el problema actual, la visitas a los clientes de entrega y recogida puede realizarse en cualquier secuencia dentro de la ruta. Este problema se suele presentar en los casos que o bien las mercancías no son pesadas ni voluminosas, o bien las áreas de carga y descarga están separadas.

Siguiendo la lógica de proponer un esquema metaheurístico basado en la aleatoriedad del algoritmo de ahorros, en este caso se presenta un algoritmo que sigue el esquema de los arranques múltiples con el objetivo de simplificar el método lo más posible reduciendo el número de parámetros requeridos al máximo. Se utiliza un grupo de instancias conocidos en la literatura para contrastar la eficiencia del método propuesto y se constata que los resultados obtenidos son competitivos en calidad y tiempo de ejecución.

4.2. Formulación del problema.

El VRPMB es una nueva extensión del problema de rutas de vehículos con retorno, en el cual se admite la visita a los clientes de recogida en cualquier momento de la ruta. En esta versión del problema, el número de rutas utilizadas no está restringido. Tiene una formulación equivalente a la expuesta en la sección 4.1 donde se han excluido las restricciones siguientes:

- K es el número de vehículos con capacidad Q están disponibles en el vértice v_0 que corresponde al depósito central.
- Cada ruta de la solución contiene a lo sumo un arco $(v_i, v_j) \in A$ que conecta un cliente de entrega con uno de recogida.
- Hay al menos un arco $(v_i, v_j) \in A$ en cada ruta de la solución que contiene un cliente de entrega.

El objetivo del VRPMB es diseñar un conjunto de rutas de tal forma que se minimice el coste total del conjunto de rutas generadas permitiendo que cada ruta puede visitar a clientes de entrega y clientes de recogida en cualquier orden secuencial.

4.3. Metodología.

Con el objetivo de simplificar lo más posible el esquema ILS mostrada en el capítulo anterior, hemos adaptado su estructura para conseguir reducir lo más posible el número de parámetros utilizados. Hemos pasado de tres parámetros en el esquema de ILS que son λ , β y p , a sólo un parámetro en nuestro nuevo esquema. De esta forma, el algoritmo propuesto para solucionar el problema presentado en este capítulo sólo considera un parámetro, β , el

relacionado con la distribución de probabilidad que se utiliza para proporcionar la característica sesgada a la lista de ahorros utilizada. Podría utilizarse una distribución que no tiene ningún parámetro (la triangular descendente) pero los experimentos iniciales han mostrado un comportamiento menos eficiente en las soluciones. Para ello, hemos pasado de un esquema ILS a un esquema multi-arranque (*Multi-start* o MS).

En el algoritmo ILS-BRCW mostrado en el capítulo anterior se trabajaba con una solución base que se va modificando durante las diferentes iteraciones, sin embargo, en el caso del algoritmo MS-BRCW se realiza un reinicio completo de la solución inicial en cada iteración. En el caso de ILS dicha solución base se actualiza en cada iteración cuando se cumple el criterio de aceptación establecido. Mantener una solución base en cada iteración, supone partir de una solución de calidad en lugar de una nueva solución aleatoria generalmente, lo cual implica normalmente un menor grado de diversificación y un mayor grado de intensificación del algoritmo ILS-BRCW respecto al MS-BRCW.

El diagrama de flujo de alto nivel de nuestro algoritmo se muestra en la [Figura 4.1](#). En primer lugar, se obtiene una solución trivial en la que se asigna un viaje de ida y vuelta desde el depósito a cada cliente. Una vez que tenemos una solución inicial, comienza el proceso de multi-arranque. La lista de ahorros se altera introduciendo la aleatoriedad sesgada de una forma equivalente a como se ha mostrado en el capítulo anterior. Como se discutió en [Juan et al \(2010\)](#), al realizar una asignación aleatoria de la lista de ahorros, los arcos se seleccionan en un orden diferente en cada iteración del proceso. En nuestro caso, se emplea una distribución geométrica para proporcionar el comportamiento aleatorio sesgado. La distribución utiliza un parámetro beta ($0 < \beta < 1$). Mientras que el algoritmo de CWS siempre elige el arco con el mayor ahorro, el enfoque presentado, en cambio, asigna una probabilidad de selección a cada arco de la lista.

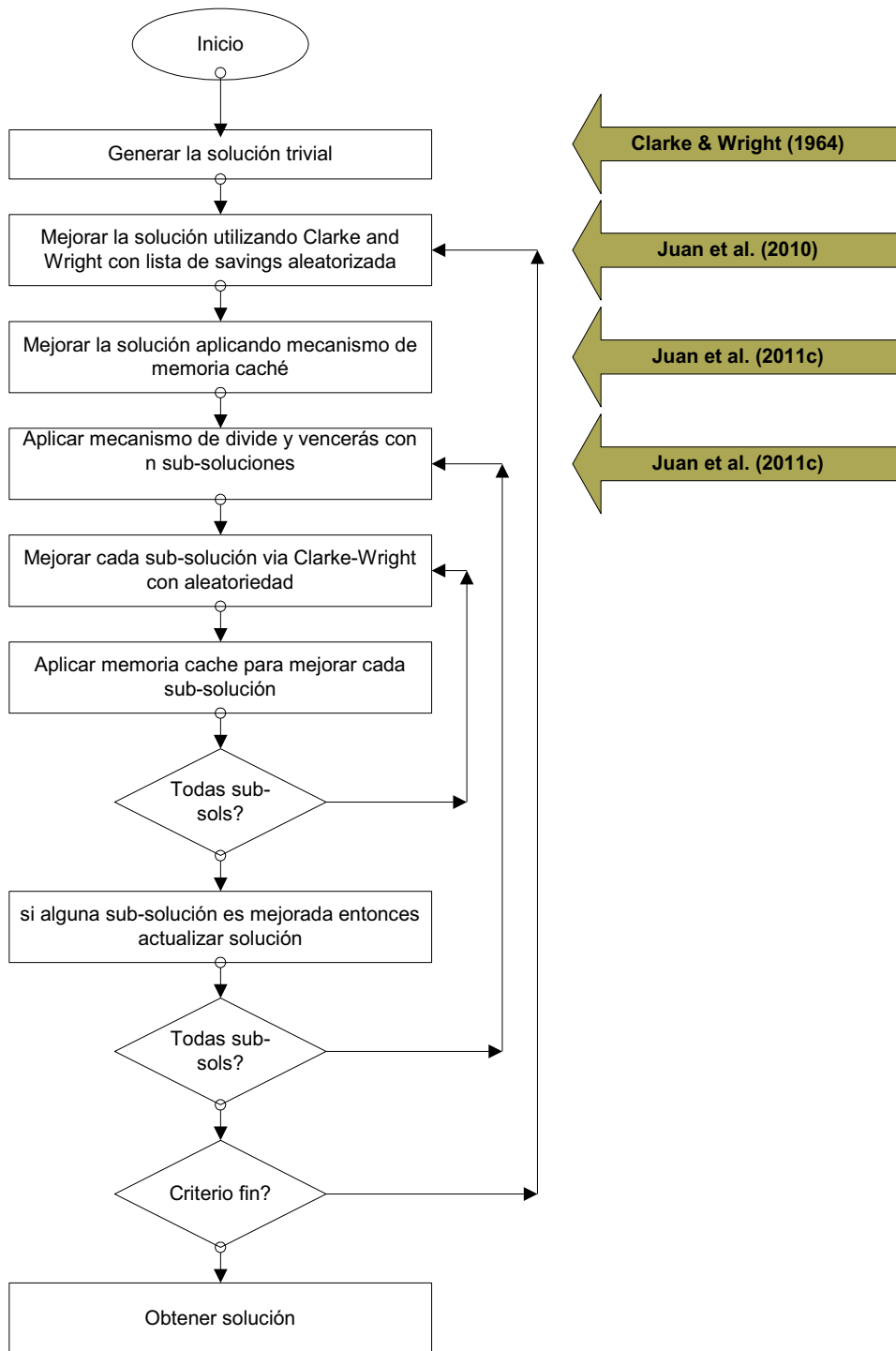


Figura 4.1: Diagrama de la metodología propuesta para la resolución del VRPMB.

Además, esta probabilidad debería ser coherente con el valor del ahorro asociado a cada arco, es decir, los arcos con ahorros mayores tienen una mayor probabilidad de ser seleccionados de la lista que aquellos con ahorros menores. De lo contrario, si, por ejemplo, se

utiliza una distribución de probabilidad uniforme, la lógica que existe detrás del heurístico se pierde y la solución resultante no es competitiva en absoluto. En este punto, con la lista de ahorros preparada se inicia el proceso de fusión de rutas siguiendo la metodología de CWS. Cuando este proceso finaliza, una nueva solución factible está lista. A diferencia del algoritmo metaheurístico presentado en el capítulo 3, la solución actual es factible y no hay que realizar ninguna comprobación adicional de forma explícita. Esto se debe a que la restricción relajada en el capítulo anterior para el VRPCB (tamaño de la flota) ya no es válida por definición para el VRPMB y se ha convertido en una variable de decisión. Esta solución puede ser mejorada usando la técnica de memoria caché descrita en la sección 4.3. El procedimiento de memoria caché debe adaptarse al considerarse en este caso que los clientes de recogida y entrega pueden alternarse en cualquier secuencia. Para evitar la creación de rutas ilegales, cuando el método propone una secuenciación de las visitas que mejora la anterior, debe realizarse de forma explícita la comprobación de que la restricción de capacidad del vehículo sigue siendo respetada, ya que una ruta puede ser factible en un sentido, pero no en el sentido contrario. Esta memoria se implementa utilizando una tabla hash indexada que permite un acceso rápido a los registros. Este caché se actualiza constantemente siempre que se encuentre un orden mejor con un coste menor para un conjunto dado de nodos.

Después de esta mejora, se introduce el uso de políticas de división del problema o también llamadas de *divide y vencerás* (*Divide and Conquer* o D&C) para obtener distintos subproblemas de tamaño más reducido que el problema original. En el capítulo anterior donde utilizábamos un esquema ILS la estrategia de descomposición respondía a un esquema de *decrece y vencerás*. La diferencia es que ahora en nuestro esquema de multi-arranque no modificamos sólo una parte de la solución sino la solución por completo. En general, la estrategia *decrece y vencerás* reduce el problema en un subproblema mientras que la estrategia *divide y vencerás* reduce el problema en varios sub-problemas.

A esto se debe la diferencia entre *decrece* (utilizado en el esquema ILS) y *divide* (utilizado en el esquema multi-arranque). Esta estrategia soluciona el problema utilizando la siguiente secuencia:

1. Dividir el problema en sub-problemas considerando un criterio determinado. Cada sub-problema es una instancia más pequeña, pero del mismo tipo que el problema original. Para

seleccionar el conjunto de clientes y vehículos que formaran parte de cada subproblema, se elige un conjunto de rutas de acuerdo con una regla de proximidad geométrica. Para cada una de estos subproblemas, se disuelven las rutas asociadas y se extraen los clientes y vehículos asociados, para construir los datos de entrada en el nuevo subproblema.

2. Comienza un proceso iterativo en el que se utiliza la metodología escogida que puede ser la misma que la del problema original, durante un periodo de tiempo que puede ser controlado por tiempo o por número de iteraciones. El objetivo es encontrar una solución “local” mejorada de cada subproblema correspondiente de forma recursiva. Debido a su tamaño reducido, cada subproblema es menos complejo.

3. Combinar de manera adecuada las sub-soluciones “locales” obtenidas para obtener una solución “global”.

El objetivo aquí es utilizar algún criterio basado en la proximidad para dividir el conjunto original de nodos en subconjuntos disjuntos y luego para resolver cada uno de estos subconjuntos mediante la aplicación de la misma metodología descrita anteriormente (es decir, una combinación de CWS aleatorio con una memoria caché). Se pueden utilizar diferentes políticas de división, pero el principio básico es siempre el mismo: una vez que se ha encontrado una solución CWS aleatorizada, usamos un criterio de proximidad geométrico para seleccionar un subconjunto de rutas y sus nodos correspondientes. Si el procesamiento de cualquier región obtiene una mejora de la solución inicial, se actualiza la solución global.

En nuestro caso se utiliza el criterio de proximidad para hacer la descomposición del problema. Se divide el plano en dos, tres o cuatro regiones de manera que el algoritmo se aplica a cada región individualmente. Los nodos se asignan a cada región en base a la ruta a la que pertenecen, es decir, todos los nodos en cada ruta están asociados a la misma región. Por tanto, se asocian en primer lugar las rutas a las regiones. Una ruta está representada por un solo punto que es el centro geométrico de todos los puntos que la componen.

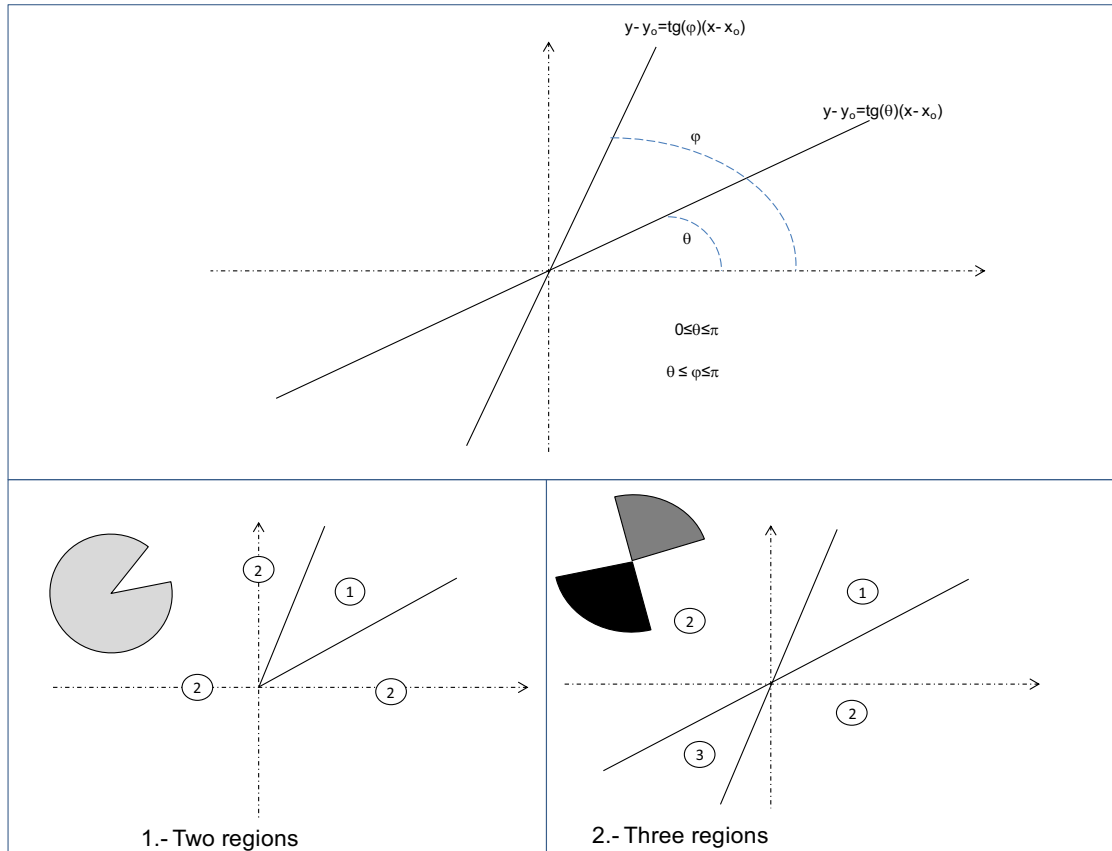


Figura 4.2: Procedimiento de divide y vencerás utilizando rectas y ángulos.

La asignación de las rutas (utilizando su centro geométrico) a las regiones, se basa en el uso de rectas que dividen el plano euclidiano en un número de regiones determinado. La Figura 4.2 muestra el caso considerando dos rectas en el plano. Los ángulos θ y φ determinan la amplitud de cada región y el número de regiones depende de la política específica en vigor. Como se muestra en la Figura 4.2, el plano puede dividirse en: (i) dos regiones (ii) tres regiones si se aumenta el número de rectas. Si el nivel de detalle requerido para el problema específico requiere más de cuatro regiones, entonces se puede aumentar el número de rectas, por ejemplo, utilizando los ejes x e y como las rectas $y=0$ y $x=0$ respectivamente. En nuestro enfoque, se utilizan dos rectas. Utilizamos la ecuación en forma de punto-pendiente que pasa por el punto (x_0, y_0) y forma un ángulo φ con el eje x como:

$$y - y_0 = tg(\varphi) * (x - x_0)$$

Para cada una de estas regiones, se aplican los mismos mecanismos para obtener una solución, aunque en este caso normalmente la dimensión es menor. Si se mejora alguna de las

soluciones parciales a través del proceso, entonces la solución global se mejora en consecuencia. Concluida esta fase, tenemos la mejor solución asociada al arranque actual. El proceso se reinicia de nuevo con una nueva solución de CWS aleatorizada hasta que se cumplan los criterios parada. El uso de este enfoque con la lista de ahorros aleatorios contribuye a escapar de los mínimos locales ya que se contribuye a la diversificación en la búsqueda de los resultados. Además, a diferencia del esquema ILS presentado en el capítulo 3 donde sólo se recalcula un porcentaje de las rutas, en el caso del esquema multi-arranque, se recalcula la solución para todas las áreas del espacio euclidiano que se han formado. Esta es una de las características que diferencia a ambos planteamientos. Como se puede ver, para resolver el VRPMB sólo se utiliza un parámetro:

- El que define la distribución geométrica utilizada durante el proceso de aleatorización parcial, β .

El coeficiente de penalización para los arcos interfaz (λ) no se utiliza ya que en el VRPMB se permite más de un arco interfaz y por tanto ya no tiene sentido la secuenciación de grupos. Por tanto, la restricción de precedencia (entregas primero, después recogidas) se ha eliminado. Igualmente, el operador correctivo recursivo (RCO) no es invocado ya que el número de rutas en la solución ha pasado de ser una restricción a ser una de las variables de decisión del problema. Además, el porcentaje de destrucción de rutas (p) no se utiliza como consecuencia de la utilización del esquema multi-arranque. A diferencia del esquema ILS visto anteriormente donde sólo un porcentaje de rutas era modificado en cada iteración, en el esquema actual de multi-arranque, todas las rutas son modificadas según el área a la que pertenezcan.

4.4. Pseudocódigo del algoritmo.

Esta sección presenta una descripción más detallada del algoritmo propuesto. El **Algoritmo 4.1** muestra el pseudocódigo asociado con el esquema multi-arranque que se plantea (MS-BRCW), el procedimiento principal de nuestro enfoque.

Algoritmo 4.1 Pseudocódigo del procedimiento principal del MS-BRCW.

```
procedure MS-BRCW(inputs,  $\beta$ )
01   savingsList <- CreateSavingsList(inputs)
```

```
02   cwsSol <- ConstructCWSSol(inputs, false)// aleatorio off
03   bestSol <- cwsSol
04   while {condición final no alcanzada} do //tiempo o límite de iteraciones
05       newSol <- ConstructRandomSol(inputs,β, true) // aleatorio on
06       newSol <- improveRoutesUsingHashTable(newSol)
07       newSol <- improveRoutesUsingSplitting(aTest, inputs, cache,
newSol)//aplicar divide y vencerás
08       ourBestSol <- updateBestSolsList(bestSol, newSol)
09   end while
10   return bestSol
End
```

El algoritmo recibe los datos de entrada en la variable `inputs`. Esta variable contiene las coordenadas cartesianas de cada nodo (clientes y deposito), las demandas de los clientes, la semilla, la capacidad máxima del tipo de vehículo y los clientes que reciben material y los que lo envían. Asimismo, recibe el valor del único parámetro considerado, β tal y como se ha explicado anteriormente. En primer lugar, siguiendo la lógica del heurístico basada en ahorros de [Clarke y Wright \(1964\)](#), el procedimiento genera la lista de arcos y ahorros correspondientes, así como la solución inicial (líneas 01 y 02) que es el resultado del proceso de fusión de rutas. Esta solución será la base y referencia para todas las operaciones posteriores y hasta que no se produzca alguna mejora, nuestra mejor solución (línea 03).

En este momento, se inicia el proceso multi-arranque (líneas 04-09). Aunque el objetivo principal que tenemos para utilizar este esquema es intentar reducir el número de parámetros al máximo, hay otras características que son también muy interesantes:

- Permite que el algoritmo escape de mínimos locales gracias a la fase de diversificación que se completa en cada arranque.
- Permite la ejecución en paralelo del algoritmo.

De hecho, el algoritmo puede ser ejecutado en paralelo, asignando a cada ejecución una semilla diferente que será utilizada por el generador de números aleatorios. Estas ejecuciones se pueden implementar en diferentes entornos paralelos ya que cada arranque es independiente de los otros. Este apartado se incluye como una de las líneas de investigación futuras que aparecen en la última sección de esta tesis. De esta forma, en cada iteración del proceso multi-

arranque (arranque independiente), se obtiene una nueva solución factible (línea 05). El número de iteraciones o arranques puede basarse en el tiempo, o en el número de iteraciones.

A continuación, se aplica un proceso de búsqueda local basada en una memoria hash rápida (línea 06). En esta búsqueda local se procesa individualmente cada ruta intentando buscar una mejor secuencia en el mismo conjunto de nodos y que fue obtenida en iteraciones anteriores. La solución actual se puede mejorar adicionalmente utilizando la técnica de división de problemas que ya ha sido explicada en el capítulo anterior. En este momento, hay que decidir si aplicamos esta técnica a todas las soluciones obtenidas o tan solo a las prometedoras. En nuestra versión hemos decidido aplicarla a todas las soluciones (línea 07) incluso aquellas que no mejoran la solución actual. Hemos relajado esta condición para evitar que algunas buenas soluciones sean rechazadas demasiado pronto. Este procedimiento de división intenta mejorar los resultados considerando primero diferentes subconjuntos de rutas y actuando sobre cada uno de estos subconjuntos. Este proceso iterativo de segundo nivel hace uso de nuevo tanto de los procedimientos CWS aleatorizados como de los procedimientos de cache.

Cuando tanto los procesos de memoria como el de división de problemas han finalizado, ya disponemos de una solución para este arranque que será comparada con las anteriores (línea 08) para dar como resultado la mejor solución obtenida hasta el momento. De esta forma, el algoritmo realiza el siguiente arranque y la única información que conserva del anterior es:

- Coste de la mejor solución.
- Rutas que componen la solución y su composición.

De esta solución sólo utilizará el coste para compararlo con los costes obtenidos en futuras iteraciones. La composición de rutas de la solución actual no se utiliza como entrada del siguiente arranque ya que en el nuevo arranque se calcula una nueva solución sin tener en cuenta ningún dato anterior. Cuando se ha alcanzado la condición de parada, o bien en número de iteraciones o bien en tiempo, el algoritmo devuelve la mejor solución obtenida y finaliza (línea 11).

Algoritmo 4.2 Pseudocódigo del Procedimiento de divide y vencerás utilizando rectas y ángulos.

```

procedure improveRoutesUsingSplitting(inputs, sol)
01 for L1 {rotando  $2\pi$  en intervalos  $\pi/5$ } //Línea 1
02 for L2 {rotando  $2\pi$  en intervalos  $\pi/5$ } //Línea 2
03   subSol <- splitSol(inputs, sol, L1, L2)//dividir rutas en areas
04   for each subSol
05     newSubSol <- ConstructRandomSol(subSol)//solucionar cada area
06     newSubSol <- cache(newSubSol)//aplicar cache
07     if(cost(newSubSol)<cost(subSol) then update(sol)
08   end for
09 end for
10 end for
11 return sol
end procedure

```

El **Algoritmo 4.2**, proporciona una idea del enfoque utilizado para reducir la dimensión del problema. Para cada solución completa o de primer nivel proporcionada por el CWS aleatorizado, se usan estrategias de división para seleccionar diferentes subconjuntos de rutas a partir de él. Se barre el espacio euclidiano con dos rectas que lo recorren de forma asíncrona. Esto se consigue gracias a los dos bucles anidados (líneas 01 y 02). En este momento se produce la asignación de rutas a áreas según la localización relativa de ambas (línea 03). Con esta asignación se ha dividido el problema original en tantos sub-problemas como áreas tengamos y se procesa cada área por separado (líneas 04-08). Para cada sub-problema, los nodos asociados se utilizan para definir una nueva instancia reducida o de segundo nivel. La misma metodología utilizada para la instancia de primer nivel se aplica ahora para cada una de estas instancias de segundo nivel. Es decir, se resuelve el sub-problema con el mismo algoritmo que se ha utilizado para el problema original (línea 05) y se le aplica el procedimiento cache a las rutas obtenidas en el sub-problema (línea 06). Si se produce alguna mejora en algún sub-problema, se actualiza la solución general. El algoritmo acaba con la devolución de la solución (línea 11). En conjunto, esta estrategia iterativa de dos niveles puede ser considerada como una metodología de divide y vencerás que tiene como objetivo mejorar la solución global resolviendo casos reducidos del problema original.

Una de las partes principales del procedimiento es la asignación de cada ruta a las diferentes áreas creadas en el proceso de división. Las rutas se asignan a cada área considerando su centro geométrico. El **Algoritmo 4.3** muestra el procedimiento que realiza esta asignación utilizando los ángulos de las dos rectas en la forma de punto pendiente de cada una de ellas. Usando esta formulación, la ecuación de una recta que pasa por el punto (x_0, y_0) con y forma un ángulo de φ radianes con eje x es:

$$y - y_0 = tg(\varphi) * (x - x_0)$$

En nuestro enfoque, el punto (x_0, y_0) no es el origen de las coordenadas, sino el centro del VRP representado por el centro geométrico de todos los nodos del problema que se calcula evaluando el valor medio de todos los nodos. Un punto (x_i, y_i) es asignado a un área específica, según el valor positivo o negativo de la expresión:

$$y_i - y_0 - tg(\varphi) * (x_i - x_0)$$

Si es positivo, ese punto está situado por encima de la recta, de lo contrario, ese punto está situado debajo de la recta y es asignado al área correspondiente. Dado que consideramos 2 rectas, los puntos pueden ser:

1. Arriba (recta 1)-bajo (recta 2).
2. Arriba-arriba.
3. Abajo-abajo.
4. Abajo-arriba.

Esta lógica se muestra en el pseudocódigo mostrado en el **Algoritmo 4.3**. El algoritmo recibe como parámetros la ruta a clasificar, así como el centro del VRP y los ángulos actuales de las dos rectas que se utilizan en la clasificación. A continuación, líneas 02, 06 y 10, se compara el centro de la ruta actual con la ubicación de las rectas para determinar a qué área pertenece según la ecuación punto pendiente, líneas 04, 08 y 12.

Algoritmo 4.3 Asignación de rutas a áreas utilizando los ángulos de las rectas.

```
procedure getRouteArea(vrpC, r, angl, ang2)
```

```

01  area = 0//vrpC es el origen del problema como media de las
    coordenadas de todos los nodos
02  if( r.getCenter()[1] - vrpC[1] > tang(ang1)*(vrpC[0] -
r.getCenter()[0]) && r.getCenter()[1] - vrpC[1] > tang(ang2)*(vrpC[0] -
r.getCenter()[0]))
04      area = 1
05  else
06      if( r.getCenter()[1] - vrpC[1] < tang(ang1)*(vrpC[0] -
r.getCenter()[0])&&r.getCenter()[1] - vrpC[1] < tang(ang2)*(vrpC[0] -
r.getCenter()[0]))
08          area = 2
09      else
10          if( r.getCenter()[1] - vrpC[1] < tang(ang1)*(vrpC[0] -
r.getCenter()[0])&& r.getCenter()[1] - vrpC[1] >tang(ang2)*(vrpC[0] -
r.getCenter()[0]))
12              area = 3
13          end if
14      endif
15  end if
16  return area
end procedure

```

4.5. Experimentos computacionales.

Hemos probado el rendimiento de nuestro algoritmo utilizando las instancias de referencia de la literatura para este problema. El conjunto de instancias fue propuesto por [Salhi y Nagy \(1999\)](#). Tres subconjuntos de problemas se generan como T, Q y H de 14 instancias de VRP. En cada conjunto, cada segundo, cuarto y décimo cliente se declara como cliente de recogida (*backhaul*), y se le asigna una demanda del mismo tamaño que la demanda original. El número de clientes en estos conjuntos de datos varía de 50 a 199, y los problemas CMT6-10 y CMT13-14 imponen una restricción de longitud de ruta máxima. Comparamos nuestros resultados con las BKS reportadas en [Vidal et al. \(2014\)](#). Obsérvese que, a pesar de que nuestro algoritmo no fue específicamente diseñado para resolver el VRPMB, después de una rápida adaptación para reducir el número de parámetros es capaz de obtener diferencias muy bajas para las instancias empleando tiempos de computación razonablemente bajos, con un promedio por debajo del 0,4%.

Instance	n-lh	n-bh	Q	K	Best10		Avg10		BKS	Gap:(Best10-BKS)/BKS
					cost	time (s)	cost	time (s)		
CMT01H	25	25	160	4	465,02	60	465,02	60	465,02	0,00%
CMT01Q	38	12	160	6	489,74	45	489,74	45	489,74	0,00%
CMT01T	45	5	160	7	520,06	42	520,06	42	520,06	0,00%
CMT02H	38	37	140	5	662,63	65	663,43	65	662,63	0,00%
CMT02Q	57	18	140	7	732,66	110	733,71	110	731,26	0,19%
CMT02T	68	7	140	9	782,77	73	782,77	73	782,77	0,00%
CMT03H	50	50	200	3	702,47	145	704,78	145	700,94	0,22%
CMT03Q	75	25	200	4	747,15	133	747,15	133	747,15	0,00%
CMT03T	90	10	200	5	800,34	103	802,95	93	798,07	0,28%
CMT12H	50	50	200	6	636,65	91	641,65	89	629,37	1,16%
CMT12Q	75	25	200	8	729,25	111	738,25	106	729,25	0,00%
CMT12T	90	10	200	9	788,02	62	791,27	59	787,52	0,06%
CMT11H	60	60	200	4	829,50	75	834,83	71	818,05	1,40%
CMT11Q	90	30	200	6	943,83	121	948,83	112	939,36	0,48%
CMT11T	108	12	200	7	1.001,94	188	1.012,40	161	998,80	0,31%
CMT04H	75	75	200	6	835,92	402	840,03	403	828,12	0,94%
CMT04Q	113	37	200	9	922,07	421	926,21	394	915,27	0,74%
CMT04T	135	15	200	11	1.001,07	164	1.010,57	155	990,39	1,08%
CMT05H	100	99	200	9	987,67	379	997,81	366	978,74	0,91%
CMT05Q	150	49	200	12	1.119,33	452	1.131,66	447	1.104,87	1,31%
CMT05T	180	19	200	15	1.226,90	523	1.243,35	554	1.218,77	0,67%
CMT06H	25	25	160	7	555,43	24	555,43	24	555,43	0,00%
CMT06Q	38	12	160	7	555,43	31	555,43	31	555,43	0,00%
CMT06T	45	5	160	7	555,43	31	555,43	31	555,43	0,00%
CMT07H	38	37	140	14	900,54	83	900,54	83	900,12	0,05%
CMT07Q	57	18	140	14	900,69	79	900,69	79	900,69	0,00%
CMT07T	68	7	140	14	903,05	51	903,05	51	903,05	0,00%
CMT08H	50	50	160	10	865,50	103	869,95	92	865,50	0,00%
CMT08Q	75	25	160	10	865,50	91	869,95	91	865,50	0,00%
CMT08T	90	10	160	10	865,54	87	874,06	85	865,54	0,00%
CMT14H	50	50	200	11	821,75	104	830,88	104	821,75	0,00%
CMT14Q	75	25	200	11	821,75	63	832,75	52	821,75	0,00%
CMT14T	90	10	200	11	826,77	84	833,91	84	826,77	0,00%
CMT13H	60	60	200	12	1.545,17	191	1.553,16	183	1.542,86	0,15%
CMT13Q	90	30	200	12	1.545,17	150	1.554,35	159	1.542,86	0,15%
CMT13T	108	12	200	12	1.545,17	196	1.556,02	198	1.542,86	0,15%
CMT09H	75	75	200	16	1.169,41	251	1.178,74	245	1.160,68	0,75%
CMT09Q	113	37	200	16	1.169,41	235	1.175,55	228	1.161,24	0,70%
CMT09T	135	15	200	16	1.172,16	231	1.176,33	279	1.162,55	0,83%
CMT10H	100	99	200	20	1.390,96	403	1.401,10	398	1.372,47	1,35%
CMT10Q	150	49	200	20	1.395,85	421	1.399,35	458	1.374,18	1,58%
CMT10T	180	19	200	20	1.395,85	389	1.398,02	379	1.381,04	1,07%
									Average	0,39%

Tabla 4.1: Resultados comparativos para las instancias de Salhi y Nagy (1999).

4.6. Conclusiones de la metodología para el VRPMB y su relación con la Cooperación Horizontal.

Como hemos visto en este capítulo la adaptación de la metodología presentada en el capítulo 3, adaptándola a su versión multi-arranque nos ha permitido simplificar el algoritmo de tal forma que hemos utilizado sólo un parámetro. Hemos mostrado la flexibilidad del enfoque utilizado ya que después de una adaptación menor hemos podido resolver el problema

de rutas de vehículos con retornos mezclados, en el que los clientes de entrega y recogida pueden aparecer en cualquier orden dentro de una ruta. A pesar de que el algoritmo no fue inicialmente diseñado para solucionar este problema, muestra resultados competitivos tanto en la calidad de la solución como en los tiempos de computación, con una separación media por debajo del 0,5% con respecto a las soluciones más conocidas para un conjunto de referencias clásicas.

En el proceso de rutas con retornos, se combinan los dos procesos principales de la distribución, el reparto y la recogida. En su versión agrupada (VRPCB) la combinación se hacía sin solapamiento de actividades mientras que en la versión mezclada (VRPMB) presentada en este capítulo la integración es total. En la versión presentada en este capítulo las actividades de reparto y las de abastecimiento se pueden alternar dentro de la misma ruta con la única restricción de respetar la capacidad de los vehículos. Una característica que añade el VRPMB es que los clientes de recogida se convierten potencialmente en almacenes o depósitos de tal forma que pueden servir no sólo al depósito central sino también a otros clientes de la misma ruta. Así que este modelo representa la transición entre los modelos que consideran las rutas que empiezan y acaban en el depósito central, y los que no consideran el depósito en sus actividades. Esta separación se puede ver claramente en la clasificación de la [Figura 1.6](#).

En su vertiente cooperativa supone un paso adelante según se puede ver en la estructura de capas presentada en la [Figura 3.9](#). Ahora la integración es total y el orden en que se visita a los clientes no está determinado por su naturaleza (entrega o recogida) sino solamente por su ubicación geográfica. De esta forma las posibilidades de cooperación y compartición de clientes aumentan. En la [Figura 4.3](#) se puede ver la comparación de un escenario de no cooperación versus uno cooperativo. De la misma forma que en el caso agrupado, se considera un solo depósito como origen y fin de todas las rutas. Lo que diferencia el modelo mezclado frente al agrupado no es sólo la distribución física de los clientes sino también la capacidad de carga del camión. La suma por separado de las demandas de los clientes de entrega y recogida pueden alcanzar la capacidad máxima del vehículo en el caso agrupado. En el caso mezclado, es posible que no se alcance ya que la visita alternada de clientes de distinto tipo hace que pueden convivir en el vehículo al mismo tiempo mercancías que se deben entregar a los clientes con mercancías que se han recogido. Por tanto, la carga del vehículo fluctúa según la naturaleza de las visitas realizadas haciendo que la capacidad sobrante fluctúe en la misma medida.

El modelo mezclado completa el agrupado y permite analizar la mejor composición de las visitas teniendo en cuenta la función objetivo elegida que normalmente es la minimización de la distancia recorrida.

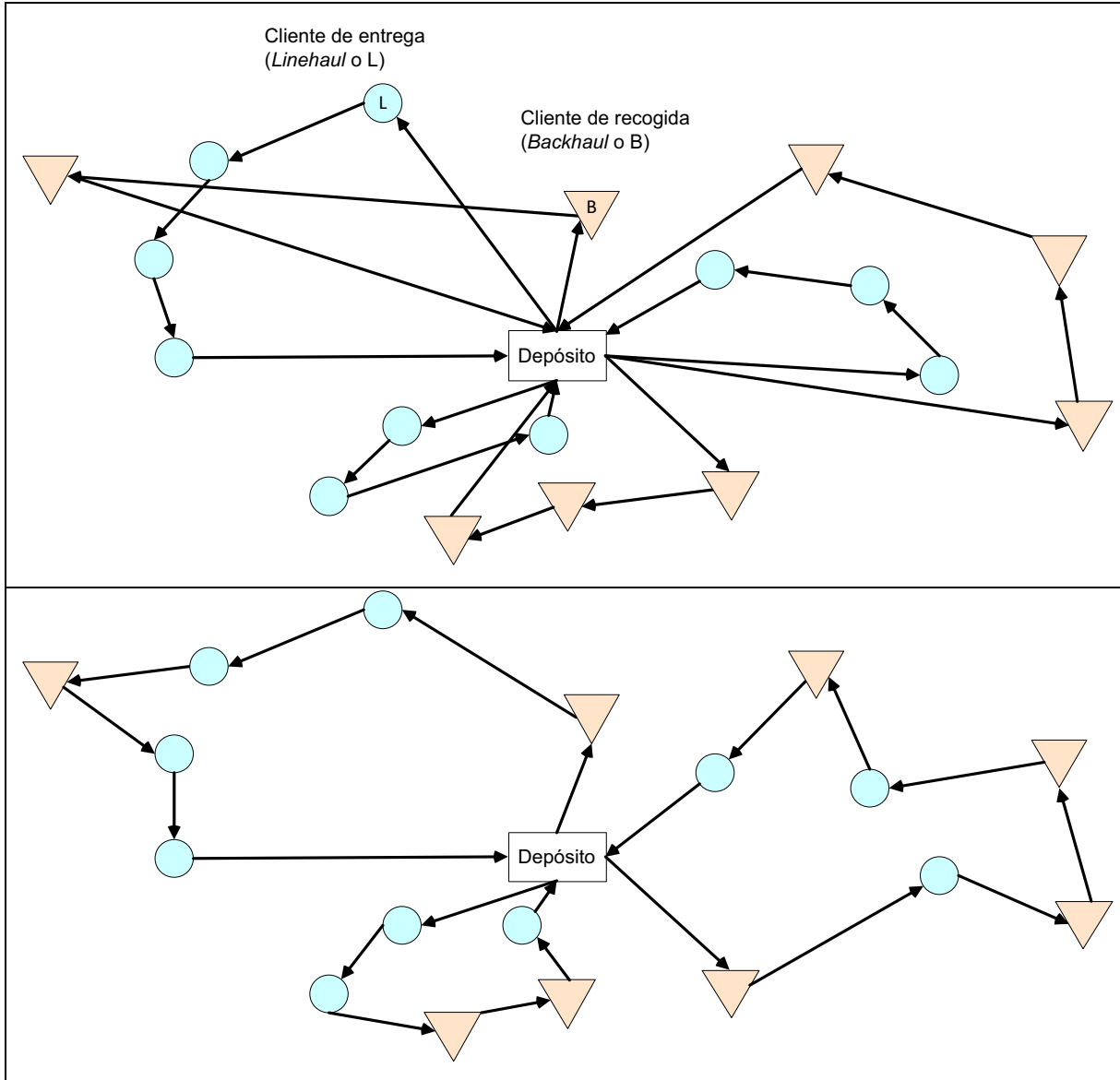


Figura 4.3: No cooperativo (arriba) vs. Cooperativo (abajo) utilizando el VRPMB.

Capítulo 5.

El problema de las rutas de vehículos con retornos y flota heterogénea (HVRPB).

Partes de este capítulo se han extraído de la publicación aceptada en la revista ITOR:

Belloso, J., Juan, A., Faulin, J. 2017. An iterative biased-randomized heuristic for the fleet size and mix vehicle routing problem with backhauls. *International Transactions in Operational Research*. DOI: 10.1111/itor.12379.

5.1. Introducción.

En este capítulo se analiza el problema de rutas de vehículos con retornos y flota heterogénea, denominado en la literatura como problema de rutas de vehículos con retornos y flota heterogénea (*Heterogeneous fleet Vehicle Routing Problem with Backhauls* o HVRPB). Esta variante, amplía el modelo del problema de rutas de vehículos con retornos, mediante la posibilidad de utilizar una flota de vehículos heterogénea. Disponer de vehículos con diferentes capacidades, permite una mejor adaptación a la demanda (Yepes y Medina, 2002). Dicha flota

heterogénea está compuesta por vehículos de diferentes capacidades, tamaños, costes fijos y costes variables. A pesar de que las flotas heterogéneas son frecuentes en las empresas logísticas o de transporte, existe una carencia de trabajos publicados en la literatura científica sobre el HVRPB. El problema considera varias variantes dependiendo de que la composición de la flota sea una variable de decisión (con un número ilimitado de vehículos de cada tipo) o sea un parámetro del problema (la flota es dada y hay que utilizarla de la mejor forma posible). En particular, nuestro planteamiento considera la flota ilimitada, es decir, se considera una flota heterogénea de vehículos y un número ilimitado de cada tipo. Por tanto, la solución consistirá en encontrar el tamaño de la flota y composición que minimice los costes fijos y variables asociados a las tareas de distribución y recogida de materiales entre los clientes. Esta variante se denomina tamaño y composición de la flota en el problema de rutas de vehículos con retornos (*Fleet Size and Mix Vehicle Routing Problem with Backhauls* o FSMVRPB). La literatura es escasa en la resolución de este problema y el único artículo que lo aborda es el de [Salhi et al \(2013\)](#).

El algoritmo propuesto utiliza un método iterativo que se basa en la resolución de una serie de instancias más pequeñas de la versión homogénea del problema y luego utiliza estas sub soluciones como soluciones parciales para la instancia heterogénea original. Con el fin de guiar mejor la exploración del espacio de soluciones, el algoritmo emplea varios procesos de aleatoriedad: el primero para seleccionar un tipo de vehículo. El segundo para ordenar la lista de ahorros, y un tercero para definir el número de rutas que deben seleccionarse de la sub solución de flota homogénea. Los experimentos muestran que nuestro enfoque es competitivo y capaz de proporcionar 20 nuevas mejores soluciones para las 36 instancias recientemente propuesto por [Salhi et al. \(2013\)](#).

5.2. Descripción del problema.

El HVRPB es una nueva extensión del problema de rutas de vehículos con retornos, en el cual se dispone de una flota de vehículos heterogénea. El modelo del problema tiene una descripción similar a la expuesta en la sección 4.1, con las siguientes diferencias:

- El transporte de la mercancía se realiza mediante una flota heterogénea de T tipos diferentes de vehículos, inicialmente localizados en el depósito.
- El número de vehículos disponibles de cada tipo presenta dos posibles variantes:

- Está limitado y fijado de antemano.
- Es ilimitado y por tanto configurable en tamaño y composición de tipos.
- Cada tipo de vehículo t ($t = 1, 2, \dots, T$) tiene las siguientes características asociadas:
 - coste fijo f_t .
 - coste variable por unidad de distancia recorrida v_t .
 - capacidad máxima de carga Q_t .

De tal forma que, el coste fijo correspondiente a una ruta usando un vehículo del tipo t será f_t , mientras que el coste variable de la ruta se calcula como el producto de la distancia recorrida por la ruta multiplicada por v_t . En nuestro caso hemos desarrollado la variante en que la flota es ilimitada en tamaño y composición. La extensión se conoce como tamaño de la flota y composición en problemas de vehículos con retornos (*Fleet Size and Mix Vehicle Routing Problem with Backhauls* o FSMVRPB). Por lo tanto, a diferencia del problema VRPB, en la extensión FSMVRPB, se debe buscar la configuración de vehículos y rutas factibles que minimice el coste total de la solución.

5.3. Metodología.

La metodología utilizada para el caso en que tenemos vehículos de distinto tipo se basa en el conocido método de las aproximaciones sucesivas (*Successive Approximation Method* o SAM), que ha sido utilizado anteriormente por otros autores ([Juan et al., 2014a](#)). El método de forma esquemática consiste en:

1. Se consideran los datos iniciales y se elige un tipo de vehículo de los disponibles. El tipo de vehículo se marca como ya utilizado y no se volverá a usar en las siguientes iteraciones.
2. Se considera que el número de vehículos disponibles del tipo seleccionado es ilimitado y entonces se soluciona el problema como si fuera homogéneo (VRPB).
3. De las rutas obtenidas, se seleccionan aleatoriamente un número de ellas para formar parte de la solución final.
4. Para el resto de rutas, se liberan sus nodos y se consideran como pendientes.
5. El problema se ha convertido en dos sub-problemas:
 - 5.2 Rutas ya construidas y que pertenecen a la solución final.
 - 5.3 Nodos pendientes y tipos de vehículos pendientes.

6. Se selección un nuevo tipo de vehículo que no se había utilizado todavía, así como los nodos pendientes de incluir en las rutas y se vuelve a paso 2 hasta que no quedan ya tipos de vehículos por utilizar o todos los nodos están ya en alguna ruta de la solución.
7. Se evalúa la solución en comparación con las obtenidas anteriormente y se guarda la mejor de ellas en términos de distancias.
8. FIN. El proceso acaba.

Se puede utilizar cualquier algoritmo eficiente para solucionar cada problema homogéneo del punto 2. Se puede ver un esquema de este algoritmo en la [Figura 5.1](#).

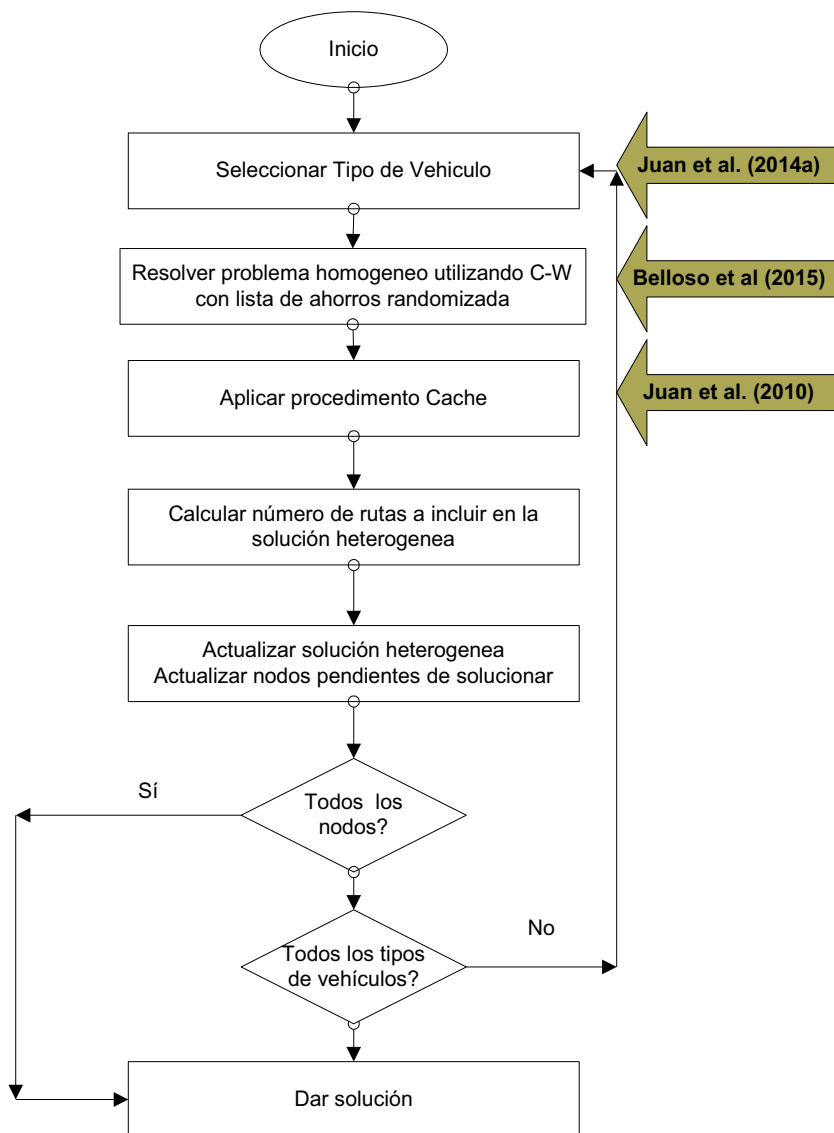


Figura 5.1: Diagrama de la metodología propuesta para la resolución del VRPMB.

5.4. Pseudocódigo del algoritmo utilizado.

La metodología presentada en la sección anterior se implementa a través del operador iterativo de problemas homogéneos (*Iterative Homogeneous Problems Operator* o IHPO), que es invocado a través de un procedimiento multi-arranque cada vez que se necesita resolver un VRPB homogéneo. Este procedimiento de inicio múltiple finaliza cuando se alcanza un criterio de terminación basado en el tiempo. Cada nueva solución generada por este procedimiento para un VRPB dado se compara con la mejor obtenida hasta ahora para el mismo problema. El **Algoritmo 5.1** muestra el pseudocódigo del procedimiento. Las líneas 01-03 representan la inicialización de un problema homogéneo particular. Específicamente, la línea 03 define el problema homogéneo una vez que se ha elegido el tipo de vehículo en la línea 01 de forma uniformemente aleatoria. Los clientes a visitar son definidos como parámetros del procedimiento. La línea 04 llama al heurístico que genera una solución para el problema homogéneo. La línea 05 llama a la búsqueda local basada en memoria que intenta mejorar la solución actual. En este punto, parte de la solución homogénea está lista para formar parte de la solución heterogénea. Las líneas 06 a 10 gestionan la selección de rutas y actualizan la lista de tipos de vehículos disponibles. Si el tipo de vehículo actual es el último de la lista de vehículos, todas las rutas de la solución homogénea se transferirán a la solución final. De lo contrario, se selecciona un número aleatorio entre 1 y el número de rutas obtenidas en la solución homogénea. La línea 11 resume el proceso de promoción de rutas hacia la solución final, mientras que las líneas 12 y 13 representan el valor actualizado de clientes y vehículos. La línea 14 comprueba si se cumplen los criterios de terminación del procedimiento. Si este es el caso, el procedimiento iterativo devuelve la solución actual. De lo contrario (Línea 16) se invoca de nuevo el procedimiento con los clientes no servidos y los tipos de vehículos no utilizados como parámetros de entrada.

Algoritmo 5.1 Pseudocódigo del algoritmo principal.

```
procedure IHPO(customers, vehicles) // Operador iterativo para problemas
homogeneos
01  newVehType <- selectRandom(vehicles) // Seleccionar un tipo de vehículo
aleatoriamente
02  vehCap <- getCapacity(newVehType)
03  newHomVRPB <- Initialization(customers, vehCap)
```

```
04 homSol <- solveHoVRPB(newHomVRPB) // Soluciona un problema homogeneo
utilizando un algoritmo eficiente
05 homoSol <- improveRoutesUsingHashTable(homoSol) // Utilizar la
estructura cache para mejorar las rutas calculadas
// Seleccionar el número de rutas (numroutes) para ser transferidas a la
solución heterogenea
06 if {pending vehicles is 0} then
07   numroutes <- routes(homoSol)
08 else
09   numroutes <- selectRandom(1...routes(homoSol))
10 end if
11 heteroSol <- addRoutes(homoSol(numroutes))
12 pendingCustomers <- extractCustomers(homoSol(routes(homoSol)-numroutes))
13 pendingVehicles <- extractVehicles(newVehType)
// Devolver la solución final
14 if {pendingVehicles es vacío} then
15   return heteroSol
16 else
17   return heteroSol & IHPO(pendingCustomers, pendingVehicles)
end procedure
```

5.5. Experimentos computacionales.

Las instancias del FSMVRPB que consideramos en este documento son las recientemente propuestas por [Salhi et al. \(2013\)](#), que se generan utilizando los datos de la flota heterogénea de [Golden et al. \(1984\)](#) y los datos del VRPB de [Toth y Vigo \(1997\)](#). A partir de los datos de la flota heterogénea, estas instancias toman los datos de la flota (distancias, coste fijo, coste variable y capacidad del vehículo), mientras que se toman los porcentajes de *backhaul* de las instancias del VRPB. Cabe señalar que la fuente original de este conjunto de datos FSMVRPB es el conjunto de datos inicialmente propuesto por [Christofides y Eilon \(1969\)](#). [Salhi et al. \(2013\)](#) generó un conjunto de 36 casos FSMVRPB utilizando los 12 problemas propuestos por [Golden et al. \(1984\)](#) que consideran entre 20 y 100 clientes. Para cada instancia, siguiendo las convenciones utilizadas en [Toth y Vigo \(1997\)](#), se generaron tres nuevas instancias utilizando porcentajes de entregas y recogidas (LH / BH) de 50/50, 67/33 y 80/20. Esto se hizo tomando, como cliente de recogida, cada primer cliente de cada de dos, tres y cinco clientes, respectivamente. El algoritmo descrito en este documento se ha implementado como una aplicación Java. Hemos resuelto el conjunto de 36 casos con el fin de probar la eficacia del método propuesto.

Para realizar estas pruebas se utilizó un ordenador personal estándar, Intel R Core TM i7 CPU M 640 @ 2,80 GHz, 3,42 GB de RAM y un sistema operativo MS XP Professional. Los resultados se resumen en la **Tabla 5.1**, que contiene la siguiente información para cada instancia: nombre de la instancia, número de clientes, número de clientes de línea directa, número de clientes de recogida, mejor solución conocida (BKS) según lo reportado por **Salhi et al. (2013)**, nuestra mejor solución (*Our Best Solution* o OBS) después de diez ejecuciones del algoritmo, la diferencia porcentual entre ambos valores (calculado como $(OBS-BKS) / BKS$), tiempo computacional empleado para obtener OBS y solución óptima o límite inferior que se reporta en **Salhi et al. (2013)**. Se observa que nuestro algoritmo es capaz de mejorar 20 BKS y por tanto proporcionar nuevas BKS. En promedio ofrece un gap de -0.54% con respecto a los valores BKS anteriores. La **Figura 5.2** también muestra la eficiencia de nuestro enfoque mostrando una comparación visual entre los valores de BKS, OBS y los valores óptimos o los límites inferiores para cada instancia (en caso de no tener los óptimos). Los resultados proporcionados por nuestro algoritmo son dignos de mención teniendo en cuenta los bajos tiempos de cálculo utilizados.

Problem	Size	Line	Back	BKS	OBS	Gap:(OBS-BKS)/BKS	Time (s)	Optimal* or Lower Bounds
HWS1	20	10	10	726,48	734,03	1,04%	10	720,7*
HWS2	20	13	7	818,12	820,99	0,35%	15	818,12*
HWS3	20	16	4	848,59	848,32	-0,03%	1	848,23*
HWS4	20	10	10	4.350,65	4.342,48	-0,19%	2	4.342,48*
HWS5	20	13	7	5.366,39	5.357,98	-0,16%	8	5.357,98*
HWS6	20	16	4	5.875,23	5.872,52	-0,05%	6	5.421,63*
HWS7	20	10	10	767,93	729,50	-5,00%	15	729,50*
HWS8	20	13	7	872,97	838,11	-3,99%	16	838,11*
HWS9	20	16	4	903,18	890,76	-1,38%	18	890,76*
HWS10	20	10	10	4.365,44	4.349,13	-0,37%	2	4.349,12*
HWS11	20	13	7	5.414,50	5.363,58	-0,94%	21	5.363,58*
HWS12	20	16	4	5.928,78	5.523,50	-6,84%	102	5.465,40
HWS13	50	25	25	1.625,70	1.632,60	0,42%	23	1.480,75
HWS14	50	33	17	1.811,63	1.793,26	-1,01%	14	1.705,44
HWS15	50	40	10	2.018,93	2.030,37	0,57%	65	1.937,24
HWS16	50	25	25	5.561,67	5.562,94	0,02%	16	5.537,24
HWS17	50	33	17	6.570,39	6.571,77	0,02%	4	6.547,93*
HWS18	50	40	10	7.599,08	7.413,27	-2,45%	19	7.026,15
HWS19	50	25	25	1.704,41	1.659,86	-2,61%	71	1.509,41
HWS20	50	33	17	2.037,23	2.038,22	0,05%	120	1.926,03
HWS21	50	40	10	2.340,09	2.341,17	0,05%	97	2.196,32
HWS22	50	25	25	1.774,71	1.746,28	-1,60%	11	1.697,38
HWS23	50	33	17	2.166,52	2.164,65	-0,09%	110	2.028,35
HWS24	50	40	10	2.430,88	2.425,83	-0,21%	23	2.335,56
HWS25	75	37	38	1.332,02	1.313,59	-1,38%	246	1.228,48
HWS26	75	50	25	1.421,04	1.412,69	-0,59%	95	1.341,97
HWS27	75	60	15	1.534,65	1.543,10	0,55%	145	1.466,24
HWS28	75	37	38	1.617,85	1.631,70	0,86%	65	1.486,89
HWS29	75	50	25	1.799,76	1.808,46	0,48%	45	1.660,15
HWS30	75	60	15	1.990,46	2.001,79	0,57%	197	1.853,72

HWS31	100	50	50	4.943,29	4.990,84	0,96%	310	4.870,54
HWS32	100	66	34	6.035,96	6.019,82	-0,27%	126	5.901,57
HWS33	100	80	20	7.601,29	7.593,25	-0,11%	43	6.886,44
HWS34	100	50	50	2.465,41	2.516,30	2,06%	241	2.369,80
HWS35	100	66	34	2.971,98	2.998,21	0,88%	156	2.730,67
HWS36	100	80	20	3.533,90	3.563,17	0,83%	253	3.139,87
<i>Averages</i>						<i>-0,54%</i>	<i>75,31</i>	

Tabla 5.1: Resultados computacionales para las instancias el FSMVRPB.

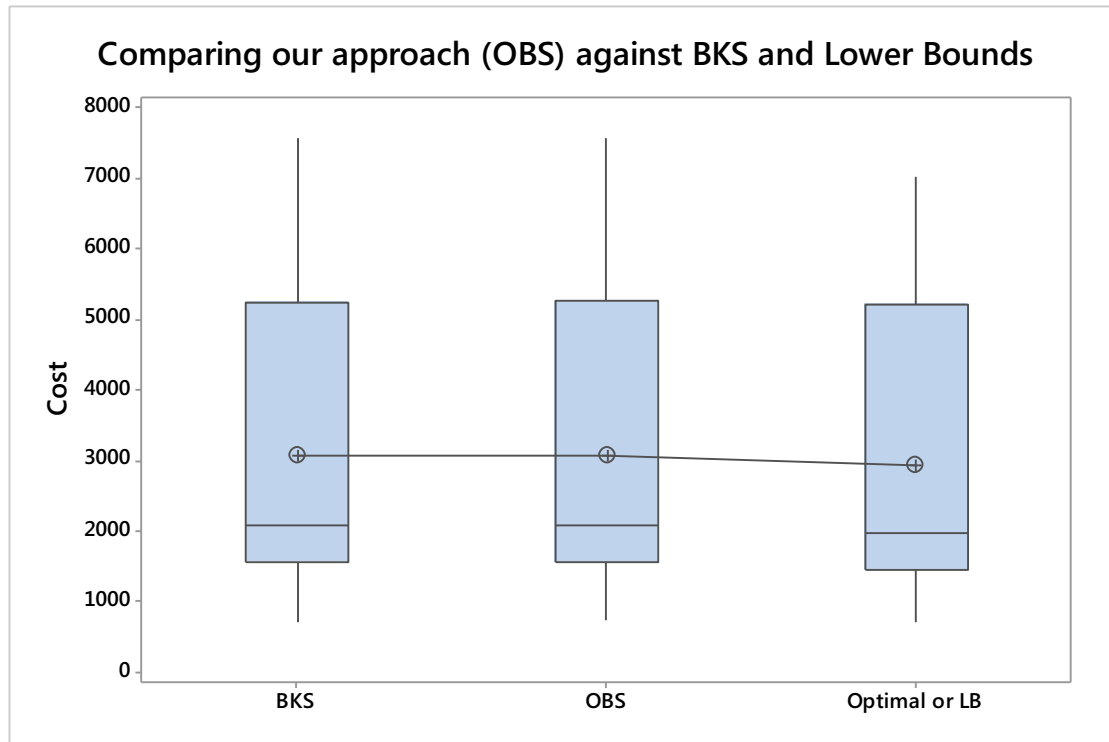


Figura 5.2: Comparación de nuestros resultados con BKS y los valores óptimos (si existen) o LB (*Lower Bound*).

5.6. Conclusiones de la metodología para el HVRPB y su relación con la Cooperación Horizontal.

En este capítulo, se ha presentado un método de aproximaciones sucesivas para resolver el HVRPB. La idea principal detrás la metodología propuesta es transformar (descomponer) el problema de tal forma que su resolución consiste en resolver una serie de VRPBs homogéneos. Este enfoque de descomposición permite resolver variantes del HVRP incluyendo ventanas de

tiempo, demandas estocásticas, carga bidimensional, costes asimétricos, multi-depósitos, etc. Simplemente combinando nuestro algoritmo con cualquier algoritmo eficiente desarrollado para la variante correspondiente.

En los capítulos 4 y 5 hemos visto los modelos que comparten clientes de distintos tipos y estas versiones se complementan con este capítulo que permite la compartición de flotas las cuales es muy posible que tengan cierto grado de heterogeneidad.

Permitir la heterogeneidad de la flota supone un paso adelante en la optimización de las rutas y según la opción escogida, puede abordarse de dos formas:

- Adaptar las rutas a las características de la flota.
- Adaptar la flota a las características de los clientes a visitar.

En el primer caso la flota es fija y el objetivo es su utilización de la forma más eficiente posible, mientras que en el segundo (el tratado en este capítulo) se trata de elegir la mejor composición de flota para servir a los clientes.

Hemos elegido este segundo modelo ya que está más enfocado a las prácticas de cooperación horizontal. Cuando dos o más clientes se plantean cooperar compartiendo el proceso de ruteo, lo más probable es que la flota necesaria tenga un tamaño menor que la suma de las flotas de los participantes y por tanto tendrán que hacer el ejercicio de encontrar la mejor composición de flota para operar en el nuevo escenario de cooperación.

En su vertiente cooperativa supone un paso adelante según se puede ver en la estructura de capas presentada en la [Figura 3.9](#). La capa que contiene este modelo cubre a las anteriores e integra los vehículos de los participantes en la cooperación para obtener la mejor composición de flota que opere en el nuevo escenario. De esta forma las posibilidades de cooperación de forma existosa aumentan.

En la [Figura 5.3](#) se puede ver un escenario básico no cooperativo frente a uno cooperativo.

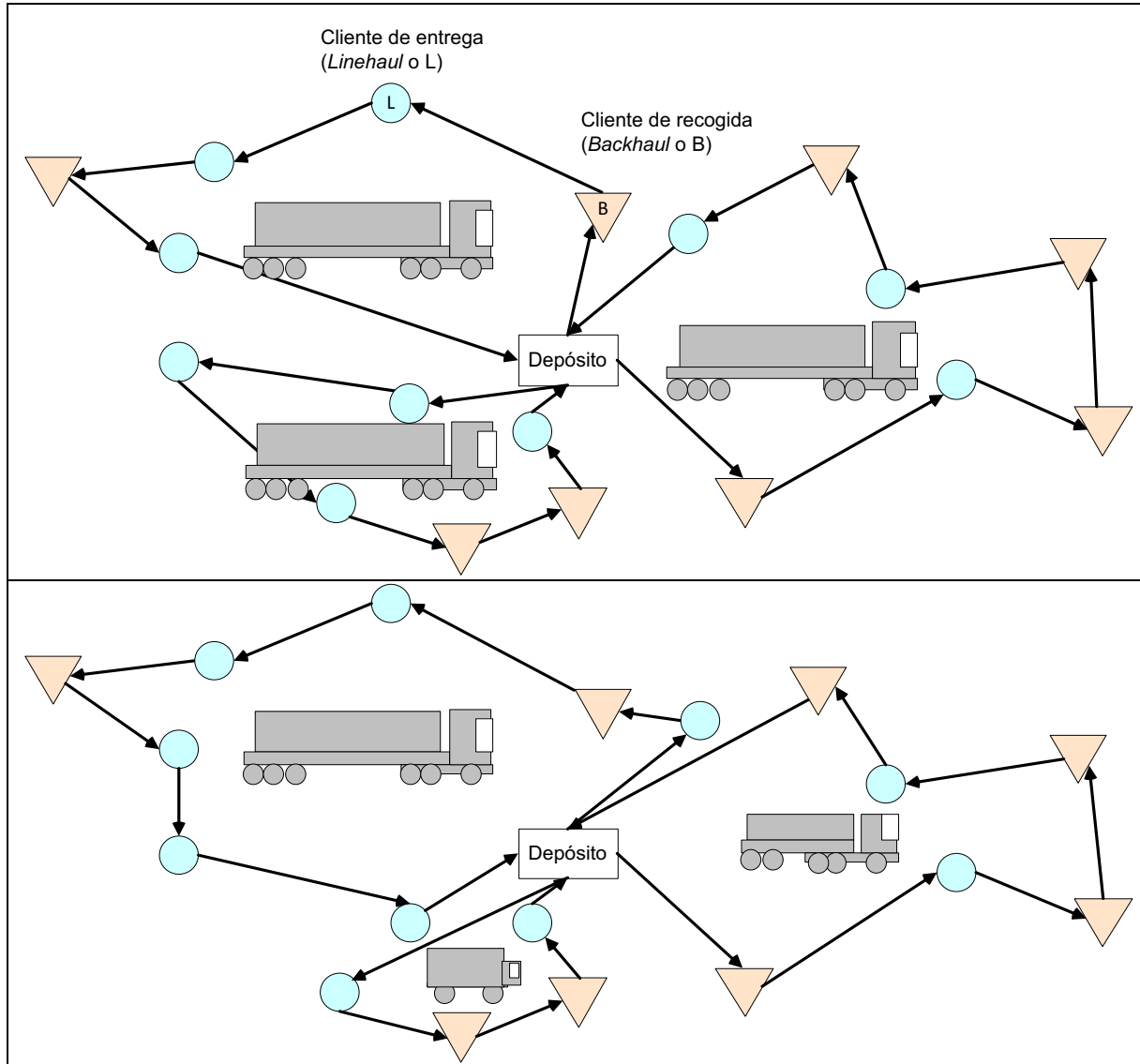


Figura 5.3: No cooperativo (arriba) vs. Cooperativo (abajo) utilizando el HVRPB.

Capítulo 6.

Conclusiones finales.

6.1. Conclusiones.

El propósito de la tesis consiste en el estudio de las mejores prácticas de cooperación horizontal en el terreno de la logística y la distribución. La modelización de éstas prácticas es el camino obligado para poder cuantificar las economías de escala que se producen como consecuencia de su aplicación en el entorno empresarial. Tras el estudio correspondiente, se ha seleccionado un problema (que es abordado en tres subproblemas) para su solución por métodos metaheurísticos dada su complejidad computacional. El problema es NP-complejo y por tanto hasta la fecha no se conocen algoritmos en tiempo polinomial que lo solucionen y por ello se ha utilizado una metodología híbrida. El problema considerado como paradigma de la cooperación horizontal es el del tamaño de la flota y composición de las rutas de vehículos con retornos en sus diferentes vertientes. Se ha abordado el problema en tres partes diferenciadas, en primer lugar, con la consideración de las recogidas agrupadas, en segundo lugar, se ha relajado la restricción de agrupación permitiendo la mezcla entre recogidas y entregas.

Finalmente se ha relajado la restricción de la flota homogénea permitiendo distintos tipos de vehículos para desarrollar las tareas de entregas y recogidas de materiales.

En el capítulo 1 se establecen los argumentos que motivan la tesis debido a la importancia de la cooperación en las actividades empresariales y en particular en las actividades de L&D. Esta cooperación y su trascendencia económica motiva la necesidad de desarrollar técnicas que resuelvan de forma eficiente los problemas de esta índole. El propósito es doble, por un lado, se trata de diseñar nuevos algoritmos que resuelvan estos problemas de forma robusta, flexible, rápida y que no requiere un gran esfuerzo de ajuste de parámetros para solucionar casos que se acerquen a situaciones reales. Por otro lado, los métodos tienen que poderse comparados con otros métodos publicados anteriormente para contrastar su eficiencia. Asimismo, se introducen los problemas de optimización combinatoria como marco que engloba a los problemas abordados. Los problemas de distribución que se solucionan en esta tesis son de alta complejidad debido a su naturaleza combinatoria. Esta es la razón por la que se han propuesto numerosos métodos aproximados para solucionar los problemas en los cuales no se dispone del tiempo necesario para aplicar algoritmos exactos. Se plantea uno de los problemas matemáticos del milenio, ¿Es $P=NP$? (*P versus NP*). El problema no se ha solucionado hasta la fecha y mientras tanto tenemos que diseñar algoritmos de tipo aproximado para solucionar los problemas que hasta la fecha están catalogados como *NP*. Es decir, no se conoce algoritmo de tipo polinomial que lo resuelva.

En el capítulo 2 se hace una descripción del estado del arte de los dos aspectos contenidos en la tesis: la cooperación horizontal y los problemas de rutas de vehículos. Las conclusiones de esta revisión han sido los modelos seleccionados para ser solucionados en esta tesis. En primer lugar, se han seleccionado los problemas reales como mejores prácticas de cooperación en la primera parte del capítulo, para a continuación revisar el estado del arte particularizado a los modelos que definen estas mejores prácticas.

En la **Figura 6.1** se puede ver un resumen gráfico de los tres ejes sobre los que se edifica esta tesis doctoral. Los capítulos 1 y 2 son los que han abordado los tres temas cuyo resultado son los capítulos 3, 4 y 5 donde se solucionan los problemas seleccionados.



Figura 6.1: Resumen de los temas abordados en esta tesis.

Los capítulos siguientes detallan las metodologías utilizadas en cada uno de los problemas seleccionados. En el capítulo 3 la metodología utilizada combina un esquema que guía el proceso de búsqueda de tipo metaheurístico con una adaptación del conocido algoritmo de los ahorros que incluye aleatoriedad. Esta aleatoriedad convierte al algoritmo anterior en no determinista de tal forma cada vez que es invocado se obtiene una solución distinta y posiblemente mejor que la anterior. La aplicación de cierta aleatoriedad a los criterios que generan los conjuntos de soluciones permite la diversificación de la exploración presentando extremos de calidad mayor que los creados de forma determinista. Este esquema es introducido dentro de un marco de búsqueda local iterada, ILS que parte de una solución base que es mejorada en cada iteración del proceso. Se introduce también un operador que corrige el número de rutas obtenidas ya que la naturaleza del algoritmo de los ahorros en su construcción no permite controlar esta restricción.

En el Capítulo 4, para mostrar la flexibilidad de nuestro planteamiento, tras una pequeña adaptación se ha utilizado para resolver el VRP con mezcla de recogidas en el que los clientes de recogida y entrega se pueden visitar en cualquier orden dentro de una ruta. A diferencia del

anterior, este metaheurístico se ha embebido en un esquema multi-arranque que reinicia los resultados cada vez que el metaheurístico es invocado.

Con el objetivo de solucionar estos problemas permitiendo asimismo que la flota sea heterogénea, el Capítulo 5 presenta un algoritmo que integra distintas estrategias. La principal es el método de aproximaciones sucesivas que reduce el problema heterogéneo para solucionar iterativamente una serie de problemas homogéneos. La principal característica de este algoritmo es que permite que la solución de cada problema homogéneo se obtenga con cualquier algoritmo que sea eficiente. En nuestro caso se utiliza el metaheurístico presentado en el capítulo 3, aunque cualquier otro podría haber sido utilizado.

Se ha realizado una serie extensiva de experimentos para estos problemas y como se puede ver en los artículos relacionados nuestro algoritmo es capaz de igualar las mejores soluciones reportadas en la literatura en la mayoría de los casos y mejorarlas en más de veinte casos. Esto además se realiza en tiempos de ejecución razonablemente bajos teniendo en cuenta la simplicidad del algoritmo que requiere según el caso entre uno y tres parámetros tan solo.

Además de los problemas abordados, el planteamiento de la presente tesis permite solucionar entre otros los problemas que se enumeran en la [Tabla 6.1](#).

Acrónimo	Denominación	Característica
AVRP	Asymmetric Vehicle Routing Problem	La distancia entre clientes puede depender del sentido de la ruta.
CVRP	Capacitated Vehicle Routing Problem	La capacidad del vehículo no se puede superar.
VRPMDR	Vehicle routing problem with multiple driving ranges	La longitud de la ruta tiene un límite máximo.
FSMVRP	Fleet Size and Mix Vehicle Routing Problem	Varios tipos de vehículos con distintos costes fijos y variables. No hay limitación en número de cada tipo.
HVRP	Heterogeneous Vehicle Routing Problem	Varios tipos de vehículos con distintos costes fijos y variables. La flota es fija.

VRPCB	Vehicle Routing Problem with Clustered Backhauls	Los clientes pueden ser de recogida y de entrega. Las entregas han de realizarse antes de las recogidas.
VRPMB	Vehicle Routing Problem with Mixed Backhauls	Los clientes pueden ser de recogida y de entrega. Las visitas pueden hacerse en cualquier secuencia.
VRPDDP	<i>Vehicle Routing Problem with Divisible Deliveries and Pick-ups</i>	Los clientes pueden ser visitados más de una vez.
VRPSDP	<i>Vehicle Routing Problem with Simultaneous Deliveries and Pick-ups</i>	Se puede realizar la recogida y la entrega en el mismo cliente en una sola visita.
VRPSD	<i>Vehicle Routing Problem with Stochastic Demand</i>	Las demandas no son deterministas.

Tabla 6.1: Problemas clásicos del VRP que se pueden resolver con el planteamiento de esta tesis doctoral.

6.2. Trabajos futuros.

El presente trabajo de investigación proporciona el punto de partida para el análisis de beneficios obtenidos como consecuencia de la aplicación de prácticas de cooperación horizontal basadas en las variantes que se incluyen. Por tanto, el análisis de escenarios del tipo, “que-pasa-si” (“*What if*”) puede utilizar las metodologías presentadas para su cuantificación. Esta sería la aplicación futura más evidente e inmediata de esta tesis doctoral. En cuantos a nuevos modelos que se pueden adaptar a los aquí propuestos, se podrían destacar:

- Adaptación del modelo al caso de flotas heterogéneas fijas siguiendo la propuesta presentada en este trabajo.
- Inclusión de demandas estocásticas dentro de un marco del tipo simheurístico como se describe en [Grasas et al. \(2016\)](#).

- Paralelización del algoritmo para reducir los tiempos computacionales y poder obtener soluciones de alta calidad en tiempo real.
- Comparación de los resultados de los modelos en entornos de cooperación horizontal, i.e., cuáles serían los resultados para los mismos datos en un escenario de no colaboración.
- Consideración del impacto medioambiental en los modelos tratados.
- Resolución de otros modelos del ámbito de los VRP.

Estos algoritmos permitirán la adaptación de problemas reales más complejos a modelos existentes y por tanto permitirán su optimización siguiendo las reglas presentadas en esta tesis doctoral.

6.3. Artículos.

6.3.1. Análisis del problema VRPCB y VRPMB en el artículo aceptado de Networks.

Belloso, J., Juan, A.; Faulin, J. Martinez, E. (2017): A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls. *Networks* (indexed in ISI SCI, 2014 IF = 0.830, Q3; 2014 SJR = 1.036, Q1). ISSN: 0028-3045.

En este artículo se analiza el modelo conocido como problema de rutas de vehículos con retornos, donde se consideran dos tipos de clientes, los de recogida (envían materiales al depósito) y los de entrega (reciben materiales desde el depósito). Ambos son servidos desde un depósito central. El depósito representa el almacén donde están almacenados los productos a entregar y donde se almacenan los productos que se recogen de los clientes. Inicialmente se estudia la versión conocida como retornos agrupados, en la cual los clientes de entrega en cada ruta deben ser servidos antes de que se visiten los clientes de recogida. Se presenta un metaheurístico relativamente simple de implementar a la vez que eficiente, que integra una versión aleatoria del conocido algoritmo de los ahorros dentro de un marco metaheurístico. Se utiliza la aleatoriedad sesgada para inducir un efecto parcialmente aleatorio en la ordenación de la lista de ahorros de los arcos del problema. Asimismo, se asigna un coste de penalización a los arcos que conectan un cliente de recogida con uno de entrega con el objetivo de retrasar lo

más posible la creación de rutas mixtas (con clientes de ambos tipos). Esta penalización promueve a su vez el orden secuencial de las rutas primero entregas, luego recogidas. En el metaheurístico propuesto, la restricción del tamaño de la flota se relaja de forma implícita. Esta es una restricción que no está directamente relacionada con el modelo pero que todos los autores han considerado en sus experimentos. Esta relajación permite al proceso de perturbación explorar regiones no factibles en el espacio de soluciones. Estas soluciones no factibles en conjunto pueden contener sub-soluciones que pueden formar parte de soluciones factibles. Cuando esto sucede, se utiliza un operador recursivo corrector para transformar soluciones no válidas en válidas. Esto se realiza utilizando internamente un evaluador de rutas, i.e., un elemento que decide individualmente si una ruta puede formar parte de una solución factible o no. Este operador se ha explicado en las secciones anteriores y supone una de las contribuciones originales de este trabajo. El algoritmo de ahorros por construcción no puede fijar el número de rutas que contiene la solución, con nuestro operador, esto es posible. Con el objetivo de comparar nuestro algoritmo con otros algoritmos conocidos, hemos utilizado dos conjuntos de instancias que son utilizadas por el resto de autores igualmente. En estas dos instancias, nuestro algoritmo ha sido capaz de obtener una mejor solución conocida hasta ahora.

En la sección final hemos analizado la flexibilidad de nuestro planteamiento en el modelo que permite retornos mezclados, i.e los clientes de recogida y entrega pueden ser alternados en cualquier secuencia dentro de cada ruta. El planteamiento que se propone es incluso más sencillo que el anterior al contener un parámetro menos.

La contribución del artículo es principalmente el desarrollo de un algoritmo relativamente fácil de implementar y eficiente. El algoritmo contiene tan solo tres parámetros para resolver el problema de las rutas de vehículos con retornos agrupados. Internamente, el algoritmo está basado en: un marco metaheurístico, un componente que utiliza un mecanismo de memoria (cache) de rutas y una versión aleatoria de un conocido heurístico. Esta aleatoriedad es inducida por una distribución de probabilidad sesgada (la geométrica) en lugar de utilizar una distribución uniforme. Los beneficios de utilizar este tipo de distribución son la velocidad y el uso de menos parámetros con lo que el algoritmo es más fácil de implementar y de ajustar.

En la sección final del artículo se discute cómo nuestro algoritmo se puede extender de forma sencilla para resolver el problema general en que los clientes de recogida y entrega se

pueden visitar en cualquier orden. Se realiza esta adaptación y se resuelve el problema utilizando unas instancias conocidas en la literatura para este nuevo modelo.

A pesar de que el algoritmo estaba inicialmente designado para resolver el problema de los retornos agrupados, los resultados del algoritmo para el nuevo modelo muestran resultados competitivos con bajos tiempos de computación. Esta es una característica muy positiva sobre todo teniendo en cuenta que el algoritmo es relativamente fácil de implementar y requiere muy pocos ajustes de parámetros.

6.3.2. Análisis del problema FSMVRPB en el artículo aceptado de la revista ITOR.

Belloso, J.; Juan, A.; Faulin, J. (2017): An Iterative Biased-Randomized Heuristic for the Fleet Size and Mix Vehicle Routing Problem with Backhauls. *Int. Transactions in Operational Research* (indexed in ISI SCI, 2015 IF = 1.255, Q2; 2015 SJR = 1.179, Q1). ISSN: 0969-6016. ISSN: 0969-6016.

En este artículo se analiza el modelo conocido como Tamaño de la Flota y composición en Problemas de vehículos con retorno. Este es una realística variante del popular problema de rutas de vehículos en el que tenemos dos tipos de clientes, de recogida y de entrega que son servidos desde un depósito central utilizando una flota heterogénea. Esta flota tiene además la característica de ser configurable, tanto en tamaño como en composición de cada vehículo. El objetivo en este caso es la optimización (minimización de costes) de las rutas empleando la flota adecuada dentro de los tipos de vehículos disponibles. El algoritmo propuesto utiliza un método iterativo que resuelve una serie de instancias más pequeños de la versión del problema con flota homogénea. Utilizando estas sub-soluciones como soluciones parciales disjuntas se compone la solución propuesta. Una de las principales aportaciones de este artículo es que cada problema homogéneo puede resolverse utilizando cualquier algoritmo que sea competitivo. En particular utilizamos el presentado en Belloso et al (2015). Con el objetivo de guiar la exploración del espacio de soluciones, el algoritmo emplea varios procesos de aleatoriedad sesgada.

- Seleccionar un tipo de vehículos dentro de la lista de disponibles.

- Ordenación de la lista de ahorros.
- El número de rutas de cada sub-solución homogénea que pasan a formar parte de la solución heterogénea.

Para poder comparar la eficiencia de nuestro algoritmo con otros algoritmos conocidos para el mismo problema, hemos utilizado dos conjuntos de instancias que son utilizadas por el resto de autores. Nuestros experimentos muestran que nuestra propuesta es competitiva ya que obtiene 20 nuevas mejores soluciones de un total de las 36 instancias propuestas recientemente en la literatura.

El metaheurístico propuesto está particularizado para el caso en que la flota es configurable pero también es válido para el caso en que la flota es fija y heterogénea. En este segundo caso el objetivo pasa a ser la mejor utilización de la flota disponible, i.e., todos los vehículos disponibles deben ser utilizados en las rutas. Para que nuestro metaheurístico se pueda utilizar en esta variante, bastaría eliminar el paso aleatorio mencionado anteriormente en tercer lugar ya que el número de rutas utilizadas de cada solución homogénea pasa a ser un número fijo determinado por la flota disponible. Esta variable aleatoria pasa a ser otra que determine cuál(es) de las rutas de la solución homogénea pasan a formar parte de la solución heterogénea. La selección de las rutas determinará la calidad de la solución final. El carácter aleatorio de esta variable, y el planteamiento multi-arranque propuesto garantiza la obtención de buenas soluciones que en cualquier caso deberán ser analizadas en futuros trabajos como se propone en la sección trabajos futuros.

6.3.3. Análisis del problema VRPCB en el artículo publicado de la revista Lecture Notes in Management Science.

Belloso, J., Juan, A., Faulin, J., Serrano, A. 2015. Using Multi-Start Biased Randomization of Heuristics to Solve the Vehicle Routing Problem with Clustered Backhauls. *Lecture Notes in Management Science* 7, pp. 15–20.

En este artículo se analiza el problema de las rutas de vehículos con Backhauls (VRPB), donde los clientes de entrega y recogida son servidos desde un depósito central. En particular,

el grupo de clientes de entrega tiene que ser servido antes de que el primer cliente de recogida pueda ser visitado. Así, el problema pertenece a una subclase del VRP con entregas y recogidas agrupadas VRPCB. Nuestro procedimiento de resolución utiliza un enfoque multi-arranque diseñado para evitar los mínimos locales y para ser fácilmente paralelizable. El algoritmo emplea una versión pseudoaleatoria de una versión del heurístico de ahorros, junto con algunos procesos de búsqueda local. Durante el proceso de construcción de soluciones, los arcos que conectan un cliente de entrega con un cliente de recogida se penalizan para ser elegidos en una etapa posterior. El orden de la lista de ahorros se altera utilizando una distribución de probabilidad sesgada.

La metodología propuesta aplica una aleatorización basada en arranques múltiples combinada con un mecanismo de aprendizaje de memoria y estrategias de división del problema en problemas reducidos. Posiblemente este es el primer algoritmo aleatorio que es capaz de competir con otros metaheurísticos en la resolución del VRPB. Además, gracias a la naturaleza multi-arranque del enfoque constructivo, se pueden generar muchas soluciones alternativas según las necesidades del experimento. Así, el decisor puede evaluar diferentes soluciones considerando no sólo los criterios apriorísticos sino también otros conocidos en una etapa posterior. El algoritmo presentado aquí es fácilmente paralelizable y no requiere ningún proceso particular de ajuste fino o configuración. Estas propiedades hacen de este procedimiento una alternativa interesante a otros enfoques más complejos. Se ha utilizado un conjunto de instancias de referencia para probar nuestro algoritmo, y los resultados computacionales obtenidos muestran que nuestro enfoque es capaz de proporcionar soluciones de alta calidad en tiempo real a la mayoría de las instancias.

6.4. Congresos.

6.4.1. Congresos Internacionales.

1. Belloso, J., Mallor, F. y Azcarate, C. 2016. Bull running in Pamplona: A methodological approach to deliver the Risk Response Plan. Proceedings of ORAHS 2016, The European Working group on O.R. Applied to Health Services. Universidad Pública de Navarra, 26 de Julio, 2016. Sesión plenaria.
2. Belloso, J., Juan, A. y Faulin, J. 2015. A Sim-Heuristic algorithm for the Single Period Inventory Routing in Vehicle Routing Problems with Backhauls and stochastic consums

- (IRPB). Proceedings of III Simposio Internacional CEA de Modelado y Simulación. Universidad Pública de Navarra, Campus de Tudela, 16 de Junio, 2016.
3. Belloso, J., Juan, A. y Faulin, J. 2015. A Recursive Corrective Operator for fixing the Fleet Size constraint in Vehicle Routing Problems with Backhauls. Proceedings of Green COOP-CYTED Workshop. Universidad Politécnica de Madrid, October 16th 2015.
 4. Belloso, J., Juan, A. y Faulin, J. 2015. A Successive Approximation Framework for Solving the Fleet Size and Mix Vehicle Routing Problem with Backhauls. GreenCOOP-CYTED Workshop. Universidad Politécnica de Madrid, September 3-4, 2015.
 5. Belloso, J., Juan, A. y Faulin, J. 2015 Solving the Fleet Size and Mix Vehicle Routing Problem with Backhauls using a Multi-Round Biased Randomized Heuristic Method. Proceedings of Verolog 2015, The fourth meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization. University of Vienna, 8-10 June 2015.
 6. Belloso, J., Juan, A. y Faulin, J. 2015. Fleet Size and Mix Vehicle Routing Problem with Backhauls: A Multi-Start Approach. Proceedings of ICAOR 2016-8th International Conference on Applied Operational Research. Rotterdam, The Netherlands, 28-30 June 2016.
 7. Belloso, J., Juan, A. y Faulin, J. 2015. Solving the Fleet Size and Mix Vehicle Routing Problem with Backhauls: A successive approximation approach. Proceedings of INFORMS Annual meeting – Philadelphia 2015. November 1-4, 2015 Philadelphia Convention Center.
 8. Belloso, J., Juan, A. y Faulin, J. 2015. Solving the Green Capacitated Vehicle Routing Problem with Backhauls at Eroski: A Revisited Case Study. Proceedings of EURO 2015 – European Conference on Operational Research. University of Strathclyde, Glasgow 12-15 July 2015.
 9. Belloso, J., Juan, A. y Faulin, J. 2015. Fleet Size and Mix Vehicle Routing Problem with Backhauls: A random successive approximations method. Proceedings of EURO 2015 – European Conference on Operational Research. University of Strathclyde, Glasgow 12-15 July 2015.
 10. Serrano A., Faulín, J. y Belloso, J. 2015. Locating and Designing a Biorefinery Supply Chain under Uncertainty in Navarre: a Stochastic Facility Location Problem Case. Proceedings of EWGT 2015 – 18th Euro Working Group on Transportation. Delft, The Neetherlands. Delft University of Technology, Holland, 2015, 14th-16th July.

11. Belloso, J., Juan, A. y Faulin, J. 2015. Fleet Size and Mix Vehicle Routing Problem with Backhauls: A Multi-Stat approach. Proceedings of Odysseus 2015, Sixth International Workshop on Freight Transportation and Logistics. Ajaccio, France – Palais des congres – May 31- June 5.
12. Belloso, J., Juan, A. y Faulin, J. 2014. A Multi-Start Biased Method for the Vehicle Routing Problem with Backhauls and Environmental Impact. Proceedings of INFORMS annual meeting – San Francisco 2014. Hilton San Francisco Union Square & Parc 55 Wyndham San Francisco, California. November 9-12, 2014.
13. Belloso, J., Juan, A. y Faulin, J. 2014. Optimizing Routing Problems with Backhauls Using a Biased-Randomized Algorithm. Proceedings of IFORS 20th Conference of the International Federation of Operational Research Societies. Barcelona 2014, 13th-18th July.
14. Belloso, J., Juan, A. y Faulin, J. 2014. A Multi-Start Biased algorithm for the Vehicle Routing Problem with Backhauls. Proceedings of Vehicle Routing and Logistics Optimization 2014. In The third meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization – VeRoLog. Oslo, Norway, June 22-25.
15. Pérez-Bernabeu, E., Juan, A., Faulin, J. y Belloso, J. 2014. Reducing Pollutant Emissions by using Horizontal Cooperation in Road Transportation. Proceedings of SDM'2014. International Conference on Sustainable Design and Manufacturing, 28, 29 & 30 April 2014 Cardiff University, Wales, UK.
16. Belloso, J., Juan, A. y Faulin, J. 2013. A Simulation-based Heuristics for the Vehicle Routing Problem with Backhauls. 2013 CYTED-HAROSA International Workshop on Simulation-Optimization & Internet Computing. UOC-Harosa-IN3, November 21-22, 2013
17. Juan, A., Faulin, J. y Belloso, J. 2013. Horizontal Cooperation in Road Transportation: A Multi-Depot Case Illustrating Savings in Distances and Pollutant Emissions. Proceedings of the 2013 ICSO-HAROSA International Workshop on Simulation-Optimization & Internet Computing. UOC-Harosa-IN3, July 10-12, 2013 - Barcelona, SPAIN.
18. Belloso, J., Juan, A. y Faulin, J. 2013. Green Horizontal Cooperation with Nonsmooth Objective Functions in Multicriteria Decision Making. Proceedings of the EURO-INFORMS, 26th European Conference on Operational Research. SAPIENZA Università di Roma, July 1-4 Rome 2013, Dipartimento di scienze Statistiche.

19. Pérez-Bernabeu, E., Juan, A., Faulin, J. y Belloso, J. 2013. Horizontal Cooperation with Environmental Criteria in Vehicle Routing Problems. Proceedings of the Mathematics of Operations and Logistics Conference (MOLC). Universidad de Zaragoza – Zaragoza Logistic center, June 26-28, 2013.

6.4.2. Congresos nacionales.

1. Belloso, J., Juan, A. y Faulin, J 2016. Tamaño y Composición de Flotas de Vehículos en Problemas de Rutas con Retornos: Un Método Heurístico con Aleatoriedad Sesgada. 1º Congreso de Doctorandos. Universidad Pública de Navarra. Universidad Pública de Navarra, edificio del Sario, 5 de Mayo, 2016.
2. Belloso, J., Mallor, F. y Azcarate, C 2016. El encierro de Pamplona: un acercamiento matemático cuantitativo al plan de respuesta a los riesgos. 8º Ciclo de Charlas de Divulgación Científica. Universidad Pública de Navarra. Escuela Técnica superior de Ingenieros industriales y de telecomunicación. Universidad Pública de Navarra, campus de Tudela, 20 de Abril, 2016
3. Belloso, J., Mallor, F. y Azcarate, C 2016. Bull running in Pamplona: A methodological approach to deliver the Risk Response Plan. 1º Congreso de Doctorandos. Universidad Pública de Navarra. Universidad Pública de Navarra, edificio del Sario, 6 de Mayo, 2016
4. Belloso, J., Mallor, F. y Azcarate, C 2016. El encierro de Pamplona: ¿Podemos ordenar los riesgos?. Ciencia en el Bar. Universidad Pública de Navarra, Universidad Pública de Navarra, 3 de Julio, 2016
5. Belloso, J., Faulin, J. y Serrano A. 2015. Tamaño y Composición de Flotas de Vehículos en Problemas de Rutas con Retornos: Un Método Heurístico con Aleatoriedad Sesgada. Proceedings of SEIO 2015 – XXXV Congreso Nacional de Estadística e Investigación Operativa IX Jornadas de Estadística Pública. Edificio El Sario, Universidad Pública de Navarra, Pamplona. Del 26 al 29 de mayo de 2015.
6. Belloso, J., Faulin, J. y Serrano A. 2015. Solving a Location Routing Problem to Pinpoint a Biorefinery Plant in Navarre. Proceedings of SEIO 2015 – XXXV Congreso Nacional de Estadística e Investigación Operativa, IX Jornadas de Estadística Pública. Edificio El Sario, Universidad Pública de Navarra, Pamplona. Del 26 al 29 de mayo de 2015.

7. Belloso, J., Juan, A. y Faulin, J. 2015. Tamaño y Composición de Flotas de Vehículos en Problemas de Rutas con Retornos: Un Método Heurístico con Aleatoriedad Sesgada. 7º Ciclo de Charlas de Divulgación Científica. Universidad Pública de Navarra. Escuela Técnica superior de Ingenieros industriales y de telecomunicación. Universidad Pública de Navarra, campus de Tudela. 13 de Abril, 2015.
8. Belloso, J., Juan, A. y Faulin, J. 2015. Tamaño y Composición de Flotas de Vehículos en Problemas de Rutas con Retornos: Un método heurístico con Aleatoriedad sesgada. MAEB 2015 - X Congreso Español de Meta-heurísticos, Algoritmos Evolutivos y Bioinspirados. Centro Universitario de Mérida - Universidad de Extremadura. Mérida-Almendralejo, 4 al 6 de Febrero de 2015.

Bibliografía.

- Adenso-Díaz, B., García-Carbajal, S., y Lozano, S. 2012. A GRASP Algorithm for the Routes Conjoint Planning in Horizontal Cooperation Scenarios. Proceedings of the Eighth Spanish Meeting on Metaheuristics, Evolutionary and Bio- inspired Algorithms, 8-10 February 2012, Albacete (Spain), pp. 561-566 (in Spanish).
- Adenso-Díaz, B., Lozano, S., Garcia-Carbajal, S., Smith-Miles, K. 2014. Assessing partnership savings in horizontal cooperation by planning linked deliveries. *Transportation Research Part A: Policy and Practice* 66, pp. 268–279.
- Adler N. 2001. Competition in a deregulated air transportation market. *European Journal of Operational Research* 129, pp. 337–345.
- Álvarez-SanJaime, O., Cantos-Sánchez, P., Moner-Colonques, R. and. Sempere-Monerris, J. 2013a. Vertical integration and exclusivities in maritime freight transport. *Transportation Research Part E: Logistics and Transportation Review*, 51, pp. 50–61.
- Álvarez-SanJaime, O., Cantos-Sánchez, P., Moner-Colonques, R. y Sempere-Monerris, J. 2013b. Competition and horizontal integration in maritime freight transport. *Transportation Research Part E: Logistics and Transportation Review*, 51, pp. 67–81.
- Anily, S. 1996. The vehicle-routing problem with delivery and back-haul options. *Naval Research Logistics* 43, pp 415–434.
- Apfelstädt, A., Dashkovskiy, S. y Nieberding, B. 2016. Modeling, Optimization and Solving Strategies for Matching Problems in Cooperative Truckload Networks ,IFAC-PapersOnLine 49 (2), pp. 18-23.
- Archetti, C. y Speranza, M.G. 2014. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, vol. 2, no. 4, pp. 223–246.
- Bahinipati, B. K., Kanda, A. y Deshmukh. S.G. 2009. Horizontal collaboration in semiconductor manufacturing industry supply chain: An evaluation of collaboration intensity index. *Computers & Industrial Engineering*, Vol. 57, pp. 880-895.

- Bahrami, K. 2002. Improving Supply Chain Productivity Through Horizontal Cooperation: The Case of Consumer Good Manufacturers. In *Cost Management in Supply Chains*, edited by Seuring, S., y M. Goldbach, 213–232. Heidelberg: Physica-Verlag.
- Bailey E., Unnikrishnan A., Lin D-Y. 2011. Models for Minimizing Backhaul Costs through Freight Collaboration. *Journal of Transportation Research Board*, 2224, pp. 51-60.
- Baldacci, R., Battarra, M. y Vigo D. 2008. Routing a Heterogeneous Fleet of Vehicles. In: *Golden BL. Raghavan S. Wasil EA. Editors. New York: Springer.*
- Barbarino, S. 2011. From CO₂ to CO₃. Presentation at the 2nd horizontal collaboration summit. <http://events.eft.com/SCHC/past-presentations.shtml>. (Last access 25 Oct 2016).
- Bektas, T., and Laporte. G. 2011. The Pollution Routing Problem. *Transportation Research Part B*, 45 (8), pp.1232-1250.
- Belloso, J., Juan, A., Faulin, J., Serrano, A. 2015. Using Multi-Start Biased Randomization of Heuristics to Solve the Vehicle Routing Problem with Clustered Backhauls. *Lecture Notes in Management Science* 7, pp. 15–20.
- Bengtsson, M. y Kock, S. 1999. Cooperation and Competition in Relationships between Competitors in Business Networks. *The Journal of Business and Industrial Marketing*, 14 (3), pp. 178-194.
- Bengtsson, M. y Kock, S. 2000. Cooperation in business networks--to cooperate and compete simultaneously. *Industrial Marketing Management*, 29 (5). pp. 411-426.
- Bengtsson, M. y Raza-Ullah, T. 2016. A systematic review of research on coopetition: Toward a multilevel understanding. *Industrial Marketing Management* 57, pp. 23-39.
- Berbeglia, G., Cordeau, J.F., Gribkovskaia I. y Laporte, G. 2007. Static pickup and delivery problems: a classification scheme and survey. *TOP* 15 (1), pp. 1–31.
- Boctor F.F y Renaud, J. 1993. A tour improvement procedure for the travelling salesman problem. Working paper 93-37, Faculté des Sciences de l'Administration, Université Laval, Canada.

- Boese, K. D., Kahng, A. B., y Muddu. S. 1994. A new adaptive multi-start technique for combinatorial global optimization. *Operations Research Letters*, 16, pp. 101–113.
- Brandao, J. 2006. A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research* 173, pp. 540–555.
- Bresina, J.L. 1996. Heuristic-biased stochastic sampling. In *Proceedings of the AAAI-96*, pp. 271-278.
- Brimberg, J. y Love, R.F. 1998. Solving a Class of Two-Dimensional Uncapacitated Location-Allocation Problems by Dynamic Programming. *Operations Research* 46, pp. 702-709.
- Burke, E. K., Kendall, G. y Whitwell, G. 2004. A new placement heuristic for the orthogonalstock-cutting problem. *Operations Research*, 52(4), pp. 655–671.
- Caceres, J., Juan, A., Grasman, S., Bektas, T. y Faulin, J. 2012. Combining monte carlo simulation with heuristics for solving the inventory routing problem with stochastic demands. In *Proceedings of the 2012 Winter Simulation Conference*, Berlin, Germany.
- Cachon, G. y M. Lariviere. 1999. Capacity choice and allocation: strategic behavior and supply chain performance. *Management Science*. 45(8), pp. 1091-108.
- Capgemini and Global Commerce Initiative. 2008. *Future Supply Chain 2016*. Tech. Rep. Global Commerce Initiative and Capgemini.
- Caputo, M. y Mininno, V. 1996. Internal, vertical and horizontal logistics integration in Italian grocery distribution. *International Journal of Physical Distribution and Logistics Management*, 26(9), pp. 64-89.
- Chiambaretto, P., Gurău, C. y Le Roy, F. 2016. Coopetitive branding: Definition, typology, benefits and risks. *Industrial Marketing Management* 57, pp. 86-96.
- Choi, E. y Tcha, D. W. 2007. A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem. *Computers and Operations Research* 34 (7), pp. 2080–95.

- Christofides, N. y Eilon, S. 1969. An Algorithm for the Vehicle Routing Dispatching Problem. *Operations Research Quarterly*, 20, pp. 309-318.
- Chu, C.-W. 2005. A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research*, 165, pp. 657–667.
- Clarke, G. y Wright, J. 1964. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research* 12 (4), pp. 568–81.
- Clay Mathematics Institute. 2000. Millenium problems. <http://www.claymath.org/millennium/>. (Último acceso 23 de Enero 2017).
- Cordeau, J.F., Gendreau, M., Laporte, G., Potvin, J. y Semet, F. 2002. A guide to vehicle routing heuristics. *Journal of Operations Research Society*, 53, pp. 512-522.
- Crujssen, F. 2006. Horizontal cooperation in transport and logistics, PhD thesis, Faculty of Economics and Business Administration, Center Dissertation Series, Tilburg University, Tilburg. Downloaded from <http://arno.uvt.nl/show.cgi?fid=55815> (Last access 25 Oct 2016).
- Crujssen, F., Cools, M. y Dullaert, W. 2007a. Horizontal Cooperation in Logistics: Opportunities and Impediments. *Transportation Research Part E*, 43, pp.129-142.
- Crujssen, F., Dullaert, W. y Fleuren, H. 2007b. Horizontal cooperation in transport and logistics: A literature review. *Transportation Journal*, 46 (3), pp. 22-39.
- Cuervo, D.P. Goos, P., Sörensen, K. y Arraiz, E. 2014. An Iterated Local Search Algorithm for the Vehicle Routing Problem with Backhauls. *European Journal of Operational Research*, 237, pp. 454-464.
- Dahl S. y Derigs U. 2011. Cooperative planning in express carrier networks-An empirical study on the effectiveness of a real-time Decision Support System. *Decision Support Systems*, 51, pp. 620-626.

- Dai, B. y Chen H. 2011. Price-Setting Based Combinatorial Auction Approach for Carriers. In ICAART 2011 - Proceedings of the 3rd International Conference on Agents and Artificial Intelligence, pp. 407-413.
- Dantzig, G.B. y Ramser, J.H. 1959. The truck dispatching problem. *Management Science*, 6:80.
- Dantzig, G.B. y Wolfe, P. 1960. Decomposition Principle for Linear Programs. *Operations Research* 8(1), pp. 101-111.
- Dantzig, G.B. y Wolfe, P. 1961. The decomposition Algorithm for Linear Programs. *Econometrica* 29(4), pp. 767-778.
- Deif, I. y Bodin, L. 1984. Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauls. In Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management, Babson Park, pp. 75–96.
- Dekker, R., Bloemhof, J. y Mallidis. I. 2012. Operations Research for Green Logistics. An Overview of Aspects, Issues, Contributions and Challenges. *European Journal of Operational Research*, 219 (3), pp. 671-679.
- Dell’Amico, M., Righini, G., y Salani, M. 2006. A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transport Sci.*, 40, pp 235–247.
- Denebourg, J.L., Pasteels, J.M. y Verhaeghe, J.C. 1983. Probabilistic Behaviour in Ants: a Strategy of Errors? *Journal of Theoretical Biology*, 105.
- Dethloff, J. 2002. Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society* 53, pp. 115–118.
- Dodgson, M. 1992. Technological collaboration: Problems and pitfalls. *Technology Analysis & Strategic Management*, 4(1), pp. 83-88.

- Dominguez, O., Juan, A., Barrios, B., Faulin, J. y Agustin, A. 2016. Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, 236 (2), pp. 383-404, doi: 10.1007/s10479-014-1551-4.
- Dorigo, M. 1992. Optimization, learning and natural algorithms. PhD thesis, Politécnico de Milán, Italia.
- Dorigo, M. y Stutzle, T. 2010. Ant colony optimization: overview and recent advances. In Gendreau, M., and Potvin, J. (eds.), *Handbook of metaheuristics. International series in operations research management science*, 146, 2nd edition, Kluwer Academic, Dordrecht, pp. 227-264.
- Eekhout, B.van. 2001. *Global Traffic Management, beheersen van intercontinental transport als voorwaarde voor het realiseren van efficiënte logistieke ketens*, TenHagen & Stam, The Hague (in Dutch).
- Engvall, S., Göthe-Lundgren, M. y Värbrand, P. 1998. The traveling salesman game: An application of cost allocation in a gas and oil company. *Annals of Operations Research* 82, pp. 453-471.
- Engvall, S., Göthe-Lundgren, M. y Värbrand, P. 2004. The heterogeneous vehicle-routing game. *Transportation Science* 38 (1), pp. 71–85.
- Erdmann, M. 1999. Consolidation of freight forwarding potential collaborations: A simulation-based analysis. Deutscher Universitäts-Verlag, Köln. (In German).
- Erdogan, S. y Miller-Hooks, E. 2012. A Green Vehicle Routing Problem. *Transportation Research Part E: Logistics and Transportation Review*, 48 (1), pp. 100-113.
- European Commission 2011. *EU transport in figures*. Publications Office of the European Union. Downloaded from <http://ec.europa.eu/transport/facts-funding/statistic/pocketbook-2016.pdf> (ultimo acceso 10 Octubre 2016).
- European Union 2001. Guidelines on the Applicability of Article 81 of the AC Treaty to Horizontal Cooperation Agreements. European Commission Notice 2001/C 3/02.

- Fan, T., Vigeant-Langlois, L., Geissler, C., Bosler, B. y Wilmking, J. 2001. Evolution of global airline strategic alliance and consolidation in the twenty-first century. *Journal of Air Transport Management*, 7 (6), pp. 349-360.
- Faulin, J., Juan, A., Grasman, S. y Fry, M. 2012. *Decision Making in Service Industries: A Practical Approach*. CRC Press – Taylor & Francis, Clermont, FL, USA.
- Festa, P. y Resende, M. 2009a. An annotated bibliography of GRASP – Part I: algorithms. *International Transactions in Operational Research*, 16, pp. 1-24.
- Festa, P. y Resende, M. 2009b. An annotated bibliography of GRASP – Part II: applications. *International Transactions in Operational Research*, 16, pp. 1-24.
- Ford E., Wells R. y Bailey B. 2004. Sustainable network advantages: A game theoretic approach to community-based health care coalitions. *Health Care Mngt Rev* 29, pp. 159–169.
- Frisk, M., Göthe-Lundgren, M., Jörnsten, K. y Rönnqvist, M. 2010. Cost allocation in collaborative forest transportation. *European Journal of Operational Research*, 205(2), pp. 448-458.
- Gajpal, Y. y Abad, P. 2009. Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European Journal of Operational Research* 196, pp. 102–117.
- Garcia-Najera, A. 2012. The vehicle routing problem with backhauls: a multi-objective evolutionary approach. In: Hao, J.-K., Middendorf, M. (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 7245. Springer, pp. 255–266.
- Garey, M.R. y Johnson, D.S. 1979. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H: Freeman and Company.
- Gendreau, M. 1999. A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem. *Computers & Operations Research* 26, pp. 1153–1173.

- Gendreau, M., Hertz, A. y Laporte, G. 1992. New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, 40 (60), pp. 1086-1093.
- Gillett, B. y Miller, L. 1974. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 22 (2), pp. 340–349.
- Glover, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computational Operations Research*, 13, pp. 533–549.
- Glover, F. 1989. Tabu Search — Part I. *ORSA Journal on Computing* 1989 1: 3, pp. 190-206.
- Glover, F. 1990. Tabu Search — Part II. *ORSA Journal on Computing* 1990 2: 1, pp. 4-32.
- Goetschalckx, M. y Jacobs-Blecha, C. 1989. The Vehicle Routing Problem with Backhauls. *European Journal of Operational Research*, no. 42, pp. 39–51.
- Goetschalckx, M. y Jacobs-Blecha, C. 1993. The vehicle routing problem with backhauls: Properties and solution algorithms. Technical Report MHRC-TR-88-13, Georgia Institute of Technology, Atlanta.
- Goldberg, D.E. 1989. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. *Genetic Algorithms in Search, Optimization and Machine Learning*.
- Golden, B., Alfaro, J., Baker, E. y Schaffer, J. 1985. The vehicle routing problem with backhauling: two approaches. In: *Proceedings of the 21st Annual Meeting of the Southeast Institute of Management Science*, pp. 90–92.
- Golden, B., Assad, A. y Levy, L. 1984. The Fleet Size and Mix Vehicle Routing Problem. *Computers & Operations Research* 11, pp. 49–66.
- Gomory, R.E. 1958. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Society*, 64, pp. 275-278.
- Grasas, A., Juan, A. y Ramalhinho H. 2016. SimILS: A Simulation-based extension of the Iterated Local Search metaheuristic for Stochastic Combinatorial Optimization. *Journal of Simulation*, 10(1), pp. 69-77.

- Guajardo, M. y Rönnqvist, M. 2016. A review on cost allocation methods in collaborative transportation. *International Transactions in Operational Research*, 23, pp. 371–392.
- Hagedoorn, J 1994. The effects of strategic technology alliances on company performance. *Strategic Management Journal*, 15, pp. 291-309.
- Halse, K. 1992. Modeling and Solving Complex Vehicle Routing Problems. PhD thesis, Institute of Mathematical Statistics and Operations Research (IMSOR), Technical University of Denmark.
- Hamers, H., Borm, P., van de Leensel, R. y Tijs, S. 1999. Cost allocation in the Chinese postman problem. *European Journal of Operational Research* 118, 1, pp. 153–163.
- Held, M. y Karp, R.M. 1970. The traveling salesman problem and minimum spanning trees. *Operations Research* 18, pp. 1138-1162.
- Held, M. y Karp, R.M. 1971. The traveling salesman problem and minimum spanning trees:Part II. *Mathematical Programming* 1, pp. 6-25.
- Hoff, A., Andersson, H., Christiansen, M., Hasle, G. y Løkketangen, A. 2010. Industrial Aspects and Literature Survey: Fleet Composition and Routing. *Computers & Operations Research* 37 (12), pp. 2041–61.
- Hoffman, K. L. y Ralphs, T. K. 2013. Integer and combinatorial optimization. In *Encyclopedia of Operations Research and Management Science*, pp. 771-783. Springer US.
- Hsu, J.-L. y Lai Y.-C. 2014. Automatic playlist generation by applying Tabu Search. *International Journal of Machine Learning and Cybernetics* 5(4), pp. 553-568.
- Hundt, R. 2011. Loop recognition in c++/java/go/scala. *Proceedings of Scala Days*. Disponible: <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37122.pdf>.
- Irnich, S. 2000. A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. *European Journal of Operational Research*, 122(2), pp. 310-328.

- Jabali, O., Van Woensel, T. y De Kok, A.G. 2012. Analysis of Travel Times and CO2 Emissions in Time-Dependent Vehicle Routing. *Production and Operations Management*, 21 (6), pp. 1060-1074.
- Juan, A., Faulin, J., Caceres, J., Barrios, B. y Martinez, E. 2014a. A Successive Approximations Method for the Heterogeneous Vehicle Routing Problem: analyzing different fleet configurations. *European Journal of Industrial Engineering*, 8(6), pp. 762–788.
- Juan, A., Faulin, J., Ferrer, A., Lourenço, H.R. y Barrios, B.B. 2011a. MIRHA: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *TOP*, pp. 1-24. ISSN 1134-5764. doi: 10.1007/s11750-011-0245-1. URL <http://dx.doi.org/10.1007/s11750-011-0245-1>.
- Juan, A., Faulin, J., Grasman, S., Rabe, M. y Figueira, G. 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2, pp. 62-72.
- Juan, A., Faulin, J., Grasman, S., Riera, D., Marull, J. y Mendez, C. 2011b. Using safety stocks and simulations to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C*, 1, pp. 751-765.
- Juan, A., Faulin, J., Jorba, J., Riera, D., Masip, D. y Barrios, B. 2011c. 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* 62(6), pp. 1085-1097.
- Juan, A., Faulín, J., Pérez-Bernabeu, E. y Jozefowicz, N. 2014b. Horizontal cooperation in Vehicle Routing problems with backhauling and environmental criteria. *Procedia-Social and Behavioral Sciences*, 111, pp 1133-1141.
- Juan, A., Faulin, J., Ruiz, R., Barrios, B. y Caballe, S. 2010. The SRGCWS hybrid algorithm for solving the capacitated vehicle routing. *Applied Soft Computing*, 10, pp. 215–224.
- Kirkpatrick, S., Gelatt, C. D. y Vecchi, M. P. 1983. Optimization by Simulated Annealing. *Science* 220 (4598), pp. 671-680.

- Kotlet, P. 1991. *Marketing Management. Analysis, Planning, Implementation and Control*. Prentice Hall International. United Kingdom.
- Krajewska, M., Kopfer, H., Laporte, G., Ropke, S. y Zaccour, G. 2008. Horizontal cooperation of freight carriers: request allocation and profit sharing. *Journal of the Operational Research Society*, 59, pp. 1483–1491.
- Kuhn, H.W. 1955. The Hungarian method for the assignment algorithm. *Naval Research Logistics Quarterly* 1, pp. 88–97.
- L'Ecuyer, P., Meliani, L. y Vaucher, J. 2002. A framework for stochastic simulation in java. En *Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers, WSC '02*, pp. 234–242. Winter Simulation Conference.
- Lawler, E. L. 2001. *Combinatorial optimization: networks and matroids*. Courier Corporation.
- Lee, Y.H., Kim, J.I., Kang, K.H. y Kim, K.H. 2008. A heuristic for vehicle fleet mix problem using tabu search and set partitioning. *Journal of the Operational Research Society* 59, pp. 833–841.
- Leitner, R., Meizer, F., Prochazka, M. y Sihn, W. 2011. Structural concepts for horizontal cooperation to increase efficiency in logistics. *CIRP Journal of Manufacturing Science and Technology*, 4 (3), pp. 332-337.
- Lenstra, J. y Rinnooy Kan, A. 1981. Complexity of Vehicle Routing and scheduling problems. *Networks* 11, pp 221-227.
- Leon, R. y Juan, A. 2015. Promoting Corporate Social Responsibility in Logistics throughout Horizontal Cooperation. *Managing Global Transitions* 12 (1), pp. 79–93.
- Lera-López, F., Faulin, J. y Sánchez. M. 2012. Determinants of the Willingness-to-Pay for Reducing the Environmental Impacts of Road Transportation. *Transportation Research Part D*, 17(3), pp. 215-220.
- Lin, S. 1965. Computer solutions of the travelling salesman problem. *Bell System Technical Journal* 44, pp. 2245–2269.

- Liu, F-H. y Shen, S-Y. 1999. The Fleet Size and Mix Vehicle Routing Problem with Time Windows. *Journal of the Operational Research Society* 50 (7), pp. 721–32.
- Liu, P., Wu, Y. y Xu, N. 2010. Allocating Collaborative Profit in Less-than-Truckload Carrier Alliance. *Journal of Service Science & Management*, 3, pp. 143-149.
- Lourenço, H. Martin, O. y Stutzle, T. 2010. Iterated local search: framework and applications. In, Gendreau, M. and Potvin, J. (eds.), *Handbook of metaheuristics. International series in operations research management science*, 146, 2nd edition, Kluwer Academic, Dordrecht, pp. 363-397.
- Lourenço, H. R., Martin, O. C. y Stutzle, T. 2003. Iterated local search. En Glover, F. y Kochenberger, G., editores, *Handbook of Metaheuristics*, volumen 57 de International Series in Operations Research & Management Science, pp. 320–353. Springer US.
- Lozano, S., Moreno, P., Adenso-Díaz, B. y Algaba, E. 2013. Cooperative game theory approach to allocating benefits of horizontal cooperation. *European Journal of Operational Research*, 229 (2), pp. 444-452.
- Lyons, A. C., Mondragon, A. E. C., Piller, F. y Poler, R. 2012. Network Collaboration: Vertical and Horizontal Partnerships. Book chapter in Lyons, A. C., Mondragon, A. E. C., Piller, F., and Poler, R., *Customer-Driven Supply Chains*, pp. 95-111.
- Menedeme, L. 2011. Prepare Your Supply Chain for Horizontal Collaboration. Presentation at the 2nd horizontal collaboration summit. <http://events.eft.com/SCHC/past-presentations.shtml>. (último acceso 2 Febrero 2017).
- Montoya, J., Juan, A., Huaccho, L., Faulin, J. y Rodríguez-Verjan, G. 2011. Hybrid algorithms for service, computing and manufacturing systems: routing and scheduling solutions. IGI Global, Hershey.
- Myerson, R.B. 1992. On the Value of Game Theory in Social Science. *Rationality and Society*, 4 (1), pp. 62-73.

- Nadarajah, S. 2008. Collaborative Logistics in Vehicle Routing. Master thesis, University of Waterloo, Ontario, Canada. Downloaded from <http://uwspace.uwaterloo.ca/bitstream/10012/4083/1/nadarajah-thesis.pdf>. (último acceso 25 Oct 2016).
- Nagy G. y Salhi, S. 2005. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* 162, pp 126–141.
- Nikolaev, A. y Jacobson, S. 2010. Simulated annealing. In, Gendreau, M. and Potvin, J. (eds.), *Handbook of metaheuristics*. International series in operations research management science, 146, 2nd edition, Kluwer Academic, Dordrecht, pp. 1-40.
- Osman, I. y Wassan, N. 2002. A Reactive Tabu Search for the Vehicle Routing Problem with Backhauls. *Journal of Scheduling* 5, no. 4, pp. 263–85.
- Oum, T., Park, J. y Zhang, A. 2000. *Globalisation and Strategic Alliances: the Case of the Airline Industry*. Elsevier Science, London.
- Palander, T. y Väättäinen, J. 2005. Impacts of Inter-enterprise Collaboration and Backhauling on Wood Procurement in Finland. *Scandinavian Journal of Forest Research* 20 (2), pp. 177–183.
- Papadimitriou, C. y Steiglitz, K. 1982. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, New Jersey.
- Parragh, S., Doerner, K. y Hartl, R. 2008. A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *Journal für Betriebswirtschaft* 58 (1), pp. 21–51.
- Pérez-Bernabeu, E., Juan, A., Faulin, J. y Barrios, B. 2015: Horizontal Cooperation in Road Transportation: A Case Illustrating Savings in Distances and Greenhouse Gas Emissions. *International Transactions in Operational Research*, 22, pp. 585–606.

- Pulleman, P. 2011. Maatschappelijk Verantwoord Kosten Besparen Door Bundelen Van Goederenstromen. Internet article. <http://www.goederenstromen.nl/pers-downloads.html>. (ultimo acceso 20 octubre 2016).
- Quintero-Araujo, C., Gruler y Juan, A. 2016. Quantifying Potential Benefits of Horizontal Cooperation in Urban Transportation Under Uncertainty: A Simheuristic Approach. *Advances in Artificial Intelligence*. Volume 9868 of the series *Lecture Notes in Computer Science*, pp 280-289.
- Reeves, C. 2010. Genetic Algorithms. In, Gendreau, M. and Potvin, J. (eds.), *Handbook of metaheuristics*. International series in operations research management science, 146, 2nd edition, Kluwer Academic, Dordrecht, pp. 109-140.
- Reimann, M. y Ulrich, H. 2006. Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research* 14, pp. 105–123.
- Renaud, J. y Boctor, F. F. 2002. A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem. *European Journal of Operational Research* 140 (3), pp. 618–28.
- Resende M.G. y Ribeiro, C.C. 2010. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pp. 283-319. Springer US.
- Rieck J. y Zimmermann, J. 2010. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181 (1), pp. 337-358.
- Ropke, S. y Pisinger, D. 2006. A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research*, 171, pp. 750–775.
- Ropke, S., Cordeau, J.F. y Laporte, G. 2007. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49, pp. 258–272.

- Salhi S. y Nagy, G. 1999. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society* 50, pp 1034–1042.
- Salhi, S., Wassan, N. y Hajarat, M. 2013. The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-Based Heuristics. *Transportation Research Part E: Logistics and Transportation Review* 56 (September), pp. 22–35.
- Sanchez-Rodrigues, V., Harris, I. y Mason, R. 2015. Horizontal logistics collaboration for enhanced supply chain performance: an international retail perspective, *Supply Chain Management: An International Journal*, Vol. 20 No. 6, pp. 631-647.
- Sánchez-Soriano, J, López, M.A. y García-Jurado, I. 2001. On the core of transportation games. *Mathematical social sciences*, 41 (2), pp. 215-225
- Schrijver, A. 1986. *Theory of Linear and Integer Programming*. Wiley, Chichester.
- Shepperd, E. y Seidman, D. 2001. Ocean shipping alliances: the wave of the future? *International Journal of Maritime Economics*, Vol. 3 No. 4, pp. 351-367.
- Sheu, J.B. y Talley, W.K. 2011. Green supply chain management: trends, challenges, and solutions. *Transportation Research Part E* 47, pp. 791–792.
- Simatupang, T. y Sridharan, R. 2002. The collaborative supply chain. *International Journal of Logistics Management*, 13 (1), pp. 15-30.
- Smullyan, R. M. 1992. *Godel's incompleteness theorems*. Oxford University Press.
- Subramanian, A. y Penna, P. 2012. A Hybrid Algorithm for the Heterogeneous Fleet Vehicle Routing Problem. *European Journal of Operational Research* 221 (3), pp. 513-20.
- Taillard, E. D. 1999. A Heuristic Column Generation Method for the Heterogeneous Fleet VRP. *RAIRO Recherche Operationnelle* 33 (1), pp. 1–14.
- Talbi, E (ed.) 2009. *Metaheuristics: From Design to Implementation*, Wiley.

- Tarantilis, C. D., Kiranoudis, C. T. y Vassiliadis, V. S. 2003. A List Based Threshold Accepting Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem. *Journal of the Operational Research Society*. 54 (1), pp. 65–71.
- Tavasszy, L., Ruijgrok, C. y Thissen M. 2003. Emerging global logistics networks: implications for transport systems and policies. *Growth and Change: A Journal of Urban and Regional Policy*, 34 (4), pp. 456-472.
- Tillemans, T. 2011. How to Spread Horizontal Collaboration Across Your Supply Chain Networks – A HJ Heinz Case Study. Presentation at the 2nd horizontal collaboration summit. <http://events.eft.com/SCHC/past-presentations .shtml>. (último acceso 12 Enero 2017).
- Toth, P. y Vigo, D. 1996. A heuristic algorithm for the vehicle routing problem with backhauls. In L. Bianco & P. Toth (Eds.), *Advanced methods in transportation analysis*, pp. 585–608. Springer.
- Toth, P. y Vigo, D. 1997. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science* 31 (4) (1997), pp. 372–385.
- Toth, P. y Vigo, D. 1999. A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, 113, pp 528-54.
- Toth, P. y Vigo, D. 2014. *Vehicle Routing: Problems, Methods, and Applications*. SIAM Philadelphia.
- Tovey, C.A. 1988. Simulated annealing. *American Journal of Mathematical and Management Sciences*, 8, pp. 389–407.
- Turing A.M. 1950. Computing Machinery and Intelligence. *Mind*, 59, pp. 433-460.
- Tütüncü, G. 2010. An Interactive GRAMPS Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem with and without Backhauls. *European Journal of Operational Research* 201 (2). Elsevier B.V., pp. 593–600.

- Ubeda, S., Arcelus, F. J. y Faulin, J. 2011. Green Logistics at Eroski: A Case Study. *International Journal of Production Economics* 131 (1), pp. 44–51.
- Valender, R. 2011. The Future of Collaborative Supply Chains. Presentation at the 2nd horizontal collaboration summit. <http://events.eft.com/SCHC/past-presentations.shtml>. (último acceso 12 Enero 2017).
- van den Nouweland A, Borm P, van Golstein Brouwers W, Groot Bruinderink R y Tijs S. 1996. A game theoretic approach to problems in telecommunication. *Mngt Sci* 42: pp. 294–303.
- Vanovermeire, C. y Sörensen, K. 2014. Measuring and rewarding flexibility in collaborative distribution, including two-partner coalitions. *European Journal of Operational Research* 239, pp. 157–165.
- Vanovermeire, C., Sörensen, K., Van Breedam, A., Vannieuwenhuysse, B. y Verstrepen, S. 2014. Horizontal logistics collaboration: decreasing costs through flexibility and an adequate cost allocation strategy. *International Journal of Logistics Research and Applications*. 17(4), pp. 339-355.
- Vassallo, J. M., Solis, C., Perez-Martinez, P. y Perez, Y. 2014. Balance economico, fiscal, social y medio ambiental del sector transporte de mercancías en España. Fundación Francisco Correl, Madrid.
- Vidal, T. Crainic, T., Gendreau, M. y Prins, C. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3), pp. 658–673.
- Vigo, D. 1996. A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research*, 89, pp. 108-126.
- Wade A. y Salhi, S. 2001. An ant system algorithm for the vehicle routing problem with backhauls. MIC_2001—4th Metaheuristic International Conference. Porto, Portugal, July, pp. 16-20.

- Wade A. y Salhi, S. 2002. An investigation into a new class of vehicle routing problem with backhauls, *Omega* 30, pp. 487–497.
- Wang, X. y Kopfer, H. 2014. Collaborative transportation planning of less-than-truckload freight: a route based request exchange mechanism. *OR Spectrum*, 36(2), pp. 357-380.
- Wang, X. y Kopfer, H. 2015. Rolling horizon planning for a dynamic collaborative routing problem with full-truckload pickup and delivery requests. *Flexible Services and Manufacturing Journal*, 27(4), pp. 509-533.
- Wang, X., Kopfer, H. y Gendreau, M. 2014. Operational transportation planning of freight forwarding companies in horizontal coalitions. *European Journal of Operational Research*, 237(3), pp. 1133-1141.
- Wassan, N. 2007. Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *Journal of the Operational Research Society* 58 (12), pp. 1630–1641.
- Wassan, N. Wassan, N., Nagy, G. y Salhi, S. 2016. The Multiple Trip Vehicle Routing Problem with Backhauls: Formulation and a Two-Level Variable Neighbourhood Search. *Computers and Operations Research*, <http://dx.doi.org/10.1016/j.cor.2015.12.017> .
- Wassan, N., Salhi, S. y Nagy, G. 2009. Solving the mixed VRP with backhauling using ants. *AIP Conference Proceedings* 1159, pp. 248-253.
- Williamson, O.E. 1991. Comparative economic organization: The analysis of discrete structural alternatives. *Administrative Science Quarterly*, 36, pp. 269-296.
- Xu, J. y Kelly, J.P. 1996. A network flow-based tabu search heuristic for the vehicle routing problem. *Transportation Science* 30, pp. 379–393.
- Xumari, G.L. 1967. *Introduction to dynamic programming*. Wilwy & Sons Inc., New York.
- Yazgi Tüçüntü, G., Carreto, C.A.C y Baker, B.M. 2009. A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, 37 (1), pp.138-154.

Yepes, V. y Medina, J. R. 2002. Criterio económico para la optimización de rutas con flotas heterogéneas VRPHESTW. En Actas del V Congreso de Ingeniería del Transporte, pp. 693-700.

Zachariadis, E.E. y Kiranoudis, C.T. 2012. An effective local search approach for the Vehicle Routing Problem with Backhauls. *Expert Systems with Applications* 39, pp. 3174–3184.

Apéndices.

Los artículos cuyos originales que se muestran a continuación son los siguientes:

Belloso, J.; Juan, A.; Faulin, J.; Martinez, E. (2017): A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls. ACEPTADO en Networks (indexed in ISI SCI, 2015 IF = 0.943, Q3; 2015 SJR = 1.043, Q1). ISSN: 0028-3045.

Belloso, J.; Juan, A.; Faulin, J. (2016): An Iterative Biased-Randomized Heuristic for the Fleet Size and Mix Vehicle Routing Problem with Backhauls. ACEPTADO en International Transactions in Operational Research (indexed in ISI SCI, 2015 IF = 1.255, Q2; 2015 SJR = 1.179, Q1). ISSN: 0969-6016.

Belloso, J., Juan, A., Faulin, J., Serrano, A. 2015. Using Multi-Start Biased Randomization of Heuristics to Solve the Vehicle Routing Problem with Clustered Backhauls. PUBLICADO. Lecture Notes in Management Science 7, pp. 15–20.

Carpeta actual: **VeroLog**[Desconectarse](#)[Escribir](#)[Direcciones](#)[Carpetas](#)[Opciones](#)[Buscar](#)[Ayuda](#)[Calendario](#)[Lista de mensajes](#) | [No leído](#) | [Borrar](#)[Anterior](#) | [Siguiente](#)[Reenviar](#) | [Reenviar como adjunto](#) | [Responder](#) | [Responder a todos](#)[Borrar y Atrás](#) | [Borrar y Adelante](#)Mover a: **Asunto:** Manuscript Conditionally Accepted - Please submit final updates to NET-17-0004 [email ref: ENR-AW-1-c]**De:** "Networks" <onbehalfof+jcavanagh+rhsmith.umd.edu@manuscriptcentral.com>**Fecha:** Lun, 9 de Enero de 2017, 18:17**Para:** javier.belloso@unavarra.es**Cc:** bgolden@rhsmith.umd.edu**Prioridad:** Normal**Opciones:** [Ver encabezado completo](#) | [Ver versión imprimible](#) | [Bajar este mensaje como un archivo](#) | [View Message Details](#) | [Add to Address Book](#)

09-Jan-2017

Dear Mr. Belloso:

Manuscript id: NET-17-0004

Manuscript title: A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls

Although your manuscript has been conditionally accepted for publication it is now being returned to your author center for you to review and make any final adjustments or corrections prior to production and publication.

Any special instructions will be listed below:

To go directly to Step 1 in the First Look submission process for this paper, please use the link below:

https://mc.manuscriptcentral.com/net?URL_MASK=84d2af8ad7f5471fb63134dedde4d71f

On the File Upload screen please upload the FINAL versions of all the files, including any print quality image files. Specifically, upload a PDF reading copy as well as the native (Word/LaTeX/figure) files used to produce this PDF. We prefer the submission of final figure files in .eps or .tiff formats, although figures created within LaTeX/TeX can also be used.

Please refer to the following site for guidelines on image quality requirements:

http://exchanges.wiley.com/authors/digital-artwork_335.html

Instructions for uploading files for First Look:

1. On the 'File Upload' step, click on 'Select File'.
2. Browse to locate the file and add an appropriate 'File Designation'. The PDF file must be designated as 'Main Document' and should replace any existing 'Main Document'. All other files are designated 'Source Files'.

Carpeta actual: **VeroLog**[Desconectarse](#)[Escribir](#)[Direcciones](#)[Carpetas](#)[Opciones](#)[Buscar](#)[Ayuda](#)[Calendario](#)[Lista de mensajes](#) | [No leído](#) | [Borrar](#)[Anterior](#) | [Siguiente](#)[Reenviar](#) | [Reenviar como adjunto](#) | [Responder](#) | [Responder a todos](#)[Borrar y Atrás](#) | [Borrar y Adelante](#)Mover a: **Asunto:** Manuscript Accepted - Updates Received NET-17-0004 [email ref: ENR-AW-1-f]**De:** "Networks" <onbehalfof+jcavanagh+rsmith.umd.edu@manuscriptcentral.com>**Fecha:** Mar, 10 de Enero de 2017, 15:38**Para:** javier.belloso@unavarra.es**Prioridad:** Normal**Opciones:** [Ver encabezado completo](#) | [Ver versión imprimible](#) | [Bajar este mensaje como un archivo](#) | [View Message Details](#) | [Add to Address Book](#)

10-Jan-2017

Dear Mr. Belloso:

NET-17-0004

This is an automatic email to thank you for returning your final files. These will now be checked and you will be informed if any further updates are needed. If no updates are required the files will be sent to production.

For reference, Author Response to First Look Comments/Instructions (if any) appear below:

Sincerely,
Networks Editorial Office

[Borrar y Atrás](#) | [Borrar y Adelante](#)Mover a:



A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls

Journal:	<i>Networks</i>
Manuscript ID	Draft
Wiley - Manuscript type:	Research Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Belloso, Javier; Universidad Publica de Navarra, Estadistical and Operations Research Juan, Angel; Universitat Oberta de Catalunya, Computer Science Dept Faulin, Javier; Universidad Publica de Navarra Martinez, Enoc; Universitat Oberta de Catalunya
Keywords:	Vehicle Routing Problem, Vehicle Routing Problem with Backhauls, Biased Randomization of Heuristics, Metaheuristics, Combinatorial Optimization Problem, Horizontal cooperation
Abstract:	This paper analyzes the Vehicle Routing Problem with Backhauls, where delivery and pick-up customers are served from a central depot. Initially, we focus in the version with clustered backhauls (VRPCB), where all the customers in the delivery group of the same route have to be served before the first customer in the pick-up group can be visited. The paper presents a relatively simple-to-implement yet efficient metaheuristic algorithm that integrates a biased-randomized version of the popular savings heuristic within a metaheuristic framework. A skewed probability distribution is used to induce a biased (oriented) randomization effect on the savings list of routing edges, and a penalty cost is assigned to those edges connecting delivery customers with pick-up customers in order to promote a sequential order in delivery and pick-up activities. Some classical benchmark instances for the VRPCB are selected in order to compare our approach with other state-of-the-art algorithms, and a new best-know solution for a classical benchmark set is obtained. Finally, we analyze the flexibility of our approach by employing it, after a minor adaptation, in the Vehicle Routing Problem with Mixed Backhauls, where linehaul and backhaul customers might appear in any order during a route.

SCHOLARONE™
Manuscripts

A Biased-Randomized Metaheuristic for the Vehicle Routing Problem with Clustered and Mixed Backhauls

Javier Belloso¹, Angel A. Juan², Javier Faulin³, Enoc Martinez²

(1) Department of Mathematical Engineering and Computing. Public University of Navarre, 31006 Pamplona, SPAIN. e-mail: javier.belloso@unavarra.es

(2) Department of Computer Science, Multimedia, and Telecommunication. Open University of Catalonia – IN3, 08018 Barcelona, SPAIN. e-mail: {ajuanp, emartinezmasip}@uoc.edu

(3) Department of Statistics and Operations Research. Public University of Navarre, 31006 Pamplona, SPAIN. e-mail: javier.faulin@unavarra.es

This paper analyzes the Vehicle Routing Problem with Backhauls, where delivery and pick-up customers are served from a central depot. Initially, we focus in the version with clustered backhauls (VRPCB), where all the customers in the delivery group of the same route have to be served before the first customer in the pick-up group can be visited. The paper presents a relatively simple-to-implement yet efficient metaheuristic algorithm that integrates a biased-randomized version of the popular savings heuristic within a metaheuristic framework. A skewed probability distribution is used to induce a biased (oriented) randomization effect on the savings list of routing edges, and a penalty cost is assigned to those edges connecting delivery customers with pick-up customers in order to promote a sequential order in delivery and pick-up activities. In the proposed approach, the fleet size constraint is implicitly relaxed. This feature allows the perturbation process to explore unfeasible but promising regions in the solution space. Then, whenever the number of routes does not match the number of available vehicles, a ‘recursive corrective operator’ is employed to transform the new solution into a feasible one. Some classical benchmark instances for the VRPCB are selected in order to compare our approach with other state-of-the-art algorithms, and a new best-know solution for a classical benchmark set is obtained. Finally, we analyze the flexibility of our approach by employing it, after a minor adaptation, in the Vehicle Routing Problem with Mixed Backhauls, where linehaul and backhaul customers might appear in any order during a route.

Keywords: Vehicle Routing Problem with Backhauls; Biased Randomization of Heuristics; Metaheuristics.

1. INTRODUCTION

The distribution of raw materials, goods, and products is a core process that constitutes the backbone of most modern supply chains. It represents an important portion of a company total running cost and time, which has contributed to raise the interest of researchers in developing effective solutions for optimizing real-life logistics operations. Road transportation is the predominant way of distributing goods in many parts of the world. Direct costs associated with this type of transportation have experienced a significant increase since 1970, and more so during the decade 2005-2015. Furthermore, road transportation is intrinsically associated with indirect and variable costs, and the consequences are easily observable –congestion, contamination, security-related and safety-related costs, mobility, delay time costs, etc.– but are usually left unaccounted because it is not easy to quantify them (Sinha and Labi [37]). For example, traffic jams in metropolitan areas constitute a serious challenge to the competitiveness of European industry. According to some studies (Bastiaans et al. [2]), external costs generated by traffic congestion represented at the beginning of the last decade about 2% of the European Gross Domestic Product. This percentage has followed an increasing trend during the last years (Demir et al. [11]). In addition to these tangible costs, it is possible to consider others such as environmental costs due to the production and use of fossil fuel. Therefore, there is a need for developing efficient models and methods that support decision-making processes in the road transportation arena so that optimal (or near-optimal) strategies can be made. This necessity for optimizing road transportation, which affects both the public and private sectors, constitutes a major challenge for most industrialized regions around the world.

The Vehicle Routing Problem (VRP) is a well-known model to support decision making in the logistics and transportation sector (Laporte [27], Toth and Vigo [41]). The basic version of the VRP aims at generating the minimum-cost set of routes for a fleet of vehicles that have to satisfy the customers' requests. Vehicles may have the same capacity (homogeneous fleet) or not (heterogeneous fleet). Usually, the generated routes originate and finish at a central depot. Each customer must be visited by a single vehicle only once. Additionally, there is a cost matrix that measures the costs associated with moving a vehicle from one node (customer or depot) to another. These costs usually represent distances, traveling times, number of vehicles employed, or a combination of these factors. Based on this basic VRP, some researchers have proposed and examined a large number of variants that capture the special requirements of practical logistics processes (Caceres et al. [6]). The Vehicle Routing Problem with Backhauls (VRPB) considers both deliveries and pick-ups. The VRPB belongs to the General Pick-up and Delivery Problem (Savelsbergh and Sol [34]), and it takes advantage of the unused capacity of a vehicle on the trip back to the depot. Customers are divided into two clusters: linehaul customers –who receive goods from the depot–, and backhaul or pick-up customers –who ship goods to the depot. In this paper we focus on the clustered version of the problem (VRPCB), where all the customers in the delivery group of the same route have to be served before the first customer in the pick-up group can be visited.

The clustered variant of the VRPB is motivated by the fact that, in real-life practice, vehicles are often rear loaded. Hence, the on-board load rearrangement required by a mixed delivery and pick-up service is difficult –or even impossible– to carry out at customer locations. Moreover, even when some trucks are not always restricted to be rear loaded, in many real-life applications linehaul customers have a higher service priority than backhaul customers. Finally, the VRPCB is becoming increasingly popular due to the current trend towards recycling and products re-use. In order to solve the VRPCB, we propose a hybrid algorithm that combines a biased-randomized version of the popular savings heuristic (Juan et al. [23]) with a metaheuristic framework. Our algorithm employs a reduced number of parameters, is relatively simple to understand and implement in practice, and provides highly competitive results. A skewed probability distribution is used to induce a biased (oriented) randomization effect on the savings list of routing edges, and a penalty cost is assigned to those edges connecting delivery customers with pick-up customers in order to promote a sequential order in delivery and pick-up activities. In the proposed approach, the fleet size constraint is implicitly relaxed. This feature allows the perturbation process to explore unfeasible but promising regions in the solution space. Then, whenever the number of routes does not match the number of available vehicles, a ‘recursive corrective operator’ is employed to transform the new solution into a feasible one.

The remainder of the article is structured as follows: Section 2 provides a short description of the VRPCB. Section 3 reviews the relevant literature on the VRPB. Section 4 presents an overview of the proposed methodology, while Section 5 includes the low-level (pseudo-code) details. A series of numerical experiments that serve to both illustrate and validate our approach are discussed in Section 6. Section 7 compares our results against those provided by other state-of-the-art algorithms. Section 8 shows how our algorithm can be easily adapted to efficiently solve the Vehicle Routing Problem with Mixed Backhauls. Finally, Section 9 summarizes the main contributions and results of this work.

2. A SHORT DESCRIPTION OF THE VRP WITH CLUSTERED BACKHAULS

The VRPCB model can be defined on a complete graph $G = (V, A)$ where $V = \{v_0\} \cup L \cup B$ is the node set and A is the edge set. Thus, $L = \{v_1, v_2, \dots, v_l\}$ represents the linehaul customer set and $B = \{v_{l+1}, v_{l+2}, \dots, v_{l+b}\}$ denotes the backhaul customer set. Also, K vehicles with capacity Q are available at vertex v_0 that corresponds to the central depot. Linehaul customer $v_i \in L$ has an associated delivery demand quantity, d_i , that must be transported from the depot to the customer, while each backhaul customer $v_j \in B$ has an associated pick-up quantity, p_j , which must be shipped to the depot (Figure 1). Each arc $(v_i, v_j) \in A$ has an associated fixed nonnegative cost c_{ij} which is the cost involved for traveling from location v_i to v_j . We consider the symmetric case where $c_{ij} = c_{ji}$. The goal of the VRPCB is to design a set of routes such that: (i) the total cost of the generated routes set is minimized; (ii) the size of the generated route set is equal to K (the number of vehicles to use is given in advance and routes with fewer or more vehicles are not accepted); (iii) if a route contains both linehaul and backhaul customers, the backhaul customers can

be served only after all the linehaul customers have been visited; (iv) a route contains at least one linehaul customer; (v) every customer is visited once; and (vi) the vehicle capacity Q is not exceeded. For a detailed mathematical formulation of the VRPB model, the reader can review the works of Goetschalckx and Jacobs-Blecha [17] and Toth and Vigo [39].

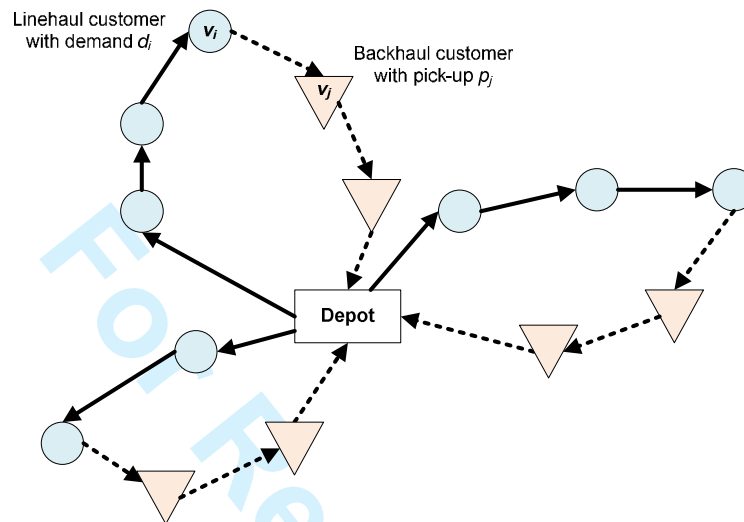


Fig. 1. A visual representation of the VRP with clustered backhauls

Our interest in the VRPB is motivated by its great practical and theoretical importance. From the practical perspective, VRPB is frequently encountered by large companies that must transport goods from their production sites to other inter-company sites (linehauls). At the same time, the production site must be supplied from vendors (backhauls) and/or other inter-company production sites located within the same geographic region (Goetschalckx and Jacobs-Blecha [17], Goetschalckx [19]). A common example can be found on linehaul customers that must return empty containers to their respective senders. Another example can be found in the grocery industry, where supermarkets and shops are the linehaul customers and grocery suppliers are the backhaul customers (Ubeda et al. [42]). In the retail industry, large companies own many outlets to be supplied from the depot, and at the same time the depots must be resupplied by the vendors located in the same region by visiting backhaul customers in distribution routes (Golden et al. [20]). It has been widely recognized that a significant saving in transportation costs can be achieved in this mixed distribution-collection context. Other applications can be considered too, e.g.: return of empty bottles, pallets, used batteries, etc.; delivery of new appliances accompanied by the pick-up of the old ones; recovery of defective or obsolete products, and many more. In addition, pro-environmental practices raise the necessity of bi-directional product flows modeled by the VRPB. Products are transported from the production site to the retailers, while at the same time used and expired products are collected from the retailers and sent back to the

production site in order to be recycled, disassembled, or appropriately processed before being disposed. From the theoretical perspective, the VRPB is a challenging optimization problem. It reduces to the basic VRP when only linehaul customers are considered. Thus, being a generalization of the basic VRP, the VRPB is also an NP-hard combinatorial optimization problem.

3. LITERATURE REVIEW ON THE VRPB

The VRPB has been addressed in the literature mainly as an extension of the basic VRP, so the main characteristics of the literature review to consider are strongly connected. The natural decomposition of the VRPB problem into two VRP sub-problems (one for delivery customers and a second one for pick-up customers) has been the main point of inspiration for different authors. Thus, a traditional approximation to solve the VRPB has consisted in the following steps: firstly, to solve the linehaul problem; secondly, to solve the backhaul problem; and thirdly, to combine both solutions in some effective way. Considering these three sub-problems, the first two ones are equivalent and have been extensively documented in the literature, whereas the third one is almost exclusive of the VRPB variant. Deif and Bodin [10] proposed one of the first heuristic algorithms for solving this problem. These authors suggested an extension of the Clarke and Wright [9] savings (CWS) heuristic that was conveniently adapted to the VRPCB case: the clustering and precedence constraints (backhaul customers are only served when all deliveries to linehaul customers have been completed), reduce the number of feasible solutions to be considered. In other words, the final solution can only contain one “interface” edge that shifts the route from delivery to pickup mode. Vigo [45] presented an adaptation of this procedure to the asymmetric case. Also, Golden et al. [20] proposed a heuristic procedure based on the insertion principle. Initially, the distribution routes are generated for linehaul customers and afterwards the backhaul customers are inserted into routes according to an insertion criterion. The precedence constraint is assured with the introduction of a penalty factor that makes the backhaul customers to be introduced at the end of the route. The selection of the interface edge is one of the key factors for the success in the insertion algorithms. Goetschalckx and Jacobs-Blecha [17] proposed an algorithm that uses a space-filling curve transformation with the purpose of mapping points from the Euclidean space into points along a unit line. This mapping preserves sequences and two different clusters are respectively created for linehaul and backhaul customers. The two sequences are managed separately to form feasible routes, which are joint in a later stage according to the Euclidean distance. Similarly, Goetschalckx and Jacobs-Blecha [18] proposed a cluster-first route-second algorithm, considering the cluster creation as a generalized assignment problem and the route creation as a Travelling Salesman Problem with the constraint that only one interface from delivery to pick-up customers is allowed. The best sequence of customers for the route is highly dependent on the link selected as interface. Anily [1] introduced another cluster-first and route-second algorithm. The clustering phase is accomplished by a modified circular partitioning heuristic, followed by the construction of traveling salesman tours through the different clusters (each cluster contains either

only linehaul or only backhaul customers). Then, linehaul clusters are assigned to backhaul clusters using the assignment algorithm proposed by Kuhn [26]. Finally, a route generation phase is in charge of determining the optimal connections between depot, linehaul, and backhaul clusters. Another algorithm for the VRPB that falls under the cluster-first, route-second category has been presented by Toth and Vigo [38]. Their algorithm is based on a K-tree Lagrangian relaxation presented for the VRP by Fischer [13]. Toth and Vigo [40] introduced the idea of symmetric and asymmetric costs between two customers and proposed another cluster-first route-second algorithm for solving both the symmetric and the asymmetric VRPB. In the symmetric version of the problem, the distance between each pair of customers is the same in both directions, whereas in the asymmetric version this assumption does not hold. In Toth and Vigo [39] the authors make use of the information included in infeasible solutions associated with a lower-bound obtained with the Lagrangian relaxation.

According to Toth and Vigo [41] the VRPB model is included within the General Pick-up and Delivery Problem (GPDP). The GPDP is divided into two problem classes. The first class, the VRPB, refers to scenarios where goods are transported from one or several depots to the customers and from the customers to the depot(s). The second class refers to problems where goods are transported between pick-up and delivery customers. This problem is known as VRP with Pick-ups and Deliveries (VRPPD). The VRPB can be divided into four subclasses. In the first two subclasses, customers are either delivery or pick-up customers. The first problem is the VRPCB. The second one does not respect any order in the visit of the clusters and it is known as VRP with Mixed Backhauls (VRPMB). In the third VRPB subclass, customers can require both pick-up and delivery operations, and each customer can be visited more than once. This is known as VRP with Divisible Delivery and Pick-up (VRDPD). Cases where each customer can only be visited once are referred to as VRP with Simultaneous Delivery and Pick-up (VRPSDP), representing the fourth subclass of VRPB. A detailed explanation of the classification of the GPDP can be found in Parragh et al. [31], which provides a meaningful table of General Pickup and Delivery Problems classification. These authors show that the most general problem following this codification is the VRPMB, for which many of the solving methods are based on the insertion concept.

Halse [21] proposed a heuristic with the sequence cluster-first and route-second. Salhi and Nagy [35] solved the VRPMB by allocating first linehauls and, afterwards, backhauls inserted in clusters one by one. For the same problem, Wade and Salhi [46] proposed an Ant System to find the best route. Furthermore, Dethloff [12] developed an insertion heuristic based on the concept of residual capacity of the vehicle that was initially introduced for the VRPSDP. The insertion criterion is based not only on the distance between customers, but also on the measure of the residual capacity of the route. Another heuristic for the same problem is presented by Nagy and Salhi [29]. In addition, Wade and Salhi [47] introduced a new variant that allows to determine the position along a route from which the first backhaul customer may be visited. This is done according to: the user's experience, the vehicle capacity, the type of products, and the type of vehicle used. All these methods, even not specifically thought for the

VRPB, can be easily adapted to the precedence constraint. A similar comment is applicable to the improvement/exchange heuristics, with the only restriction that linehaul and backhaul customers should not be mixed in the process. This is the case of 2-opt and 3-opt procedures (Lin [28]), which can be easily applied separately to the sets of linehauls and backhauls within each route. The same is true for the 4-opt procedure designed by Boctor and Renaud [4]. The heuristics belonging to the cluster-first and route-second type, like the sweep algorithm from Gillett and Miller [16], can also be included into this group. The extended sweep algorithm can be easily applied to the VRPB by truncating clusters if linehaul or backhaul is exceeded.

Literature since 2000 is mainly devoted to metaheuristic methods. This is the case of Osman and Wassam [30], who were the first one in developing a Tabu Search (TS) algorithm for the VRPB. The authors proposed two route construction methods which are based on savings, insertion, and assignment approaches. The results are then used with their Reactive TS algorithm. Similar approximations were performed by Brandao [5], with his TS algorithm, and Wassan [48], who developed a TS enhanced by adaptive memory programming. Ropke and Pisinger [33], unified VRPB problems as extension of what they called the Rich Pick and Delivery Problem with Time Windows (PDPTW). They transformed all the problems by considering the time restriction and proposed an algorithm based on a local search heuristic. Wassan et al. [49] and Gajpal and Abad [14] used an Ant Colony System to solve the problem. A comparison of backhauling strategies using ants can be found in Reimann and Ulrich [32]. A different approach is the one presented by Yazgi Tütüncü et al. [51], who described a decision support system based on a Greedy Randomized Adaptive Memory Programming Search. The VRPB with clustered backhauls and the VRPMB were solved using an interactive environment. García-Najera [15] generalized the problem as a multi-objective problem and solved it with an adapted version of a previously proposed Evolutionary Algorithm. Zachariadis and Kiranoudis [52] solved the problem using a metaheuristic approach. These authors explore the neighborhoods by exchanging variable-length customer sequences. The initial set of feasible VRPB routes is obtained through a savings heuristic that stochastically adjusts the Euclidean savings.

Vidal et al. [44] introduces a solution for multi-attribute VRPs. Attributes are characteristics or constraints that describe the specifications of real-life VRPs. They propose a general-purpose algorithm called Unified Hybrid Genetic Search (UHGS), which uses adaptive assignment, sequencing, and route evaluation operators to solve 29 variants of VRPs. For the tested cases, UHGS matches or outperforms the current best-known solutions. A very comprehensive review of the main classes of attributes and several remarkable metaheuristics can be found in Vidal et al. [43]. Cuervo et al. [8] proposed an Iterated Local Search (ILS) algorithm where the main component is an oscillating search heuristic that performs constant transitions between feasible and infeasible solutions. One of the main characteristic of their approach is the statistical analysis of the algorithm components, which allows them to obtain very competitive results in terms of solution quality and computing times. Wassan et al. [50] introduced a new variant called Multiple Trip Vehicle Routing Problem with Backhauls and solve large instances using a two-level

Variable Neighborhood Search algorithm. From a practical point of view, the VRPB is also considered into the field of reverse logistics and green logistics. Significant cost reduction can be achieved by combining linehaul with backhaul customers, i.e., having less empty vehicles returning to the depot and causing, as an indirect effect, the reduction of gas emissions related to the normal use of the fleet (Berbeglia et al. [3], Parragh et al. [31], Sheu and W.K. Talley [36], Ubeda et al. [42]).

4. OVERVIEW OF OUR SOLUTION APPROACH

In this section, we provide an overview of the proposed algorithm and how we combine different heuristics and local search processes from the VRP literature. The high-level flowchart of our algorithm is shown in Figure 2. Firstly, an initial base solution is obtained through the CWS heuristic (Clarke and Wright [9]). As proposed in this heuristic, the savings list is sorted using the savings value associated with each edge. This saving calculation is adapted to backhauls using the approach introduced by Deif and Bodin [10]. The idea behind this adaptation is to delay the eligibility of the interface links, so that both the linehaul and backhaul routes are “mature” enough (have conveniently evolved in number of customers) to be combined. In order to achieve this goal, the interface links are penalized as described next. The parameter p is a penalty coefficient ranging between 0 and 1, and S is an estimation of the maximum saving value that can be attained. According to Deif and Bodin [10], the best results were obtained with p in a range from 0.05 to 0.20. Before the introduction of the interface edges, the problem can be divided into two sub-problems: the linehaul route creation and the backhaul route creation. Following the main ideas behind the CWS heuristic, once the initial ‘dummy’ solution has been created, the savings are computed and the merging process starts. During this merging process, routes with backhaul customers are added at the end of routes with linehaul customers throughout the use of interface edges (Figure 3).

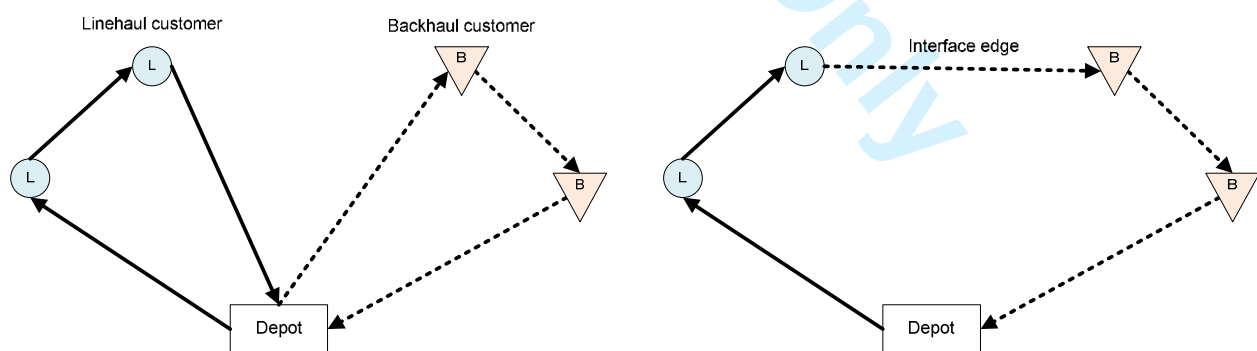


Fig. 3. Using an interface edge to connect a linehaul route with a backhaul route

The order of the list is also altered by applying a biased (oriented) randomization. As discussed in Juan et al. [24] and Juan et al. [25], by performing a biased randomization of the savings list, edges are selected in a different order at each iteration of the multi-start process. In our case, a geometric distribution is employed to provide the biased-randomized behavior. The distribution process uses only one parameter, β ($0 < \beta < 1$). While the CWS heuristic always chooses the edge with the largest savings value, the presented approach, instead, assigns a selection probability to each edge in the savings list. Moreover, this probability should be coherent with the savings value associated with each edge, i.e., edges with larger savings will be more likely to be selected from the list than those having smaller savings.

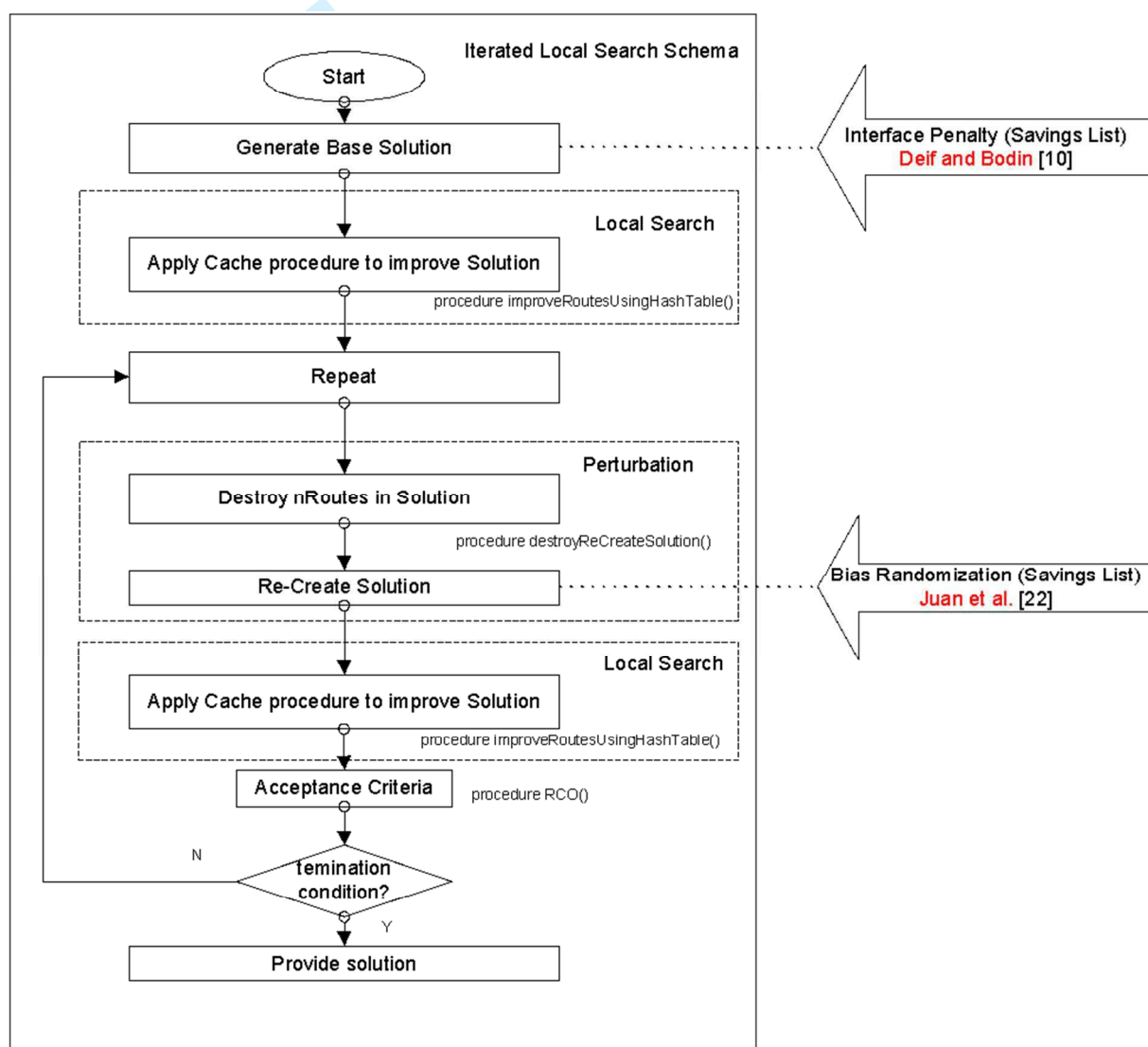


Fig. 2. Framework of our metaheuristic approach

Following the proposed framework, once the base solution is known, a cache or memory-based procedure is initiated. For each solution, all its routes are checked to determine if there is a better known way (one obtained in a previous iteration and stored in memory) to sort the associated nodes. If this is the case, the solution is updated. After this quick improvement, the iteration is started until a finishing condition is met. Then, a perturbation is applied to the base solution in order to generate a new one. This perturbation process is basically composed of two steps. The first one consists in ‘destroying’ some selected routes in the solution, while the second one will ‘re-construct’ the solution by applying the heuristic to the nodes that were located in the destroyed routes. This procedure is similar to the one proposed in Juan et al. [24]. Once the perturbation is completed, the cache procedure is applied again. The solution obtained is then checked to determine if the acceptance criterion is simultaneously accomplished by the following two conditions: (i) the cost of the solution improves the best solution found so far; and (ii) the number of routes in the solution corresponds to the fleet size.

In our approach, the fleet size constraint is implicitly relaxed during the searching process. Therefore, the aforementioned condition needs to be checked because the solution can require a different number of routes than the available number of vehicles to serve all the customers, Toth and Vigo [40], Wassan [48]. We take advantage of this characteristic and allow the searching process to visit new regions of the solution space and reach infeasible solutions. If this is the case a corrective operator (RCO) with the main goal of restoring the fleet size constraints is triggered. The pseudocode of this operator is included and explained in the next section. The idea of considering infeasible solutions in the searching space is not new and has been considered by different authors already, such as Toth and Vigo [41] and Cuervo et al. [8]. As explained by Vidal et al. [44], one of the parameters that influences the performance of any metaheuristic is related to its capability of evaluating the created routes during the searching process. We have used this idea to develop our own operator. Thus, once the CWS heuristic creates a solution, the RCO evaluates each component route depending on its demand. The evaluation of the routes will conclude that some routes are valid, and some are invalid. For the first ones, they will be promoted to be part of the final solution, while the others will be destroyed and the correspondent nodes will be marked as pending to be routed. This can be seen as a perturbation process as well. This new perturbation, instead, divides the formed routes in valid or invalid depending on the total demand they are serving. The invalid ones are destroyed and re-built, whereas the valid ones are kept.

5. ADDITIONAL DETAILS OF OUR APPROACH

This section presents a more detailed description of the procedures that implement the proposed framework, including their pseudo-codes. Figure 4 shows the pseudo-code associated with the biased-randomized metaheuristic framework, which is the main component of our algorithm. The procedure requires only three parameters:

- (i) the penalty coefficient for the interface links (λ)
- (ii) the one defining the geometric distribution used during the biased-randomization process (β)
- (iii) the percentage of routes to be destructed during the perturbation stage (p)

The input variables contain the information related to the specific instance: demands, vehicle capacity, number of vehicles, Euclidean coordinates for all customers and the depot, number of linehaul customers, and seed for the pseudo-random number generator. Once it is loaded within the program, the CWS heuristic is invoked (line 02) using the savings list that is obtained after considering the penalization factor (line 01). At this point, we start the iterative process to generate new solutions outperforming the CWS (line 04). At each iteration, a new solution is constructed by using again the CWS heuristic (line 05) but now the savings list has been randomized. This randomization allows selecting the edges in a different order each time the procedure is invoked. The construction of the savings list and the mechanism for selecting the edges preserve the logic of the savings heuristic. The biased order of the edges is induced by a Geometric distribution that uses one single parameter β . As explained in Section 4, we have set this parameter as uniformly random in the interval (0.06, 0.23). After generating a solution, a cache (memory-based) procedure which stores the best routing patterns from previous iterations is employed (line 06). This can be seen as a learning mechanism that is implemented through a classical ‘hash map’ data structure. This data structure, sometimes called ‘dictionary’, is common in all modern programming languages and it generalizes the concept of dynamic array by using ‘keys’ instead of number-style indexes to refer to stored values or objects. In practice, this allows for quick search of any value / object throughout its key (usually a hash code). The cache always keeps the best-known order to visit a set of customers in a route. The routing component of the solution can be improved further by using the “destroy and re-create” procedure (line 07). This procedure tries to improve the results by first considering different subsets of routes (i.e., by reducing the problem dimension) and then applying an iterative process in these subsets. This second-level iterative process makes use again of both the randomized CWS and the cache procedures.

```

procedure BR(inputs,  $\lambda$ ,  $\beta$ ,  $p$ )
01 savingsList <- CreatePenalizedSavingsList(inputs,  $\lambda$ )
02 cwsSol <- CWS(savingsList, inputs)
03 while {terminationCondition is not met} do
04   savingsList <- CreateRandomPenalizedSavingsList(inputs,  $\lambda$ ,  $\beta$ ) % biased randomization
05   newSol <- CWS(savingsList, inputs)
06   newSol <- improveRoutesUsingHashTable(newSol) % cache
07   newSol <- destroyReCreateSolution(inputs, newSol,  $p$ ) % perturbation stage
08   ourBestSol <- acceptanceCriteria(ourBestSol, newSol, cwsSol)
09 end while

```

```

10 return ourBestSol
end

```

Fig.4. Main procedure of our algorithm

Figure 5 shows the pseudo-code for the creation of the randomized saving list with an interface link penalty. Following Deif and Bodin [10] firstly, the maximum saving value (*maxsaving*) is computed. Afterwards, this value is multiplied by the penalty parameter λ . The original saving value for the interface link decreases linearly by this amount. As explained in Section 4 we have set this parameter as uniformly random in the interval (0.05, 0.20). Once the savings list is obtained, the bias randomization is applied (line 08) and the procedure is completed.

```

procedure CreateRandomPenalizedSavingsList(inputs,  $\lambda$ ,  $\beta$ )
01 savingsList <- createSavingsList % list of savings following CWS heuristic
02 maxSaving <- obtainMaxSaving(savingsList)
03 for each edge in savingsList do
04   if {edge is interface} then % edge connects one linehaul node with backhaul node
05     currentSaving(edge) <- currentSaving(edge) % maxSaving
06   end if
07 end for
08 savingsList <- OrderDesc(savingsList) % Sort savings list
09 for each edge in savingsList do % Randomize savings list using the seed and @
10   index <- obtainBiasedRandom(inputs, seed,  $\beta$ , #savingsList)
11   randomPenalizedSavingsList <- addEdge(removeEdge(savingsList, edge), index)
12 end for
13 return randomPenalizedSavingsList
end

```

Fig.5. Randomized saving list creation with interface penalty

Figure 6 shows the pseudo-code of the ‘recursive corrective operator’ (RCO) that is used for transforming infeasible solutions into feasible ones. This operator is triggered whenever the number of routes in a solution does not coincide with the number of available vehicles. The algorithm explained in the previous pseudo-codes is based on the CWS heuristic, which implies that the number of routes in the final solution will only depend on the routes merging process. Therefore, the fleet size constraint is not considered in the searching process. As shown in Figure 6, initially the operator calculates the routes in the current infeasible solution that can be part of a feasible solution. This selection is done by the select routes procedure. In the case that the solution contains a number of routes inferior to the available number of vehicles, we first sort the routes in a descending order of demand. The condition for any route to become part of the final solution is that the aggregated demand assigned so far is less than or equal to the total demand that can be served by the available vehicles. If this is the case, the route is added to the final solution. If the solution provided by the algorithm has more routes than available vehicles, the routes are sorted in ascending order of demand. Now, the condition for a route to be considered as part of the final solution is that the remaining

demand (of all the routes except the current one) can be served by the remaining vehicles. Once this selection process is completed, we have a partial solution composed of the feasible routes. For all the remaining routes, their nodes are marked as pending to be routed and the algorithm is then invoked again in order to generate a new solution. If the solution is feasible in terms of fleet size, the solution is returned and the process is finished. Otherwise, the procedure is called recursively.

```

procedure RCO(solution, numVehicles,  $\lambda$ ,  $\beta$ )
01 parcSolution <- SelectRoutes(solution, NumVehicles)
02 pendingNodes <- ExtractNodes(solution, parcSolution)
03 numVehicles <- numVehicles - #Routes(parcSolution)
04 savingsList <- CreateRandomPenalizedSavingsList(inputs(pendingNodes),  $\lambda$ ,  $\beta$ )
05 newSol <- CWS(savingsList, inputs(pendingNodes))
06 if (#Routes(newSol) = numVehicles) then
07   return (newSol)
08 else
09 return (parcSolution & RCO(newSol, numVehicles))
end

```

Fig. 6. Recursive Corrective Operator

6. EXPERIMENTAL RESULTS

The algorithm described in this paper has been implemented as a Java application. A standard desktop with an Intel Intel Core i7 CPU at 2.79 GHz and 3.42 GB RAM was used to run all the experiments. The compiled program was run under Netbeans IDE over MS Windows 7. A loop-intensive algorithm implemented in Java might execute somewhat slower than the same algorithm implemented in optimized C/C++. According to some studies (Hundt [22]), the Java code could be about 6 times slower in loop-intensive algorithms. However, Java offers other advantages for algorithm development, such as rapid implementation and higher reproducibility of results (the performance of a code implemented in C/C++ is usually more dependent on the programmer skills as well as on the specific configuration of the compiled version to be run).

We have solved the GJB set of 62 instances generated and introduced by Goetschalekx and Jacobs-Blecha [17] and the TV set of 33 instances introduced by Toth and Vigo [40] in order to test the effectiveness of the proposed method. For the instances in the first set, the depot is located at coordinates (12000; 16000), while the customer coordinates are randomly generated, using a uniform probability distribution, in the rectangle [0, 24000] x [0, 32000]. Customers' demands are randomly generated from a normal distribution with mean 500 and standard deviation 200. Fourteen values for the total number of customers (between 25 and 150) are considered, being the percentage of linehaul customers equal to 50%, 66%, and 80% of the total customers. For each of these values, the vehicle capacity is assigned such in a way that between 3 and 12 vehicles are able to serve all the demands. The set of benchmarks is available in the following website: <http://www2.isye.gatech.edu/~mgoetsch/lineback.html>.

Regarding the instances in the second set, the original source is the VRP data initially proposed by Christofides and Eilon [7]. Following the conventions used by the authors, three new instances are generated using a linehaul/backhaul (LH/BH) mix of 50/50, 67/33, and 80/20, respectively (i.e., the first customer every two, three, or five customers is considered to be a backhaul customer). The distances between each pair of customers have been calculated using double precision notation. Additionally, for the TV set, each distance has been rounded to the nearest integer. This has been a convention used by most of the algorithms for VRPB in the literature and, in particular, by Cuervo et al. [8].

Next, we will discuss the values that have been assigned to each of the parameters in our algorithm. In the case of the β parameter, which defines the specific geometric distribution to use during the biased-randomization process, we have set this parameter as uniformly random in the interval (0.06, 0.23), as suggested by Juan et al. [22]. Regarding the parameter λ , we have set this parameter as uniformly random in the interval (0.05, 0.20), as suggested by Deif and Bodin [10]. In the case of the percentage of destruction during the perturbation stage, p , we have set this parameter as uniformly random in the interval (0.10, 0.50). Notice that we have just selected very general values for these parameters, without the need to perform complex and time-expensive fine tuning processes. Finally, the termination time has been fixed to 25 seconds for instances up to 50 nodes, to 75 seconds for instances up to 100 nodes, and to 300 seconds for instances with over 100 nodes.

Table 1 shows the results obtained for the GJB instances, while Table 2 shows the results obtained for the TV instances. The basic information contained in each table is: name of the instance, number of linehaul customers, number of backhaul customers, vehicle capacity (Q), number of routes in the solution (K), the cost of our best and average solution after 10 runs (Best10 and Avg10), and the gap (in percentage) between our best solution and the best-known solution in the literature (BKS). Also, in order to illustrate the contribution of the biased randomization process to the quality of the obtained solutions, we have included two columns in Tables 1 and 2. The first column contains the gap between our solution and the best-known solution obtained without using biased randomization. In other words, in this case the savings list is ordered by the savings value and following a strict diminishing criterion. The second column represents the corresponding gap when the biased randomization process described in this paper is used. Notice that when this process is not employed, then: (i) the average gap with respect to our best-known solution is over 6%; and (ii) 24 benchmarks are not properly solved (in terms of the number of required vehicles), 20 in the GJB set and 4 in the TV set.

Table 1. Results for the GJB instances

Instance	n-lh	n-bh	Q	K	NoBR	Gap NoBR - Best10	Best10		Avg10		Gap Best10 - BKS
							cost	time (s)	cost	time (s)	
A1	20	5	1,550	8	230,730	0.37%	229,886	1	229,886	1	0.00%
A2	20	5	2,550	5	185,228	2.76%	180,119	1	180,119	1	0.00%
A3	20	5	4,050	4	-	-	163,405	3	163,405	3	0.00%
A4	20	5	4,050	3	169,723	8.21%	155,796	4	155,796	4	0.00%
B1	20	10	1,600	7	257,426	7.13%	239,080	1	239,080	1	0.00%

Networks

B2	20	10	2,600	5	211,496	6.36%	198,048	2	198,048	2	0.00%
B3	20	10	4,000	3	178,343	5.03%	169,372	2	169,372	2	0.00%
C1	20	20	1,800	7	266,187	5.87%	250,557	2	250,557	2	0.00%
C2	20	20	2,600	5	216,996	0.91%	215,020	2	215,020	2	0.00%
C3	20	20	4,150	5	-	-	199,346	3	199,346	3	0.00%
C4	20	20	4,150	4	203,987	4.23%	195,367	1	195,367	1	0.00%
D1	30	8	1,700	12	-	-	322,530	1	322,530	1	0.00%
D2	30	8	1,700	11	336,598	5.91%	316,709	1	316,709	1	0.00%
D3	30	8	2,750	7	243,846	1.79%	239,479	4	239,479	4	0.00%
D4	30	8	4,075	5	208,685	1.37%	205,832	2	205,832	2	0.00%
E1	30	15	2,650	7	244,487	2.29%	238,880	2	238,880	2	0.00%
E2	30	15	4,300	4	218,597	2.90%	212,263	1	212,263	1	0.00%
E3	30	15	5,225	4	216,307	4.46%	206,659	1	206,659	1	0.00%
F1	30	30	3,000	6	292,802	10.12%	263,174	3	263,274	3	0.00%
F2	30	30	3,000	7	-	-	265,214	4	265,655	4	0.00%
F3	30	30	4,400	5	258,238	6.63%	241,121	5	241,121	5	0.00%
F4	30	30	5,500	4	245,621	4.79%	233,862	1	234,606	1	0.00%
G1	45	12	2,700	10	-	-	306,305	9	306,421	16	0.00%
G2	45	12	4,300	6	250,104	1.86%	245,441	1	245,441	1	0.00%
G3	45	12	5,300	5	241,383	4.92%	229,507	1	230,736	5	0.00%
G4	45	12	6,400	6	-	-	232,521	7	232,642	16	0.00%
G5	45	12	6,400	5	-	-	221,730	1	221,730	4	0.00%
G6	45	12	8,000	4	233,665	8.65%	213,457	17	215,241	6	0.00%
H1	45	23	4,000	6	285,618	5.84%	268,933	2	268,933	12	0.00%
H2	45	23	5,100	5	271,493	6.68%	253,365	5	253,462	27	0.00%
H3	45	23	6,100	4	269,395	8.15%	247,449	6	247,593	25	0.00%
H4	45	23	6,100	5	-	-	250,221	3	251,984	9	0.00%
H5	45	23	7,100	4	263,617	6.64%	246,121	9	246,169	24	0.00%
H6	45	23	7,100	5	-	-	249,135	4	250,178	19	0.00%
I1	45	45	3,000	10	-	-	350,245	17	351,245	14	0.00%
I2	45	45	4,000	7	320,635	3.33%	309,944	14	310,468	45	0.00%
I3	45	45	5,700	5	-	-	294,507	23	296,785	42	0.00%
I4	45	45	5,700	6	309,547	4.38%	295,988	2	296,920	13	0.00%
I5	45	45	5,700	7	-	-	301,236	39	302,649	19	0.00%
J1	75	19	4,400	10	345,697	3.09%	335,007	39	336,009	30	0.00%
J2	75	19	5,600	8	326,193	4.84%	310,417	47	310,871	23	0.00%
J3	75	19	8,200	6	294,206	5.09%	279,219	62	282,613	53	0.00%
J4	75	19	6,600	7	312,163	4.87%	296,947	32	297,088	40	0.14%
K1	75	38	4,100	10	423,361	6.92%	394,071	61	395,041	63	0.00%
K2	75	38	5,200	8	396,466	8.66%	362,130	105	364,830	99	0.00%
K3	75	38	5,200	9	-	-	365,694	112	372,452	90	0.00%
K4	75	38	6,200	7	378,999	7.93%	348,949	38	351,035	64	0.00%
L1	75	75	4,400	10	-	-	418,309	143	422,131	57	0.10%
L2	75	75	5,000	8	443,219	9.26%	402,155	85	408,817	176	0.23%
L3	75	75	5,000	9	-	-	405,083	89	407,730	128	0.60%
L4	75	75	6,000	7	412,740	6.81%	384,636	173	387,857	188	0.00%
L5	75	75	6,000	8	-	-	387,565	96	391,740	67	0.00%
M1	100	25	5,200	11	-	-	399,628	122	405,337	40	0.26%
M2	100	25	5,200	10	436,978	9.06%	397,407	91	401,905	128	0.12%
M3	100	25	6,200	9	409,948	8.24%	375,864	119	378,430	24	0.04%
M4	100	25	8,000	7	387,304	10.11%	348,140	71	351,850	100	0.00%
N1	100	50	5,700	11	-	-	408,101	111	416,326	264	0.00%
N2	100	50	5,700	10	465,112	12.27%	408,065	166	414,607	167	0.00%
N3	100	50	6,600	9	437,896	9.95%	394,338	109	402,158	120	0.00%
N4	100	50	6,600	10	-	-	394,788	331	402,362	267	0.00%
N5	100	50	8,500	7	411,866	8.75%	375,835	261	378,328	150	0.63%
N6	100	50	8,500	8	-	-	375,803	174	382,981	156	0.55%
Average	53	28	4,847	7		5,89%	290,742	46	292,486	46	0.04%

Table 2. Comparative results for TV instances

Instance	n-lh	n-bh	Q	K	NoBR	Gap NoBR - Best10	Best10		Avg10		Gap Best10 - BKS
							cost	time (s)	cost	time (s)	
eil22_50	11	10	6,000	3	378	1.85%	371	1	371	1	0.00%
eil22_66	14	7	6,000	3	413	11.38%	366	3	366	3	0.00%
eil22_80	17	4	6,000	3	383	2.09%	375	2	375	2	0.00%
eil23_50	11	11	4,500	2	728	6.32%	682	5	682	5	0.00%
eil23_66	15	7	4,500	2	704	7.81%	649	1	649	1	0.00%
eil23_80	18	4	4,500	2	670	7.01%	623	1	623	1	0.00%
eil30_50	15	14	4,500	2	539	7.05%	501	1	501	1	0.00%
eil30_66	20	9	4,500	3	566	5.12%	537	2	539	2	0.00%
eil30_80	24	5	4,500	3	557	7.72%	514	2	514	2	0.00%
eil33_50	16	16	8,000	3	776	4.90%	738	3	739	3	0.00%
eil33_66	22	10	8,000	3	811	7.52%	750	2	750	2	0.00%
eil33_80	26	6	8,000	3	742	0.81%	736	4	736	4	0.00%
eil51_50	25	25	160	3	591	5.41%	559	9	561	9	0.00%
eil51_66	34	16	160	4	576	4.86%	548	5	548	5	0.00%
eil51_80	40	10	160	4	579	2.42%	565	4	565	4	0.00%
eilA76_50	37	38	140	6	791	6.57%	739	2	740	19	0.00%
eilA76_66	50	25	140	7	800	4.00%	768	16	768	35	0.00%
eilA76_80	60	15	140	8	862	9.40%	781	80	794	41	0.00%
eilB76_50	37	38	100	8	875	8.46%	801	4	802	21	0.00%
eilB76_66	50	25	100	10	935	6.63%	873	13	875	14	0.00%
eilB76_80	60	15	100	12	967	4.96%	919	20	930	11	0.00%
eilC76_50	37	38	180	5	751	5.06%	713	6	715	13	0.00%
eilC76_66	50	25	180	6	777	5.53%	734	1	736	8	0.00%
eilC76_80	60	15	180	7	770	4.81%	733	14	738	17	0.00%
eilD76_50	37	38	220	4	736	6.25%	690	6	690	31	0.00%
eilD76_66	50	25	220	5	762	6.17%	715	2	717	10	0.00%
eilD76_80	60	15	220	6	722	3.88%	694	6	702	19	0.00%
eilA101_50	50	50	200	4	-	-	831	46	846	85	0.00%
eilA101_66	67	33	200	6	911	7.14%	846	44	860	89	0.00%
eilA101_80	80	20	200	6	-	-	856	41	872	85	0.00%
eilB101_50	50	50	112	7	-	-	923	23	935	34	0.00%
eilB101_66	67	33	112	9	-	-	982	27	996	30	-0.10%
eilB101_80	80	20	112	11	1069	5.71%	1008	99	1022	65	0.00%
<i>Average</i>	39	20	103	5		5.75%	700.6	15	704.8	20	0.00%

It should be noted that the F1, F2, F3, F4, and L1 instances have been solved using two different configurations in the literature. In our case, we have considered for the instances in group F that the customer at position (5103, 11065) has a demand of 101, while the vehicle capacity in L1 is equal to 4000.

7. DISCUSSION AND COMPARISON WITH OTHER APPROACHES

Our algorithm has found a new best solution for instance EILA101_66. Also, it is able to obtain very competitive solutions in very short computing times. Thus, for the GJB instances the average gap and computing times are 0.04% and 46 seconds, respectively. Even better results have been also obtained for the TV instances, with an average gap of 0.00% in 15 seconds. At this point, it is worthy to remember that the best-known solutions are provided by different algorithms in the literature, which means that our approach seems to be quite competitive when compared

against each individual algorithm. To confirm this statement, we have directly compared the performance of our algorithm against the performance of the following state-of-the-art metaheuristics: the tabu search (BTS) from Brandão [5], the large neighborhood search (LNS) from Ropke and Pisinger [33], the reactive-adaptive memory programming search (RTS-AMP) from Wassan [48], the multi-ant colony system (MACS) from Gajpal and Abad [14], the route promise algorithm (RPA) from Zachariadis and Kiranoudis [51], the multi-attribute general purpose algorithm (UHGS) from Vidal et al. [44], the iterated local search (ILS-1000) from Cuervo et al. [8] and the Two-Level VNS algorithm from Wassan et al [50]. For the GJB instances, Figure 7 shows a multiple box-plot graph comparing the gaps provided by each of the considered metaheuristics with respect to the BKS (as usual in these type of graphs, the central boxes represent the values between the first and third quartiles of the data distribution, while outliers are marked with a star symbol). The BR column contains the results associated with our algorithm. Likewise, Figure 8 shows a similar comparison among different metaheuristics for the TV instances. Notice that all these metaheuristics are providing quite similar results and all of them are very close to the values of the best-known solutions. It seems that the RTS-AMP, BTS, and LNS algorithms are performing slightly worse than the rest for the GJB set, while the ones providing the best results for this set of benchmarks are the RPA, ILS, and UHGS. In the case of the TV set, however, our BR algorithm is the one providing the best results, together with the ILS and the Two-Level VNS.

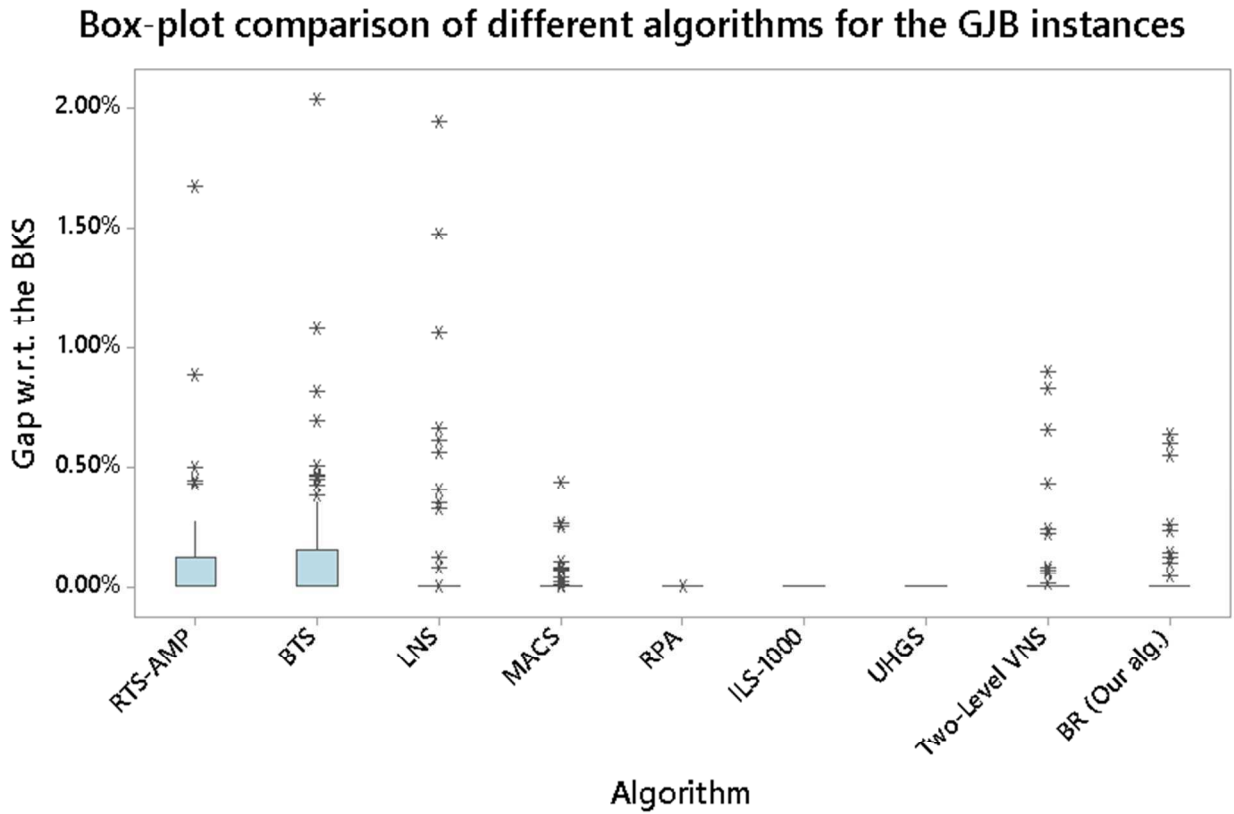


Fig. 7. Solution-quality performance comparison for the GJB set

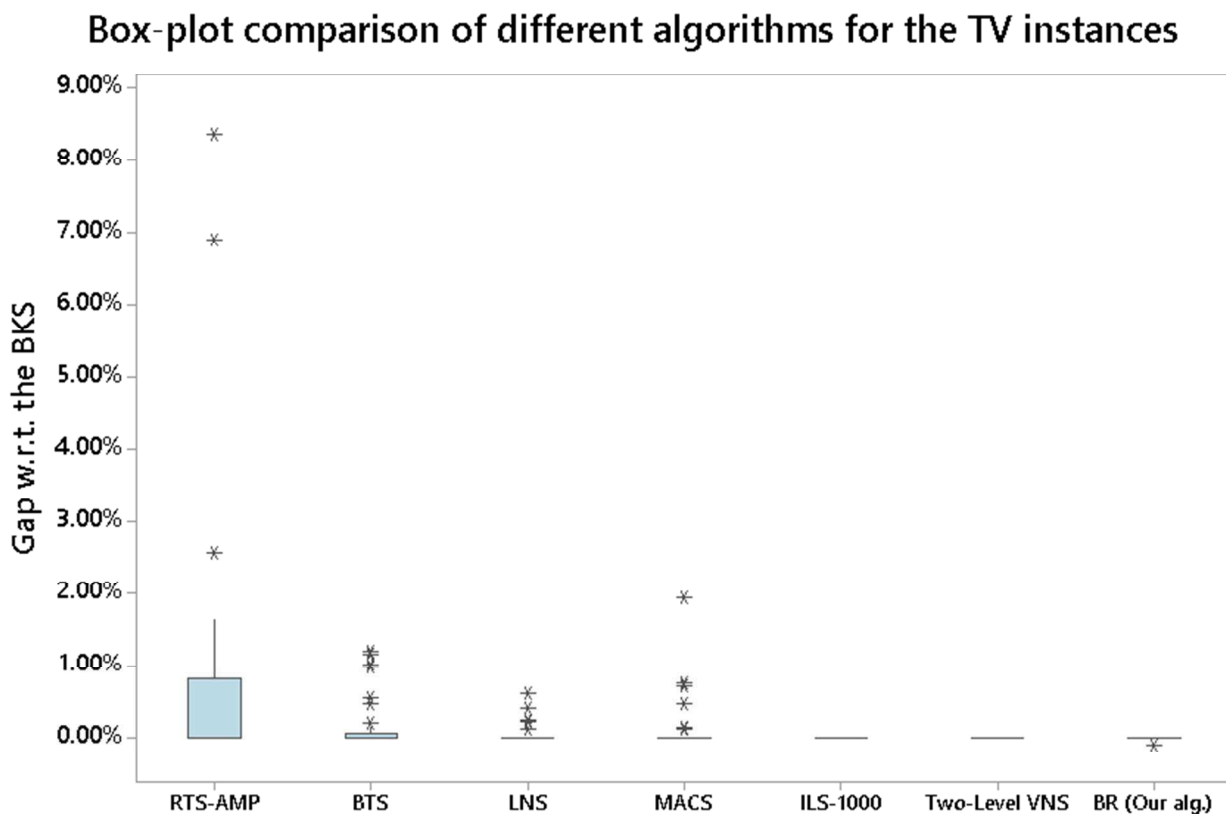


Fig. 8 Solution-quality performance comparison for the TV set

In order to make a fair comparison of the computational times employed by each algorithm, there are several objective factors that should be considered, e.g.: processor (CPU), memory, operating system, programming language, etc. The only information that has been reported by all the authors in the compared papers is the processor name and the programming language. With this information, a relatively fair performance comparison can be done using the benchmarks system proposed in <https://www.cpubenchmark.net/>. One of the goals of this application is to compare the performance of one computer with others using a different processor. The aforementioned web assigns an average mark to each CPU class that is calculated by the reports of thousands of performance tests.

Thus, we executed the benchmarks through our computer and the maximum CPU mark obtained was 2703. Whenever available, for the rest of computers we used the CPU mark reported in the aforementioned website. For those computers not included in the web, we employed the same correction factors as in Cuervo et al. [8]. Finally, following the conclusions of Hundt [22] about the speed of Java vs. that of C/C++, we applied a (quite conservative) correction factor of 0.5 to the times obtained with our Java code. Figure 9 summarizes the results we obtained.

Notice that the computing times shown by our BR algorithm are quite competitive. At this point, it is important to highlight that one of the main strengths of our algorithm is its relative simplicity, including the fact that only three parameters are needed in order to obtain competitive results in very low computing times.

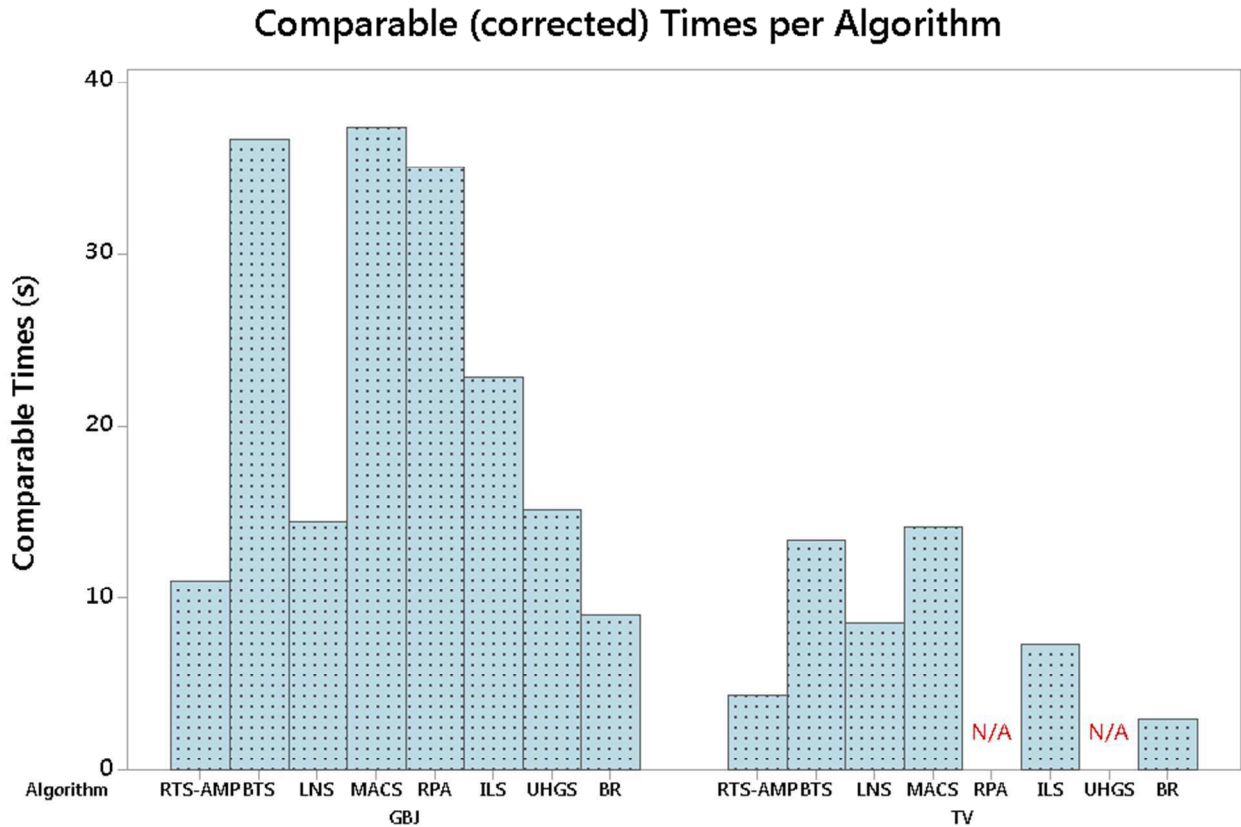


Fig. 9 Time performance comparison for the different algorithms

8. EXTENDING OUR APPROACH FOR THE VRPMB

In order to show that the presented framework is quite flexible and can be easily adapted to solve other variants of the VRP with backhauls, we have adapted our algorithm so it can deal with the Vehicle Routing Problem with Mixed Backhauls (VRPMB). The VRPMB is derived from the VRPB by relaxing constraints (ii), (iii), and (iv) described in Section 2. Thus, in the VRPMB the backhaul customers can be visited at any time (i.e., it is not necessary to deliver first to all the linehaul customers as in the VRPCB), while the fleet size is not limited. A single route in a VRPMB solution can be composed of just pick-up customers, just delivery customers, or both. The fact that linehaul and backhaul customer can be mixed in any order inside a route adds some complexity to the capacity checking, since the

vehicle load fluctuates up and down during the route. Moreover, some of the classical benchmarks for the VRPMB consider a maximum time allowed for any route, while the time employed in covering a route includes both the travel time between any pair of customers as well as the service time. As in the VRPCB, also in the VRPMB it is assumed that each customer can only be visited once and that it has a unique demand (either pick-up or delivery, but not both). Table 3 shows the results obtained by our algorithm after some minor adjustments and without any additional fine tuning process. In fact, for solving the VRPMB only the following parameters were used:

- (i) the one defining the geometric distribution used during the biased-randomization process (β)
- (ii) the percentage of routes to be destructed during the perturbation stage (p)

The penalty coefficient for the interface links (λ) is no longer applied since in the VRPMB more than one interface link is allowed and the precedence constraint (deliveries first, then pick-ups) does not hold anymore. Additionally, the recursive corrective operator (RCO) is not triggered as the number of routes in the solution is not fixed in advance. The rest of the procedures remain intact, which makes our approach even more simple than the one used for solving the VRPCB. The performance of our algorithm is tested using a well-known benchmark problem from the literature. The problem set was proposed by Salhi and Nagy [35]. Three problem subsets are generated as T, Q, and H from 14 VRP instances. In each set, every second, fourth, and 10th customer is declared as a backhaul customer, and a pick-up demand (of the same size as the original demand) is assigned to it. The number of customers in these data sets varies from 50 to 199, and problems CMT6-10 and CMT13-14 impose a maximum route length restriction. We compare our results with the BKS reported in Vidal et. al. [44]. Notice that, despite our algorithm was not specifically designed for solving the VRPMB, after a quick adaptation it is able to obtain very low gaps for more instances employing reasonably low computing times, with an average below 0.4%.

Table 3. Comparative results for VRPMB instances

Instance	n-lh	n-bh	Q	K	Best10		Avg10		BKS	Gap Best10 - BKS
					cost	time (s)	cost	time (s)		
CMT01H	25	25	160	4	465.02	60	465.02	60	465.02	0.00%
CMT01Q	38	12	160	6	489.74	45	489.74	45	489.74	0.00%
CMT01T	45	5	160	7	520.06	42	520.06	42	520.06	0.00%
CMT02H	38	37	140	5	662.63	65	663.43	65	662.63	0.00%
CMT02Q	57	18	140	7	732.66	110	733.71	110	731.26	0.19%
CMT02T	68	7	140	9	782.77	73	782.77	73	782.77	0.00%
CMT03H	50	50	200	3	702.47	145	704.78	145	700.94	0.22%
CMT03Q	75	25	200	4	747.15	133	747.15	133	747.15	0.00%
CMT03T	90	10	200	5	800.34	103	802.95	93	798.07	0.28%
CMT12H	50	50	200	6	636.65	91	641.65	89	629.37	1.16%
CMT12Q	75	25	200	8	729.25	111	738.25	106	729.25	0.00%
CMT12T	90	10	200	9	788.02	62	791.27	59	787.52	0.06%
CMT11H	60	60	200	4	829.50	75	834.83	71	818.05	1.40%
CMT11Q	90	30	200	6	943.83	121	948.83	112	939.36	0.48%
CMT11T	108	12	200	7	1001.94	188	1,012.40	161	998,80	0.31%
CMT04H	75	75	200	6	835.92	402	840.03	403	828.12	0.94%

CMT04Q	113	37	200	9	922.07	421	926.21	394	915.27	0.74%
CMT04T	135	15	200	11	1,001.07	164	1,010.57	155	990.39	1.08%
CMT05H	100	99	200	9	987.67	379	997.81	366	978.74	0.91%
CMT05Q	150	49	200	12	1,119.33	452	1,131.66	447	1,104.87	1.31%
CMT05T	180	19	200	15	1,226.90	523	1,243.35	554	1,218.77	0.67%
CMT06H	25	25	160	7	555.43	24	555.43	24	555.43	0.00%
CMT06Q	38	12	160	7	555.43	31	555.43	31	555.43	0.00%
CMT06T	45	5	160	7	555.43	31	555.43	31	555.43	0.00%
CMT07H	38	37	140	14	900.54	83	900.54	83	900.12	0.05%
CMT07Q	57	18	140	14	900.69	79	900.69	79	900.69	0.00%
CMT07T	68	7	140	14	903.05	51	903.05	51	903.05	0.00%
CMT08H	50	50	160	10	865.50	103	869.95	92	865.50	0.00%
CMT08Q	75	25	160	10	865.50	91	869.95	91	865.50	0.00%
CMT08T	90	10	160	10	865.54	87	874.06	85	865.54	0.00%
CMT14H	50	50	200	11	821.75	104	830.88	104	821.75	0.00%
CMT14Q	75	25	200	11	821.75	63	832.75	52	821.75	0.00%
CMT14T	90	10	200	11	826.77	84	833.91	84	826.77	0.00%
CMT13H	60	60	200	12	1,545.17	191	1,553.16	183	1,542.86	0.15%
CMT13Q	90	30	200	12	1,545.17	150	1,554.35	159	1,542.86	0.15%
CMT13T	108	12	200	12	1,545.17	196	1,556.02	198	1,542.86	0.15%
CMT09H	75	75	200	16	1,169.41	251	1,178.74	245	1,160.68	0.75%
CMT09Q	113	37	200	16	1,169.41	235	1,175.55	228	1,161.24	0.70%
CMT09T	135	15	200	16	1,172.16	231	1,176.33	279	1,162.55	0.83%
CMT10H	100	99	200	20	1,390.96	403	1,401.10	398	1,372.47	1.35%
CMT10Q	150	49	200	20	1,395.85	421	1,399.35	458	1,374.18	1.58%
CMT10T	180	19	200	20	1,395.85	389	1,398.02	379	1,381.04	1.07%
									<i>Average</i>	<i>0.39%</i>

9. CONCLUSIONS

This paper focuses on the Vehicle Routing Problem with Clustered Backhauls, which is found in practical applications of some real-world transportation activities. The solving methodology presented combines a metaheuristic framework with a biased (oriented) randomization version of the well-known savings heuristic. This randomization process allows for quick generation of different promising solutions each time it is run. An operator correcting the number of routes, if this quantity does not meet the fixed size of the fleet, is also introduced.

An extensive series of experiments has been carried out with the two sets of benchmarks available in the literature. According to these tests, our algorithm is able to obtain the best known solution in most of the cases in reasonably low execution times. Additionally, a new best solution has been found for one instance in the TV benchmark set. The algorithm presented here is relatively easy to implement, with just three parameters, and is effortlessly parallelizable. Also, it has been applied successfully to all tested instances without requiring any specific fine-tuning or setup process. These computational properties make this procedure an interesting alternative to other state-of-the-art approaches which are more complex, require more fine-tuning processes, and more difficult to implement in practice.

Moreover, in order to show the flexibility of our approach, after a minor adaptation it has also been used to solve the Vehicle Routing Problem with Mixed Backhauls, in which linehaul and backhaul customers can appear in any

order inside a route. Despite the algorithm was not initially designed to solve this problem, it shows competitive results both in solution quality and computing times, with average gap below 0.5% with respect to the best-known solutions for a set of classical benchmarks.

As future research lines than can emerge from this work, we can highlight the following ones: *(i)* the inclusion of heterogeneous fleets of vehicles and stochastic demands; and *(ii)* the parallelization of the algorithm in order to reduce even further the required computational times and be able to provide ‘real-time’ solutions of high quality.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness and FEDER (TRA2013-48180-C3-P and TRA2015-71883-REDT), and the Ibero-American Program for Science and Technology for Development (CYTED2014-515RT0489). Likewise, we want to acknowledge the support received by the Department of Universities, Research & Information Society of the Catalan Government (2014-CTP-00001) and by the CAN Foundation in Navarre, Spain (CAN2014-3758 and CAN2015-70473).

REFERENCES

- [1] S. Anily, S., The vehicle-routing problem with delivery and back-haul options. *Naval Research Logistics* 43 (1996):415–434.
- [2] R. Bastiaans, Reducing traffic jams, pollution and accidents. *RTD Info*, 25, <http://ec.europa.eu/research/rtdinfo/en/25/05.html> (2000), accessed 26 October 2015.
- [3] G. Berbeglia, J.F. Cordeau, I. Gribkovskaia, G. Laporte, Static pickup and delivery problems: a classification scheme and survey. *TOP* 15 (1) (2007), 1–31.
- [4] F.F. Boctor and J. Renaud, A tour improvement procedure for the travelling salesman problem. Working paper 93-37, Faculté des Sciences de l'Administration, Université Laval, Canada (1993).
- [5] J. Brandao, A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research* 173 (2006), 540–555.
- [6] J. Caceres, P. Arias, D. Guimarans, D. Riera and A. Juan, Rich Vehicle Routing Problem: a Survey. *ACM Computing Surveys*, 47(2) (2014), doi: 10.1145/2666003.
- [7] N. Christofides and S. Eilon, An Algorithm for the Vehicle Routing Dispatching Problem. *Operations Research Quarterly*, 20 (1969): 309-318.
- [8] D.P. Cuervo, P. Goos, K. Sörensen, and E. Arraiz, An Iterated Local Search Algorithm for the Vehicle Routing Problem with Backhauls. *European Journal of Operational Research*, 237 (2014): 454-464.
- [9] G. Clarke and J. Wright, Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 42 (1964), 39–51.
- [10] I. Deif and L. Bodin, Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauls. In: Kidder, A.E. (Eds.), *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park (1984), pp. 75–96.
- [11] E. Demir, T. Bektas and G. Laporte, A review of recent research on green road freight transportation. *European Journal of Operational Research* 237 (2014), 775-793.
- [12] J. Dethloff, Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society* 53 (2002), 115–118.

- [13] M. Fisher, Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research* 42 (1994), 626-642.
- [14] Y. Gajpal and P. Abad, Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European Journal of Operational Research* 196 (2009), 102-117.
- [15] A. Garcia-Najera, The vehicle routing problem with backhauls: a multi-objective evolutionary approach. In: Hao, J.-K., Middendorf, M. (Eds.), *Evolutionary Computation in Combinatorial Optimization*, Lecture Notes in Computer Science, vol. 7245. Springer (2012), pp. 255-266.
- [16] B. Gillett and L. Miller, A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 22 (2)(1974), 340-349.
- [17] M. Goetschalckx and C. Jacobs-Blecha, The vehicle routing problem with backhauls. *European Journal of Operational Research* (1989), 42:39-51.
- [18] M. Goetschalckx and C. Jacobs-Blecha, The vehicle routing problem with backhauls: Properties and solution algorithms. Technical Report MHRC-TR-88-13, Georgia Institute of Technology, Atlanta (1993).
- [19] M. Goetschalckx. Lineback Computational Cases and Results. Personal communication (2011).
- [20] B. Golden, J. Alfaro, E. Baker and J. Schaffer, The vehicle routing problem with backhauling: two approaches. In: *Proceedings of the 21st Annual Meeting of the Southeast Institute of Management Science* (1985), pp. 90-92.
- [21] K. Halse., Modeling and Solving Complex Vehicle Routing Problems. PhD thesis, Institute of Mathematical Statistics and Operations Research (IMSOR) (1992), Technical University of Denmark.
- [22] Hundt, R.: Loop recognition in c++/java/go/scala. In: 2011 Proceedings of Scala Days (2011). Available at: <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/37122.pdf>
- [23] A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip and B. Barrios, 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* 62(6) (2011), 1085-1097.
- [24] A. Juan, J. Faulin, R. Ruiz, B. Barrios and S. Caballe, The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing. *Applied Soft Computing*, 10 (2010), 215-224.
- [25] A. Juan, J. Faulin J., R. Ruiz, B. Barrios, M. Gilibert and X. Vilajosana, Using oriented random search to provide a set of alternative solutions to the capacitated vehicle routing problem. In: Chinneck J, Kristjansson B and Saltzman M (eds). *Operations Research and Cyber-Infrastructure*. Operations Research/Computer Science Interfaces Series, Vol. 47. Springer: New York: USA (2009), pp 331-346.
- [26] H.W. Kuhn, The Hungarian method for the assignment algorithm. *Naval Research Logistics Quarterly* 1 (1955), 88-97.
- [27] G. Laporte, What you should know about the Vehicle Routing Problem. *Naval Research Logistics*, 54 (2007), 811-819.
- [28] S. Lin, S., Computer solutions of the travelling salesman problem. *Bell System Technical Journal* 44 (1965), 2245-2269.
- [29] G. Nagy, S. Salhi, Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research* 162 (2005), 126-141.
- [30] I. Osman and N. Wassan, A reactive tabu search for the vehicle routing problem with backhauls. *Journal of Scheduling* 5 (4) (2002), 263-285.
- [31] S. Parragh, K. Doerner and R. Hartl, A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *Journal für Betriebswirtschaft* 58 (1) (2008), 21-51.
- [32] M. Reimann and H. Ulrich, Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research* 14 (2006), 105-123.
- [33] S. Ropke and D. Pisinger, A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research* (2006), 171750-775.
- [34] M. Savelsbergh and M. Sol. The general pickup and delivery problem. *Transportation Science*, 29 (1995), 17-29.

- [35] S. Salhi and G. Nagy. A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society* 50 (1999), 1034–1042.
- [36] J.B. Sheu and W.K. Talley. Green supply chain management: trends, challenges, and solutions. *Transportation Research Part E* 47 (2011), 791–792.
- [37] C. Sinha, and S. Labi, *Transportation Decision Making: Principles of Project Evaluation and Programming*. Wiley: Hoboken, NJ (2007).
- [38] P. Toth and D. Vigo, A heuristic algorithm for the vehicle routing problem with backhauls. In L. Bianco & P. Toth (Eds.), *Advanced methods in transportation analysis* (1996), pp. 585–608. Springer.
- [39] P. Toth and D. Vigo, An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science* 31 (4) (1997), 372–385.
- [40] P. Toth and D. Vigo, A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operational Research*, 113 (1999), 528–54.
- [41] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications* (2014). SIAM Philadelphia.
- [42] S. Ubeda, F.J. Arcelus and J. Faulin, Green Logistics at Eroski: A Case Study. *International Journal of Production Economics*, 131(1) (2011), pp. 44–51.
- [43] T. Vidal, T. Crainic, M. Gendreau and C. Prins, Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1) (2013), 1–21.
- [44] T. Vidal, T. Crainic, M. Gendreau and C. Prins, A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3) (2014), 658–673.
- [45] D. Vigo, A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research*, 89 (1996), 108–126.
- [46] A. Wade and S. Salhi, An ant system algorithm for the vehicle routing problem with backhauls. MIC_2001—4th Metaheuristic International Conference. Porto, Portugal, July. 16–20, 2001.
- [47] A. Wade and S. Salhi. An investigation into a new class of vehicle routing problem with backhauls, *Omega* 30 (2002), 487–497.
- [48] N. Wassan, Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *Journal of the Operational Research Society* 58 (12) (2007), 1630–1641.
- [49] N. Wassan, S. Salhi and G. Nagy, Solving the mixed VRP with backhauling using ants. *AIP Conference Proceedings* 1159 (2009), pp. 248–253.
- [50] N. Wassan, N. Wassan, G. Nagy and S. Salhi, The Multiple Trip Vehicle Routing Problem with Backhauls: Formulation and a Two-Level Variable Neighbourhood Search. *Computers and Operations Research* (2016), <http://dx.doi.org/10.1016/j.cor.2015.12.017i>
- [51] G. Yazgi Tüçüntü, C.A.C. Carreto and B.M. Baker, A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, 37 (1) (2009), 138–154.
- [52] E.E. Zachariadis and C.T. Kiranoudis, An effective local search approach for the Vehicle Routing Problem with Backhauls. *Expert Systems with Applications* 39 (2012), 3174–3184.

Carpeta actual: **ENTRADA**[Desconectarse](#)[Escribir](#)[Direcciones](#)[Carpetas](#)[Opciones](#)[Buscar](#)[Ayuda](#)[Calendario](#)[Resultado de la búsqueda](#) | [No leído](#) | [Borrar](#)[Reenviar](#) | [Reenviar como adjunto](#) | [Responder](#) | [Responder a todos](#)**Asunto:** IMPORTANT: Your article accepted in International Transactions in Operational Research**De:** cs-author@wiley.com**Fecha:** Jue, 1 de Diciembre de 2016, 21:03**Para:** JAVIER.BELLOSO@UNAVARRA.ES**Prioridad:** Normal**Opciones:** [Ver encabezado completo](#) | [Ver versión imprimible](#) | [Bajar este mensaje como un archivo](#) | [View Message Details](#) | [Add to Address Book](#)

Dear JAVIER BELLOSO,

Article: An iterative biased-randomized heuristic for the fleet size and mix vehicle routing problem with backhauls

Journal: International Transactions in Operational Research

Congratulations on the acceptance of your article for publication in International Transactions in Operational Research.

Your article has been received by production. Please click [here](#) or paste this link into your browser to register for Wiley Author Services.http://authorservices.wiley.com/index.html#register-invite/TgWVZAQno_jKystTftENWM2S4E1bKstvd6SzR48nI9g=

Once you've registered, you can:

- Sign your license agreement (REQUIRED)
- Track your article's progress to publication
- Access your published article
- Invite colleagues to view your published article

Sincerely,
Wiley Author Services



An iterative biased-randomized heuristic for the fleet size and mix vehicle routing problem with backhauls

Journal:	<i>International Transactions in Operational Research</i>
Manuscript ID	ITOR-Mar-16-OA-0065.R3
Manuscript Type:	Original Article
Date Submitted by the Author:	n/a
Complete List of Authors:	Belloso, Javier; Universidad Publica de Navarra, Mathematical Engineering and Computing Juan, Angel; Open University of Catalonia, Computer Science, Multimedia, and Telecommunication Faulin, Javier; Public University of Navarre, Statistics and Operations Research
Keywords:	Vehicle routing problem with backhauls, Heuristics, biased randomization, multi-start algorithms, fleet size and mix vehicle routing problem
Main Contributions:	conceptual simplicity, original methaheuristic, 20 new best known solution out of 36, Integrate different strategies: (i)sucsesive approximation method, (II)biased-randomization, (III) memory based local search

SCHOLARONE™
Manuscripts

Only

An iterative biased-randomized heuristic for the fleet size and mix vehicle routing problem with backhauls

Javier Belloso¹, Angel A. Juan², Javier Faulin¹

(1) Dept. of Statistics and Operations Research. Public University of Navarre, 31006 Pamplona, Pamplona, SPAIN. e-mail: javier.belloso@unavarra.es, javier.faulin@unavarra.es

(2) Dept. of Computer Science, Multimedia, and Telecommunication – IN3. Open University of Catalonia, 08018 Barcelona, SPAIN. e-mail: ajuanp@uoc.edu

Abstract

This paper analyzes the fleet mixed vehicle routing problem with backhauls (FSMVRPB), a rich and realistic variant of the popular vehicle routing problem in which both delivery and pick-up customers are served from a central depot using a heterogeneous and configurable fleet of vehicles. After a literature review on the issue and a detailed description of the problem, a solving approach based on a multi-start biased-randomized heuristic is proposed. Our algorithm uses an iterative method that relies on solving a series of smaller instances of the homogeneous-fleet version of the problem and then using these sub-solutions as partial solutions for the original heterogeneous instance. In order to better guide the exploration of the solutions space, the algorithm employs several biased-randomized processes: a first one for selecting a vehicle type; a second one for sorting the savings list; and a third one to define the number of routes that must be selected from the homogenous-fleet sub-solution. The computational experiments show that our approach is competitive and able to provide 20 new best-known solutions for a 36-instance benchmark recently proposed in the literature.

Keywords: *Vehicle routing problem with backhauls; heuristics; biased randomization; multi-start algorithms; fleet size and mix vehicle routing problem.*

1 Introduction

Road transportation is the predominant way of transporting goods in many world regions. This explains the relevance of rich and real-life vehicle routing problems (VRPs), since efficient route planning can significantly reduce transportation costs and offer a better service to customers (Caceres et al., 2014). This paper analyzes the fleet mixed vehicle routing problem with backhauls (FSMVRPB), in which both delivery and pick-up customers are served from a central depot using a heterogeneous and configurable fleet of vehicles. This rich and realistic variant combines the aspects of two well-known VRP versions, i.e.: the fleet size and mix VRP (FSMVRP), initially described in Golden et al. (1984), and the VRP with backhauls (VRPB), introduced in Golden et al. (1985).

The FSMVRP relates to finding the best fleet composition considering that vehicles costs can be divided into fixed and variable ones. Fixed costs are directly allocated when a vehicle of certain type is assigned to a route –despite the characteristics of the assigned route–, whereas variable costs might depend upon the type of vehicle employed and the route length. The objective here is the minimization of the total costs required to deliver the ordered products to the customers. Having to deal with heterogeneous fleets is quite usual in most practical situations. In fact, when considering fixed and variable costs, even two vehicles with the same capacity can be considered as different –in terms of costs– due to factors such as depreciation, insurance, maintenance, or operating costs (Hoff et al., 2010). In the VRP with backhauls, customers are also allowed to return goods to the depot, although it is assumed that each specific customer cannot receive and return items simultaneously –if that were the case, it would be considered as two different customers during the problem-modeling process.

In a recent work, Salhi et al. (2013) introduced the combined problem, motivated its importance, and proposed a solving approach for the FSMVRPB. These authors also proposed a new set of 36 instances for benchmarking purposes. Following their seminal work, this paper offers a review of related work and proposes a multi-start biased-randomized heuristic to solve the FSMVRPB. Our algorithm uses an iterative method that relies on solving a series of smaller instances of the homogeneous-fleet version of the problem and then using these sub-solutions as

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

partial solutions for the original heterogeneous instance. Also, the algorithm employs several biased-randomized processes in order to better guide the search process.

The remaining of the paper is organized as follows: Section 2 reviews the related works. Section 3 provides a description of the FSMVRPB. Section 4 explains the main ideas of our algorithm. Section 5 describes the pseudo-code details –which might be especially useful for implementation purposes. Section 6 contains the computational experiments carried out in order to test the efficiency of our approach. Finally, Section 7 summarizes the main contributions of our work.

2 Literature review

The FSMVRPB considering both fixed and variable costs was formally introduced by [Salhi et al. \(2013\)](#). Before that, [Tütüncü \(2010\)](#) analyzed a simplified version of the problem including heterogeneous fleet as well as fixed and variable costs, but without considering the fleet size and mix component –i.e., the fleet size and composition was given in advance. Our work follows the one by [Salhi et al. \(2013\)](#) and, since no other previous work in the literature has examined the FSMVRPB, we will review separately the two components of the problem: the VRPB and the FSMVRP.

2.1 The VRP with Backhauls

The most common and intuitive approach shown in the literature to solve this problem follows the next scheme: (a) solving the linehaul problem; (b) solving the backhaul problem; and, (c) combining both partial solutions into a global one. The two first sub-problems are, in fact, the same problem, whereas the third one is exclusive of the backhauling variant. One of the first methods proposed to solve the VRPB was designed by [Deif and Bodin \(1984\)](#). These authors follow the aforementioned sequential approach and explore the solutions space by adapting the popular savings heuristic (CWS) developed by [Clarke and Wright, \(1964\)](#). The savings list used by the CWS heuristic is adjusted to the VRPB case. This is done by incorporating the so-called precedence constraint, i.e.: backhaul customers cannot be visited until all deliveries have been completed. The criteria for the selection of the interface edge (the one

connecting the linehaul route with the backhaul route) is critical in this scenario, since it can tremendously improve the quality of each solution.

A similar approach is presented in [Golden et al. \(1985\)](#), who proposed a heuristic procedure based on the insertion principle: routes are initially generated for linehaul customers; then, backhaul customers are inserted into routes according to an insertion criterion. They use a penalty factor to impose the precedence constraint. This penalization forces the backhaul route to be introduced at the end of the linehaul route. [Goetschalckx and Jacobs-Blecha \(1989\)](#) proposed an algorithm that makes use of a mapping method to introduce the precedence constraint and generate two required clusters, one for linehaul customers and another one for backhaul customers. The two sequences are managed separately to create feasible routes, and then routes are merged. Another approach, proposed by [Goetschalckx and Jacobs-Blecha \(1993\)](#), is classified as a ‘cluster-first-and-route-second’ algorithm. The cluster creation is managed as a generalized assignment problem and the route creation as a travelling salesman problem (TSP) with the constraint that only one interface from delivery to pick-up points is permitted in each route. As explained before, the best sequence of customers’ visits in the route is highly dependent on the selection of the link interface. Using the same ‘cluster-first-and-route-second’ logic, [Anily \(1996\)](#) introduced another algorithm in which the clustering phase is accomplished by a modified circular partitioning heuristic. Once the customers are assigned to clusters, the process continues with the construction of travelling salesman tours through all the clusters (either linehaul or backhaul but not both). Finally, linehaul clusters are linked to backhaul clusters, and a route generation phase is initiated in order to determine the optimal connections between the depot and the clusters. The popular sweep algorithm from [Gillett and Miller \(1974\)](#) can be classified into this type of ‘cluster-first-and-route-second’ approaches. Also, [Toth and Vigo \(1999\)](#) proposed another heuristic based on a K-tree Lagrangian relaxation formulated by [Fisher \(1994\)](#) to the VRP. They introduce the idea of asymmetric costs between two customers, and make use of lower bounds produced by a Lagrangian approach.

Some other methods designed for the general VRP, can easily include the precedence constraint. This is the case of the improvement/exchange heuristics, which add the restriction that linehaul and backhaul customers should not be mixed during the process. In this direction,

we can consider 2-opt and 3-opt procedures proposed by [Lin \(1965\)](#), which can be easily applied separately to the sets of linehauls and backhauls within each route.

VRPB literature since 2000 is extensively focused on metaheuristic methods. In this context, [Osman and Wassan \(2002\)](#) developed a tabu search (TS) algorithm. They solve the problem by using a metaheuristic based on savings, insertion, and assignment approaches. [Brandão \(2006\)](#) proposed a similar solution with his TS algorithm. [Wassan \(2007\)](#) designed a TS enhanced by adaptive memory programming. [Ropke and Pisinger \(2006\)](#) analyzed the VRPB as an extension of the rich pick and delivery problem with time windows. They presented a large neighborhood search (LNS) heuristic to solve it. A detailed new classification of the problem was introduced by [Parragh et al. \(2008\)](#). Different backhauling strategies using ants are compared in [Reimann and Ulrich \(2006\)](#). [Wassan et al. \(2009\)](#) and [Gajpal and Abad \(2009\)](#) proposed an ant colony system to solve the VRPB. [Tütüncü et al. \(2009\)](#) proposed a decision support system (DSS) based on a greedy randomized adaptive search with memory programming (GRASP). [Zachariadis and Kiranoudis \(2012\)](#) proposed a metaheuristic local search approach where different customer sequences were exchanged. [Cuervo et al. \(2014\)](#) proposed an iterated local search algorithm that produces high quality results.

In the recent years, environmental issues have attracted the attention of many researches due to the potential reduction of gas emissions and therefore costs. This result can be achieved by combining pickups and deliveries in the same route to avoid empty vehicles during the return-to-depot stage. This characteristic has been analyzed in papers like [Berbeglia et al. \(2010\)](#), [Ubeda et al. \(2011\)](#), and [Sheu and Talley \(2011\)](#). Similarly, our literature review on the VRPB can be complemented with the extensive review completed by [Toth and Vigo \(2014\)](#).

2.2 Fleet Size and Mix Vehicle Routing Problem (FSMVRP)

In the FSMVRP, all customers receive materials from the depot. The problem has several variants depending on the assumptions made on the fleet size (either limited or unlimited) and the consideration or not of both fixed and variable costs.

The FSMVRP was introduced by [Golden et al. \(1984\)](#), and it considers variable costs. These costs are computed as the distance traveled. In this case, a limited fleet size is considered. These authors use a constructive heuristic based on the aforementioned saving heuristic. Using a

similar cost structure and fleet configuration, several authors have used TS algorithms for solving the FSMVRP (Gendreau, 1999; Osman and Wassan 2002). Taillard (1999) applies the column generation method to obtain homogenous routes for all kind of vehicles. Then, he solves the set partitioning problem to obtain the final solution by ensuring that each node is visited exactly once. Renaud and Boctor (2002) apply a similar algorithm, but with some improvements.

Choi and Tcha (2007) use a column generation method to solve the problem. Liu and Shen (1999) apply a genetic algorithm and Subramanian and Penna (2012) implement a hybrid algorithm by combining an iterated local search and a set partitioning mechanism. Some authors also consider variable costs, as in Salhi et al. (2013). Taillard (1999) adopts a similar approach. The goal for the version of the problem when the fleet size is limited is to obtain a suitable utilization of the existing fleet. Taillard (1999) adapts his algorithm to the limited fleet version. Tarantilis et al. (2003) obtain an initial solution with a constructive heuristic, achieving improvements through random movements that are evaluated following different criteria. They use a simulated annealing methodology. Finally, excellent reviews on this problem can be found in Baldacci and Battarra (2008) and in Hoff et al. (2010).

3 Problem description

The FSMVRPB model considered in this article is an extension of the model proposed in Salhi et al. (2013), being the main difference that in our model not only fixed but also variable costs associated to each vehicle can be taken into account. Our model also extends the model of Lee et al. (2008) by incorporating the presence of backhauls. Consider a graph $G = (V, A)$ where: $V = \{v_0, v_1, \dots, v_L, v_{L+1}, \dots, v_n\}$ is a set of nodes including the depot, v_0 , L delivery customers, and $B = n - L$ pick-up customers; and A is a set of arcs connecting those nodes. Each arc $(i, j) \in A$ is associated with a travel distance d_{ij} that is known in advance. Located at the depot, there is a heterogeneous fleet of vehicles of K different types. The capacity of each vehicle v is denoted by Q_v . Also, each vehicle v has both fixed costs, f_v , as well as variable costs, c_v . Each customer i has an associated demand d_i . The goal of the FSMVRPB is to design a set of routes minimizing total costs (fixed plus variable ones) of serving the customers (both linehauls and

backhauls) subject to the following constraints: (a) each route is assigned to exactly one vehicle; (b) each route departs from the depot, serves its assigned customers, and returns to the depot; (c) each customer is visited once by exactly one vehicle; and (d) the maximum load carried by each vehicle during a route cannot exceed the vehicle capacity.

4 Overview of our solving methodology

This section offers an overview of the solving approach we have developed for the FSMVRPB, which integrates some methods and techniques developed in our previous works (for different VRP variants) and then adds a final combination stage in which different routes are mixed to build complete solutions. As depicted in [Figure 1](#), our approach successively solves homogenous versions of the problem –each of them using a different type of vehicle– in order to solve the heterogeneous version of the problem. Then, it combines parts of the homogeneous solutions to generate a solution for the heterogeneous case. This methodology builds up on the “successive approximations method” (SAM) originally proposed by [Juan et al. \(2014\)](#) to solve the heterogeneous VRP. SAM is a multi-round process. At each round, a new subset of nodes and a new type of vehicle are selected following some specific criteria. Then, assuming an unlimited fleet of vehicles of this type, the associated homogeneous-fleet vehicle routing problem is solved. After several rounds, a global solution for the heterogeneous VRP is obtained by merging routes from different homogeneous VRP solutions. Notice that the SAM approach transforms the challenge of solving a complex heterogeneous VRP into the challenge of solving a series of related homogeneous VRPs, each of them smaller than the original heterogeneous VRP. The process shown in [Figure 1](#) starts by randomly selecting the type of vehicle (each type of vehicle has a different capacity) to be used in the homogeneous version of the problem. Notice that only vehicles with a capacity larger than the maximum demand to be served can be considered at this stage. Then, the problem becomes homogeneous and it is solved assuming an unlimited number of vehicles of this type (CVRPB). In order to solve the homogeneous version of the problem we have used the algorithm presented in [Belloso et al. \(2015\)](#), which is based on a variant of the aforementioned CWS heuristic. Thus, the adapted heuristic makes use of a savings list which is manipulated in two steps. In the first step, the interface links (edges connecting one linehaul customer with one backhaul customer) are penalized using the approach

1
2
3
4 proposed in [Deif and Bodin \(1984\)](#). This penalization makes them to be chosen at a later stage
5 than if the traditional CWS heuristic was employed instead. Therefore, the precedence constraint
6 (linehauls first, then backhauls) is satisfied.
7

8
9
10 In the second step, the savings list obtained in the CWS heuristic is randomized using a
11 geometric probability distribution ([Juan et al., 2011, 2015](#), [Dominguez et al., 2014](#)). There are
12 different approaches for randomizing a heuristic with the goal of better exploring the solutions
13 space ([Hemmati and Hvattum, 2016](#), [Santos et al., 2016](#)). One of these is the use of a greedy
14 randomized adaptive search procedure (GRASP) ([Resende and Ribeiro, 2010](#)). GRASP is a
15 multi-start process that randomizes a given heuristic to better explore the solution space. The
16 solution is obtained by following a constructive sequence. At each step, a restricted list of
17 candidates is ordered following a greedy function that evaluates the benefit of selecting each
18 element. Then, a candidate is randomly selected for that restricted list, thus allowing the
19 algorithm to generate different solutions at each iteration of the multi-start process. In a classical
20 GRASP, randomness is typically provided by a symmetric uniform distribution. However, in our
21 approach we use the geometric probability distribution (a skewed one) to induce a biased-
22 randomized selection process. That way, we add some degree of randomness into the CWS
23 heuristic without destroying the logic behind it. This procedure improves and extends the initial
24 ideas proposed in [Faulin and Juan \(2008\)](#). In other words, this biased-randomization technique
25 ensures that edges with higher savings will have also a higher probability of being selected at
26 each iteration of the solution-construction process ([Juan et al. 2015](#), [Quintero et al. 2016](#)). Other
27 variants of biased-randomization techniques have been also applied in the improvement of
28 genetic algorithms. Some examples of this can be found in [Reis et al. \(2011\)](#), [Morán-Mirabal et
29 al. \(2014\)](#), [Gonçalves and Resende \(2014\)](#), [Brandão et al. \(2015\)](#) and [Gonçalves et al. \(2016\)](#).
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

45 At this stage, the algorithm provides a solution for the homogeneous version of the problem.
46 This solution is then improved throughout a memory-based local search process in which any
47 route in the current solution is checked against a cache memory (hash map data structure in
48 programming terminology) storing the best-known way of routing the set of nodes in the given
49 route. If the same set of nodes is found in memory (hash map) and the cost is improved, the route
50 included into the current solution is updated using the improved equivalent route stored in
51 memory. Otherwise, the memory is updated with the route in the current solution. More technical
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

details of this technique can be found in [Juan et al. \(2011\)](#). Once this local search process is finished, the next step is to randomly determine which routes in the homogeneous solution will be used as ‘building blocks’ for the heterogeneous solution. The selected routes determine a set of customers already served and type of vehicles already used. Thus, the same procedure is iteratively run –using a different type of vehicle for solving the homogeneous version of the problem– until all customers have been served. At the end of this iterative approach, a complete solution for the heterogeneous version is obtained by aggregating all the routes selected from the solutions to the different homogeneous problems.

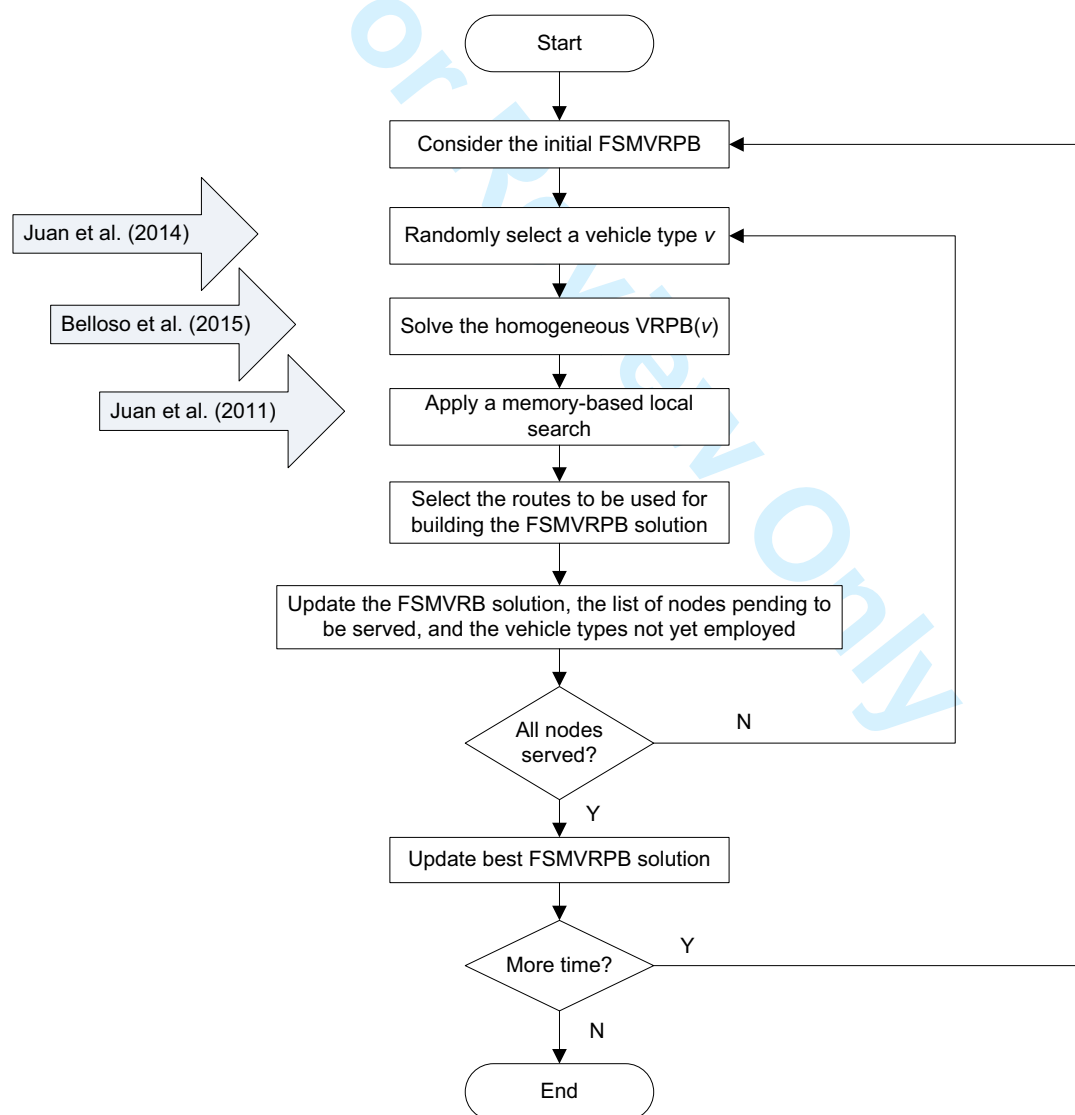


Figure 1: Flowchart for the proposed methodology.

5 Algorithm pseudo-code

The methodology presented in the previous section is implemented through the ‘iterative homogeneous problems resolution operator’ (IHPO), which is called from a multi-start procedure each time a homogeneous VRPB needs to be solved. This multi-start procedure ends when a time-based termination criteria is reached. Each new solution generated by this procedure for a given VRPB is compared with the best one obtained so far for the same problem. **Figure 2** shows the algorithm pseudo-code. Lines 1, 2, and 3 represent the initialization of a particular homogeneous problem. Specifically, line 3 defines the homogeneous problem once the vehicle type is chosen in line 1. The customers to be visited are defined as inputs of the procedure. Line 4 calls to the heuristic that generates a solution for the homogeneous problem. Line 5 calls to the memory-based local search that tries to improve the current solution. At this point, part of the homogeneous solution is ready to be promoted to the heterogeneous solution. Lines 6 to 10 manage the selection of routes and update the list of available types of vehicles. If the current vehicle type is the last one in the list of vehicles, all routes in the homogeneous solution will be transferred to the final solution. Otherwise, a random number between 1 and the number of routes is selected. Line 11 summarizes the process of promoting routes to the final solution, while lines 12 and 13 represent the updated value of customers and vehicles. Line 14 checks if the termination criteria of the procedure is met. If this is the case, the iterative procedure returns the current solution; otherwise (Line 16) the procedure is invoked again with the non-served customers and non-used vehicle types as input parameters.

```

procedure IHPO(customers, vehicles) // Iterative Homogeneous Problems Operator
// Iteratively solves homogeneous problems until all customers are served
01  newVehType <- selectRandom(vehicles) // Select new type of vehicle randomly
02  vehCap <- getCapacity(newVehType)
03  newHomVRPB <- Initialization(customers, vehCap)
// Solve homogeneous problem using an efficient algorithm
04  homSol <- solveHoVRPB(newHomVRPB)
// Use a cache structure to recall previously computed subroutes
05  homoSol <- ImproveNodesOrder(homoSol)
// Select number of routes (numroutes) to be transferred to heterogeneous solution

```

```

06  if {pending vehicles is 0} then
07      numroutes <- routes(homoSol)
08  else
09      numroutes <- selectRandom(1...routes(homoSol))
10  end if
11  heteroSol <- addRoutes(homoSol(numroutes))
12  pendingCustomers <- extractCustomers(homoSol(routes(homoSol)-numroutes))
13  pendingVehicles <- extractVehicles(newVehType)
14  // Return final solution
15  if {number of pendingVehicles is zero} then
16      return heteroSol
17  else
18      return heteroSol & IHPO(pendingCustomers, pendingVehicles)
19  end procedure

```

Figure 2: Pseudo-code for the main algorithm.

6 Experimental results

The FSMVRPB instances we consider in this paper are the ones recently proposed by [Salhi et al. \(2013\)](#), which are generated using the mixed fleet data set of [Golden et al. \(1984\)](#) and the VRPB data set of [Toth and Vigo \(1997\)](#). From the mixed fleet data set these instances take the mix fleet attributes (distances, fixed and variable costs, and vehicle capacities), while they take the backhaul percentages from the VRPB data set. It should be noted that the original source of this FSMVRPB data set is the data set initially proposed by [Christofides and Eilon \(1969\)](#). [Salhi et al. \(2013\)](#) generated a set of 36 FSMVRPB instances using the 12 test problems of [Golden et al. \(1984\)](#) ranging from 20 to 100 customers. For each instance, following the conventions used in [Toth and Vigo \(1997\)](#), three new instances were generated using linehaul/backhaul (LH/BH) percentages of 50/50, 67/33, and 80/20. This was done by taking, as a backhaul customer, every first customer in series of two, three, and five customers, respectively. The algorithm described in this paper has been implemented as a Java application. We have solved the set of 36 instances in order to test the effectiveness of the proposed method.

A standard personal computer, Intel R Core TM i7 CPU M 640 @ 2.80 GHz, 3.42 GB RAM, and a MS XP Professional Operating system, was used to perform these tests. The results are summarized in [Table 1](#), which contains the following information for each instance: name of the instance, number of customers, number of linehaul customers, number of backhaul customers,

1
2
3
4 best-known solution (BKS) as reported by [Salhi et al. \(2013\)](#), our best solution (OBS) after ten
5 runs of the algorithm, the percentage gap between both values (computed as $(OBS-BKS)/BKS$),
6
7
8 computing time employed in obtaining OBS, and optimal solution or lower bound as reported in
9
10 [Salhi et al. \(2013\)](#). Notice that our algorithm is able to provide 20 new best-known solutions and
11 offers an average negative gap of -0.54% with respect to the previous BKS values. [Figure 3](#) also
12 contributes to illustrate the efficiency of our approach by showing a visual comparison among
13 the BKS, OBS, and the lower bound values for each instance. The results provided by our
14 algorithm are particularly noteworthy taking into consideration the low computation times
15 employed.
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Table 1: Computational results for the FSMVRPB

Problem	Size	Line	Back	BKS	OBS	Gap (%)	Time (s)	Optimal* or Lower Bounds
HWS1	20	10	10	726.48	734.03	1.04%	10	720.7*
HWS2	20	13	7	818.12	820.99	0.35%	15	818.12*
HWS3	20	16	4	848.59	848.32	-0.03%	1	848.23*
HWS4	20	10	10	4,350.65	4,342.48	-0.19%	2	4,342.48*
HWS5	20	13	7	5,366.39	5,357.98	-0.16%	8	5,357.98*
HWS6	20	16	4	5,875.23	5,872.52	-0.05%	6	5,421.63*
HWS7	20	10	10	767.93	729.50	-5.00%	15	729.50*
HWS8	20	13	7	872.97	838.11	-3.99%	16	838.11*
HWS9	20	16	4	903.18	890.76	-1.38%	18	890.76*
HWS10	20	10	10	4,365.44	4,349.13	-0.37%	2	4,349.12*
HWS11	20	13	7	5,414.50	5,363.58	-0.94%	21	5,363.58*
HWS12	20	16	4	5,928.78	5,523.50	-6.84%	102	5,465.40
HWS13	50	25	25	1,625.70	1,632.60	0.42%	23	1,480.75
HWS14	50	33	17	1,811.63	1,793.26	-1.01%	14	1,705.44
HWS15	50	40	10	2,018.93	2,030.37	0.57%	65	1,937.24
HWS16	50	25	25	5,561.67	5,562.94	0.02%	16	5,537.24
HWS17	50	33	17	6,570.39	6,571.77	0.02%	4	6,547.93*
HWS18	50	40	10	7,599.08	7,413.27	-2.45%	19	7,026.15
HWS19	50	25	25	1,704.41	1,659.86	-2.61%	71	1,509.41
HWS20	50	33	17	2,037.23	2,038.22	0.05%	120	1,926.03
HWS21	50	40	10	2,340.09	2,341.17	0.05%	97	2,196.32
HWS22	50	25	25	1,774.71	1,746.28	-1.60%	11	1,697.38
HWS23	50	33	17	2,166.52	2,164.65	-0.09%	110	2,028.35
HWS24	50	40	10	2,430.88	2,425.83	-0.21%	23	2,335.56
HWS25	75	37	38	1,332.02	1,313.59	-1.38%	246	1,228.48
HWS26	75	50	25	1,421.04	1,412.69	-0.59%	95	1,341.97
HWS27	75	60	15	1,534.65	1,543.10	0.55%	145	1,466.24
HWS28	75	37	38	1,617.85	1,631.70	0.86%	65	1,486.89
HWS29	75	50	25	1,799.76	1,808.46	0.48%	45	1,660.15
HWS30	75	60	15	1,990.46	2,001.79	0.57%	197	1,853.72
HWS31	100	50	50	4,943.29	4,990.84	0.96%	310	4,870.54
HWS32	100	66	34	6,035.96	6,019.82	-0.27%	126	5,901.57
HWS33	100	80	20	7,601.29	7,593.25	-0.11%	43	6,886.44
HWS34	100	50	50	2,465.41	2,516.30	2.06%	241	2,369.80
HWS35	100	66	34	2,971.98	2,998.21	0.88%	156	2,730.67
HWS36	100	80	20	3,533.90	3,563.17	0.83%	253	3,139.87
Averages						-0.54%	75.31	

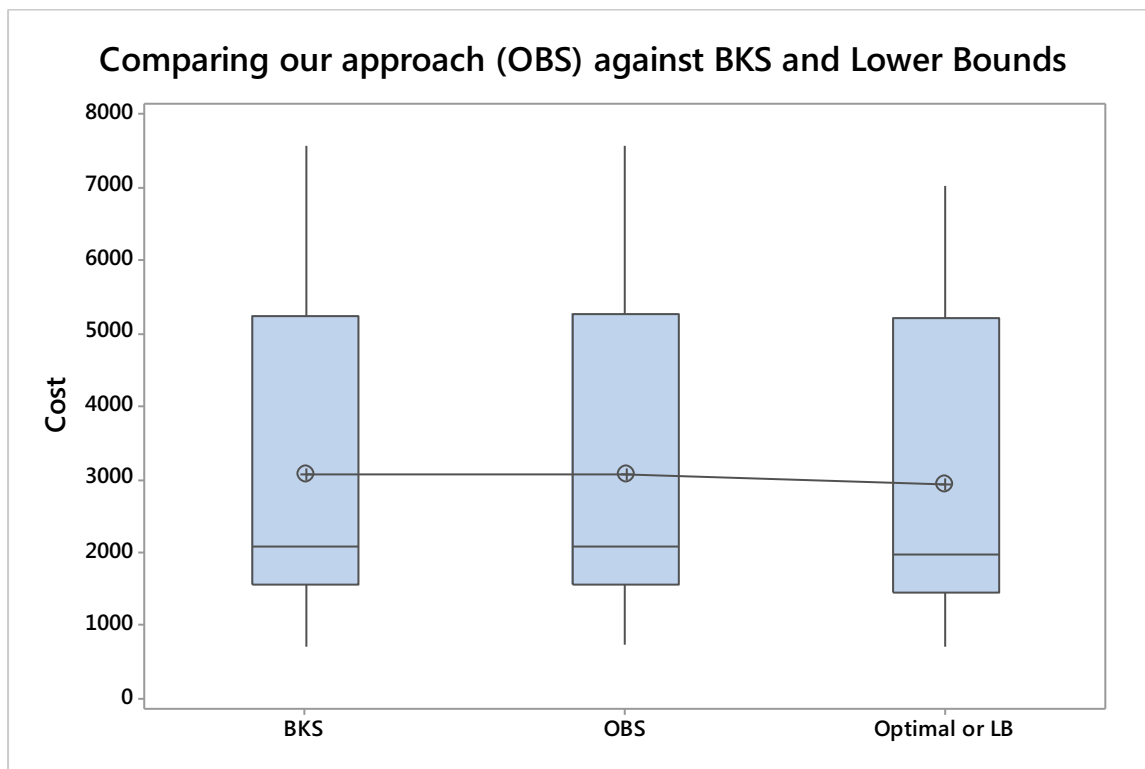


Figure 3: Comparison of our approach with BKS and Optimal or LB.

7 Conclusions

This paper presents an original and efficient metaheuristic approach to solve the fleet size and mix vehicle routing problem with backhauls (FSMVRPB), a realistic variant of the vehicle routing problem that has been rarely considered in the literature due to its inherent complexity. Our approach integrates different strategies that contribute to effectively deal with the problem, including: a successive approximation method that reduces the FSMVRPB to iteratively solving a series of homogeneous vehicle routing problems with backhauls (VRPB), a biased-randomized metaheuristic algorithm for solving the VRPB, and a memory-based local search. Despite its conceptual simplicity, the proposed approach is able to provide noticeable results when compared with other previous works, and it attains 20 new best-known solutions for the 36-instance set considered, also showing negative average gap and low computing times.

As a future research line, we plan to incorporate even more realistic conditions to the problem by considering stochastic demands for the customers, which we plan to address by transforming the current metaheuristic approach into a simheuristic one as described in [Juan et al. \(2015\)](#).

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (TRA2013-48180-C3-P, TRA2015-71883-REDT), FEDER, and the Ibero-American Program for Science and Technology for Development (CYTED2014-515RT0489). Likewise, we want to acknowledge the support received by the Department of Universities, Research & Information Society of the Catalan Government (2014-CTP-00001) and by the CAN Foundation in Navarre (CAN2014-3758, CAN2015-70473).

References

- Anily, S. 1996. "The Vehicle-Routing Problem with Delivery and Back-Haul Options." *Naval Research Logistics* 43 (3): 415–34.
- Baldacci, R., Battarra, M. and Vigo D. 2008. "Routing a Heterogeneous Fleet of Vehicles." In: Golden BL, Raghavan S, Wasil EA. Editors . New York: Springer.
- Belloso, J., Juan, A., Faulin, J., Serrano, A. 2015. "Using Multi-Start Biased Randomization of Heuristics to Solve the Vehicle Routing Problem with Clustered Backhauls." *Lecture Notes in Management Science* 7: 15–20.
- Berbeglia, G., Cordeau, J. F. and Laporte, G. 2010. "Dynamic Pickup and Delivery Problems." *European Journal of Operational Research* 202 (1): 8–15.
- Brandão, J. 2006. "A New Tabu Search Algorithm for the Vehicle Routing Problem with Backhauls." *European Journal of Operational Research* 173 (2): 540–55.
- Brandão, J.S., Noronha, T.F., Resende, M.G.C., Ribeiro, C.C. 2015: "A biased random-key genetic algorithm for single-round divisible load scheduling." *International Transactions in Operational Research*. 22 (5), 823–839.
- Caceres, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. 2014. "Rich Vehicle Routing Problem: a Survey". *ACM Computing Surveys*, 47(2), doi: 10.1145/2666003.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
- Choi, E., and Tcha, D. W. 2007. "A Column Generation Approach to the Heterogeneous Fleet Vehicle Routing Problem." *Computers and Operations Research* 34 (7): 2080–95.
- Christofides, N., and Eilon, S.. 1969. "An Algorithm for the Vehicle- Dispatching Problem." *Operational Research Quarterly* 20 (3): 309–18.
- Clarke, G., and Wright, J.. 1964. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points." *Operations Research* 12 (4): 568–81.
- Cuervo, D. P., Goos, P., Sørensen, K. and Arráiz, E. 2014. "An Iterated Local Search Algorithm for the Vehicle Routing Problem with Backhauls." *European Journal of Operational Research* 237 (2). Elsevier: 454–64.
- Deif, I., and Bodin, L.. 1984. "Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauls." In *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, 75–96.
- Dominguez, O., Juan, A. A. and Faulin, J. 2014. "A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations." *International Transactions in Operational Research.*, 21 (3): 375–398.
- Fisher, M. 1994. "Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees." *Operations Research* 42: 626–42.
- Gajpal, Y, and Abad, P. L. 2009. "Multi-Ant Colony System (MACS) for a Vehicle Routing Problem with Backhauls." *European Journal of Operational Research* 196 (1): 102–17.
- Gendreau, M. 1999. "A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem." *Computers & Operations Research* 26: 1153–73.
- Gillett, B., Miller, L. 1974. "A Heuristic Algorithm for the Vehicle-Dispatch Problem." *Operations Research* 22 (2): 340–49.
- Goetschalckx, M. Jacobs-Blecha, C. 1989. "The Vehicle Routing Problem with Backhauls." *European Journal of Operational Research*, no. 42: 39–51.
- Goetschalckx, M. Jacobs-Blecha, C. 1993. *The Vehicle Routing Problem with Backhauls: Properties and Solution Algorithms*. Technical Report MHRC-TR-88-13.
- Golden, B., Assad, A. and Levy, L. 1984. "The Fleet Size and Mix Vehicle Routing Problem." *Computers & Operations Research* 11, 49–66.
- Golden, B., Alfaro, J., Baker, E. and Schaffer, J. 1985. "The Vehicle Routing Problem with Backhauling: Two Approaches." In *Proceedings of the 21st Annual Meeting of the Southeast Institute of Management Science*, 90–92.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
- Gonçalves, J.F. and Resende, M.G., 2014. "An extended Akers graphical method with a biased random-key genetic algorithm for job-shop scheduling." *International Transactions in Operational Research*, 21(2), 215-246.
- Gonçalves, J.F., Resende, M.G., Dias C.M., 2016. "A biased random-key genetic algorithm for the minimization of open stacks problem." *International Transactions in Operational Research* 23, 25–46.
- Hemmati, A. and Hvattum, L. M. 2016. "Evaluating the importance of randomization in adaptive large neighborhood search." *International Transactions in Operational Research*. doi:10.1111/itor.12273.
- Hoff, A., Andersson, H., Christiansen, M., Hasle, G. and A. Løkketangen. 2010. "Industrial Aspects and Literature Survey: Fleet Composition and Routing." *Computers & Operations Research* 37 (12): 2041–61.
- Juan, A., Faulin, J., Caceres, J., Barrios, B., Martinez, E. 2014. "A Successive Approximations Method for the Heterogeneous Vehicle Routing Problem: analyzing different fleet configurations". *European Journal of Industrial Engineering*, 8(6): 762–788.
- Juan, A., Faulin, J., Grasman, S., Rabe, M., Figueira, G. 2015. "A review of Simheuristics: extending metaheuristics to deal with stochastic optimization problems". *Operations Research Perspectives*, 2: 62–72.
- Juan, A., Faulin, J., Jorba, J., Riera, D., Masip, D., Barrios, B., 2011. "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* 62(6), 1085-1097.
- Juan, A., Pascual, I., Guimarans, D. and Barrios, B. 2015. "Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem". *International Transactions in Operational Research*, 22 (4): 647–667. doi:10.1111/itor.12101
- Lee, Y. H., Kim, J.I., Kang, K.H. and Kim, K.H. 2008. "A Heuristic for Vehicle Fleet Mix Problem Using Tabu Search and Set Partitioning." *Journal of the Operational Research Society* 59 (6): 833–41.
- Lin, S. 1965. "Computer Solutions of the Travelling Salesman Problem." *Bell System Technical Journal* 44: 2245–69.
- Liu, F-H, and Shen, S-Y. 1999. "The Fleet Size and Mix Vehicle Routing Problem with Time Windows." *Journal of the Operational Research Society* 50 (7): 721–32.
- Morán-Mirabal, L.F., González-Velarde, J.L., Resende, M. 2014. "Randomized heuristics for the family traveling salesperson problem." *International Transactions in Operational Research* 21, 41–57.

- 1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
- Osman, I., and Wassan, N.. 2002. "A Reactive Tabu Search for the Vehicle Routing Problem with Backhauls." *Journal of Scheduling* 5, no. 4: 263–85.
- Parragh, S.N., Doerner, K. F. and Hartl, R. F. 2008. "A Survey on Pickup and Delivery Problems: Part II: Transportation between Pickup and Delivery Locations." *Journal Fur Betriebswirtschaft* 58 (2): 81–117.
- Quintero-Araujo, C. L., Caballero-Villalobos, J. P., Juan, A. A. and Montoya-Torres, J. R. 2016. "A biased-randomized metaheuristic for the capacitated location routing problem." *International Transactions in Operational Research*. doi:10.1111/itor.12322
- Reimann, M., and Ulrich, H. 2006. "Comparing Backhauling Strategies in Vehicle Routing Using Ant Colony Optimization." *Central European Journal of Operations Research* 14 (2): 105–23.
- Reis, R., Ritt, M., Buriol, L.S., Resende, M. 2011. "A biased random-key genetic algorithm for OSPF and DEFT routing to minimize network congestion." *International Transactions in Operational Research* 18, 401–423.
- Renaud, J., and Boctor, F. F. 2002. "A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem." *European Journal of Operational Research* 140 (3): 618–28.
- Resende M.G. and Ribeiro, C.C. "Greedy randomized adaptive search procedures: Advances, hybridizations, and applications." In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 283–319. Springer US, 2010.
- Ropke, S., and Pisinger, D. 2006. "A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls." *European Journal of Operational Research* 171 (3): 750–75.
- Salhi, S., Wassan, N. and Hajarat, M. 2013. "The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-Based Heuristics." *Transportation Research Part E: Logistics and Transportation Review* 56 (September): 22–35.
- Santos, H. G., Toffolo, T. A.M., Silva, C. L.T.F. and Vanden Berghe, G. 2016. "Analysis of stochastic local search methods for the unrelated parallel machine scheduling problem." *International Transactions in Operational Research*. doi:10.1111/itor.12316
- Sheu, J-B., and Talley, W. K. 2011. "Green Supply Chain Management: Trends, Challenges, and Solutions." *Transportation Research Part E: Logistics and Transportation Review* 47 (6). Elsevier Ltd: 791–92.

- 1
2
3
4 Subramanian, A., and Penna, P.. 2012. "A Hybrid Algorithm for the Heterogeneous Fleet Vehicle
5 Routing Problem." *European Journal of Operational Research* 221 (3):513-20.
6
7 Taillard, E. D. 1999. "A Heuristic Column Generation Method for the Heterogeneous Fleet VRP."
8 *RAIRO Recherche Operationnelle* 33 (1): 1–14.
9
10 Tarantilis, C. D., Kiranoudis, C. T., and Vassiliadis, V. S. 2003. "A List Based Threshold Accepting
11 Metaheuristic for the Heterogeneous Fixed Fleet Vehicle Routing Problem." *Journal of the*
12 *Operational Research Society*. 54 (1): 65–71.
13
14 Toth, P., and Vigo, D. 1997. "An Exact Algorithm for the Vehicle Routing Problem with Backhauls."
15 *Transportation Science* 31 (4): 372–85.
16
17 Toth, P., and Vigo, D. 1999. "A Heuristic Algorithm for the Symmetric and Asymmetric Vehicle Routing
18 Problems with Backhauls." *European Journal of Operational Research* 113 (3): 528–43.
19
20 Toth, P., Vigo, D. 2014. *Vehicle Routing: Problems, Methods, and Applications*. 2a ed. Volume 18 of
21 *MOS-SIAM Series on Optimization*.
22
23 Tütüncü, G. 2010. "An Interactive GRAMPS Algorithm for the Heterogeneous Fixed Fleet Vehicle
24 Routing Problem with and without Backhauls." *European Journal of Operational Research* 201 (2).
25 Elsevier B.V.: 593–600.
26
27 Tütüncü, G, Carreto, C. and Aker, B. 2009. "A Visual Interactive Approach to Classical and Mixed
28 Vehicle Routing Problems with Backhauls." *Omega* 37 (1): 138–54.
29
30 Ubeda, S., Arcelus, F. J. and Faulin, J. 2011. "Green Logistics at Eroski: A Case Study." *International*
31 *Journal of Production Economics* 131 (1): 44–51.
32
33
34
35
36
37
38 Wassan, N. 2007. "Reactive Tabu Adaptive Memory Programming Search for the Vehicle Routing
39 Problem with Backhauls." *Journal of the Operational Research Society* 58 (12): 1630–41.
40
41 Wassan, N., Salhi, S., Nagy, G. 2009. "Solving the Mixed VRP with Backhauling Using Ants." In *AIP*
42 *Conference Proceedings* 1159, 248–53.
43
44
45 Zachariadis, E., and Kiranoudis, C. T. 2012. "An Effective Local Search Approach for the Vehicle
46 Routing Problem with Backhauls." *Expert Systems with Applications* 39 (3): 3174–84.
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Using multi-start biased randomization of heuristics to solve the vehicle routing problem with clustered backhauls

Javier Belloso¹, Angel A. Juan², Javier Faulin¹ and Adrián Serrano¹

¹ Public University of Navarra, 31006 Pamplona, Spain

{javier.belloso, javier.faulin, adrian.serrano}@unavarra.es

² Open University of Catalonia, 08018 Barcelona, Spain

ajuarp@uoc.edu

Proc. ICAOR 2015
Vienna, Austria

Abstract

Keywords:

Biased randomization

Clustered backhauls

Heuristics

Multi-start algorithms

Vehicle routing problem

We consider the Vehicle Routing Problem with Backhauls (VRPB), where delivery and pick-up customers are to be served from a central depot. In particular, the group or cluster of delivery customers has to be served before the first pickup customer can be visited. Thus, the problem belongs to a sub-class called VRP with Clustered Backhauls (VRPCB). Our resolution procedure uses a multi-start approach designed to avoid the local minima and to be easily parallelizable. The algorithm employs a biased-randomized version of the classical savings heuristic, together with some local search processes. During the solution-construction process, the edges that connect one delivery customer with a pick-up customer are penalized to be chosen at a later stage. The savings list of edges is randomized using a skewed probability distribution. Some classical benchmark instances for the VRPB were selected in order to compare the efficiency of our approach.

Este artículo ha sido eliminado por restricciones de derechos de autor.