Randomized Algorithms for Rich Vehicle Routing Problems: From a Specialized Approach to a Generic Methodology

José de Jesús Cáceres Cruz

Department of Computer Science, Multimedia and Telecommunication,
Universitat Oberta de Catalunya - Internet Interdisciplinary Institute
(UOC-IN3)

Directors: Dr. Angel A. Juan P. & Dr. Daniel Riera T.
Commitee Member: Dr. Helena R. Lourenço

PhD. Thesis
Information and Knowledge Society Doctoral Programme
Network and Information Technologies

July 2013

Abstract

The Vehicle Routing Problem (VRP) is a well known domain in optimization research community. Its different basic variants have been widely explored in the literature. Some studies have considered specific combinations of real-life constraints to define the emerging Rich VRP scopes. This work deals with the integration of heuristics, biased probability, simulation, parallel & distributed computing techniques, and constraint programming. The proposed approaches are tested for solving some variants of VRPs, namely, first, the deterministic families: Heterogeneous VRP (HVRP), Heterogeneous VRP with Variable cost (HVRP-V), Heterogeneous fleet VRP with Multi-trips (HVRPM), Asymmetric cost matrix VRP (AVRP), Heterogeneous fleet with Asymmetric cost matrix VRP (HAVRP), VRP with Time Windows (VRPTW), and Distance-Constrained VRP (DCVRP); second, the stochastic nature families: VRP with Stochastic Demands (VRPSD), and Inventory Routing Problem with Stochastic Demands (IRPSD). An extensive literature review is performed for all these variants, focusing on the main contributions of each work. A first approach proposes a biased-randomization of classical heuristics for solving the deterministic problems addressed here. A second approach is centered on the combination of randomized heuristics with simulation (*Simheuristics*) to be applied on the commented stochastic problems. Finally, a third approach based on the joined work of randomized heuristics with constraint programming is proposed to solve several types of routing problems. The developed heuristic algorithms are tested in several benchmark instances —between these, two real-life case studies in Spain are considered— and the results obtained are, on average, highly promising and useful for decision makers.

Keywords: Rich Vehicle Routing Problems, Biased Randomized Heuristics, Metaheuristics, Real-Life Applications, Optimization, Logistics.

Resumen

El Problema de Enrutamiento de Vehículos (VRP) y sus diferentes variantes básicas son un dominio ampliamente estudiado en la comunidad científica de optimización. Algunos estudios han utilizado combinaciones específicas de restricciones encontradas en la vida real para definir los emergentes VRP Enriquecidos. Este trabajo aborda la integración de heurísticas, probabilidad sesgada, simulación, técnicas de computación distribuida & paralelas, y programación con restricciones. Los enfoques propuestos han solucionado algunas variantes del VRP: en primer lugar, las familias deterministas: VRP con flotas Heterogéneas (HVRP), VRP con flotas Heterogéneas y costo variable (HVRP-V), VRP con flota Heterogénea y Múltiples viajes (HVRPM), VRP con matriz de costo Asimétrica (AVRP), VRP con flota Heterogénea y matriz de costo Asimétrica (HAVRP), VRP con ventanas de Tiempo (VRPTW), y VRP Distancia limitada (DCVRP); en segundo lugar, las familias de naturaleza estocástica: VRP con Demandas estocásticas (VRPSD), y Problemas de Inventario y Enrutamiento de Vehículos con Demandas estocásticas (IRPSD). Una extensa revisión bibliográfica se ha realizado para cada una de estas variantes. Un primer enfoque propone la combinación de una aleatorización sesgada con heurísticas clásicas para la solución de problemas deterministas. Un segundo enfoque se centra en la combinación de heurísticas aleatorias con simulación (*Simheuristics*) para ser aplicados sobre los problemas estocásticos comentados. Por último, se propone un tercer enfoque basado en el trabajo conjunto de heurísticas aleatorias con programación de restricciones para resolver varios tipos de problemas de enrutamiento. Los algoritmos heurísticos desarrollados han sido aplicados en varios casos de referencia —entre ellos, dos estudios de casos reales de distribución en España— y los resultados obtenidos son, en general, prometedores y útiles para los decisores.

Palabras claves: Problemas Enriquecidos de Enrutamiento de Vehículos, Heurísticas Aleatorias y Sesgadas, Metaheurísticas, Aplicaciones Reales, Optimización, Logística.

To my lovely family and mainly to next generations
(Mia, Mathias and Jorge Alejandro) who would have
more opportunities and the duty of making the right decisions.

To all dreams of latino-american students.

In memory of my father.

Aimer, c'est agir (Victor Hugo).

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisors Dr. Angel Juan and Dr. Daniel Riera for their continuous support of my research, for his patience, motivation, enthusiasm, and knowledge. I could not have imagined having a better mentors for my Ph.D study. Besides my directors, I would like to thank my external advisor: Dr. Helena R. Lourenço for her encouragement, insightful comments, and challenging remarks. To the Doctoral Grant Programme of **UOC-IN3** and all people inside of this institution: Josep Llados, Jordi Ferran, Yolanda González, Lin Win, Isabel Egea, Monserrat Mir, Felisa Cabeza, Andrea, Alejandra Pérez, etc. Thanks to all PhD candidates of the programme (new and old generations) to share the good and hard moments of this path: Jordi, Denise, Debora, Nadia, Guillem, Sergio, Ola, Marc, Cecilia, etc. Particularly, thanks to Enosha and Barry for helping me with patience to improve my english. Also thanks to all scientific collaborators who help me to develop the content of this thesis, specially to Daniel Guimarans and Pol Arias to show me the path of constraint programming.

The work in this thesis has been also part of the MICINN project with reference TRA2010-21644-C03-02 ("Algorítmos y software distribuido para el Diseño de Rutas Óptimas en PYMES") of my research group **DPCS** for the "Plan Nacional I+D+i 2011-2013", the **CYTED-HAROSA** Network (CYTED2010-511RT0419), in the context of the **IN3-ICSO** program and the **IN3-HAROSA** Knowledge Community (`http://dpcs.uoc.edu`). Also thanks to contacted people in distribution companies involved on the studies of this thesis for their time and cooperation.

Gracias a toda mi familia quienes siempre me han brindado incondicionalmente su apoyo y cariño para recorrer este exigente camino de la academia:

mi madre Margarita; tias Thialy, Mima, Noris y Thamara; tios Carlos, Roque, Goyo y Luis; primos Jesus Enrique, Lucho y Carlos; y primas Dayana y Oriana; gracias por siempre estar ahí y enseñarme que el espíritu del venezolano es luchador y generoso por naturaleza propia. Gracias en particular a la familia Reyes-Zuloaga (en especial a Marcos) por todo el apoyo brindado durante el desarrollo del doctorado.

Gracias a todos mis amigos esparcidos entre Europa y América. A esos amigos venezolanos que me han ayudado a extender el concepto de familia a nuevos límites: Victor, Roselyn, Alejandro y Aleana. A los catalanes por recibirme en su tierra (Albert). En especial a mis compañero del voleyball en **Panteres Grogues de Barcelona** quienes con su dinamismo y espíritu entusiasta me han concedido un espacio lo suficientemente abierto y diverso como para mantener mi mente y cuerpo en sano equilibrio durante mi estadía en la ciudad de las grandes obras de Gaudí. Félix, gracias por diseñar la portada.

Merci á mes cher amis en France: Edwin, Gloria, Alfonso, Quentin et Jan, pour toujours me donner de l'énergie et de la espoir pour suivre ce long parcours.

Contents

List of Figures

List of Tables

Abbreviation	Description
ACO	Ant Colony Optimization
ACS	Ant Colony System
ALNS	Adaptive Large Neighborhood Search
AVRP	Asymmetric cost matrix Vehicle Routing Problem
BKS	Best Known Solution
BPP	Bin Packing Problem
CG	Column Generation
COP	Combinatorial Optimization Problem
CP	Constraint Programming
CSP	Constraint Satisfaction Problem
CVRP	Capacitated Vehicle Routing Problem
CWS	Clarke-and-Wright Savings Heuristic
DCVRP	Distance-Constrained Vehicle Routing Problem
DPCS	Distributed- and Parallel-Computing Systems
FSMVRP	Fleet Size and Mix Vehicle Routing Problem
GRASP	Greedy Randomized Adaptive Search Procedure
HAROSA	Hybrid Algorithms for solving Realistic rOuting, Scheduling and reliability/Availability problems
HAVRP	Heterogeneous fleet with Asymmetric cost matrix Vehicle Routing Problem
HBSS	Heuristic-Biased Stochastic Sampling
HVRP	Heterogeneous fleet Vehicle Routing Problem
HVRP-V	Heterogeneous fleet Vehicle Routing Problem with Variable cost minimization
HVRPM	Heterogeneous fleet Vehicle Routing Problem with Multiple trips
ICT	Internet and Communication Technologies
ILS	Iterated Local Search
IRPSD	Inventory Routing Problem with Stochastic Demands
LNS	Large Neighborhood Search
LS	Local Search

Table 1: Summary of abbreviations used on this dissertation.

Abbreviation	Description
MCS	Monte-Carlo Simulation
MDVRP	Multi-Depots Vehicle Routing Problem
MILP	Mixed Integer Linear Programming
MIRHA	Multi-start biased Randomization of classical Heuristics with Adaptive local search
OVRP	Open Vehicle Routing Problem
OBS	Our Best found Solution
PDCS	Parallel and Distributed Computing Systems
PDVRP	Pickup-and-Delivery Vehicle Routing Problem
PVRP	Periodic Vehicle Routing Problem
RVRP	Rich Vehicle Routing Problem
RC	Retail Center
RNG	Random Number Generator
SAM	Successive Approximations Method
Simheuristic	Simulation-optimization using Heuristics
SR-GCWS	Simulation in Routing via the Generalized Clarke-and-Wright Savings heuristic
SR-GCWS-CS	SR-GCWS with Cache and Splitting local searches
SVRP	Site-Dependent Vehicle Routing Problem
SDVRP	Split-Delivery Vehicle Routing Problem
SME	Small- and Medium-Enterprises
SP	Set Partitioning
T&L	Transportation and Logistics
TS	Tabu Search
VMI	Vendor Managed Inventory
VNS	Variable Neighborhood Search
VRP	Vehicle Routing Problem
VRPB	Vehicle Routing Problem with Backhauls
VRPM	Vehicle Routing Problem with Multitrips
VRPSD	Vehicle Routing Problem with Stochastic Demands
VRPTW	Vehicle Routing Problem with Time Windows

Table 2: Summary of abbreviations used on this dissertation (*continuation*).

Symbol	Description
Ω	Set of customers, vertices or nodes including the depot
$\Omega*$	Set of customers, vertices or nodes without the depot
n	Total number of nodes on a data instance
i, j, u	Given nodes on a data instance
A	Set of arcs or edges between all nodes
d_{ij}	Distance (eucledian) between nodes i and j
c_{ij}	Cost associated to the connected arc between nodes i and j
q_i	Demand of node i
$E[D_i]$	Expected random demand of node i
D_i	Random variable for the aggregated demand of node i
d_i	Deterministic (mean) demand of node i
y_{ij}	Vehicle load arriving at node j after visiting node i
x_{ij}^k	Binary variable indicating that arc (i, j) is used by a vehicle type k
R	Set of routes in the generated routing solution
M	Number of routes in the generated routing solution
k, l	Given vehicles of types k and l
Q	Capacity of available homogeneous vehicle fleet
Q_k	Capacity of vehicle type k
Q'_k	Theoretical capacity of vehicle type k
K	Total available number of vehicles at the depot
m_k	Available number of vehicles of type k at the depot
MQ_k	Aggregated total capacity of vehicles of type k
AM_k	Allowed number of multiple trips for vehicles of type k
MQM_k	Aggregated total capacity of vehicles of type k considering multiple trips
F_k	Fixed cost of using a vehicle of type k
γ_k	Factor cost per unit of distance associated to vehicles of type k
c_{ij}^k	Cost associated to the connected arc between clients i and j using a vehicle type k
L_i	Current inventory level on node i
$\hat{L}_i$	Maximum allowable inventory level on node i
L_i^*	Chosen inventory level (delivery policy) for node i
b_{i_j}	Waiting time to begin the service at node i given that j is on the route
b_i	Waiting time to begin the service at node i
l_i	Lastest time for service time windows on node i
s_i	Number of time units for serving on node i

Table 3: Mathematical notation used in this dissertation.

1

Introduction

Transportation & Logistics (T&L) issues have a major economic and environmental impact in most countries and regions over the world. For instance, the EU land transport policy aims at promoting a "sustainable mobility that is efficient, safe and with reduced negative effects on the environment" (Janic, 2006; Steg and Gifford, 2005; Whiteing and Stantchev, 2008). Several international organizations have developed projects for transportation optimization. Likely since 2011, the Inter-American Development Bank has supported for programs to modernize logistical and freight transport systems in several countries of south- and central-america (Bate, 2012; Constance, 2011; Funez, 2012).

Road transportation is the predominant way of transporting goods in Europe and in other parts of the world. Direct costs associated with this type of transportation have increased significantly since 2000, and more so in recent years due to rising oil prices. Furthermore, road transportation is intrinsically associated with a good deal of indirect or external costs, which are usually easily observable congestion, contamination, security- and safety-related costs, mobility, delay time costs, etc. However, these costs are usually left unaccounted because of the difficulty of quantifying them (Kumares and Labi, 2007). For example, traffic jams in metropolitan areas constitute a serious challenge for the competitiveness of European industry: according to some studies (Bastiaans, 2000), external costs due to traffic jams could represent about 2% of the European GDP, a percentage which continues to increase. In addition to these easily observable costs, many others might be considered. In this scenario, it becomes evident that new methods must be developed to support the decision-making process so that optimal (or quasi-optimal) strategies can be chosen in road transportation. This necessity for optimizing the road transportation affects both the public and private sectors, and constitutes a major challenge for most industrialized regions.

Recent advances on Information and Communications Technologies (ICT) —such as the growing use of GPS and smart-phone devices, Internet-scale (distributed) systems, and Internet computing technologies—, open new possibilities for optimizing the planning process of road transportation (Orozco, 2011). In particular, when combined with advanced Simulation and Optimization techniques, Distributed- and Parallel-Computing Systems (DPCS) allow the practical development and implementation of new ICT-based solutions to support decision-making in the T&L arena. ““Real-world applications, both in North America and in Europe, have widely shown that the use of computerized procedures generates substantial savings (generally from 5% to 20%) in the global transportation costs” (Toth and Vigo, 2001).” Road-transportation optimization (cost-saving) issues are especially critical in the case of Small and Medium Enterprises (SME), since they are rarely able to obtain the economic and human resources required to implement, maintain, and manage efficient routing-optimization methods. Similarly, those companies have difficulties to access the appropriate technologies —e.g., computer clusters and expensive commercial software—, which would help them to improve their productivity level and to reduce the unnecessary costs, thus making a more sustainable business model.

In this context, the goal of the so-called Vehicle Routing Problem (VRP) is to *optimize* the routing design (distribution process from depots to customers) in such a way that customers' demand of goods is satisfied without violating any problem-specific constraint —e.g., route maximum distance or time-related restrictions (Golden et al., 2008). The VRP has many variants depending on the parameters and constraints considered. Its most basic variant is the so-called Capacitated Vehicle Routing Problem (CVRP). The CVRP assumes the existence of a homogeneous fleet (same capacity for all vehicles) and a central storehouse. It also assumes that customers' demands are given in advance. Even in its apparent simplicity, the CVRP is a combinatorial explosion problem. This implies that, in practice, it will not be possible to guarantee the (mathematically) optimal solution except in the case of *small* problems with no more than 75 customers. Here is where heuristic and metaheuristic algorithms can make an outstanding contribution by providing quasi-optimal solutions, in reasonable computing times, for medium- and large-scale problems and even when considering real-life constraints. Notice, however, that as a general rule, the closer the VRP constraints are to real-life scenarios the more difficult it is to obtain quasi-optimal solutions. Unfortunately, real-world T&L environments are complex and rich in nature. In recent years, due to the fast development of new and more efficient optimization and computing methods, the interest of academics and practitioners has been shifting towards realistic VRP variants, which are commonly known as Rich VRP. These problems deal with realistic (and sometimes multi-objective) optimization functions, uncertainty (e.g., stochastic or fuzzy behaviour), dynamism, along with a wide variety of real-life constraints related to time and distance factors, use of heterogeneous fleets, linkage with inventory and scheduling problems, integration with ICT, environmental and energy issues, etc. Of course, there exists commercial software which has been developed to support transportation companies when designing their routing (distribution) plans (Drexl, 2012). However, these tools do not satisfy all the routing requirements of SME, they usually require some experts' support, and they can be unaffordable for all except the largest corporations.

In most existing works, the core optimization task is mainly focused in the minimization of time, costs, CO_2 emissions, and risk. Alternatively, it is focused on the maximization of profit, quality, and efficiency (Talbi, 2009). Since most real-life optimization problems are complex and difficult to solve, many researchers have approached transportation problems by developing efficient heuristics and metaheuristics. Following these trends, a wide set of randomized algorithms have promoted and published (Faulín and Juan, 2008; González et al., 2010; Juan et al., 2009, 2010). These algorithms, which combine simulation-optimization, heuristics, and computer-parallelization techniques, have been able to efficiently solve several VRP variants. Accordingly, the main goal of this thesis is the development of new open-source, hybrid, and randomized algorithms and methods which provide efficient support to decision-making in the Rich VRP context. As a consequence, it is expected that these algorithms can be potentially interesting not only for the academic community but also for real SME in the T&L business sector.

1.1 Structure of this Thesis

This thesis discusses several issues concerning the Rich Vehicle Routing Problem (RVRP). The general presentation will be focusing on providing the reader with a theoretical basis for studying the RVRP. Also it provides the practitioner with the implementation of tailored techniques as well as generic solution methods for solving the RVRP. A substantial portion of the problem data in a RVRP is subject to deterministic sources. Uncertainty is a real feature demanded by real-life companies scenarios. In fact, it is hard to consider into the optimization models and approaches. For this, we use this feature as a primary division of the approaches developed in this thesis. So we propose a broad division of four blocks for grouping chapters:

- *Block I: Introduction, the classical VRP and its applied methodologies, and finally the Rich VRP context* (chapters 1–4). In this block, the relevance of the road transportation is discussed. Also some VRP methodologies are introduced. The classical Vehicle Routing Problem (VRP) and its Rich counterpart —the RVRP— are introduced and we give a discussion of the differences between them. We provide a survey of the existing literature dealing with the RVRP. The main goal of this block is to provide the reader with a consistent overview of the work on the RVRP and the progress made within this area throughout the past 15 years.

- *Block II: Tailored approaches for some deterministic VRPs* (chapters 5–8). In this block, we discuss how to deal with some deterministic cases (like HVRP, HVRPM, AVRP, HAVRP and VRPTW) when analyzing VRP and RVRP scenarios using biased-randomized solution techniques. Also a detailed literature review of specific studied variants is provided.

- *Block III: Tailored approaches for some stochastic VRPs* (chapters 9–11). In this block, we consider stochastic variables in the resolution of RVRP scenarios (like the VRPSD and IRPSD) using simple simulation techniques. Some solution techniques for the RVRP cases with Stochastic Demands are provided.

- *Block IV: Generic Approach for Rich VRPs* (chapters 12–13). In this block, the creation of a generic framework based on constraint programming is discussed. This can solve some variants of RVRP without an additional coding phase. A generic approach is then developed and preliminary tested to illustrate its performance against tailored techniques.

Finally, in the last chapter we give our conclusions in a brief summary of the discussions of this thesis, as well as the importance of knowledge transfer to SME, and a list of the scientific contributions included in this dissertation.

In all these chapters we could appreciate the adaptation of some heuristics to different routing contexts and its constraints. We have explored several integration of heuristics, simulation, biased probability, parallel and distributed computing, and constraint programming. The application of proposed methodologies have allowed to solve two real-life enterprise cases and some other theoretical known instances. On these instances, several phases of the supply-chain were addressed which offers useful and fast tools to the decision-maker. Some quantitative methods were used to analyze the generated results where remarkable savings on distance, money and time were obtained. The global study developed on this dissertation can be summarized in the context of Rich VRP. A large and detailed literature review of the evolution of this emerging research line is presented. The studies related to this optimization line have the particular feature of being inspired on real-life situations where an enterprise is interested on applying new advanced techniques to solve a given problem with complex constraints. The addressed Rich VRPs cover a set of both deterministic and stochastic routing problems. This is a major contribution of this dissertation since few studies have covered both of these routing optimization families. The way we pretend to design algorithms for Rich VRPs is proposing several methodologies mainly based on biased-randomized heuristics.

1.2 Relevance of this Topic

Transportation has had a key role in human history. It is related to migrations, economic development, military moves, etc. Since the XX century, the development of technology has changed this sector forever. The real time of routing planning and all available information associated to demands, locations, times, etc. has created new opportunities to optimize. Some numbers related to the last ten-twelve years will help to contextualize the current importance of transportation. As discussed in the Introduction, Vehicle Routing —as part of the supply chain process (see Fig. 1.1)— is one

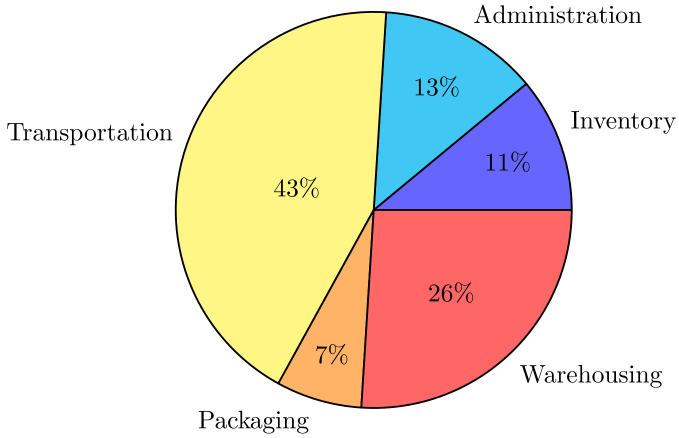


Figure 1.1: Cost of Logistic Activities as a Percentage of Total Logistics Costs (Source: Differentiation for Performance: Excellence in Logistics, 2004, ELA/AT Kearney).

of the most important and complex activities in modern economies. Being a complex combinatorial problem, efficient solving of VRP instances can only be attained by combining knowledge from different areas such as Computer Science, Operations Research, and Applied Mathematics. A successful planning of this activity might result in significant cost reductions and higher service levels to the customer (see Fig. 1.2). However, real-life vehicle routing involves a wide range of variables, uncertainty, and complex constraints. Therefore the Rich VRP is an emerging research area which constitutes a relevant topic for current researchers and practitioners (Drexl, 2012).

Most of the works in the literature are focused on theoretical analysis (Laporte, 2007, 2009). Many real-life instances are unsolved and a great interest is growing up between public and private sectors to invest in this kind of studies. In addition, the use of hybrid algorithms and new computing paradigms —e.g., the use of GPUs or the use of Internet-scale computing— are changing the research scenario and new work lines have been created (Crainic, 2008; Crainic and Toulouse, 2003; Talbi, 2012). During the last three years, several companies have manifested their interest in the potential applications for the routing optimization algorithms, among others: Tech Ideas, Evolution Algorithms (Logisplan), Corporación Alimentaria de Guissona, Eptisa, and ITENE (Instituto Tecnológico del Embalaje, Transporte y Logística).

In the EU17 countries, the turnover of freight transport by road represent an important percentage of the national turnover (see Fig. 1.3). In particular, Spain has

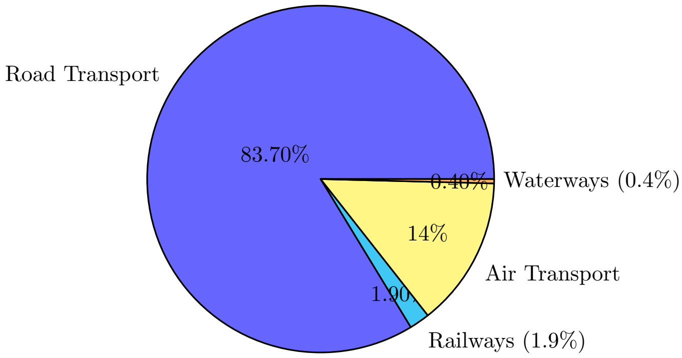


Figure 1.2: Total External Costs in 2000 of Transportation in UE17 Countries (Source: INFRA/IWW 2004, Germany-Switzerland).

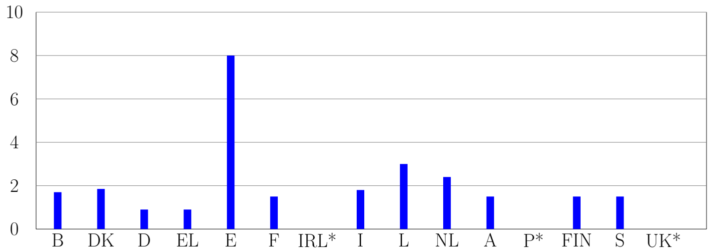


Figure 1.3: Turnover of freight transport by road as a percentage of national turnover (Source: EU road freight transport sector; Work and employment conditions. European Foundation for the Improvement of Living and Working Conditions, 2004) [* No information available].

the highest level of this percentage. For instance, in Table 1.1, we can appreciate the road transportation portion against other types of transport inside of Spain for 2009. Once again the road transportation sector takes the greatest value which represent a remarkable sector to be optimized and all aspects waterfalls down. There are many real applications where the transport optimization represents a significant saving — i.e., logistic, retailing, bottle distribution, garbage collection, food production, among others. In general, the research community states that using advanced techniques in routing-distribution could improve this current context (Laporte, 2009).

	Rail	Road	Boat	Airplane
Number of Enterprises	10	134,915	80	22
Turnover (thousands of Euros)	2,289,659	33,108,840	935,270	199,151
Employed persons (annual average)	20,770	382,070	2,888	821

Table 1.1: Main magnitudes by type of transportation sectors (Source: Service Annual Survey, CNAE-2009, National Institute of Statistics, Spain).

1.3 Objectives

In general, a desirable or efficient optimization algorithm for a VRP context should be able to generate results in a short period of time (seconds or minutes); produces good quality solutions; is simple to configure; flexible to be adapted to new constraints or new computing architectures; and easy to understand/explain to other researchers (Cordeau et al., 2002). Therefore these can be categorized as the main requirements of any VRP algorithm. The main goal of this research is to develop hybrid randomized algorithms and methods which combine simulation-optimization, heuristics, and computing techniques in order to efficiently support decision-making processes in the Rich VRP arena. To reach this general goal, some specific objectives are considered:

- To design, implement, and test (validate) new hybrid randomized algorithms for solving different variants of Rich VRPs. These algorithms will combine simulation-optimization methods with heuristics and metaheuristics.

- To adapt the developed algorithms so that they can benefit from parallel-computing, multi-agent, and other related techniques. This, in turn, will contribute to significantly reduce the wall-clock time necessary to obtain high-quality solutions to Rich VRP instances.

- To promote the knowledge transfer to SME, so that they can improve their competitiveness by using these algorithms when designing their road distribution planning.

Our main hypothesis is to check if it is possible to develop efficient techniques for a broad set of VRPs using randomized methods. So the main idea is to extend the VRPs to more *rich* ones, as originally proposed Toth and Vigo (2001). In fact, this has been the main direction for the Operation Research field, as the increasing number

Authors	Year	Problem	Number of instances	Number of requests
Christofides and Eilon	1969	CVRP	3	25–100
Christofides et al.	1979	CVRP	14	50–100
Golden et al.	1984	HVRP	20	12–100
Solomon	1987	VRPTW	168	25–100
Fisher	1994	CVRP	3	45–135
Fischetti et al.	1994	AVRP	20	10–300
Augerat et al.	1995	CVRP	74	16–101
Golden et al.	1998	CVRP-DCVRP	20	200–480
Taillard	1999	HVRP	8	50–100
Li and Lim	2001	VRPTW	56	200
Prins	2002	HVRP	20	100
Olivera and Viera	2007	VRPM	104	50–100
Li et al.	2007a	HVRP	5	200–360
Rodríguez and Ruiz	2012	AVRP	540	50–500

Table 1.2: Some VRP Benchmarks.

of papers could confirm (Golden et al., 2008; Laporte, 2009). However the randomized features have an intrinsic potential that could add some interesting solutions to the state-of-the-art in this field. One important sub-objective of this study is to make a literature review on each routing variant addressed.

Some issues could be found when a Rich VRP approach is developed, like not having data to execute tests. Many studies use real data provided by distribution companies even when this used to be private or hard to access. However, some studies proposed the generation of instances following random aspects or specific ones. In Lahyani et al. (2011), the authors analyze different design factors for instances on the context of Rich VRP with heterogeneous fleet, time windows and multiple products. They test an instance generator with an exact method in order to help companies to identify best policies. In our case, real data from several interested enterprises will be used for testing the performance of developed algorithms. Also several well-known and public benchmarks (see Table 1.2) will be used to test the proposed techniques on this dissertation. Each of these benchmarks have been developed for a specific VRP branch. In the literature, some of these have been solved using methods inspired on heuristic, meta-heuristics, exact methods, hybrids, etc. as we will appreciate on next chapters.

1.4 Chapter Conclusions

In this first chapter, we have defined the context and motivation of this thesis. We have presented the relation of routing optimization contexts to human economy activities and its impact on different sectors. Also the research objectives of this study on its limited scope have been pointed out. Next chapters will help to understand the academic definition of routing optimization, the mathematical notation or modelling of the VRP as well as its most important variants in the literature.

2

Capacitated VRP

In the Capacitated Vehicle Routing Problem (CVRP), first defined by Dantzig and Ramser (1959), a homogeneous fleet of vehicles supplies customers using resources available from a depot or central node (see Fig. 2.1). Each vehicle has the same capacity (homogeneous fleet) and each customer has a certain demand that must be satisfied. Additionally, there is a cost matrix that measures the costs associated with moving a vehicle from one node to another. These costs usually represent distances, travelling times, number of vehicles employed or a combination of these factors.

2.1 Definition

More formally, we assume a set Ω of $n + 1$ nodes, each of them representing a vehicle destination (depot node) or a delivery point (demanding node). The nodes are numbered from 0 to n, node 0 being the depot and the remaining n nodes are the delivery points ($\Omega * = \Omega - \{0\}$). A demand $q_i > 0$ of some commodity has been assigned to each non-depot node i ($1 \leq i \leq n$). In the other hand, $A = \{(i, j)/i, j \in \Omega; i < j\}$ represents the set of the $n \cdot (n + 1)/2$ existing edges connecting the $n + 1$ nodes. Each of these links has an associated aprioristic cost, $c_{ij} > 0$, which represents the cost of sending a vehicle from node i to node j. In this original version, these c_{ij} are assumed to be symmetric ($c_{ij} = c_{ji}, 0 \leq i, j \leq n$), and they are frequently expressed in terms of the Euclidean distance, d_{ij}, between the two nodes. The delivery process is to be carried out by a fleet of K vehicles ($K \geq 1$) with equal capacity, Q >> $\max\{q_i/1 \leq i \leq n\}$.

These K vehicles are responsible of M routes. Some additional constraints associated to the CVRP are the following (Laporte et al., 2000):

- Each non-depot node is supplied by a single vehicle.
- All vehicles begin and end their routes at the depot (node 0).
- A vehicle cannot stop twice at the same non-depot node.
- No vehicle can be loaded exceeding its maximum capacity.

In this generic formulation, useful for both symmetrical and asymmetrical issues as well as for both homogeneous and heterogeneous fleet, $O(n^2K)$ binary variables x are used. This is the main advantage of the three-index model representation proposed by Toth and Vigo (2001) and then used in Baldacci et al. (2008) for the heterogeneous fleet VRP variant. The variable x_{ij}^k indicating the arc (i,j) $(i,j \in \Omega)$ is used or travelled by a vehicle type k $(k \in 1,\dots,K; K \leq M)$ in the optimal solution (2.8). Each vehicle type k has a capacity defined by Q_k, and a number of available vehicles m_k. In addition, there are $O(nK)$ binary variables y. The variable y_{ij} represents the load in the truck arriving at customer j after visiting customer i in terms of units of commodity.

$$\min \sum_{k=1}^{M} \sum_{i \in \Omega} \sum_{j \in \Omega} c_{ij}^k \cdot x_{ij}^k \tag{2.1}$$

subject to:

$$\sum_{i \in \Omega*} x_{i0}^k = \sum_{j \in \Omega* \setminus \{i\}} x_{0j}^k \qquad \forall k \in \{1,\dots,M\} \tag{2.2}$$

$$\sum_{k=1}^{M} \sum_{i \in \Omega} x_{ij}^k = 1 \qquad \forall j \in \Omega* \tag{2.3}$$

$$\sum_{i \in \Omega \setminus \{u\}} x_{iu}^k = \sum_{j \in \Omega \setminus \{u,i\}} x_{uj}^k \qquad \forall u \in \Omega*, \forall k \in \{1,\dots,M\} \tag{2.4}$$

$$\sum_{j \in \Omega*} x_{0j}^k \leq m_k \qquad \forall k \in \{1,\dots,M\} \tag{2.5}$$

$$\sum_{i \in \Omega} y_{ij} + q_j = \sum_{i \in \Omega} y_{ji} \qquad \forall j \in \Omega* \tag{2.6}$$

$$0 \leq q_i \; x_{ij}^k \leq y_{ij} \leq (Q_k - q_j) x_{ij}^k \qquad \forall i,j \in \Omega, \forall k \in \{1,\dots,M\} \tag{2.7}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall i,j \in \Omega, i \neq j, \forall k \in \{1,\dots,M\} \tag{2.8}$$

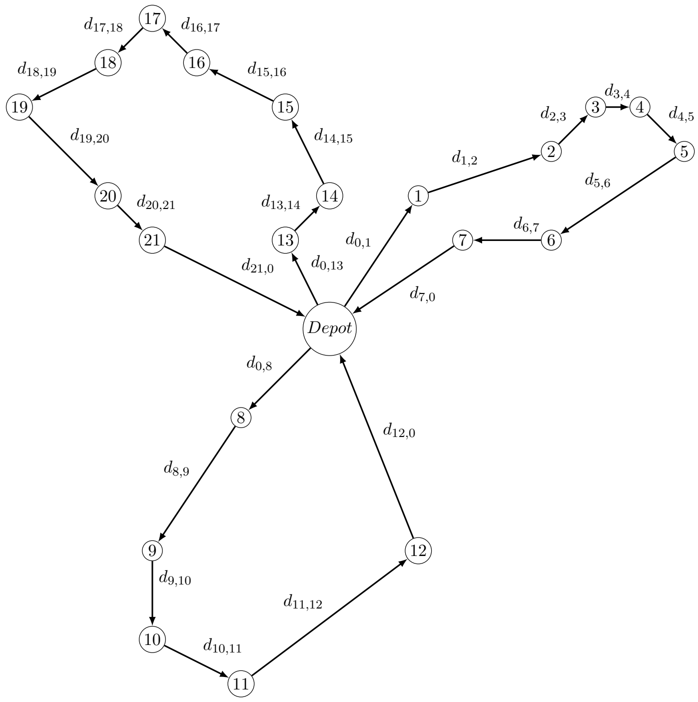


Figure 2.1: Representation of a VRP example where it designs the routes through a group of nodes.

The objective function in Eq. 2.1 minimizes the total cost distance of the arcs used by all M routes generated. Constraint Eq. 2.2 implies that the number of vehicles leaving the depot is the same as the number of vehicles returning to it. Constraint Eq. 2.3 and 2.4 require that each customer is visited exactly once, and that the vehicle k arrives and leaves each h customer location respectively. Constraint Eq. 2.5 imposes that the number of used vehicles does not exceed the number of available vehicles. Constraint Eq. 2.6 states that the quantity of products in the truck arriving at customer j, y_{ij}, plus the demand of that customer, equals the quantity of products in the truck leaving it after the service has been completed. Constraint Eq. 2.7 guarantees lower and upper bounds ensuring that: the quantity of products in the truck leaving customer i, y_{ij}, is equal to or greater that its demand, q_i; and the total demand served by each vehicle k does not exceed the service capacity Q_k. All the mathematical notation used in this dissertation is summarized in the Glossary.

In computational complexity theory, the classical version of VRP and its variants (for extension) are NP-hard (non-deterministic polynomial-time hard). This is a general classification which means that there is no known deterministic algorithm that can solve the problem in a polynomial number of steps (Garey and Johnson, 1978). NP-hard problems may be of any type: decision problems, search problems, or optimization problems. Some practical examples could be found in Data mining, Scheduling, Planning, Decision support, etc. (Lenstra and Rinnooy-Kan, 1981). For more information and definitions related to computational complexity theory, the reader can consult Garey and Johnson (1979).

2.2 VRP Variants

Different variants of the Vehicle Routing Problem (VRP) have been studied in the last fifty years (Laporte, 2009). In the literature, the variants of the VRP include a large family of specific optimization problems. As their main common feature, they are focused in considering one or few constraints into their mathematical models; this has created a huge set of separated branches of VRP research lines with long abbreviation names. Each research line has been identified by the acronym of the considered constraints or attributes inside of the optimization problem. Many of these individual branches have been recombined creating new 'basic' branches. The main variants of the VRP could be found in Golden et al. (2008); Toth and Vigo (2001). So far the most common current extensions studied in the literature are described here:

- *Asymmetric cost matrix VRP* (AVRP): The cost for going from customer a to b is different for going from b to a.

- *Distance-Constrained VRP* (DCVRP): The total length of the arcs in a route cannot exceed a maximum route length. This constraint can either replace the capacity constraint or supplement it.

- *Heterogeneous fleet VRP* (HVRP): The company uses different kinds of vehicles and the routes have to be designed according to the capacity of each vehicle. Some costs could be considered and the number of vehicles could be limited or not creating different contexts. When the number of vehicles is unlimited then it is called Fleet Size and Mix VRP (FSMVRP). If a specific type of vehicle can not reach some clients for any accessible reason then the problem become Site-Dependent VRP (SVRP). Also if a vehicle is allowed to perform more than one trip then we are solving a HVRP with Multiple use of vehicles (HVRPM).

- *Multiple Depots VRP* (MDVRP): A company has several depots from which they can serve their customers. Therefore, some routes will have different starting/ending points.

- *Open VRP* (OVRP): The planned routes can end on several points distinct to the depot location.

- *Periodic delivery VRP* (PVRP): The optimization is done over a set of days. The customers may not have to be visited each day. Customers can have different delivering frequencies.

- *Pickup-and-delivery VRP* (PDVRP): Each customer is associated by two quantities, representing one demand to be delivered at the customer and another demand to be picked up at the customer and returned to the depot. In addition to the constraint that the total pickup and total delivery on a route cannot exceed the vehicle capacity, also it has to ensure that this capacity is not exceeded at any point of the route. One variant of the pickup and delivery problem is when the pickup demand is not returned to the depot, but should be delivered to another customer —e.g., transport of people. In some cases, the vehicles must pickup and deliver items to the same customers in one visit (Simultaneous Pickup-and-delivery VRP) —i.e., new and returned bottles.

- *Split-delivery VRP* (SDVRP): The same customer can be served by different vehicles if it will reduce the overall cost. This relaxation of the basic problem is important in the cases where a customer order can be as large as the capacity of the vehicle.

- *Stochastic VRP*: There is a realistic aspect of the routing problem where a random behaviour is considered. So far, this uncertainty aspect has shown to be a key aspect for future demanding developments. This can be the demand of each customer, if the customer itself is present (VRPSD) or the service or travel times between the customers. This last one is also known as Time-Dependent VRP (TDVRP).

- *VRP with Backhauls* (BVRPB): As in the VRPPD, the customers are divided into two subsets. The first subset contains the linehaul customers, which are customers requiring a given quantity of product to be delivered. The second subset contains the backhaul customers, where a given quantity of inbound product must be picked up. Then all linehaul customers have to be visited before the backhaul customers on a route.

- *VRP with Time Windows* (VRPTW): Each customer is associated with a time interval and can only be served within this interval. In this problem the dimension of time is introduced and one has to consider the travel time and service time at the customers. A set of time windows for each customer could be also considered (VRP with Multiple Time-Windows). Also these time windows could be flexible depending on some extra costs (VRP with Soft Time-Windows).

Several hybrid variants have been created in the literature from these 'basic' variants which are also inspired in real-life scenarios. A large number of VRP acronyms have been developed to refer to these combinations of routing restrictions. However, all these can be encompassed in the larger family of Rich VRP, as we will explain later.

2.3 Chapter Conclusions

In this second chapter, we have presented the basic definition used on routing optimization, the three-index mathematical notation of the VRP as well as its most important variants in the literature. A wide number of routing families have been created in the last years. Thus, new routing family acronyms have been often proposed. However, this different routing scopes can be summarized into a global research area called Rich VRPs. The next chapters will introduce some of the most important proposed methodologies developed for the VRP so far; and also the demanded evolution of VRPs to the Rich VRPs.

3

VRP Methodologies

Different approaches to VRPs have been explored during the last years. These approaches range from the use of pure optimization methods, such as mathematical programming, for solving small- to medium-size problems with relatively simple constraints, to the use of heuristics and metaheuristics that provide near-optimal solutions for medium and large-size problems with more complex constraints. Metaheuristics serve three main purposes: solving problems faster, solving larger problems, and obtaining more robust algorithms.

> They “are a branch of optimization in Computer Science and Applied Mathematics that are related to algorithms and computational complexity theory. Metaheuristics provide *acceptable* solutions in a reasonable time for solving hard and complex problems” (Talbi, 2009).

Even though the VRP has been studied for decades and a large set of efficient optimization methods, heuristics and metaheuristics have been developed (Golden et al., 2008; Laporte, 2007), more realistic or Rich VRP problems —such as the VRP with Stochastic Demands or the Inventory VRP— are still in their infancy. There is a large set of methods applied to the VRP. Following the proposed division of Talbi (2009), this huge family could be preliminary summarized in a balanced tree presented in Fig. 3.1. For practical reasons, only the used techniques are depicted.

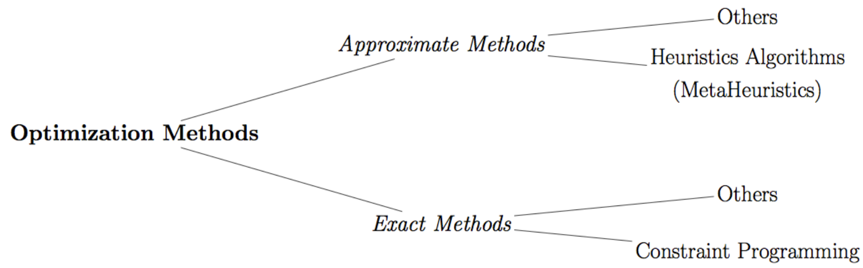


Figure 3.1: Representation of relation of Classical Optimization Methods.

3.1 Exact Methods

From Talbi (2009), "Exact methods obtain optimal solutions and guarantee their optimality". This type of technique is often applied to small-size instances. This family includes a broad set of methods. There are methods like the family of Branch-and-X (where the X represent the different variants) used for solving Integer Linear Programming (ILP) and Mixed Integer Linear Programming problems (MILP); and also Dynamic Programming which focus on solving complex problems by breaking them down into simpler subproblems. Likely, Column Generation is a popular technique used for solving larger linear programming problems, which consists in splitting the given problem into two problems: the master problem and the subproblem (Desaulniers et al., 2005). This allows to simplify the original problem with only a subset of variables in the master problem. While a new variable is created in the subproblem, which will be minimized in the objective function with respect to the current dual variables and the constraints naturally associated to the new variable. The Set Partitioning modelling is another binary variable formulation for each feasible route. This technique is quite general and can consider several constraints at the same time (Subramanian, 2012; Subramanian et al., 2012). In this thesis, we will use Constraint Programming (CP) as a complete checking technique of the feasibility of generated solutions. CP has an intrinsic flexibility to check different complex routing constraints at the same time in a short period of time. So the idea will be to complement other routing techniques with the advantages of CP. This has been wide used in several domains and is based on a tree search combined with logical implications.

3.2 Approximate Methods

From Talbi (2009):

> "Heuristics find *good* solutions on large-size problem instances. They allow to obtain acceptable performance at acceptable costs in a wide range of problems. They do not have an approximation guarantee on the obtained solutions. They are tailored and designed to solve a specific problem or/and instance. Meta-heuristics are general-purpose algorithms that can be applied to solve almost any optimization problem. They may be viewed as upper level general methodologies that can be used as a guiding strategy in designing underlying heuristics".

The author also proposes that two contradictory criteria must be taken into account: exploration of the search space (diversification) and the exploitation of the best solutions found (intensification). Promising regions are determined by the obtained *good* solutions. In the intensification, the promising regions are explored more thoroughly in the hope to find better solutions. In diversification, non-explored regions must be visited to be sure that all regions of the search space are evenly explored and that the search is not confined to only a reduced number of regions.

There are many metaheuristic inspired in natural process like Evolutionary Algorithms (including Genetic Algorithms, GA) and Ant Colony Optimization (ACO). For instance the ACO metaheuristic is inspired from the communication and cooperation mechanisms among real ants that allow them to find short paths from their nest to food sources. The communication medium is a chemical compound (pheromone). The amount of pheromone is represented by a weight in the algorithm (Gendreau et al., 2008). In ACO algorithms, the range $[min, MAX]$ of pheromone trail values can be controlled. This type of technique can be also classified as population-based metaheuristics because they iteratively improve a population of solutions. Another member of this wide group is the deterministic strategy of Scatter Search which recombines selected solutions from a known set to create new ones (Talbi, 2009).

Other techniques are based on memory usage (short-, medium-, and long-term). Tabu Search (TS) is a local search-based metaheuristic where, at each iteration, the best solution in the neighbourhood of the current solution is selected as the new current solution, even if it leads to an increase in solution cost. A short-term memory (the Tabu list) stores recently visited solutions (or attributes) to avoid short-term cycling (Gendreau et al., 2008). This family can considered as single-solution based metaheuristics since they are focused on improving a single solution at a time. A common feature is that all include the definition of building an initial solution. Other promising techniques are Variable Neighbourhood Search (VNS) and Greedy Randomized Adaptive Search Procedure (GRASP). VNS has been wide used in several problems. It is based on a successive exploration of a set of predefined neighbourhoods to find a better solution at each step. Large Neighbourhood Search (LNS) can be interpreted as a special case of VNS where efficient procedures are designed to consider a high number of neighbourhoods at the same time. Inside of this branch, we can find one of the first techniques used for the Travelling Salesman Problem which is the Nearest Neighbour. GRASP will be explained on the next chapter. Simulated Annealing (SA) is other single-solution based method which is based in the process of heating and then slowly cooling of a substance in order to produce a strong crystalline structure. So it is typical to include a temperature element in order to control the process.

There are some approximate algorithms made with a tailored design to solve a specific problem called Heuristics. Following a systematically number of steps, they used to find an acceptable solutions. However, they do not guarantee to find the optimal solution. For instance, Clark-and-Wright Savings (CWS) (Clarke and Wright, 1964) is probably one of the most cited heuristic to solve the CVRP. As the authors propose, this procedure uses the concept of savings. In general, at each step of the solution construction process, the edge with the most savings is selected if and only if the two corresponding routes can feasibly be merged using the selected edge. The CWS algorithm usually provides relatively good solutions in less than a second, especially for small and medium-size problems. In the literature, there are several variants and improvements of the CWS (Golden et al., 1984). The original version of CWS is based on the estimation of possible savings originated from merging routes, i.e., for unidirectional or symmetric edges $Sav(i,j) = c_{i0}+c_{0j}-c_{ij}$. These savings are estimated between all nodes, and then decreasingly sorted. Then the bigger saving is always taken, and used to merge the two associated routes. In fact, new algorithms have been proposed based on CWS. For instance, Juan et al. (2010) propose a multi-start randomized approach, called Simulation in Routing via the Generalized Clarke and

Wright Savings heuristic (SR-GCWS), that could be considered a metaheuristic in this general classification.

Local Searches are another type of metaheuristics that move from one solution to another making systematic punctual or local changes. In fact, Juan et al. (2011e) propose two easy-to-use-and-to-understand local searches. The first is based in a cache memory for the best-known order to travel among the nodes that constitute one route. This cache is constantly updated whenever a better order with a lower cost is found for a given set of nodes. At the same time, the routes contained in this cache are re-used whenever possible to improve newly merged routes. Second, a Splitting local search method which divides the current solution into disjoint subsets of routes together with their previously assigned vehicles; then, each of these subsets are solved applying the same methodology described before during a given number of iterations. This tries to apply a "divide-and-conquer" approach since smaller instances could be easier to solve. So a new set of routes could be created on each partition with the previously assigned vehicles.

3.3 Chapter Conclusions

In this chapter, we have reviewed some of the most important methodologies definitions in the VRP arena. We have highlighted the CWS and SR-GCWS as the main techniques that we will use in this thesis. The next chapter will explain the definition of Rich VRP and the main studies related to this global research line.

4

Rich VRPs

In a previous chapter, we have introduced the most important variants families of VRPs. However, the objectives and contexts of these individual variants have slowly evolved towards more realistic scenarios. Therefore, these basics variants are far away from the current needs of enterprises. For instance, Sörensen et al. (2008) states: "although there is an increasing academic focus on so-called *rich* vehicle routing problems (that incorporate more complex constraints and objectives), they have not in any way caught up with the whole complexity of real-life routing problems." Considering several VRP restrictions at the same time still represents a challenge for the research community. The authors listed several characteristics of real-world VRPs. They also refer a survey made in 2006 by the magazine OR/MS Today about routing commercial software (Hall, 2006). There exists a wide product offer in the software market which has been developed to support transportation companies when designing their routing (distribution) plans. In 2010, another survey was made by the same magazine considering 16 vendors. In 2011, a survey of 28 software developing enterprises from Germany has been made (Drexl, 2012). Finally, in 2012, an OR/MS Today survey of 12 vendors in U.S. and Europe for 15 products shows the new needs demanded by enterprises. Between the new requirements highlight the connectivity, flexibility and dynamic of software (Partyka and Hall, 2012). However, many of these software products are too generic to solve the dynamic and demanding requirements of enterprises. The commercial routing tools often are not based on scientific approaches nor follows the features of efficient optimization algorithms previously commented in chapter 1.

4.1 Definition

A first attempt to define the Rich VRP (RVRP) has been made by Toth and Vigo (2001). The authors define the potential of extending the "vehicle flow formulations, particularly the more flexible three-index ones". The authors stated that models of the symmetric and asymmetric CVRP "may be adapted to model some variants of the basic versions". Other authors have given a different adjective to this realistic problem. Inside the research community, the RVRP is a generalization or union of other independent problems. As Goel and Gruhn (2005, 2006, 2008) deal with the General Vehicle Routing Problem (GVRP):

> "a combined load acceptance and routing problem which generalizes the well known Vehicle Routing Problem and Pickup and Delivery Problem... Furthermore, it amalgamates some extensions of the classical models which, up to now, have only been treated independently".

On a Special Issue explicitly specialized for Rich models, the editors summarize "non-idealized models that represent the application at hand in an adequate way by including all important optimization criteria, constraints, and preferences" (Hasle et al., 2006). In fact, Hasle and Kloster (2007) refers to this type problem as an *Industrial* or *Applied* Routing Problem. Pellegrini et al. (2007) state that:

> "in recent years, moreover, thanks to the increasing efficiency of these methods and the availability of a larger computing power, the interest has been shifted to other variants identified as Rich VRP. The problems grouped under this denomination have in common the characteristics of including additional constraints, aiming a closer representation of real cases... [Their case study] it is characterized by many different types of constraints, each of which unanimously classified as challenging even when considered alone".

For instance, Crainic et al. (2009a,b) introduces another term to refer RVRP. They deal with the multi-attribute VRP like *rich* problems. They also stated that:

> "real-world problems are generally characterized by several interacting *attributes*, which describe their feasibility and optimality structures. Many problems also display a combinatorial nature and are, in most cases of interest, both

> formally difficult and dimensionally large. In the past, the general approach when tackling a combinatorial multi-attribute, rich problem was either to frontally attack it, to address a simplified version, or to solve in a pipeline manner a series of simpler problems".

Therefore, the constraints may be known also as attributes of the RVRP (VRP with multi-attributes). More recently, Rieck and Zimmermann (2010) states that:

> "hence, research has turned to more specific and rich variants of the CVRP. The family of these problems is identified as rich vehicle routing problems. In order to model RVRPs, the basic CVRP must be extended by considering additional constraints or different objective functions".

The evolution of models can be appreciated when new needs about the models themselves emerges. On this respect, the authors stated:

> "rich vehicle routing problems are usually formulated as three-index vehicle-flow models with decision variables x_{ijk} which indicate whether an arc $(i, j)/i, j \in \Omega$ is traversed by vehicle k $(k = 1, ..., K)$. These models seem to be more flexible incorporating additional constraints, e.g., different capacities of the vehicles. In their monograph, Toth and Vigo (2001) suggest that two-index vehicle-flow formulations 'generally are inadequate for more complex versions of vehicle routing problems'. Their arguments are based on that these models are not suited for the cases where the cost or the feasibility of a circuit [each corresponding to a vehicle route] depends on the overall vertex sequence or on the type of vehicle allocated to the route".

The new models have been extended to include other features in the logistic or supply chain process. Furthermore, Schmid et al. (2013) have proposed six integrative models considering the classical version of the VRP and some important extensions in the context of supply chain management. These extensions are lotsizing, scheduling, packing, batching, inventory and intermodality. The authors state as benefit of their models that these consider an efficient use of resources as well as the inclusion of inter-dependencies among the subproblems. Lahyani et al. (2012) have point out the importance of stating a common and closed definition for RVRP scope:

> "in most papers devoted to RVRPs, definitions of rich problems are quite vague and not significantly different. There is no formal definition either criterion which leads to decide whether or not a VRP is rich. Such definition has to rely on a relevant taxonomy which can help to differentiate among numerous variants of the VRP".

In fact, the authors conclude their study with a numerical proposal for a specific definition:

> "a RVRP extends the academic variants of the VRP in the different decision levels by considering additional strategic and tactical aspects in the distribution system (4 or more) and including several daily restrictions related to the 'problem physical characteristics' (6 or more) [pure routing or operational aspects]. Therefore, a RVRP is either a VRP that incorporates many strategic and tactical aspects and/or a VRP that reflects the complexities of the real-life context by various challenges revealed daily. The state of the art of RVRP has changed since 2006. Now studies incorporate more complex aspects of reality. Therefore, some variants described as rich by their authors in 2006 may not be considered as such anymore".

So depending on the considered paper (or photography of achievements in research community), the RVRP definition will be evolving all the time.

As the reader can appreciate, the implications of the Rich VRP definition has evolved to a more precise concept during time. The new demanding needs of enterprises have forced academics to consider more complex approaches. There is a clear trend of creating generic and efficient approaches. Considering the large number of papers that have been devoted to the VRP, just a few of these could be applied to the RVRP context. There are a small number of papers that have explicitly addressed the RVRP. This fact emphasizes the emptiness in the literature as well as the opportunities that the academy sector has to collaborate with enterprises addressing real routing problems. The next section presents a brief literature review on some strategies addressed to solve Rich VRP instances with more than one constraint simultaneously.

4.2 Literature Review

In this section, we have find more than 50 papers selected because they auto-denominate *Rich* extensions of the original VRP or are related to other RVRPs, plus some few others that consider several VRP variants. They have in common that they consider one or more variants of the classical VRP. The approaches presented on these papers solve separated VRP variants or with different combinations of their constraints. One of the first-explicitly Rich VRP cases is presented on Pellegrini (2005). The authors have addressed a specific Rich VRP approach with the consideration of heterogeneous fleet, multiple time windows, the delivery cannot be offered in some intervals of time and there is a maximum time for a single tour. The author proposed two heuristic algorithms based on the well-known Nearest Neighbour (NN) heuristic procedure (Solomon, 1987) and other combined with a swap local search. The author created a Deterministic version of a NN (DNN) algorithm as well as a Randomized NN (RNN) version which adds a random behaviour to the selection of the next customer in the building process of a route. The author showed encouraging results in a short computational time with generated instances of 50, 100, 150, and 200 customers. The RNN algorithm reaches better results than the DNN version. Although the RNN version loses some efficiency as the number of customers increases.

On the other hand, Goel and Gruhn (2005, 2006) address the capacity restrictions, time windows, heterogeneous fleet with different travel times, and also multiple pickup and delivery locations, travel costs, different start and end locations for vehicles and other constraints. They propose iterative improvement approaches based on Large Neighbourhood Search (LNS). The authors have created a instance generator of 50, 100, 250 and 500 orders in order to show the performance of their approach. Likely, Goel and Gruhn (2008) consider real-life requirements —e.g., time window restrictions, a heterogeneous vehicle fleet with different travel times, travel costs and capacity, multi-dimensional capacity constraints, order/vehicle compatibility constraints, orders with multiple pickup, delivery and service locations, different start and end locations for vehicles, and route restrictions for vehicles. The authors propose an iterative improvement approach. They used a reduced Variable Neighbourhood Search (VNS) algorithm for exchanging elements between neighbourhood, and also a LNS approach for using nested neighbourhoods of different size. This combination helps to avoid local minimum.

Following the LNS research line, Ropke and Pisinger (2006a,b) propose a heuristic based on LNS as proposed by Shaw (1998). Furthermore, their approach is a unified heuristic with an adaptative layer. They are focused on the BVRP with time windows, pickup-and-delivery and multi-depots. They propose a model transformation of the BVRP to solve the simultaneous pickup-and-delivery. Nine data sets are used to test several configurations of the proposed heuristic, where more of the 50% of best known solutions for those instances are improved. Later, the same authors developed an Adaptative LNS framework (Pisinger and Ropke, 2007) for addressing the capacitated, time windows, multi-depot, split-deliveries and open routes constraints. They use several sets of instances with up to 1000 customers, and improve 183 best known solutions out of 486 benchmark tests.

Hasle et al. (2005) shortly describe four mechanisms for enhance scalability and present a generic route construction heuristic for Rich VRPs. The empirical investigation results based on standard test instances for several VRP variants show the effectiveness of this approach. Likely, Hasle and Kloster (2007) propose a generic approach to harness a modelling flexibility. The authors present a generic solver based on a unified algorithmic approach which is a combined operation of Local Searches and Metaheuristics (Variable Neighbourhood Descent and Iterated Local Search). An initial solution is generated using the parallel version of CWS, then other methods are applied. They address the capacitated constraint, the distance limitation, the pickup-and-delivery, the fleet size and mix problem as well as the time windows. They present the possibility to extend it for multi-depot and site-dependent problems. Classical benchmarks of Solomon (1987) and its modification (Li and Lim, 2001) for new contexts are also used. Their results are based on a range of customers between 50 up to 1000.

A wide classification of the Rich VRP variants is presented in a special issue published by Hartl et al. (2006). On this, seven papers were selected for covering different aspects of ampleness and illustrating novel types of VRP applications for that time. The editors state "VRP research has often been criticized for being too focused on idealized models with non-realistic assumptions for practical applications". Several optimization methods are proposed for solving problems inspired in real applications of VRP knowledge. For instance, Reimann and Ulrich (2006) addressed the VRP with backhauls and time windows. Hoff and Løkketangen (2006) is focused in the Travelling

Salesman Problem with pickup and delivery. Ileri et al. (2006) work in the pickup and delivery requests with time windows, heterogeneous fleet, and some operational constraints over the driver routes. The authors use a Set Partitioning technique and also Column Generation to solve real-life instances. Fügenschuh (2006) proposes a metaheuristic for the VRP with coupling time windows. This method combines classical construction aspects with mixed-integer preprocessing techniques, and improving with a randomized search strategy. Several randomly generated instances are used, as well as a real-world case for public bus transportation considering school times in rural areas of Germany. (Magalhães and Sousa, 2006) presents a real case of adopting a system of variable routes that are dynamically designed. Sörensen (2006) shows a bi-objective case considering marketing and financial interests for being solved using metaheuristics. Bolduc et al. (2006) addressed a multiple period horizon in an inventory context with heterogeneous fleet, multi-trips and capacity restrictions. The authors use heuristics to minimize the cost of distributing products to the retailers and the cost of maintaining inventory at the facility. Randomly generated instances were used to measure the performance of the approach with two set of small and large cases.

Pellegrini et al. (2007) have presented a case of study characterized by multiple objectives, constraints concerning multiple time windows, heterogeneous fleet of vehicles, maximum duration of the sub-tours, and periodic visits to the customers. They considered two versions of Ant Colony Optimization (ACO): (a) Multiple Ant Colony System (M-ACS) first proposed by Dorigo and Gambardella (1997); and (b) MAX-MIN Ant System (MMAS) based on Stützle and Hoos (1997). The authors compared the results with a Tabu Search (TS) algorithm and a Randomized NN (RNN) heuristic which was mentioned before. Both ACO developed algorithms perform significantly improved than the TS and RNN approaches, using an instance generator of 70-80 orders. Other ACO implementation is proposed by Rizzoli et al. (2007) which has been applied to real contexts addressing separately heterogeneous fleet, time windows, pickup and delivery, time dependent and on-line VRP. The authors have tested four ACO algorithms using data from real distribution companies between 15 and 600 customers.

In Hoff (2006), we can find four papers (Hoff and Løkketangen, 2006, 2007; Hoff et al., 2009, 2010) focused in the development of Lasso Solution Strategies using TS and heuristics for the VRP with pickup and delivery, time-depending and stochastic demands. The author has created instances with 7-262 nodes which are derived from classical ones used in CVRP. A real-life problem from a Norwegian company is also considered. In Derigs and Döhmer (2005), the authors also addressed the pickup and delivery VRP with time windows. They proposed an indirect search procedure based on sequence/permutation of tasks, cheapest insertion of a visit, and a Threshold-Accepting like a local search metaheuristic. The proposed algorithm has been implemented into a Decision Support System for a removal firm. They produce some promising preliminary results with random generated instances.

Irnich (2008) takes advantages of strong modelling capabilities and proposes a Unified Modelling and Heuristic Solution Framework. The author highlights the potential of $k - edge$ exchange neighbourhoods. This approach is intended to support efficient local search procedures for addressing all standard types of VRPs, such as the capacitated and distance-constrained, multiple depots, time windows, simultaneous delivery and pickup, backhauling, pickup-and-delivery problems, periodic VRP, fleet mix and size, site dependencies as well as mixtures and extensions of these. The author propose to integrate the efficient search blocks into different metaheuristics. Some promising results are presented for VRPTW and MDVRPTW combining a VNS with LNS strategies —inspired on Ropke and Pisinger (2006a). On large scale instances, they speed-up the results.

There is a long line of studies using exact methods or combinations with them. In Wen (2010), we can find three papers to address some variants of the Rich VRP inspired in real-life situations. The author proposes different strategies to solve each: (a) the VRP with cross-docking options through a TS based heuristic and testing with 200 pairs of suppliers and customers (Wen et al., 2008); (b) the dynamic VRP with multiple objectives over a planning horizon that consists of multiple periods through Mixed Integer Linear Programming (MILP) and a three phase heuristic (Wen et al., 2010a); and (c) the VRP with multi-period horizon, the time windows for the delivery, the heterogeneous vehicles, the drivers working regulations, and other constraints (Wen et al., 2010b). On the last work, the author proposes a MILP and treated by a multi-level VNS algorithm. Good quality solutions for solving up to 2000 orders are generated using a real case information. On this same research line, (Rieck and Zimmermann, 2010) propose a new MILP (two-index vehicle-flow) model for a Rich VRP with docking constraints. They consider time windows, simultaneous delivery and pick-up at customer locations and multiple uses of vehicles. The test instances with 10-30 customers were generated from the classical set of Solomon (1987). The proposed method solves small and medium problem instances efficiently. Other promising approach, as proposed by Doerner and Schmid (2010), consists in the combination of exact algorithms and metaheuristics search components. The author presents a survey of several hybrid techniques and also highlights some key aspects for future studies. Hybrid approaches allow conquering the obstacles observed when the individual concepts are applied independently. They present three trends of hybridization schemes: set-covering based, local branching approaches, and decomposition techniques. They addressed the periodic VRP with time windows and the multi-depot VRP with time windows, but other variants are commented. An exact solution framework based on Set Partioning (SP) modelling is proposed by Baldacci et al. (2010, 2011a,b) for individual types of VRPs. The results outperforms all other exact methods published so far and also solve several previously unsolved test instances. The preliminary step to the proposed Framework is presented on Baldacci and Mingozzi (2009) where a unified exact method based on set partitioning is introduced for solving the well-known CVRP, HVRP, FSMVRP, SVRP and the MDVRP. Computational results show the performance of their approach over the main instances from the literature of the different variants of HVRP, SVRP and MDVRP.

Several studies have developed Column Generation-based (CG) methods. Oppen et al. (2010) consider a real scenario called the Livestock Collection Problem (LCP) which is considered a Rich VRP extended with inventory constraints. This context includes duration and capacity restrictions, heterogeneous fleets, time windows, multi-trips and multi-product issues. The authors addressed it through an exact solution method based on CG. The authors have created instances with less than 30 customers' orders inspired in real-world. The CG approach has helped to find optimal solutions in different scenarios. But the authors defined limitations for finding optimal solutions to LCP instances. Another CG heuristic is proposed by Goel (2010) for addressing a VRP with time windows, heterogeneous vehicle fleet, multiple depots, and with pickup-and-delivery. Some small instances are randomly generated in order to test the heuristic performance. Ceselli et al. (2009) also propose the use of a CG combined with a dynamic programming algorithm in order to address simultaneously a heterogeneous fleet, different depots, time windows, route length, optionally opened routes, pickup and delivery and several other constraints. The authors tested their approach with 46 randomly generated instances composed by 100 orders and the results are compared with lower bounds. Under a similar restricted context, Ruinelli (2011) has compared three methods on a master thesis context: an ACS, a CG algorithm and a general purpose MILP solver. Computational results are presenting using 14 real instances from a distribution company, where the CG outperforms the other two methods.

Other generic Rich solvers have been emerged in the literature. Cordeau and Laporte (2003); Cordeau et al. (1997, 2001b, 2004) propose an Unified Tabu Search approach for VRPs with time windows, multi-period, multi-depot and site-dependent. Several real and theoretical benchmarks have been used to test the performance of this approach. Some ILS approaches are proposed by Hashimoto et al. (2006, 2008); Ibaraki et al. (2005). In fact, Subramanian (2012) propose a promising combination of ILS with Integer Programming aspects for several VRP variants. In fact, this work was extended to the FSM and HVRP research line in Subramanian et al. (2012). They have developed a hybrid algorithm composed by an Iterated Local Search (ILS) based heuristic and a Set Partitioning (SP) formulation. The SP model is solved by using a MIP solver that calls the ILS heuristic during its execution. Three benchmark instances with up to 360 customers were used to test the approach. For instance, Groër et al. (2010) implemented a library of 7 local search Heuristics for addressing several variants like the CVRP, VRPTW and MDVRP. Some classical heuristics are used —e.g., Record-to-Record, CWS. Their approach is based on simply removing and inserting customers from an existing solution (called neighbourhood ejection). Several classical benchmarks are used to show the performance of their approach. On Battarra (2011) several exact and heuristic algorithms for several routing problems are presented individual Rich VRP cases (Baldacci et al., 2009; Battarra et al., 2009). Some of the problems addressed are the Fleet Size and Mix and the HVRP with multi-trips and time windows.

Prescott-Gagnon et al. (2010) present a real-inspired oil distribution which presents a set of particular features. Some of the constraints addressed are the heterogeneous vehicle fleet, multiple depots, intra-route replenishment, time windows, driver shifts and optional customers. The authors propose three metaheuristics, namely, a TS algorithm, a LNS heuristic combined with TS heuristic, and another LNS heuristic based on a CG heuristic. Computational results indicate that both LNS methods outperform the TS heuristic. In fact, the LNS method based on CG tends to produce better quality solutions. Also Lannez et al. (2010) present a heuristic based on CG for a very particular extension of Rich VRP called Rich Arc Routing Problem, where the demand is located on the arcs and not in the nodes.

Recently, Santillán et al. (2012) solve a Routing-Scheduling-Loading using a heuristic-based system. As a first step, the proposed system applies an ACS for the Routing and Scheduling Problem, then a Bin Packing technique is used for the Vehicle Load problem. Some tests with Solomon (1987) instances are developed. Also the authors use real information from the distribution of bottles provided by a mexican company. Another hybrid approach is proposed by Vallejo et al. (2012). They apply a three-phase heuristic which merges the use of a memory-based approach with clustering techniques. The authors present promising test results using between 100 and 2000 customers comparing their approach against a Genetic Algorithm. Two particular real cases are presented next inspired on Ropke and Pisinger (2006a). First, Amorim et al. (2012) create a new Adaptative LNS for solving specific real instances of a heterogeneous fleet site dependent vehicle routing problem with multiple time windows. This case is inspired in a food distribution company in Portugal. Second, Derigs et al. (2013) propose to combine the commented ALNS with Local Searches both controlled by two metaheuristic procedures (Record-to-Record travel and attribute based hill climber) for addressing a particular real case called Rollon-Rolloff VRP (RRVRP) occurred in sanitation/waste collection. Some promising computational results are presented using previous benchmarks.

Vidal et al. (2012a) develop a study of over 64 metaheuristics comparing their benchmarks on 15 classic variants of VRP with multi-attributes. They present a classification on the types of constraints as attributes and identify promising principles in algorithmic-designing for Rich VRP. In fact, they state that the critical factors for efficient metaheuristic is the appropriate balance between intensification and diversification explorations in the solution space. The authors conclude that the combination of hybrid algorithms and parallel cooperative methods would create effective solvers. Later the same authors proposed a Unified Solution Framework called Unified Hybrid Genetic Search (UHGS) for several types of Rich VRP (Vidal et al., 2012b). The Framework uses efficient generic local search and genetic operators. This approach is also based on a giant-tour representation with a Split procedure originally proposed by

Prins (2004). The authors present interesting computational results using 39 benchmarks over 26 different Rich VRP. Furthermore, the authors apply their method combined with diversity management mechanisms to different large scale instances of Rich Time-constrained VRPs (Vidal et al., 2013). The used instances involve up to 1000 customers. The proposed framework outperforms all current state-of-the-art approaches. The approach is addressed to any combination of periodic, multi-depot, site-dependent, and duration-constrained VRP with time windows.

In Table 4.1 a summary of the state-of-the-art approaches developed for the Rich VRP is presented by authors, year of publication, type of proposed method, maximum number of customers addressed in the study. As the reader can appreciate the rows are sorted by type of method, year and last name of first author. Also we have applied a restrictive filter if the approach can solve more than one Rich VRP. The star (*) on the last column highlights the approaches that have been or can be tested with no restriction on the combination of constraints. The table is divided on two parts. The first part describes the exact methods then it follows the heuristic and meta-heuristic inspired approaches.

4.3 Classification of Rich VRP papers

Most routing constraints considered in the previous works were unified and classified. The next list presents the main distribution constraints considered on these papers. Table 4.2 shows the presence of each constraint on commented papers. This is useful to appreciate the diversity of cataloged papers as Rich VRPs. In fact, all steps in the supply-chain are being considered at the same time in order to be optimized in a general way. Then in Table 4.3 a classification of these routing constraints is done using commented studies of Lahyani et al. (2012); Vidal et al. (2012b). In Vidal et al. (2012b), the routing constraints are related to its representation point inside of the inner methodology process. For this, they propose three groups which represent the simple aspects that any solver must deal with: Assignment of customers and routes to resources, the Sequence choices, and the Evaluation of fixed sequences. The authors state that this "simply classification is intimately connected with the resolution methodology". In Lahyani et al. (2012), constraints are associated to the enterprise decision levels (operational, tactical, and strategy). The first level (strategic) includes decisions related to

Authors	Year	Method	Maximum n	Several Rich VRPs
Ruinelli	2011	Column Generation	150	
Baldacci et al.	2011a	Exact Method	200	$\surd^*$
Baldacci et al.	2011b	Exact Method	200	$\surd^*$
Baldacci et al.	2010	Exact-Solution Framework	200	$\surd^*$
Bettinelli et al.	2011	Branch-and-Cut-and-Price	144	
Doerner and Schmid	2010	MatHeuristics	-	
Goel	2010	Column Generation	250	
Oppen et al.	2010	Column Generation	27	
Rieck and Zimmermann	2010	Mixed-Integer Linear Programming	30	
Baldacci and Mingozzi	2009	Exact algorithm based on Set Partitioning	100	$\surd$
Ceselli et al.	2009	Column Generation	100	
Fügenschuh	2006	Mixed-Integer Programming	404	
Derigs et al.	2013	LS/LNS-based metaheuristic	199	
Vidal et al.	2013	Hybrid Genetic Search with Advanced Diversity Control	1000	$\surd^*$
Amorim et al.	2012	Adaptative Large Neighbourhood Search Framework	366	
Santillán et al.	2012	Ant Colony System	100	
Subramanian et al.	2012	Iterated Local Search	360	
Vidal et al.	2012b	Unified local search and Hybrid Genetic Search	480	$\surd^*$
Vallejo et al.	2012	3-phase heuristic using a memory-based and clustering techniques	2000	
Battarra	2011	Exact and Heuristic algorithms	100	$\surd$
Groër et al.	2010	Local Search Heuristic	483	
Prescott-Gagnon et al.	2010	Tabu Search, LNS+TS heuristic, LNS+CG heuristic	750	
Wen et al.	2010a	3-phase heuristic	80	
Goel and Gruhn	2008	Variable and Large Neighbourhood Searches	40	
Irnich	2008	Heuristic Framework using Local Search-Based Metaheuristics	1000	$\surd^*$
Wen et al.	2008	TS and Adaptive Memory Procedure	200	
Hasle and Kloster	2007	MetaHeuristics	199	$\surd^*$
Pellegrini et al.	2007	Multiple Ant Colony Optimization	80	
Pisinger and Ropke	2007	LNS Heuristic	1008	$\surd^*$
Rizzoli et al.	2007	Ant Colony Optimization	600	$\surd$
Bolduc et al.	2006	Heuristics	75	
Goel and Gruhn	2006	Large Neighborhood Search	500	
Hoff and Løkketangen	2006	Tabu Search Heuristic	262	
Ileri et al.	2006	Set partitioning model	130	
Magalhães and Sousa	2006	Heuristic based on clustering	450	
Reimann and Ulrich	2006	Ant Colony Optimization	100	
Ropke and Pisinger	2006a	LNS Heuristic	500	$\surd$
Ropke and Pisinger	2006b	LNS Heuristic	500	$\surd$
Sörensen	2006	Memetic algorithm with population management	199	
Derigs and Döhmer	2005	Local Search Algorithm	-	
Goel and Gruhn	2005	Large Neighborhood Search	500	
Pellegrini	2005	Nearest Neighbor	200	
Cordeau et al.	2004	Improved Unified Tabu Search heuristic	288	$\surd$
Cordeau et al.	2001b	Unified Tabu Search heuristic	1035	$\surd$
Cordeau et al.	1997	Tabu Search	288	

Table 4.1: State-of-the-art of Rich VRP methods.

the locations, the number of depots used and the data type. The tactical level defines the order type and the visit frequencies of customers over a given time horizon. Finally, the operational level considers the vehicle and the driver schedules; so the constraints are related to the distribution planning and specified for customers, vehicles, drivers and roads. Additionally, we propose a second level of classification associated to the routing element involved (depot, customer, route, vehicle, and product) in order to help for a better understanding of the classification.

- Multi-Products (**CP**): Some vehicles can carry out several types of products (fresh-cold, small-big, etc.).
- Multi-Dimensional capacity (**CD**): The capacity of vehicles is considered in 2D or 3D.
- Vehicle Capacity (**C**): The capacity of vehicles is limited.
- Homogeneous Fleet of Vehicles (**FO**): All vehicles of the fleet have the same capacity.
- Heterogeneous Fleet of Vehicles (**FE**): Several type of vehicles (capacities) can be found in the fleet.
- Unfixed Fleet of Vehicles (**VU**): The number of vehicles considered is unlimited.
- Fixed Fleet of Vehicles (**VF**): The number of vehicles considered is limited.
- Fixed Cost per Vehicle (**FC**): To use a vehicle implies an extra cost.
- Variable Cost of Vehicle (**VC**): The real cost is represented by the product of the distance assigned to a vehicle and its price per distance unit.
- Multi-Trips (**MT**): All or some vehicles of the fleet can execute more than one trip (multiple uses of vehicles).
- Vehicle Site Dependent (**DS**): Some vehicles can not visit some nodes due to geographical, compatibility or legal issues.
- Vehicle Road Dependent (**DR**): Some vehicles can not pass through some edges of the network for some legal issues.

- Duration Constraints/Length (**L**): The duration of each route can not exceed a maximum value or cost, including service times on each visited client.

- Driver Shifts/Working Regulations (**D**): The design of routes include the number of legal working hours of drivers (stops, breaks, rest, etc).

- Balanced Routes (**BR**): The load of routes or vehicles must be balanced between all.

- Symmetric Cost Matrix (**CS**): The cost matrix has a symmetric nature.

- Asymmetric Cost Matrix (**CA**): The cost matrix has an asymmetric nature.

- Intra-route replenishments (**IR**): The vehicles must be re-loaded in some point of the routes.

- Time Dependent/Dynamic/Stochastic times (**TD**): The target is minimizing time and the travelling times could vary during a day (hard or flexible). The location/distance of clients changes.

- Stochastic Demands/Dynamic (**S**): The demands of clients can change during the application of a routing solution.

- Time Windows (**WT**): The clients can not receive the orders out of a time windows. Each client has a particular time window (hard or soft).

- Multiple Time Windows (**WM**): The clients can not receive the orders out of a set of time windows. Each client has a particular set of time windows.

- Pick-up & Delivery (**PD**): The construction of routes must consider the picking up of products in some clients and the delivery to others, in a sequentially or separately way. The depot just define the starting/ending point of vehicles.

- Simultaneous Pick-up & Delivery (**SP**): The construction of routes must consider the picking up and delivery of products/persons at the same time in all nodes by the same vehicle. The depot just define the starting/ending point of vehicles.

- Backhauls (**B**): The construction of routes must consider the picking up of products in some clients and the delivery to others, in a sequentially or separately way. The critical assumption is that all deliveries must be made on each route before any pickups can be made (sometimes a client could require both a delivery and a pick-up). The re-arrangement of products could be expensive or unfeasible. The depot just define the starting/ending point of vehicles.

- Multiple Visits/Splitted deliveries (**MV**): The clients are visited several times for delivering the summatory of the original order orders. Each vehicle may deliver a fraction of a customer's demand.

- Multi-Period/Periodic (**MP**): The optimization is made over a set of days, considering several visits and each client has a different frequency of visits.

- Inventory Levels Controls (**I**): The costs of stocks are also considered to be minimized with the routing costs while the levels of stock are controlled.

- Customer Capacity (**CC**): The capacity stock of clients is also considered.

- Multi-Depot (**MD**): There is more than one depot from where the vehicles leave and arrive.

- Time Windows for the Depot (**WD**): The depot is open during a period of time. So if vehicles need to do more than one trip they need to consider this.

- Different end locations/Open Routes (**O**): The routes start at the depot but finish on the last client. The return cost is not considered or optional.

- Different start and end locations (**DA**): The vehicles start and end in different locations.

- Departure from different locations (**DD**): The vehicles start in different locations.

- Precedence constraints (**PC**): The order of visits some clients could be important for the loading and unloading of products. Its order could be important for healthy or security reasons.

- Multi-Objectives (**MO**): The study considers more than one objective function or related costs at the same time.

Authors	Year	CP	CD	C	FO	FE	VU	VF	FC	VC	MT	DS	DR	L	D	BR	CS	CA	IR	TD	S	WT	WM	PD	SP	B	MV	MP	I	CC	MD	WD	O	DA	DD	PC	MO
Derigs et al.	2013			√	√		√							√				√						√													
Vidal et al.	2013			√	√	√	√					√					√					√	√					√			√						
Amorim et al.	2012			√		√	√		√	√		√		√			√						√														
Santillán et al.	2012			√		√	√		√	√	√	√	√				√				√	√	√				√				√		√				
Subramanian	2012			√	√	√	√	√	√	√							√	√						√	√						√		√				
Subramanian et al.	2012			√		√	√	√	√	√							√																				√
Vidal et al.	2012b			√	√	√	√	√	√	√		√		√	√		√			√		√	√		√	√		√			√		√				
Vallejo et al.	2012			√		√		√			√						√					√									√						
Baldacci et al.	2011b			√	√			√									√														√						√
Baldacci et al.	2011a			√	√			√									√											√									
Battarra	2011			√		√		√	√		√						√							√													
Bettinelli et al.	2011			√		√		√	√								√					√									√						
Ruinelli	2011			√		√			√			√		√	√								√	√									√		√		
Yu and Zhen-Yang	2011			√	√			√									√					√						√									
Baldacci et al.	2010			√		√	√	√	√	√		√					√					√		√				√			√						
Doerner and Schmid	2010			√	√			√									√					√						√			√						
Groër et al.	2010			√	√			√						√			√					√									√						
Oppen et al.	2010	√		√		√	√				√			√			√					√					√		√							√	
Prescott-Gagnon et al.	2010			√		√	√							√	√		√		√		√	√						√	√		√						
Wen et al.	2010a			√	√			√									√				√							√									√
Rieck and Zimmermann	2010			√		√		√			√			√	√		√					√			√												
Goel	2010			√		√	√			√		√					√					√		√									√	√			
Ceselli et al.	2009			√		√	√		√	√		√		√	√		√					√					√				√		√				
Baldacci and Mingozzi	2009			√		√	√	√	√	√		√					√														√						
Crainic et al.	2009a			√	√			√						√			√					√						√			√						
Goel and Gruhn	2008		√	√		√	√			√		√					√					√		√									√	√			
Hashimoto et al.	2008			√	√			√						√			√			√		√															
Irnich	2008			√	√	√	√	√				√		√			√					√		√	√	√		√			√				√	√	
Wen et al.	2008			√	√			√									√					√		√													
Hasle and Kloster	2007			√	√	√	√	√						√			√					√	√	√							√						
Pellegrini et al.	2007			√		√	√		√	√		√		√	√		√						√				√	√									√
Pisinger and Ropke	2007			√	√	√		√				√					√	√				√									√		√				
Rizzoli et al.	2007			√	√			√									√			√	√	√		√													
Bolduc et al.	2006			√		√		√									√											√	√								
Fügenschuh	2006			√	√			√									√					√	√														
Hashimoto et al.	2006			√	√			√									√					√	√														
Goel and Gruhn	2006		√	√		√		√		√		√		√	√		√				√	√		√									√	√			
Hoff and Løkketangen	2006			√	√			√									√							√													
Ileri et al.	2006			√		√		√	√	√							√					√		√													
Magalhães and Sousa	2006			√		√		√									√				√																
Reimann and Ulrich	2006			√	√			√									√					√				√											√
Ropke and Pisinger	2006a			√	√			√									√					√		√	√	√					√					√	
Ropke and Pisinger	2006b			√	√			√									√					√		√													
Sörensen	2006			√	√			√						√			√																				√
Derigs and Döhmer	2005			√	√		√										√						√	√													
Goel and Gruhn	2005		√	√		√		√		√		√					√				√	√		√									√	√			
Ibaraki et al.	2005			√	√			√						√			√					√							√								
Pellegrini	2005			√		√		√									√						√														√
Cordeau et al.	2004			√	√	√		√				√					√					√						√			√						
Cordeau and Laporte	2003			√	√			√						√			√					√		√													
Cordeau et al.	2001b			√	√			√									√					√						√			√						
Cordeau et al.	1997			√	√			√									√											√			√						

Table 4.2: Relation of Rich VRP papers and considered constraints.

Restriction	Code/Id	(Vidal et al., 2012b) Classification	(Lahyani et al., 2012) Classification	Our 2nd Level Classification
Multi-Products	**CP**	Assign	Strategic	Veh-Prod
Multi-Dimensional capacity	**CD**	Assign	Strategic	Veh-Prod
Vehicle Capacity	**C**	Assign	Operational	Veh
Homogeneous Fleet of Vehicles	**FO**	Assign	Operational	Veh
Heterogeneous Fleet of Vehicles	**FE**	Assign	Operational	Veh
Unfixed Fleet of Vehicles	**VU**	Evaluation	Operational	Veh
Fixed Fleet of Vehicles	**VF**	Assign	Operational	Veh
Fixed Cost per Vehicle	**FC**	Evaluation	Operational	Veh
Variable Cost of Vehicle	**VC**	Evaluation	Operational	Veh
Multi-Trips	**MT**	Sequence	Operational	Veh
Vehicle Site Dependent	**DS**	Assign	Operational	Veh-Cust
Vehicle Road Dependent	**DR**	Assign	Operational	Veh-Route
Duration Constraints/Lenght	**L**	Evaluation	Operational	Route-Driver
Driver Shifts/Working Regulations	**D**	Evaluation	Operational	Route-Driver
Balanced Routes	**BR**	Assign	Operational	Route-Driver
Symmetric Cost Matrix	**CS**	Sequence	Operational	Route
Asymmetric Cost Matrix	**CA**	Sequence	Operational	Route
Intra-route replenishments	**IR**	Assign	Tactical	Route
Time Dependent/Dynamic/Stochastic times	**TD**	Evaluation	Tactical	Route
Stochastic Demands/Dynamic	**S**	Evaluation	Tactical	Customer
Time Windows	**WT**	Evaluation	Tactical	Customer
Multiple Time Windows	**WM**	Evaluation	Tactical	Customer
Pick-up & Delivery	**PD**	Sequence	Tactical	Customer
Simultaneous Pick-up & Delivery	**SP**	Evaluation	Tactical	Customer
Backhauls	**B**	Sequence	Tactical	Customer
Multiple Visits/Splitted deliveries	**MV**	Assign	Tactical	Customer
Multi-Period/Periodic	**MP**	Assign	Tactical	Customer
Inventory Levels Controls	**I**	Assign	Tactical	Customer
Customer Capacity	**CC**	Assign	Tactical	Customer
Multi-Depot	**MD**	Assign	Strategic	Depot
Time Windows for the Depot	**WD**	Evaluation	Strategic	Depot
Different end locations/Open Routes	**O**	Evaluation	Strategic	Depot
Different start and end locations	**DA**	Evaluation	Strategic	Depot
Departure from different locations	**DD**	Evaluation	Strategic	Depot
Precedence constraints	**PC**	Sequence	Tactical	Depot
Multi-Objectives	**MO**	Evaluation	Tactical	Depot

Table 4.3: Classification of main documented Rich VRP constraints.

4.4 Chapter Conclusions

In this last three chapters, we have reviewed the evolution of studied problems in the VRP arena. We present a variety of routing scenarios that can be found in reality. Also several methods have been developed for addressing all types of Rich VRPs. The Rich VRP domain has appeared on the first decade of XXI century; and it is shown itself as a promising research area. There are many tailored approaches for specific cases of Rich VRP. However, in the last ten years the general-purpose methods are slowly emerging keeping the previous quality features but for generic Rich VRP scenarios. Next chapters, we will study some tailored approaches for both deterministic (HVRP, HVRPM, HAVRP, AVRP, VRPTW, and DCVRP) and stochastic (VRPSD and IRPSD) scenarios (see Fig. 4.1). Progressively on the thesis, we will present three methodologies related between them in order to finally design a generic approach for the Rich VRP. The first methodology is the core of the other two which consists in the biased-randomization of classical heuristics. The second is based in the combination of Monte-Carlo simulation with biased-randomized heuristics. Finally, a generic approach is proposed joining the advantages of constraint programming validation and the solution generation of biased-randomized heuristics. In the next chapter, we discuss the main implementation aspects of the first methodology.

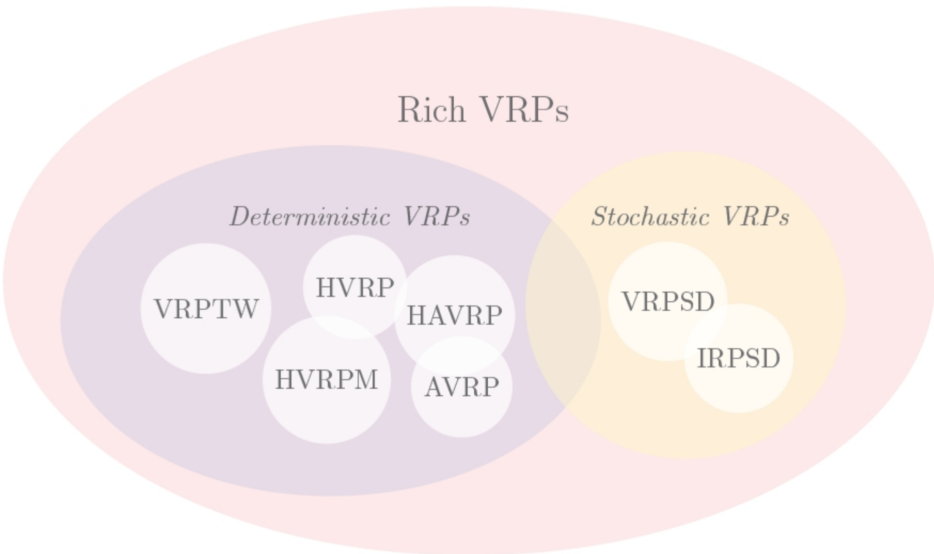


Figure 4.1: Relation of VRPs studied on this dissertation.

5

Biased Randomization of Heuristics

Parts of this chapter have been taken from the co-authored publication: Juan, Cáceres-Cruz, González-Martín, Riera, and Barrios (2014a) in Encyclopedia of Business Analytics and Optimization. IGI Global. USA.

In this chapter, we will present the basic core used in all approaches proposed in this thesis. This common aspect for addressing different variants is a biased-randomization aspect inside of the proposed methods. The potential of biased randomized heuristics for solving real problems (mainly using the CWS) is actually promoted in this thesis. This chapter discusses how to randomize classical heuristics in order to transform these deterministic procedures into more efficient probabilistic algorithms. This randomization process can be performed by using a uniform probability distribution or, even more interesting, by using a non-symmetric distribution.

Combinatorial Optimization Problems (COPs) have posed numerous challenges to the human mind throughout the past decades. From a theoretical perspective, they have a well-structured definition consisting of an objective function that needs to be minimized or maximized, and a series of constraints that must be satisfied. From a theoretical point of view, these problems have an interest on their own due to the mathematics involved in their modelling, analysis and solution. However, the main reason for which they have been so actively investigated is the tremendous amount of real-life applications that can be successfully modelled as a COP. Thus, for example, decision-making processes in fields such as logistics, transportation, and manufacturing contain plentiful hard challenges that can be expressed as COPs (Faulin et al., 2012; Montoya-Torres et al., 2012). Accordingly, researchers from different areas —e.g., Applied Mathematics, Operations Research, Computer Science, and Artificial Intelligence— have directed their efforts to conceive techniques to model, analyze, and solve COPs.

A considerable number of methods and algorithms for searching optimal or near-optimal solutions inside the solution space have been developed. In some small-sized problems, the solution space can be exhaustively explored. For those instances, efficient exact methods can usually provide the optimal solution in a reasonable time. Unfortunately, the solution space in most COPs is exponentially astronomical. Thus, in medium- or large-size problems, the solution space is too large and finding the optimum in a reasonable amount of time is not a feasible task. An exhaustive method that checks every single candidate in the solution space would be of very little help in these cases, since it would take exponential time. Therefore, a large amount of heuristics and metaheuristics have been developed in order to obtain near-optimal solutions, in reasonable computing times, for medium- and large-size problems, some of them even considering realistic constraints.

The main goal of this chapter is to present a hybrid scheme which combines classical heuristics with biased-randomization processes. As it will be discussed later, this hybrid scheme represents an efficient, relatively simple, and flexible way to deal with several COPs in different fields, even when considering realistic constraints.

5.1 Background

In the context of this section, we will refer to any algorithm which makes use of pseudo-random numbers to perform 'random' choices during the exploration of the solution space by the term randomized search method, or simply randomized algorithm. This includes most current metaheurisics and stochastic local-search processes. Thus, since it does not follow a deterministic path, even for the same input, a randomized search method can produce different outputs in different runs. Within these type of algorithms we can include, among others, the Genetic and Evolutionary Algorithms (Reeves, 2010), Simulated Annealing (Nikolaev and Jacobson, 2010), Greedy Randomized Adaptive Search Procedure or GRASP (Festa and Resende, 2009a,b), Variable Neighborhood Search (Hansen et al., 2010), Iterated Local Search (Lourenço et al., 2010), Ant Colony Optimization (Dorigo and Stützle, 2010), Probabilistic Tabu Search (Løkketangen and Glover, 1998), or Particle Swarm Optimization (Kennedy and Eberhart, 1995).

One of the most popular randomized search methods is GRASP (Resende and Ribeiro, 2010). Roughly speaking, GRASP is a multi-start or iterative process which uses uniform random numbers and a restricted candidate list to explore the solution space (Pseudo-code 1). At each iteration, two phases are executed: (a) the construction phase, which generates a new solution by randomizing a classical heuristic; and (b) a local search phase, which aims at improving the previously constructed solution. At the end of this multi-start process, the best found solution is kept as the result.

Algorithm 1 General pseudo-code for GRASP.

```
procedure GRASP(inputs)
    while stopping criterion is not satisfied do
        solution ← ConstructGreedyRandomizedSolution(inputs)
        solution ← ApplyLocalSearch(solution)
        bestSolution ← UpdateBestSolution(solution)
    end while
    return bestSolution
end procedure
```

It is interesting to notice that most of the work on randomized algorithms is based on the use of uniform random numbers, i.e., randomness is generated throughout a symmetric (non-biased) uniform distribution. Of course, other non-symmetric (i.e., biased) distributions can also be used to induce randomness into an algorithm. As far as we know, the first approach based on the use of biased randomization of a classical heuristic is due to Bresina (1996). This author proposes a methodology called Heuristic-Biased Stochastic Sampling (HBSS), which performs a biased iterative sampling of the search tree according to some heuristic criteria. Bresina applies the HBSS to a scheduling problem, and concludes that this approach outperforms greedy search within a small number of samples.

More recently, Juan et al. (2011c) proposed the use of non-symmetric probability distributions to induce randomness in classical heuristics. Their general framework was called Multi-start biased Randomization of classical Heuristics with Adaptive local search (MIRHA). In this approach, the authors propose to combine classical greedy heuristics with pseudo-random variates from different, non-symmetric, probability distributions. In particular, the algorithm induced biased-randomness to slightly perturbate the greedy behaviour of a classical heuristic, which transforms a deterministic heuristic into a probabilistic algorithm. According to the obtained results, the use of proper biased distributions -as an alternative to the use of the uniform distribution-contributes to explore the solution space in a more efficient way. Pseudo-code 2 shows the logic flow of the MIRHA general framework. Similar to GRASP, a multi-start procedure encapsulates the randomization of a heuristic, but this time a non-symmetric distribution will be employed instead. At each iteration, two processes are carried out. First, a new solution is constructed using a biased randomization version of the selected classical heuristic -which will depend on the particular problem being considered. Secondly, an adaptive local search is employed in order to improve the randomized solution. Notice that both the randomization effect and the multi-start process work together to reduce the probabilities that the procedure gets trapped into a local minimum, while ensuring that different feasible regions in the solution space are sampled and explored.

The common aspects of MIRHA with GRASP are the construction of an initial solution using randomization and, afterwards, the application of a local search. But there are relevant differences: (a) MIRHA does not use a restrictive candidate list, one main characteristic of the GRASP algorithm; and (b) MIRHA uses a non-symmetric distribution to select the next element to be included in the solution, while most GRASP implementations only consider uniform distributions. The HBSS proposed by Bresina (1996) is similar to MIRHA since it uses a biased distribution function combined with a sampling methodology. In fact, MIRHA can be seen as a natural extension/enhancement of the HBSS methodology, one which incorporates a local search step after each solution obtained by the biased sampling.

Algorithm 2 General pseudo-code for MIRHA.

```
procedure MIRHA(inputs)
    heuristic ← DefineHeuristic(inputs)                        ▷ different for each COP
    initialSolution ← GenerateSolution(heuristic, inputs)
    bestSolution ← ApplyAdaptiveLocalSearch(initialSolution)    ▷ for each COP
    probaDist ← DefineProbabilityDistribution(COP)              ▷ different for each COP
    while stopping criterion is not satisfied do
        solution ← GenerateRandomSolution(heuristic, probaDist, inputs)
        solution ← ApplyAdaptiveLocalSearch(solution)
        bestSolution ← UpdateBestSolution(solution)
    end while
    return bestSolution
end procedure
```

In general, probabilistic algorithms have been widely used to solve many combinatorial optimization problems such as, for example: Sequencing and Scheduling Problems (Pinedo, 2012), Vehicle Routing Problems (Laporte, 2009), Quadratic and Assignment Problems (Loiola et al., 2007), Location and Layout Problems (Mladenović et al., 2007), Covering, Clustering, Packing and Partitions Problems (Chaves and Nogueira-Lorena, 2010; Muter et al., 2010). They have also been used to solve real combinatorial optimization problems that arise in different industrial sectors, e.g.: Transportation, Logistics, Manufacturing, Aeronautics, Telecommunication, Health, Electrical Power Systems, Biotechnology, etc.

As described in Festa and Resende (2009b), GRASP algorithms have been applied to solve a wide set of problems like scheduling, routing, logic, partitioning, location, graph theory, assignment, manufacturing, transportation, telecommunications, biology and related fields, automatic drawing, power systems, and VLSI design.

Regarding the use of biased/skewed randomization as proposed by the HBSS and MIRHA general schemes, Juan et al. (2010) proposed a specific implementation, called SR-GCWS, for solving the Capacitated Vehicle Routing Problem. The SR-GCWS algorithm combines a biased randomization process with the Clarke Wright savings heuristic (Clarke and Wright, 1964). A geometric distribution is used to randomize the constructive process while keeping the logic behind the heuristic. Similarly, González-Martín et al. (2012) developed the RandSHARP algorithm for solving the Arc Routing Problem. This algorithm combines a savings-based heuristic for the Arc Routing Problem with a biased randomization process also guided by a geometric distribution. Other authors have also proposed the randomization of a classical heuristic for solving the Arc Routing Problem. For example, Gomes and Selman (1997) propose a randomized version of the Backtrack Search algorithm where randomization is added to break ties, and also a randomization of the Branch-and-Bound algorithm where randomization is added in the variable selection strategy by introducing noise in the ranking of the variables. However, in both cases the authors add uniformly distributed randomization to the base heuristic. Finally, Juan et al. (2012b) propose the ILS-ESP algorithm for solving the Permutation Flow-Shop Problem. The ILS-ESP uses an Iterated Local Search framework and combines the NEH heuristic (Nawaz et al., 1983) with a biased randomization process guided by a descending triangular distribution.

All in all, the proposed methodology can be used to improve the efficiency of most existing heuristics for solving combinatorial-optimization problems. This is done by transforming the greedy deterministic behaviour of the heuristic into a probabilistic behaviour which still follows the logic behind the heuristic but randomizes the construction process using a biased, non-uniform, distribution.

5.2 Applying a Biased Randomization

Most classical heuristics for solving combinatorial optimization problems employ an iterative process in order to construct a feasible —and hopefully good- solution. Examples of these heuristics are the Clarke and Wright procedure for the Vehicle Routing Problem (Clarke and Wright, 1964), the Nawaz-Enscore-Ham procedure for the Flow-Shop Problem (Nawaz et al., 1983), or the Path Scanning procedure for the Arc Routing Problem (Golden et al., 1983). Typically, a *priority* list of potential movements is traversed during the iterative process, i.e., at each iteration, the next constructive movement is selected from this list, which is sorted according to some criteria. The criteria employed to sort the list depends upon the specific heuristic being considered. Thus, any constructive heuristic could be seen as an iterative greedy procedure, which constructs a feasible 'good' solution to the problem at hand by selecting, at each iteration, the 'best' option from a list, sorted according to some logical criterion. Notice that this is a deterministic process, since once the criterion has been defined, it provides a unique order for the list of potential movements. Of course, if we randomize the order in which the elements of the list are selected, then a different output is likely to occur each time the entire procedure is executed. However, a uniform randomization of that list will basically destroy the logic behind the greedy behaviour of the heuristic and, therefore, the output of the randomized algorithm is unlikely to provide a good solution —in fact, we could run the randomized algorithm thousands of times and it is likely that all the solutions generated would be significantly worse than the one provided by the original heuristic. To avoid losing the 'common sense' behind the heuristic, GRASP proposes to consider a restricted list of candidates —i.e., a sublist including just some of the most promising movements, that is, the ones at the top of the list—, and then apply a uniform randomization in the order the elements of that restricted list are selected (see Fig. 5.1). This way, a deterministic procedure is transformed into a randomized algorithm —which can be encapsulated into a multi-start process—, while most of the logic or common sense behind the original heuristic is still respected. The MIRHA approach goes one step further, and instead of restricting the list of candidates, it assigns different probabilities of being selected to each potential movement in the sorted list. In this way, the elements at the top of the list receive a higher probability of being selected than those at the bottom of the list, but potentially all elements could be selected. Notice that by doing so, we are not only avoiding the issue of selecting the proper size of the restricted list, but we also guarantee that the probabilities of being selected are always proportional to the position of each element in the list.

Thus, it is possible to identify the following steps when transforming a classical heuristic into a probabilistic algorithm by means of biased randomization:

1. Given a COP, select a deterministic and constructive heuristic with the following characteristics: (a) it should be able to run quite fast —typically in less than a second— even for large-size problems —this is a critical requirement since the probabilistic algorithm relies in executing over and over a randomized version of the heuristic; (b) it should provide, by design, some stage able to be randomized

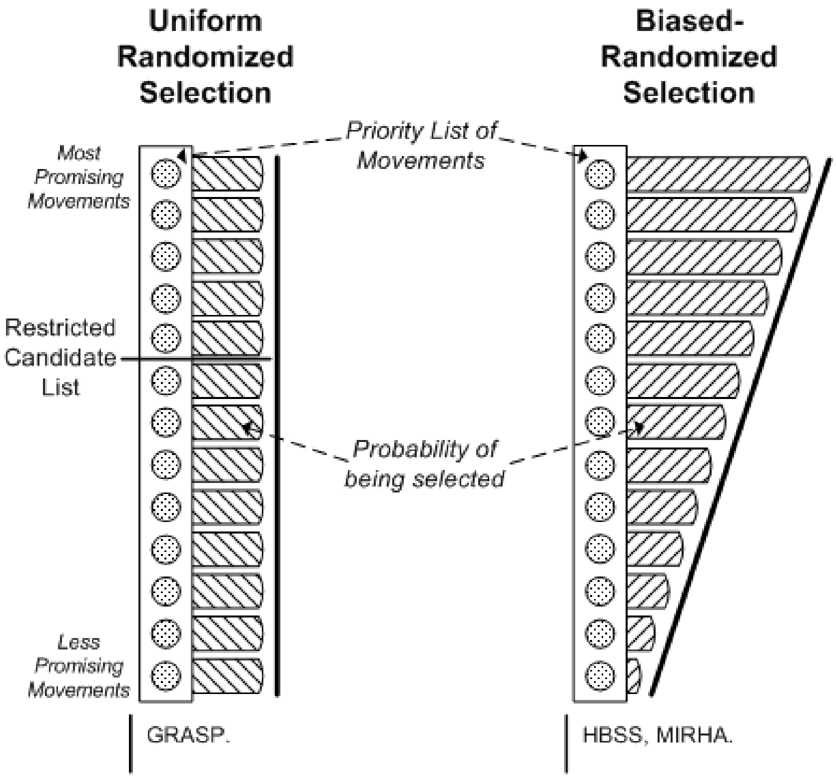


Figure 5.1: Uniform randomization vs. biased randomization.

—e.g., a priority list as the one described before; and (c) it should provide 'good' solutions which are not too far from the ones generated with more complex and time-consuming metaheuristics —e.g., average gap about 5-10%.

2. Once the base heuristic is selected, the new probabilistic algorithm should follow some kind of multi-start process —e.g., a pure multi-start or an iterated local search. At each round of this multi-start process, a new complete solution would be generated. For the construction of this solution, the base heuristic is randomized —e.g., its priority list is randomized- using a non-symmetric probability distribution. The specific distribution to employ will depend upon the specific COP being considered. Some candidate distributions to be considered are the geometric and a discrete version of the descendent triangular.

3. Optionally, a local search process can be added to the algorithm in order to improve the solution provided at each round of the multi-start process. This

local search is COP-tailored, meaning that it will be different for each COP.

The approach described above is able to quickly generate several feasible solutions with different properties. Therefore, a list containing the top 'best-found' solutions —each of them having different characteristics— can be saved and considered by the decision maker.

5.3 Benefits

The main motivation behind designing biased-randomized heuristics is to meet many of the desirable features of a metaheuristic as described by Cordeau et al. (2002), i.e.: accuracy, speed, simplicity, and flexibility. Most of the metaheuristics in literature are measured against accuracy —the degree of departure of the obtained solution value from the optimal value—, and against speed -the computation time. However, there are two other important attributes to be considered in any optimization method: simplicity and flexibility. Simplicity is related to the number of parameters to be set and the facility of implementation. This is an important feature since the method can be applied to different instances than the ones tested without losing quality or performance and without the need of performing a long run test. Flexibility is related to the possibility of accommodating new side constraints and also with the adaptation to other similar problems.

Having in mind these measured attributes, we list the main benefits of biased-randomized heuristics over other related approaches:

- They allow a simplification of the fine-tuning process, since typically the employed probability distributions require just one (e.g., the Geometric) or zero parameters (e.g., the descendent Triangular).
- Being based on classical well-tested heuristics, they are relatively simple and easy to implement methods, which can be adapted to account for new constraints (flexibility).
- Using non-uniform (biased) distributions rather than uniform distributions, they offer a more natural and efficient way to select the next movement from the priority list.

- By combining randomization with a multi-start-like process, they promote diversification during the exploration of the solution space, i.e., the search is not restricted to just a reduced number of regions.
- Likewise, the combination of randomization with a multi-start-like process promotes parallelization in an easy and natural way (Juan et al., 2013a).
- Finally, the biased-randomized heuristics can also be combined with other techniques, such as Monte-Carlo simulation, in order to consider stochastic variants of COPs (as we will see further in this thesis).

Most of the work developed so far in the area of stochastic local search and metaheuristics is based on the use of uniform randomization. As discussed in this chapter, probability distributions other than the uniform one can also be used to induce randomness inside heuristics or local search processes. In fact, the use of biased randomization, as proposed in this study, can contribute to make the search process even more efficient. For that reason, we expect to see a significant increase in the use of non-uniform distributions in all metaheuristics and probabilistic algorithms during the next few years.

5.4 Chapter Conclusions

In this chapter we have analyzed some key aspects, benefits, and examples related to the combination of randomization strategies with classical heuristics as a natural way to develop probabilistic algorithms to solve combinatorial optimization problems. Both uniform as well as non-symmetric randomization strategies have been reviewed. In particular, we have discussed how the non-symmetric or biased approach constitutes an efficient way to randomize the priority list of a constructive heuristic without losing the logic behind it. Some examples of applications to several combinatorial optimization problems have also been exposed, including: vehicle routing problems, arc routing problems, and flow-shop problems. One of the main advantages of this type of probabilistic algorithms is their relative simplicity, since they are based in well-known heuristics and they do not incorporate many additional parameters. Moreover, these algorithms are flexible, quite efficient, and can be implemented and parallelized in a natural way, which makes them an interesting alternative to more sophisticated metaheuristics in most practical applications. In next chapters, we will apply this methodology to create some tailored approaches for deterministic scenarios (see Fig. 5.2).

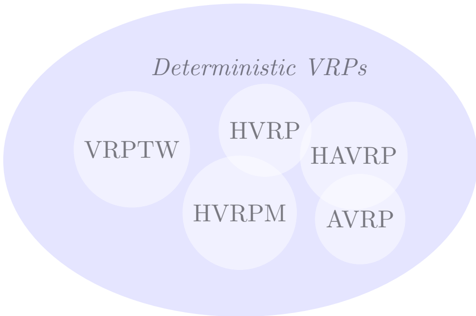


Figure 5.2: VRPs studied in this dissertation using biased-randomized heuristics.

6

Heterogeneous VRPs

Parts of this chapter have been taken from the co-authored publications:

Grasas, Cáceres-Cruz, Lourenço, Juan, and Roca (2013), OR Insight;

Juan, Faulin, Cáceres-Cruz, Barrios, and Martínez (2014b), European J. of Industrial Engineering.

In most real-life transportation scenarios, it is necessary to consider heterogeneous fleets (i.e., vehicles having different capacities) instead of homogeneous ones. In fact, most road-transportation companies own a heterogeneous fleet. This diversity in the capacity of vehicles might be due to the fact that different customers and locations might require different type of vehicles, e.g.: narrow roads in a city, available parking spaces, vehicle weight restrictions on certain roads, etc. Another reason for owning vehicles with distinct capacities is the natural diversity that arises when vehicle acquisitions are made over time. Accordingly, Privé et al. (2005); Ruiz et al. (2004) highlighted the importance of considering heterogeneous fleets while developing new vehicle routing methods. On this chapter, we will present some tailored-purpose approaches for some realistic variants of the VRP. This variant is the Heterogeneous fleet VRP where a real case application is also described. The real case includes some extra constraints like multi-trips and several involved function costs. There will be a preliminary literature review, the presentation of the proposed approach, and finally some computational results.

6.1 Definition

Several variants of the Heterogeneous fleets VRP ($HVRP$) have been proposed in the literature. For instance, Baldacci et al. (2008) presents a comprehensive description of some of them. In general, the research community has addressed the HVRP in different ways, first, considering an unlimited (i.e., $\forall k \in K$, $m_k = +\infty$) or limited number of vehicles, and second, minimizing a function cost based on a variable, fixed or both costs. Each vehicle could have a fixed cost for using it in a trip (i.e., $\forall k \in K$, $F_k \neq 0$) as well as a variable cost that is the result of multiplying a coefficient by the distance of the assigned route (i.e., $\forall k, l \in K$, $c_{ij}^k \neq c_{ij}^l$). These costs are associated to each type of vehicle. So the combinations of these aspects have created the main HVRP families, known as:

- Fleet Size and Mix VRP with fixed and variable costs (FSMVRP-FV) where an unlimited number of vehicles is considered for minimizing the addition of using a specific vehicle and the variable distance.

- Fleet Size and Mix VRP with only fixed costs (FSMVRP-F) where an unlimited number of vehicles is considered for minimizing fixed cost of all used vehicles.

- Fleet Size and Mix VRP with only variable costs (FSMVRP-V) where an unlimited number of vehicles is considered for minimizing the variable distance of all routes.

- Heterogeneous Fixed Fleet VRP with only variable costs (HVRP-V) where a limited number of vehicles is used to minimizing the variable cost.

- Heterogeneous Fixed Fleet VRP with fixed and variable costs (HVRP-FV) where a limited number of vehicles is used to minimizing both variable and fixed costs.

Notice that there are other HVRP branches considering constraints like Site-Dependent, Site-Road, etc. Another branch considers that each vehicle can optionally perform several trips (HVRPM).

6.2 Literature Review

One of the first published papers dealing with the FSMV-F is that of Golden et al. (1984). This paper defines the problem of optimal fleet design and configuration, and presents a mathematical model for it. The authors have proposed a set of benchmarks widely used. However, Taillard (1999) has improved them by adding the variable cost. This author has proposed an heuristic based on CG. Since then a large number of techniques have been developed for addressing this problem: TS techniques (Brandão, 2009; Gendreau et al., 1999; Wassan and Osman, 2002), Memetic Algorithms (MA) (Lima et al., 2004; Prins, 2009), heuristic-based method (Renaud and Boctor, 2002), and Genetic Algorithms (GA) (Liu et al., 2009).

Several techniques have been done to address the FSMV-V. Some of the main contributions can be found in Choi and Tcha (2007) (Branch-and-bound), Prins (2009), Imran et al. (2009) (Variable Neighbourhood Search) and Brandão (2011). Notice that Choi and Tcha (2007) provide an Integer Programming model. The FSMV-FV is also considered in some of the commented works, like Belfiore and Yoshizaki (2009); Choi and Tcha (2007); Imran et al. (2009); Prins (2009).

One of the most relevant works on HVRP-V is Li et al. (2007a) which developes a Record-To-Record (RTR) algorithm. Tarantilis et al. (2004) has developed two Threshold Accepting algorithms (TA). Other interesting approaches for this type of problem are presented in Brandão (2011); Ceschia et al. (2011); Euchi and Chabchoub (2010); Prins (2009); Yazgı-Tütüncü (2010).

A small number of works have considered the HVRP-FV. In fact, most of them are academic iniciatives. For instance, Baldacci and Mingozzi (2009) present a MIP model, introducing two new classes of inequalities to improve some of the variable bounds for the HVRP-FV. Li et al. (2010) has considered this type of routing problem as well.

The realistic aspect of this research line has produced recent studies considering many branches of HVRP at the same time, like that in Penna et al. (2013); Subramanian et al. (2012). These last two studies have proposed to apply methods based on an Iterative Local Search (ILS) heuristic. In order to address all families of HVRP, the first paper employs a VN Descent procedure and a random neighbourhood ordering. While the second applies a Set Partitioning (SP) formulation. However, some commented works have also addressed more than one type of HVRP. In Table 6.1, it can be

Authors	Year	Method	Heterogeneous Cap.	FSMVRP-FV	FSMVRP-F	FSMVRP-V	HVRP-V	HVRP-FV
Golden et al.	1984	Heuristics	√		√			
Gendreau et al.	1999	TS	√		√	√		
Taillard	1999	HCG	√		√		√	
Renaud and Boctor	2002	Heuristics	√		√			
Wassan and Osman	2002	TS	√		√	√		
Lima et al.	2004	MA	√		√			
Tarantilis et al.	2004	TA	√				√	
Choi and Tcha	2007	B&B	√	√	√	√		
Li et al.	2007a	RTR	√				√	
Belfiore and Yoshizaki	2009	SS	√	√				
Brandão	2009	TS	√		√	√		
Imran et al.	2009	VN	√	√		√		
Liu et al.	2009	GA	√		√			
Prins	2009	MA	√	√	√	√	√	
Euchi and Chabchoub	2010	TS	√				√	
Li et al.	2010	MetaHeuristic	√					√
Yazgı-Tütüncü	2010	MetaHeuristic	√				√	
Brandão	2011	TS	√			√	√	
Ceschia et al.	2011	TS	√				√	
Subramanian et al.	2012	ILS	√	√	√	√	√	√
Penna et al.	2013	ILS	√	√	√	√	√	√

Table 6.1: Published HVRP studies.

appreciated a summary of commented studies considering several types of HVRP. Some problems have been more studied than others. This table presents papers exclusively dedicated to different types of HVRP.

There are VRP versions of the problem that consider multi-trip, i.e., a vehicle can do more than one trip on the same planning period. This is a very common case for distribution and retailing companies, since most of these have a limited number of vehicles. Although this multi-trip feature is very relevant in practice (see Baldacci et al. (2008); Golden et al. (2008); Şen and Bülbül (2008) for more information), few authors have addressed it. The methods usually applied to solve this multi-trip version are based on the CWS method and Tabu Search approaches. Fleischmann (1990); Prins (2002) combine a Savings heuristic with a Bin Packaging Problem technique (BPP). The BPP is also used in the work of Petch and Salhi (2003), where the authors combine this technique with the savings method in a multi-phase approach. Taillard et al. (1996) propose a Tabu Search metaheuristics and also define a set of instances for this problem that have been used by other authors. Brandão and Mercer (1998); Olivera and Viera (2007) present also a Tabu Search metaheuristic for the multi-trip version of the HVRP. Multi-objective approaches have been also proposed in Lin and Kwok (2006), which is based on Tabu Search and Simulated Annealing meta-heuristics. The authors have compared their approaches with real data and simulated data. Notice that several of these authors include a real application of their algorithms to test the performance of their approach. Notice that the CWS approaches offer a simple way to develop algorithms to solve real problems (Poot et al., 2002), and unlike the Tabu Search, there is not need to perform a fine-tuning process in order to get a good performance. However the combination of heterogeneous fleets and multiple trips is quite uncommon in the literature.

Summarizing, we consider the Heterogeneous Fleet and Multitrip Vehicle Routing Problem. In particular, we consider the following additional considerations regarding the available fleet and its costs:

- The number of vehicles of each type, m_k.
- For each vehicle type:
  - The fixed costs could be ignored (i.e., $F_k = 0, \forall k \in K$) or not;
  - The routing costs could be vehicle-independent (i.e., $\forall k \in K, \gamma_k = 1$, so $\gamma_k \cdot d_{ij} = \gamma_l \cdot d_{ij} = c_{ij}^k = c_{ij}^l = c_{ij}, \forall k, l \in K, \forall i, j \in \Omega$) or not;
  - There are no restrictions on the customers they can visit (due to size or maneuverability, for example).
- Some vehicles can make multiple trips from the depot (i.e., multi-trips).

6.3 Proposed Approach

This section provides an overview of our approach for solving the HVRP as well as the HVRPM. We discusses some of its main design properties, such as: (a) the biased randomization of the MER heuristic, which allows transforming the MER deterministic heuristic (Prins, 2002) into a multi-start probabilistic algorithm; and (b) the use of two additional local search methods developed in Juan et al. (2011e), which are based on cache and splitting techniques. It should be mentioned at this point that the MER heuristic for solving the HVRP is based on the popular savings heuristic for solving the homogeneous VRP (Clarke and Wright, 1964). Fig. 6.1 depicts a high-level flowchart of the proposed algorithm, which overall description is included next.

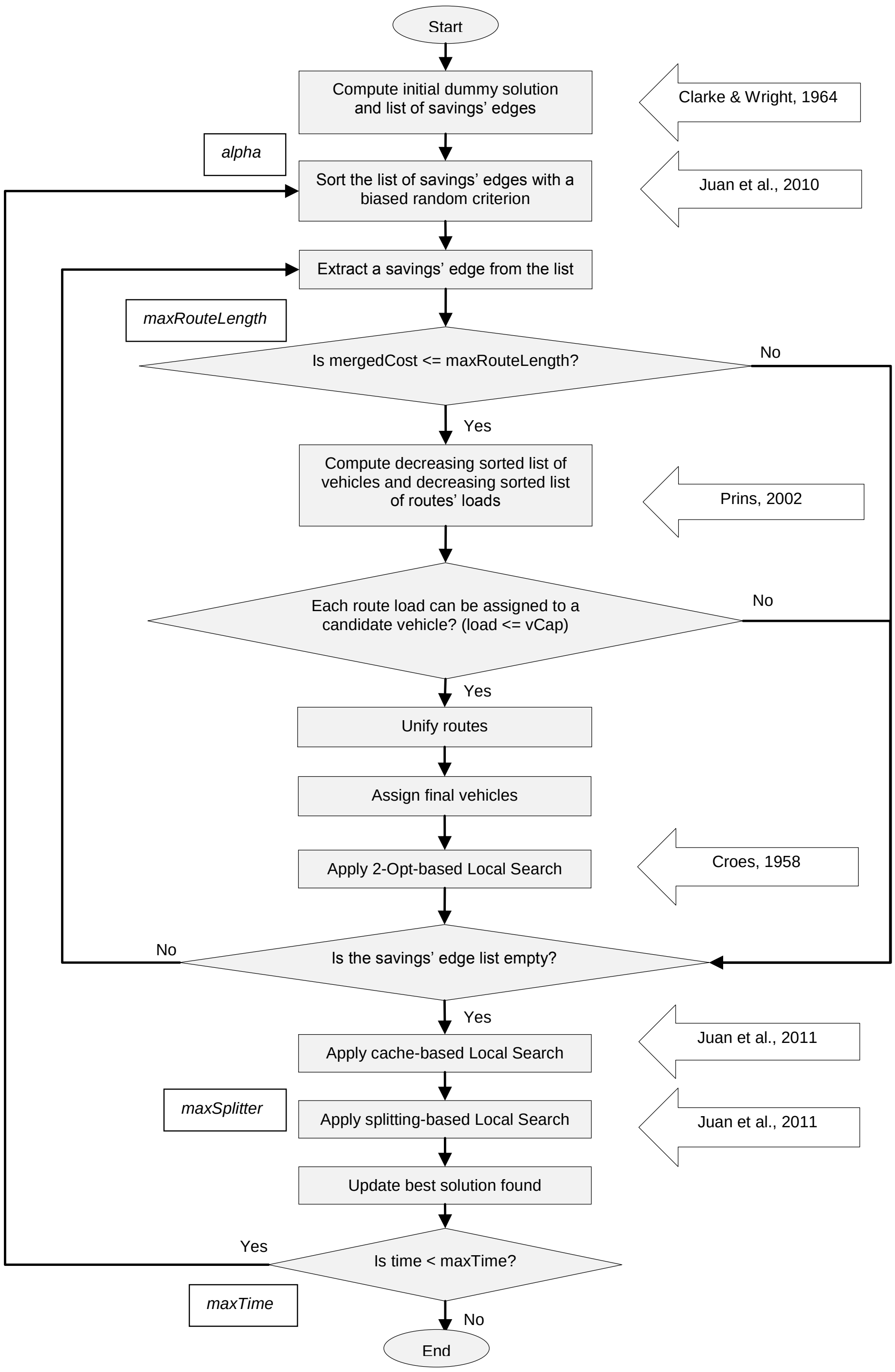


Figure 6.1: Overview of our HVRP approach.

We propose a general approach for both HVRP and HVRPM. So given a instance, first the algorithm constructs an initial solution as proposed in the classical CWS heuristic. In this initial solution, a virtual truck is assigned to each customer. Also as proposed in the aforementioned heuristic, the algorithm computes the savings associated with each edge connecting two different customers. Put in simple terms, the savings associated with a given edge are computed as the reduction in costs (distance and/or time-based) due to the use of this edge for merging two different routes into a single new one. The edges are then stored into a list, which is sorted from highest to lowest savings. At this point, a multi-start procedure begins. Typically, this procedure is executed over and over until a time-based ending condition is reached. At each iteration of this multi-start procedure, the following steps are performed:

1. A new savings list of edges is obtained by randomizing the original savings list throughout the Geometric probability distribution, as suggested in Juan et al. (2010). By randomizing the savings list, the deterministic heuristic described in step 2 is transformed into a probabilistic method. This allows obtaining different outputs at each iteration of the multi-start procedure. Furthermore, by using a biased probability distribution —the Geometric in this case— most of the logic behind the classical savings heuristic is kept, i.e.: edges with higher savings will be more likely to be selected from the list than those with lower savings.

2. Then, until the savings list gets empty, an iterative process begins in which the edge at the top of the randomized list is extracted. This edge will connect two different routes. In order to merge these two routes, the extreme points of the edge must be 'external' to their respective routes, i.e., they need to be directly connected to the depot. Moreover, both capacity and maximum-route-length constraints must be validated. A similar method to the one proposed in Prins (2002) is used to validate the capacity constraint in a heterogeneous fleet, i.e.: the list of vehicles is sorted from highest to lowest capacity, while the list of routes is sorted from highest to lowest accumulated demands; after that, a temporary assignment between the two lists is searched; if a successful match including all previously merged routes plus the new one is found, then the capacity constraint is validated and the temporary assignment becomes final; otherwise, the temporary

assignment is discarded and the merge becomes unfeasible. After each merge, a fast 2-Opt local search Croes (1958) is run over the new route.

3. Once all the edges in the savings list have been considered, the resulting solution is then improved throughout the two local search methods proposed in Juan et al. (2011e): First, a cache (hash-table) of routes is employed to quickly update any route in the current solution by the best-known route —among those routes found in previous iterations— covering the same set of nodes. Secondly, using proximity criteria, the current solution is divided into several sets of routes together with their associated vehicles; then, each of these subsets is considered as a smaller and feasible HVRP problem over which steps 1 and 2 above can be applied to find better 'local' routing plans. As a last step, the enhanced solution provided by these local search methods is compared against the best solution obtained so far by the multi-start procedure and, whenever appropriate, this best solution is updated.

Eventually, once the time-based criterion is reached, the best solution found by the described multi-start procedure is the one returned. An interesting property of this approach is that it can be naturally and easily parallelizable. In effect, due to its probabilistic nature, the searching path followed by the aforementioned procedure greatly depends upon the seed employed to initialize the pseudo-random number generator —which is used during the randomization of the savings list. Therefore, using an object-oriented programming terminology, it is possible to simultaneously run different 'instances' of the algorithm 'class' by simply changing the initial seed. These independent instances can then be run in different threads, cores, or even computers, as discussed in Juan et al. (2013a).

A lower-level description of the proposed algorithm is presented in (Pseudo-code 3), called RandCWS-Prins. The input data are the nodes information (geographical location and individual demands), the costs of moving from one node to another, and the fleet composition. This procedure requires parameters: (a) *alpha*, the one associated with the Geometric distribution employed during the randomization process of the savings list; (b) *maxTime*, the stopping time of the multi-start process; (c) *maxRouteLength*, associated with the maximum route length allowed; and (d) *maxSplitter*, associated with the maximum number of iterations for the splitting local search.

First (line 2), the procedure generates a list of edges connecting any two nodes. This list is sorted according to the savings obtained when using each edge. Then, an initial dummy solution is generated (line 3). In this solution, one round-trip route starting at the depot is considered for each client. After that, a multi-start process begins (lines 4-28). This multi-start process is especially useful for several reasons: (a) it allows the randomized algorithm to escape from local minima; and (b) it facilitates parallelization of the approach —this can be achieved by running different agents or threads of the algorithm with the same instance, each one using a different seed for the pseudo-random number generator. At each iteration of the multi-start process, a biased randomization of the savings list is produced (lines 5-6). At the end of each iteration, a new solution is iteratively constructed by merging routes, if feasible, according to the randomized list (lines 7-20). Several fast Local Search techniques are applied in order to improve values. Each merged route is improved through a classical 2-Opt local search process (line 18). Once a solution is generated, a memory-based local search process is applied (line 21). In this process, each route in the solution is checked against a cache, which contains the best found-so-far route covering the same set of nodes. Moreover, if the data instance is provided with the coordinates of nodes and the generated solution is considered as *promising*, a divide-and-conquer process is also applied (lines 22-24). Eventually, the algorithm returns the best solution produced in the entire multi-round process (line 29).

Algorithm 3 General Pseudo-code of the Randomized CWS-Prins Algorithm.

```
procedure RandCWS-Prins(nodes, costs, vehs)
    savingsList ← computeSavingsList(nodes, costs)
    initialSol ← constructInitialSol(nodes, costs)    ▷ Build dummy solution
    while stopping criterion is not satisfied do    ▷ Multi-start process defined by MaxTime parameter
        currentSol ← initialSol    ▷ Reset the dummy solution as the base
        randomSavingsList ← biasedRandomization(savingsList)    ▷ Biased randomized selection of savings
        while randomSavingsList is not empty do    ▷ Execute the route merging process
            savEdge ← selectNextEdge(randomSavingsList)
            rA ← getStartingRoute(savEdge)    ▷ Route A
            rB ← getClosingRoute(savEdge)    ▷ Route B
            mergeIsValid ← validateConstraints(savEdge, rA, rB, vehs, currentSol)    ▷ Check merging conditions for CWS-HVRP approach
            if mergeIsValid then
                unifiedRoute ← unifyRoutes(routeA, rB, savEdge)    ▷ Merge routes A and B into A
                currentSol ← deleteRoute(currentSol, rB)
                for eachRouteInCurrentSol do
                    route ← assignFinalVehicle(getCandidateVehicle(route))    ▷ Assign final vehicles to routes
                end for
                unifiedRoute ← improveWithTwoOpt(unifiedRoute)    ▷ Optimize route applying a Local Search
            end if
        end while
        currentSol ← improveSolutionUsingCache(currentSol, costs, vehs)
        if currentSol is promising then
            currentSol ← improveSolutionUsingSplitting(currentSol, costs, vehs)
        end if
        if currentSol outperforms bestHvrpSol then
            bestHvrpSol ← currentSol
        end if
    end while
    return bestHvrpSol    ▷ Return the best found solution so far
end procedure
```

One of the key steps in our approach is the vehicle assignment process that takes place during the merging of any two routes. Pseudo-code 4 shows the logic flow of the procedure employed to validate a potential merging as feasible. Notice that in order to merge two routes, three conditions must be satisfied. First (lines 2-4), both nodes in the connecting edge must be external nodes, i.e., both have to be directly connected with the depot. Secondly, the length of the merged route cannot be greater than the maximum allowed. Finally, it must be possible to cover each merged route with a truck. So we are facing an assignment problem (lines 5-15). To check this condition, all merged routes (including the new one) are sorted from the highest to the lowest aggregated demand (line 5), while all vehicles are similarly sorted by capacity (line 6). Then, starting from the top of both lists, the next vehicle is assigned to the next merged route as far as the truck capacity can cover the route demand (lines 8-15). If this assignment is not feasible after a certain point, then multi-trips are considered, i.e.: vehicles already covering one route are assigned to a second one as far as travelling times allow to cover both routes in the specified time period. Of course, if some merging routes cannot be covered by any vehicle, then the potential merging process is discarded and a new potential merging is considered as far as the edges list is not empty. Notice that our approach allows for the realistic multi-trip scenario and, at the same time, it tries to use all vehicles in the fleet before assigning additional trips to some of them. This is a relevant difference with regards the vehicle assignment proposed in Prins (2002), where multi-trips of larger vehicles are promoted and preferred over the use of the entire fleet. In our case, however, the company was interested in using the entire fleet in order to reduce total delivery times as much as possible (for customer satisfaction).

Algorithm 4 Validation for merging of two routes.

procedure VALIDATEMERGECONSTRAINTS($edge, routeA, routeB, vehicles, solution$)
  if (*nodes of edge are internals in routeA or routeB*) *OR* ($cost(routeA) + cost(routeB) - saving(edge) > MaxRouteLength$) **then**
    return false
  end if
  $candidateSolution \leftarrow getRoutesWithNewCandidate(solution, routeA, routeB)$ ▷ Vehicle Assignment Problem: create a new candidate route joining routes A and B and deleting these two from solution
  $routeList \leftarrow getSortedRouteList(candidateSolution)$ ▷ Sort list of all routes in decreasing order of loads
  $vehicleList \leftarrow getSortedVehicleList(vehicles)$
  for *each route in routeList* **do**
    $vehicle \leftarrow getFirstAvailableVehicle(vehicleList, route)$ ▷ Assign each route to the first bigger-free truck
    if $capacity(vehicle) < load(route)$ **then**
      return false
    end if
    $route \leftarrow setCandidateVehicle(vehicle)$
  end for
  return true
end procedure

6.4 Computational Results

While there are some standard benchmarks for the homogeneous (capacitated) VRP, this is not the case for the heterogeneous VRP, where several authors have proposed different sets of benchmarks depending on the specific version they are dealing with, e.g.: with or without fixed and/or variable costs associated with the use of each type of vehicle, with our without multi-trips, etc. Even worse, most authors have proposed instances which make use of probability distributions to generate the spatial coordinates of customers, as well as their associated demands. In our opinion, this is not a good practice since just from the specific probability distributions it is not possible to reproduce the exact customers' coordinates and/or demands —which makes it difficult to reproduce the experiments. For those reasons, we have decided to use three different testbeds in order to measure the performance of our approach:

1. Prins' instances: Proposed in Prins (2002), these are twenty random instances, denoted as P_i with $i \in \{1, \ldots, 20\}$. Each instance contains 100 customers uniformly distributed in a 200 x 200 km^2 grid. Each customer's demand is uniformly distributed in [1, 100]. The depot is placed at the center of the grid, and the maximum time allowed per route is 300 minutes (or 350 km at a speed of 70 km/h). The fleet is composed of $k = 9$ types of vehicles with $m_k = 2, \forall k \in \{1, 2, \ldots, 9\}$. Each type of vehicle has a capacity given by $Q_k = 600 - 50(k - 1)$.

2. Golden and Taillard instances: the first work (Golden et al., 1984) proposed 20 instances for the FSMVRP of different sizes, and the second (Taillard, 1999) defined the number of available vehicles of each type. The first 12 instances are quite small —they have less than 50 nodes-, so we have not considered them. Also, instances 13, 16, and 18 cannot be solved with the MER heuristic since they do not satisfy some of the Prins' assumptions. For our algorithm, we selected eight test instances, denoted as GT_i with $i \in \{13, \ldots, 20\}$. The number of customers in these instances, originally proposed by Christofides and Eilon (1969), is between 50 and 100. Information about the fleet composition in these instances is displayed in Table 6.2.

3. Li instances: five large-scale HVRP instances (Li et al., 2007a), inspired in the previously commented and denoted as H_i with $i \in \{1, \ldots, 5\}$. The number of customers in these instances is between 200 and 360. Each instance has a geometric symmetry, with nodes located in concentric circles around the depot. Information about the composition of the fleets for these instances is displayed in Table 6.3.

To test the aforementioned set of instances, both the MER and RandCWS-Prins algorithms have been implemented as a Java application. Notice that on this experiments our objective function is minimizing distance. These implementations have been executed on a Java Virtual Machine (JVM) version 1.6 using a computer with the following characteristics: a Windows 7 Professional SP1 64 bits operating system, an Intel Xeon E5603 1.60Ghz processor, and 8 GB of RAM.

We run 10 independent executions per instance, each of them using a different seed for the pseudo-random number generator. Each execution was run for 1 minute.

Instance	Q_A	γ_A	m_A	Q_B	γ_B	m_B	Q_C	γ_C	m_C	Q_D	γ_D	m_D	Q_E	γ_E	m_E	Q_F	γ_F	m_F	%
GT_{13}	20	1.0	4	30	1.1	2	40	1.2	4	70	1.7	4	120	2.5	2	200	3.2	1	95.39
GT_{14}	120	1.0	4	160	1.1	2	300	1.4	1										88.48
GT_{15}	50	1.0	4	100	1.6	3	160	2.0	2										94.76
GT_{16}	40	1.0	2	80	1.6	4	140	2.1	3										94.76
GT_{17}	50	1.0	4	120	1.2	4	200	1.5	2	350	1.8	1							95.38
GT_{18}	20	1.0	4	50	1.3	4	100	1.9	2	150	2.4	2	250	2.9	1	400	3.2	1	95.38
GT_{19}	100	1.0	4	200	1.4	3	300	1.7	3										76.74
GT_{20}	60	1.0	6	140	1.7	4	200	2.0	3										95.92

Table 6.2: Specifications for 8 Golden and Taillard instances with 6 Vehicle Types [Column '%': 100 x (total demand/total capacity)].

Instance	Q_A	γ_A	m_A	Q_B	γ_B	m_B	Q_C	γ_C	m_C	Q_D	γ_D	m_D	Q_E	γ_E	m_E	Q_F	γ_F	m_F	%
H_1	50	1.0	8	100	1.1	6	200	1.2	4	500	1.7	3	1000	2.5	1				93.02
H_2	50	1.0	10	100	1.1	5	200	1.2	5	500	1.7	4	1000	2.5	1				96.00
H_3	50	1.0	10	100	1.1	5	200	1.2	5	500	1.7	4	1000	2.5	2				94.76
H_4	50	1.0	10	100	1.1	8	200	1.2	5	500	1.7	2	1000	2.5	2	1500	3.0	1	94.12
H_5	50	1.0	10	100	1.2	8	200	1.5	5	500	1.8	1	1000	2.5	2	1500	3.0	1	92.31

Table 6.3: Specifications for 5 Li instances with 6 Vehicle Types [Column '%': 100 x (total demand/total capacity)].

Table 6.4 shows a comparison, for the three data sets, between the outcomes obtained with the MER heuristic and our algorithm. The first columns describe the name of the instance, the number of visited customers, and the total delivered demand. Next, for each algorithm the number of routes in the corresponding solution and the computation time are given. The last column shows the percentage gap between both solutions. Notice that our approach clearly outperforms the MER heuristic (average gap about 4.76%). However, as discussed before, the exact values obtained in Prins (2002) could not be replicated due to the use in that paper of random inputs.

Results in Table 6.4 are useful to directly compare our algorithm and the MER heuristic —which, to the best of our knowledge, is the only one considering the HVRPM. However, unlike the Prins and Li instances, instances GT_{13}, GT_{16}, and GT_{18} do not satisfy the MER assumption that any vehicle can cover any customer demand. This constraint can be somewhat unrealistic. In fact, we have been involved in a real case (described in the next section) in which a company delivers to large-size customers which cannot be covered with the smallest vehicles in its fleet.

			MER			RandCWS-Prins					
Instance	n	Total Delivered Demand	Cost (1)	M	Time (sec)	Best Cost (2)	M	Time (sec)	Gap (2-1)	Average 10 Seeds (3)	Gap (3-1)
P_1	100	4806	2537.62	10	0.09	2457.29	10	8	-3.17%	2482.2	-2.18%
P_2	100	4895	2731.76	11	0.14	2591.28	10	7.79	-5.14%	2604.8	-4.65%
P_3	100	5218	2712.66	11	0.07	2672.8	11	6.23	-1.47%	2691.01	-0.80%
P_4	100	4763	2398.98	10	0.08	2316.77	10	25.55	-3.43%	2337.92	-2.55%
P_5	100	5069	2750.92	11	0.06	2586.02	11	5.59	-5.99%	2611.97	-5.05%
P_6	100	4896	2711.17	10	0.09	2613.88	10	7.13	-3.59%	2619.71	-3.37%
P_7	100	5268	2844.43	12	0.07	2705.84	11	12.44	-4.87%	2726.04	-4.16%
P_8	100	5310	2812.29	12	0.05	2680.22	12	30.69	-4.70%	2709.45	-3.66%
P_9	100	5403	2757.86	12	0.06	2600.38	12	5.83	-5.71%	2623.62	-4.87%
P_{10}	100	4462	2426.23	10	0.09	2365.94	9	7.28	-2.48%	2379.53	-1.92%
P_{11}	100	5269	2832.11	12	0.06	2653.04	12	6.24	-6.32%	2695.42	-4.83%
P_{12}	100	4860	2645.14	10	0.07	2510.77	10	5.69	-5.08%	2548.09	-3.67%
P_{13}	100	4772	2626.43	10	0.09	2454.11	10	8.1	-6.56%	2464.2	-6.18%
P_{14}	100	5065	2658.18	11	0.08	2573.93	11	47.16	-3.17%	2584.44	-2.77%
P_{15}	100	4716	2768.28	10	0.09	2597.41	10	8.4	-6.17%	2617.78	-5.44%
P_{16}	100	5504	3013.82	13	0.05	2813.44	12	5.69	-6.65%	2839.76	-5.78%
P_{17}	100	4632	2563.45	10	0.08	2381.5	10	38.03	-7.10%	2401.9	-6.30%
P_{18}	100	4727	2517.39	11	0.08	2411.38	10	49.1	-4.21%	2427.3	-3.58%
P_{19}	100	5398	3003.3	12	0.06	2857.48	12	39.16	-4.86%	2874.62	-4.28%
P_{20}	100	4730	2539.31	10	0.10	2415.98	10	14.48	-4.86%	2452.3	-3.43%
		Average	2692.57	10.9	0.08	2562.97	10.65	16.93	-4.78%	2584.6	-3.97%
GT_{13}	50	973	NA	NA	NA	821.34	17	37.64	NA	844.93	NA
GT_{14}	50	973	569.76	7	0.01	539.01	6	39.12	-5.40%	540.9	-5.07%
GT_{15}	50	777	677.57	9	0.01	633.79	9	9.24	-6.46%	634.17	-6.40%
GT_{16}	50	777	NA	NA	NA	637.94	9	3.48	NA	638.47	NA
GT_{17}	75	1364	796.14	11	0.04	770.54	10	5.11	-3.22%	775.66	-2.57%
GT_{18}	75	1364	NA	NA	NA	787.57	12	19.87	NA	788.53	NA
GT_{19}	100	1458	853.15	6	0.17	760.24	6	73.28	-10.89%	773.94	-9.29%
GT_{20}	100	1458	1032.78	13	0.08	984.8	13	8.26	-4.65%	998	-3.37%
		Average	767.73	9.17	0.05	741.9	10.25	24.5	-6.12%	749.33	-5.34%
H_1	200	4000	9188.1	18	3.16	8459.35	18	32.25	-7.93%	8637.68	-5.99%
H_2	240	4800	6535.64	25	8.27	6457.83	24	29.89	-1.19%	6480.85	-0.84%
H_3	280	5600	11121	20	19.03	10990.18	19	76.63	-1.18%	11114.85	-0.06%
H_4	320	6400	9480.08	23	53.96	9331.28	21	59.67	-1.57%	9413.13	-0.71%
H_5	360	7200	12502.36	18	208.65	11918.75	18	143.71	-4.67%	12360.08	-1.14%
		Average	9765.44	20.8	58.61	9431.48	20	68.43	-3.31%	9601.32	-1.75%

Table 6.4: Results on 3 Datasets after 10 minutes of Execution for each instance (single trip case).

6.4.1 HVRP with Variable Costs

For considering, variable costs (see Eq. 6.1) we had made a specific change in the previously proposed algorithm. As we have commented, there is an assignment problem inside of the randomized CWS. So we thought appropriate to complement the vehicle assignment proposed by Prins with a polynomial method tailored to this type of problem. Therefore we decided to use the popular Hungarian Algorithm reviewed by Munkres (1957). This algorithm works particularly well with small instances. So we introduce a new parameter ($limitMunkres$) to decide when to use this procedure and when to used the Prins one. As starting value, we set this value on 30 because it was a reasonable value regarding the nature of instances.

$$\min \sum_{i \in \Omega} \sum_{j \in \Omega} \sum_{k=1}^{M} \gamma_k \cdot c_{ij}^k \cdot x_{ij}^k \tag{6.1}$$

As in the previous experiment, we run 10 independent executions per instance. Each execution was run for 1 minute. Table 6.5 shows a comparison, for the Golden and Taillard data set, between the outcomes obtained by Li et al. (2007a) and the new version of our algorithm. The first columns describe the name of the instance and the Best Known Solution. Next, the number of routes in the corresponding solution and the computation time are given. The last column shows the percentage gap between both solutions. The average gap of best solutions is about 2.08%. However, the values get worse while the instances get bigger. Notice that our approach is easily adaptable to different routing scenarios. Also that some distance values (e.g., GT_{13} and GT_{15}) are even worst than the previous experiments because on this we are evaluating a different objective function.

6.5 Real Case I: HVRP with Multiple Trips

The distribution company of this study distributes products from its central facilities in the Northeast of Spain to a chain of around 400 stores all over the country. Orders from every store are received daily (i.e., Monday through Saturday), and the distribution is then carried out from a central depot by a company-owned fleet of 169 vehicles. This fleet includes trucks of different capacities (see Table 6.6 for the fleet composition). At a glance, the daily distribution planning process unfolds as follows:

Instance	BKS (1)	M	Distance	Best Variable Cost (2)	Gap (2-1)	10 seeds Variable Cost (3)	Gap (3-1)
GT_{13}	1517.84	17	841.57	1522.27	0.29%	1566.29	3.19%
GT_{14}	607.53	7	537.57	609.17	0.27%	623.20	2.58%
GT_{15}	1015.29	9	655.79	1019.55	0.42%	1026.90	1.14%
GT_{16}	1144.94	9	659.39	1148.62	0.32%	1155.75	0.94%
GT_{17}	1061.96	11	785.03	1078.80	1.59%	1097.62	3.36%
GT_{18}	1823.58	12	821.42	1862.84	2.15%	1921.53	5.37%
GT_{19}	1117.51	6	767.68	1190.21	6.51%	1216.92	8.90%
GT_{20}	1534.17	13	982.31	1612.86	5.13%	1634.81	6.56%
Average		10.50	753.51	1,255.97	2.08%	1,280.38	4.01%

Table 6.5: Results on 1 Dataset after 10 minutes of Execution for each instance (single trip case with variable cost).

- Order placement: stores place orders by noon with no restriction on the number of boxes.
- Order planning: orders are received at the central depot and may be adjusted depending on product availability.
- Route planning: three route dispatchers plan routes to all stores by 2pm (see more details on route planning below).
- Distribution: vehicles load the cargo at the depot and depart to the stores. Truck loading is divided into three shifts (at 2pm, 3pm, and 5pm, respectively).
- Delivery: vehicles arrive at the stores between 5pm and 1am of the next day, and unload their cargo.
- Return to depot: after the last store in the route is served, vehicles return to the depot.

The route planning step establishes the routes that vehicles must follow to deliver the products. This phase is obviously the crucial step in the distribution process as it determines most of the total distribution costs. Currently, this task is executed manually by three route dispatchers. They divide all stores into three geographical areas, so that each dispatcher is responsible for the routes in her region (that is, each of them solves a smaller VRP). For each region, they have sets of predetermined routes that modify slightly according to daily demand and truck availability. Trucks and stores are usually assigned to one of the loading shifts, so that routes include stores in the same shift only. In addition, there exist other specific constraints on the routing

k	Q_k	m_k	MQ_k	AM_k	MQM_k
Veh_A	222	8	1,776	1	1,776
Veh_B	414	5	2,070	1	2,070
Veh_C	482	139	66,998	2	133,996
Veh_D	550	3	1,650	1	1,650
Veh_E	616	6	3,696	1	3,696
Veh_F	676	3	2,028	1	2,028
Veh_G	752	4	3,008	1	3,008
Veh_H	1,210	1	1,210	1	1,210
Total		169	75,691		149,434

Table 6.6: Fleet composition of the distribution company.

problem that make this real VRP quite unique. These need to be considered when designing routes:

- The number of trucks available each day and shift may vary due to eventualities.
- Not all types of vehicles can visit all stores. For example, large trucks cannot access some stores for maneuverability reasons.
- Some stores have restrictions on their delivery times. For example, trucks may not be allowed in some urban areas before (or after) a determined time. These are known as delivery time windows.
- Some trucks are allowed to make multiple trips (generally two). In days of high demand, for example, the total capacity of all available vehicles may not be enough to cover all demand, so that some trucks perform two trips on that same day. This implies that some stores could be visited twice by the same truck or by a different one, having to split their order in two.
- Each truck is driven by a single driver, so there is an upper bound on the route duration given by the maximum number of working hours (i.e., 8 hours).

With all this information and constraints, the three company dispatchers have about one hour every day to configure the delivery routes. The planning is done manually with some computer aid to perform simple verifications (like tracking the number of boxes yet to be assigned). As the number of stores continues to grow, the need for a scientific

Instance	n	Total Demand Delivered	Instance	n	Total Demand Delivered
A	372	77913	L	368	67875
B	366	79130	M	313	35373
C	371	91901	N	370	70199
D	372	83571	O	371	65007
E	373	85773	P	364	63078
F	372	84023	Q	315	32006
G	374	85539	R	373	71662
H	370	89596	S	372	65869
I	372	76846	T	366	62362
J	372	94892	U	314	30211
K	373	83901	V	374	67663
			W	371	63941
			X	368	61770
			Y	315	34455

Table 6.7: General Features of involved Real Instances.

method to help the decision making becomes more latent. This is a very complex problem that requires more sophisticated methods to obtain better and faster solutions that allow the company to save considerable costs in transportation. To illustrate the nature of the involved instances we summarize the information for 25 real instances from 25 business days in 2011 (see Table 6.7). However, this study only consider the multiple trips —previous fourth condition.

6.5.1 Computational Results

Before running the algorithm to solve the problem, all necessary input data had to be compiled and prepared in the appropriate format. This basically refers to all problem parameters and constraints which include data from all stores (demands and postal addresses), vehicle capacities, truck-store incompatibilities, delivery time windows and maximum time per route (i.e., at most 8 hours per route). With all addresses, including that of the depot, we constructed distance and time matrices. These matrices contained all travel distances and times between every pair of stores, and between all stores and the depot. For about 400 stores plus a depot, this implied finding around 160,000 distances and times. To automate this quest, we developed a web application `http://vrp.upf.edu` that uses Google Maps in which the user uploads an Excel file with all addresses, and the application returns, in few seconds, a plain text file with the matrices in a format ready for our algorithm. Notice that our function cost is focused on minimizing total travelling distance.

To illustrate the performance of the algorithm we summarize the results obtained for 11 multi-trips real instances (see Table 6.8) and 14 single-trips instances (see Table 6.9). We imposed a time bound of ten minutes to the algorithm ($maxTime$). For each instance, it shows the number of stores visited that day as well as the total demand (in boxes) delivered. This table compares the solution obtained by the company dispatchers with our solution, showing the total logistics cost and the number of routes employed. Notice that more routes does not necessarily imply higher costs as can be seen in some instances. In Table 6.8, multi-trips appear in almost all solutions as expected, that is, those in which the number of routes exceeds the total number of vehicles (i.e., 169). In Table 6.9, the magnitude of improvements is bigger even when the instances are smaller and single-trip. The algorithm was executed using 10 different seeds per instance. This table reports both the average cost of the 10 runs and the best solution found. The best solution for each instance was obtained in few seconds and the general average cost reduction was around 15%, which represents savings of around €6,000 per day.

	Company Solution		CWS-MofPrins				RandCWS-ModPrins							
Instance	Cost (1)	M	Cost (2)	M	Time (sec)	Gap (2-1)	Best Cost (3)	M	Time (sec)	Gap (3-1)	Gap (3-2)	Average 10 Seeds (4)	Gap (4-1)	Gap (4-2)
A	45302.40	173	40001.86	182	3.02	-11.70%	39534.11	180	63.22	-12.73%	-1.17%	39841.99	-12.05%	-0.40%
B	47184.35	182	41821.23	187	2.41	-11.37%	41072.46	183	55.20	-12.95%	-1.79%	41399.65	-12.26%	-1.01%
C	53941.43	218	50337.90	221	2.90	-6.68%	49669.31	219	66.48	-7.92%	-1.33%	50082.32	-7.15%	-0.51%
D	50872.80	197	46372.77	202	2.70	-8.84%	45485.83	200	29.73	-10.59%	-1.91%	45836.63	-9.90%	-1.16%
E	51315.40	207	46327.54	211	2.76	-9.72%	45275.62	206	7.67	-11.77%	-2.27%	45681.39	-10.98%	-1.39%
F	50492.74	200	45939.70	199	2.60	-9.02%	45165.12	197	28.53	-10.55%	-1.69%	45493.28	-9.90%	-0.97%
G	51427.10	208	45070.87	202	2.76	-12.36%	44386.64	200	65.94	-13.69%	-1.52%	44909.39	-12.67%	-0.36%
H	54446.01	215	49613.35	214	2.88	-8.88%	49053.97	212	59.57	-9.90%	-1.13%	49354.83	-9.35%	-0.52%
I	45056.40	172	39712.54	177	2.25	-11.86%	38973.19	175	29.33	-13.50%	-1.86%	39252.86	-12.88%	-1.16%
J	60246.80	245	51813.56	231	3.18	-14.00%	51145.95	229	0.54	-15.11%	-1.29%	51444.54	-14.61%	-0.71%
K	50296.10	199	44392.45	199	2.57	-11.74%	44052.29	200	43.16	-12.41%	-0.77%	44197.63	-12.13%	-0.44%
Average	50,961.96	201.45	45,582.16	202.27	2.73	-10.56%	44,892.23	200.09	40.85	-11.92%	-1.52%	45,226.77	-11.26%	-0.78%
Total	560,581.53		501,403.77				493,814.49					497,494.51		

Table 6.8: Results applying Randomized CWS-ModPrins Algorithm in Real Multi-trips Instances after 10 minutes of execution for each instance.

	Company Solution		CWS-ModPrins				RandCWS-ModPrins							
Instance	Cost (1)	M	Cost (2)	M	Time (sec)	Gap (2-1)	Best Cost (3)	M	Time (sec)	Gap (3-1)	Gap (3-2)	Average 10 Seeds (4)	Gap (4-1)	Gap (4-2)
L	41667.88	154	34476.84	148	2.24	-17.26%	34234.77	146	65.60	-17.84%	-0.70%	34359.05	-17.54%	-0.34%
M	25221.90	83	16931.76	71	1.19	-32.87%	16722.78	70	51.80	-33.70%	-1.23%	16739.59	-33.63%	-1.13%
N	41480.00	156	36039.56	155	2.27	-13.12%	35492.75	154	39.23	-14.43%	-1.52%	35708.71	-13.91%	-0.92%
O	38148.18	139	32356.92	141	2.35	-15.18%	32151.47	140	6.88	-15.72%	-0.63%	32273.13	-15.40%	-0.26%
P	36897.29	136	31697.62	136	1.73	-14.09%	31378.63	135	58.94	-14.96%	-1.01%	31543.09	-14.51%	-0.49%
Q	23676.26	78	15125.72	64	2.72	-36.11%	14919.63	63	3.69	-36.98%	-1.36%	14973.86	-36.76%	-1.00%
R	42111.20	158	36170.88	158	2.17	-14.11%	35776.78	156	7.26	-15.04%	-1.09%	35923.41	-14.69%	-0.68%
S	39644.74	144	32982.88	142	2.28	-16.80%	32579.90	141	13.07	-17.82%	-1.22%	32836.54	-17.17%	-0.44%
T	36902.82	134	31118.64	134	1.85	-15.67%	30920.33	133	37.02	-16.21%	-0.64%	31010.94	-15.97%	-0.35%
U	22319.13	71	14449.10	60	2.36	-35.26%	14260.62	59	38.27	-36.11%	-1.30%	14313.52	-35.87%	-0.94%
V	41129.10	148	34373.24	146	2.13	-16.43%	33888.81	145	13.25	-17.60%	-1.41%	34028.46	-17.26%	-1.00%
W	38782.11	141	32045.93	137	2.07	-17.37%	31812.79	136	12.86	-17.97%	-0.73%	31966.22	-17.57%	-0.25%
X	38046.72	138	31011.92	134	2.24	-18.49%	30909.63	133	37.19	-18.76%	-0.33%	30997.94	-18.53%	-0.05%
Y	23541.73	80	16325.44	69	3.78	-30.65%	15909.83	68	52.53	-32.42%	-2.55%	16061.14	-31.78%	-1.62%
Average	34,969.22	125.71	28,221.89	121.07	2.24	-20.96%	27,925.62	119.93	31.26	-21.83%	-1.12%	28,052.54	-21.47%	-0.68%
Total	489,569.06		395,106.45				390,958.72					392,735.60		

Table 6.9: Results applying Randomized CWS-ModPrins Algorithm in Real Single-trips Instances after 10 minutes of execution for each instance.

Fig. 6.2 shows the twenty longest routes obtained by proposed algorithm in instance I, providing a picture of the territorial extension supplied by the company. These routes represent a total distance of 9,057 km and 68 customers. Depot is marked with a 'D' pinpoint while customers are the remaining pinpoints.

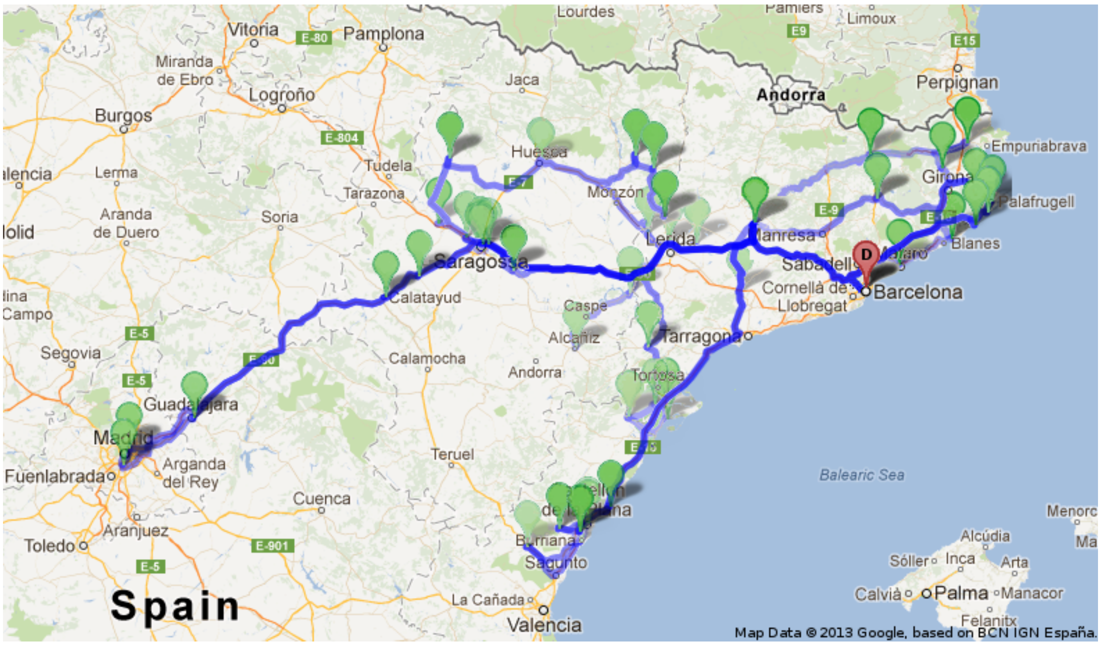


Figure 6.2: Geographical situation of the twenty longest routes of instance I, using Google Maps.

6.5.2 Sub-case: HVRPM with Real Cost Function

The previous study was focused on minimizing total distance travelling. However, the company was interested in including other related costs in the objective function. So we modified the objective function and run the algorithm over the same 25 instances. The new function cost (see Eq. 6.2) is composed by four components explained next and which values can be found in Table 6.10:

1. The variable cost based on the distance and multiplied by a factor depending on the used vehicle for a route (previously defined as γ_k),

2. A fixed cost based on the used vehicle for a route (previously defined as F_k),

3. A second fixed cost per visited store (denoted as $\lambda = 8$ €),

k	γ_k	F_k
Veh_A	0.2446	66
Veh_B	0.3195	72
Veh_C	0.3315	72
Veh_D	0.3315	90
Veh_E	0.3640	106
Veh_F	0.3640	106
Veh_G	0.3640	106
Veh_H	0.3640	106

Table 6.10: Complementary cost information of Fleet composition of the distribution companyRelated.

4. A third fixed cost per box delivered (denoted as $\kappa = 0.1115$ €) which is multiplied by a theoretical capacity of the truck (denoted as $Q'_k/Q'_k < Q_k$). This theoretical capacity was defined in the fares agreement with drivers because they consider that the vehicle goes full.

$$\min(\sum_{i\in\Omega}\sum_{j\in\Omega}\sum_{k=1}^{M}\gamma_k \cdot c_{ij}^k \cdot x_{ij}^k + \sum_{k=1}^{M} F_k + \sum_{i\in\Omega}\lambda + \sum_{k=1}^{M}\kappa \cdot Q'_k) \tag{6.2}$$

As in the previous experiment, we run 10 independent executions per instance. Each execution was run for 1 minute. Table 6.11 shows a comparison between the cost of minimizing distance versus minimizing the new cost function both obtained with our algorithm. The first columns describe the name of the instance, and column (1) with the related cost in euros of the best solutions previously found in Tables 6.9 and 6.8. Next, the number of routes in the corresponding solution, and column (2) with the travelling distance in Km are given. The gap of the distance (Km) in the new solution against the current solution of the company, and also with the best solution previously found. The associated real cost of the new solution is presented in column (3). The last column shows the percentage gap between generated and current company solutions. This average gap for best solutions is less than 1%. Notice that one more time our approach is easily adaptable to different scenarios.

On the provided instances, the total costs of current routing planning is about 877,602.61 €with a total distance of 1,050,150.59 Km. So far, minimizing only distance, we have reduced the total cost to 848,267.17 €(3.34% better) and total distance to

	Min(Distance)	Min(Real Costs)					
Instance	Real Cost (€) (1)	M	Distance Cost (Km) (2)	Gap (2-Current)	Gap (2-Best)	Real Cost (€) (3)	Gap (3-1)
A	38289.60	181	39800.00	-12.15%	0.04%	38285.68	-0.01%
B	39051.86	184	41179.97	-12.73%	0.00%	39051.86	0.00%
C	46146.12	219	50155.98	-7.02%	0.74%	46134.57	-0.03%
D	42226.99	199	45423.57	-10.71%	0.00%	42226.99	0.00%
E	43383.69	207	45777.50	-10.79%	0.52%	43351.66	-0.07%
F	41834.08	196	45345.44	-10.19%	0.00%	41834.08	0.00%
G	42420.15	202	44734.90	-13.01%	0.00%	42420.15	0.00%
H	45111.51	212	49406.72	-9.26%	0.26%	45035.03	-0.17%
I	37602.21	174	39736.12	-11.81%	1.48%	37329.90	-0.72%
J	47712.45	229	51145.95	-15.11%	0.00%	47712.45	0.00%
K	41945.63	199	44156.93	-12.21%	0.24%	41863.61	-0.20%
L	32598.75	146	34234.77	-17.84%	0.00%	32598.75	0.00%
M	17463.35	70	16726.46	-33.68%	0.20%	17347.41	-0.66%
N	33972.95	153	35815.12	-13.66%	0.91%	33960.09	-0.04%
O	31227.50	139	32329.39	-15.25%	0.55%	31156.93	-0.23%
P	30323.48	134	31422.39	-14.84%	0.10%	30216.48	-0.35%
Q	16055.47	63	14919.63	-36.98%	0.04%	15932.14	-0.77%
R	34343.53	156	35776.78	-15.04%	0.00%	34343.53	0.00%
S	31490.91	141	32579.90	-17.82%	0.00%	31490.91	0.00%
T	29953.86	132	31175.31	-15.52%	0.82%	29905.68	-0.16%
U	15230.24	59	14260.62	-36.11%	0.00%	15230.24	0.00%
V	32414.24	145	33890.79	-17.60%	0.01%	32404.95	-0.03%
W	30636.01	136	31812.79	-17.97%	0.00%	30636.01	0.00%
X	29971.28	132	30977.40	-18.58%	0.22%	29868.77	-0.34%
Y	16861.30	68	15909.83	-32.42%	0.00%	16861.30	0.00%
Average	33,930.69	155.04	35,547.77	-17.13%	0.24%	33,887.97	-0.15%
Total	848,267.17		888,694.22			847,199.18	

Table 6.11: Results of minimizing Real Cost Function on Real Instances after 10 minutes of execution for each instance.

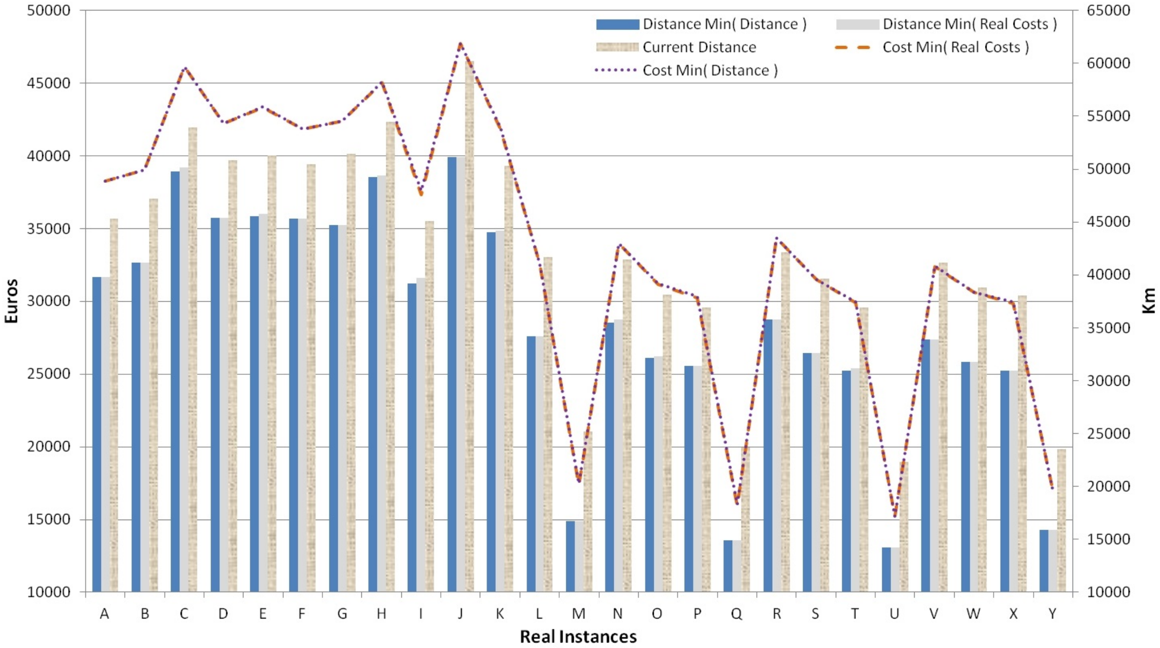


Figure 6.3: Comparison of distance travelled and real costs for each objective function considered.

884,773.21 Km (improved in 15.75%). While minimizing the real cost function, it have reduced the total cost to 847,199.18 €and total distance to 888,694.22 Km. Against the current company solution, the gap related to total costs is slightly better than the previous results (3.46%) as expected but in detriment of total distance (just 15.37% of improvement). So the use of different cost functions can impact on the optimization process of real-big instances like the used ones. In Fig. 6.3, we can appreciate how the distance is quite improved against the current company solution. There is a small distance difference (0.24%) between the solutions generated with objective functions considered, where logically the best value is obtained when distance is minimized. Even when the detail of the cost per each day is not given, the general cost is also reduced with both objective functions. The first 11 instances (with multi-trips) represent the higher values on distance and real costs. So this results open the question to what could be better for an enterprise: "have a small fleet of big vehicles or a large fleet of small capacity vehicles?". The design of a fleet is important to define then the performance of multiple trips. The next section presents a sensibility study in order to help to find a way to handle this situation.

6.6 HVRP Sensibility Analysis

There is a specific problem, commonly named Fleet Composition Problem, which focuses on how to design an optimal (heterogeneous) fleet considering the number as well as the particular properties of the different vehicles composing it (Hoff et al., 2010). So we propose to analyze how distance-based costs vary when slight deviations from the homogeneous fleet assumption are considered, i.e., how marginal costs/savings change when a few 'standard' vehicles in the homogeneous scenario are substituted by other vehicles with different loading capacity. Despite this type of 'what-if' analysis might be very interesting for decision makers, it has not been discussed before in the HVRP literature, which constitutes another important contribution of our work.

6.6.1 Proposed Approach

In this first part, we will focus on the distance-based costs, and thus we will not take into account different fixed and variable costs for different types of vehicles. Our approach for solving this variant of the HVRP is based on the combination of the so-called Successive Approximations Method (SAM) and any efficient method —either exact or approximate— for solving the CVRP. The SAM method proposes a multi-round process. At each round of this process, a new type of vehicle —e.g., the largest one available— is selected among the unused vehicles. Then, assuming an unlimited fleet of vehicles of this type (all of them with the same loading capacity), a new and smaller CVRP is solved for those nodes not yet served.

We make use of the SR-GCWS algorithm (Juan et al., 2010, 2011e) for solving the CVRP at each round. The SR-GCWS is a relatively simple, parameter-less, yet efficient approach for solving the CVRP. Notice, however, that other similar algorithms —e.g., the one by Rieck and Zimmermann (2009)— could have been employed at this stage as well. From the resulting CVRP solution, only those routes which constitute feasible routes for the entire heterogeneous routing problem are saved as partial solutions. The remaining routes are discarded, releasing the associated nodes for the next round. Once all the nodes have been served, a global routing solution is constructed as the union of the partial solutions found at each round. Notice that after each round the size of the next CVRP to be solved will be smaller. In this sense, it is possible to say that the SAM approach for the HVRP makes use of already efficient algorithms to: (a) solve a set of 'nested' CVRPs in a multi-round process; (b) save feasible routes (partial solution) at each round by assigning them to available vehicles; and (c) after the last round is performed, constructing a global solution for the original HVRP by unifying the disjoint partial solutions obtained at each round.

Pseudo-code 5 shows a logic flow for the SAM approach. Notice that the multi-round process will continue until all customers' demands have been satisfied (line 2). Thus, at each round a new type of vehicle —e.g., the largest one available— is selected from the list of available (non-used) vehicles (line 3), and its capacity is employed to define a new CVRP (lines 4 and 5). This will be a CVRP composed by: (a) the non-served customers plus the depot; and (b) a fleet of unlimited vehicles, all of them with the same capacity. Notice that the first round the CVRP may simply be the homogeneous particular case of the original HVRP stated with the capacity of the selected vehicle. On the contrary, in the remaining rounds a series of 'nested' CVRPs will be defined, i.e., smaller CVRPs in which only customers not already served and vehicles not already employed are considered. Once a new CVRP has been defined, it is solved by using any of the numerous efficient methods already available in the literature (line 6). In our case, the SR-GCWS algorithm developed by Juan et al. (2010) is employed to solve each of the nested homogeneous CVRPs. The resulting solution will contain routes designed for an imaginary fleet of vehicles with the selected capacity. For that reason, it is likely that only some of the routes in this 'virtual' solution will be feasible, i.e.: since the real fleet is composed of a limited number of vehicles with a given capacity (the one associated with the selected type of vehicle), only some of the routes in the 'virtual' solution can be implemented in practice. In order to select which routes to assign to the available vehicles, both routes and vehicles are sorted according to their total requested demand and their loading capacity, respectively (lines 7 and 8). Then, routes are assigned to vehicles following the order in the sorted lists as far as the resulting assignment is still feasible, i.e., as far as the new free vehicle in the vehicles list has enough capacity to cover the demand of the new route in the routes list (lines 9 to 16). On the one hand, the feasible routes are saved as part of a global solution, and the customers and vehicles involved in them are deleted from the lists of non-served customers and unused vehicles, respectively. On the other hand, the unfeasible routes are discarded, and the associated customers are set to be served in the next round (lines 17 to 19). At the end of the multi-round process, a global solution covering all customers with different types of vehicles will be obtained.

Algorithm 5 Pseudo-code for the SAM procedure.

```
1: procedure SAM(nodes, vehs)
2:   while list of non − served customers is not empty do    ▷ Perform a multi-round solving process until all customers are served
3:     newVehType ← selectNextType(vehs)    ▷ Select a new type of vehicle an define a new homogeneous CVRP, the largest one available
4:     vehCap ← getCapacity(newVehType)
5:     newCVRP ← defineCVRP(nodes, vehCap)
6:     sol ← solveHomogeneousCVRP(newCVRP)    ▷ Solve the new CVRP using an efficient algorithm (e.g., SR-GCWS)
7:     routes ← sortRoutes(sol)    ▷ Sort routes by total demand required and vehicles by capacity
8:     vehs ← sortVehicles(sol)
9:     i ← 0
10:    while i < size(routes) AND demand(routes[i]) ≤ getCapacity(vehs[i]) do    ▷ While feasible assign most demanding routes to largest vehicles
11:      newRoute ← assignVehicleToRoute(routes[i], vehs[i])
12:      routes[i] ← markAsUsed(routes[i])
13:      vehs[i] ← markAsUsed(vehs[i])
14:      globalSol ← addRouteToSol(newRoute, globalSol)
15:      i ← i + 1
16:    end while
17:    vehs ← deleteUsedVehicles(vehs)    ▷ Dissolve (reset) the unused routes and vehicles
18:    routes ← deleteUsedRoutes(routes)
19:    nodes ← extractNodes(routes)
20:  end while
21:  return globalSol    ▷ Return the global solution
22: end procedure
```

The SR-GCWS algorithm uses some concepts from the CWS heuristic (Clarke and Wright, 1964), such as the 'savings list of edges', the 'initial dummy solution', and the 'merging process'. The main idea behind the biased randomization process is to introduce, at each iteration of the multi-start process, a slight variation to the order in which edges in the savings list are processed. Instead of using always the edge with the most savings (deterministic approach), the savings list is processed in a slightly different order each time the multi-start process is run. However, it is important to introduce bias in this random behaviour by giving edges with higher savings a higher probability of being selected, otherwise the logic behind the heuristic would be lost. In order to introduce this biased randomization, a Geometric probability distribution, which only has one parameter, has proven to be an excellent option (Juan et al., 2010). Additionally, two local search processes can be considered: (a) the first one is based on the use of a cache of routes, so that new solutions can benefit from 'high-quality' roots already found in previous iterations —notice that this technique adds some kind of *memory* to the algorithm; and (b) the second one is based on the use of *splitting* policies, which benefits from a divide-and-conquer strategy: a given solution is split according to some geometric properties and then each of its subparts is solved separately as a new and smaller CVRP. For a more in-deep discussion on the SR-GCWS algorithm and its details, the reader is referred to Juan et al. (2011e).

6.6.2 Experimental Design

For our study we decided to perform a natural adaptation of some of the classical CVRP instances (Augerat et al., 1995). In particular, our instances use exactly the same nodes, including their location coordinates and demands, and the same number of vehicles. We then consider a heterogeneous fleet composed of *standard* vehicles —i.e., vehicles with the capacity defined in the CVRP benchmarks— and $non-standard$ vehicles with modified capacities. In our opinion, this is a natural way to adapt the homogeneous-capacity benchmarks, since it allows the decision-maker to answer sensitivity-analysis questions such as: "How would my routing costs be changed if we could employ one or two trucks with a different capacity?"

Thus, in order to test our approach, a total of fifteen classical CVRP instances were selected and adapted as 'base' HVRP instances. The selection was made at random among the set of medium- and large-size instances (in terms of number of nodes). For each base instance, six different fleet typologies were defined —thus, ninety different instances were considered in total. These fleet typologies are partially composed of *standard* vehicles, each of them with capacity Q, but they differ in their exact composition as explained in the following general rule:

- Fleet 150–125: two *standard* trucks are substituted by a *large* truck (with capacity $Q_l = 150\% \cdot Q$) and by a $large-medium$ truck (with capacity $Q_{lm} = 125\% \cdot Q$), respectively.
- Fleet 125–125: two *standard* trucks are substituted by two $large - medium$ trucks.
- Fleet 125–80: two *standard* trucks are substituted by a $large - medium$ truck and by a *small* truck (with capacity $Q_s = 80\% \cdot Q$), respectively.
- Fleet 90–90: two *standard* trucks are substituted by two $small - medium$ trucks (with capacity $Q_{sm} = 90\% \cdot Q$).
- Fleet 90–80: two *standard* trucks are substituted by a $small - medium$ truck and by a *small* truck, respectively.

Notice, however, that in some cases a reduction in the fleet capacity might cause the infeasibility of the problem, i.e., the total demand to be satisfied might be greater than the total fleet capacity. In those particular cases, an additional *standard* vehicle is added to the fleet to ensure the feasibility of the problem.

6.6.3 Computational Results

The proposed SAM algorithm has been implemented as a Java application. The computational tests have been carried out on a standard desktop computer with the MS Windows 7 operating system, an Intel Xeon E5504 at 2.00 GHz processor, and 4 GB RAM. Each instance was run twenty times using different seeds for the pseudo-random number generator. Each of these run employed a maximum time of 300 seconds. Tables 6.12, 6.13 and 6.14 contain, for each of the fifteen base instances, the following information: (a) instance name, which includes the number of nodes and necessary *standard* vehicles —e.g.: A-n80-k10 has 80 nodes and can be solved with 10 *standard* vehicles; (b) loading capacity of each *standard* vehicle; (c) problem tightness, i.e., total demand requested by nodes divided by total capacity of the available fleet of vehicles; (d) costs provided by the savings method (CWS) for the homogeneous case —i.e., fleet rule

100-100; (e) costs associated with the best-known solution (BKS) for the homogeneous case; (f) different fleet rules for the heterogeneous case, each of them defining a new routing instance; (g) capacities for vehicles 1 and 2 associated with each fleet rule (e.g. $210 = 150\% \cdot 140$); (h) costs provided by the SAM algorithm for the heterogeneous case when the CWS heuristic is employed as routing method at each round; (i) percentage gap between the BKS for the homogeneous case and the SAM-CWS solution for the heterogeneous case; (j) our best solution (OBS), i.e., costs provided by the SAM algorithm for the heterogeneous case when the SR-GCWS algorithm is employed as routing method at each round; (k) fleet configuration for OBS; (l) percentage gap between the BKS for the homogeneous case and the OBS for the heterogeneous case. Instances are distributed in the aforementioned tables according to their relative sizes.

For each base instance, it is interesting to observe the evolution of the gaps between the OBS (heterogeneous case) and the BKS (homogeneous case). Notice that for most instances it is possible to obtain notorious reductions in routing costs when two *standard* vehicles are substituted by vehicles with a somewhat larger capacity. A clear example of this are the negative gaps associated with the B-n45-k5 instance in Table 6.12. Another interesting effect that can be observed in these tables is that the number of necessary vehicles in the fleet can sometimes be reduced by employing one (or two) vehicle(s) with larger capacity. An example of this effect can be seen in Table 6.12, instance B-n50-k7, and also in Fig. 6.4 for instance A-n80-k10. On the contrary, when reducing the capacity of one or two vehicles in the fleet, it might become necessary to incorporate an additional 'standard' vehicle to obtain feasible solutions. This happens, for example, with instances E-n51-k5 or P-n55-k15 in Table 6.13. Observe that for the latter instance no feasible solution has been found, for the fleet rules 90–90 and 90–80, when combining SAM with the CWS heuristic. This difficulty in finding a feasible solution might be due to the combination of two factors: (a) the high tightness of that particular instance (99%); and (b) the fact that the CWS routing process is far from being as efficient as the SR-GCWS routing process. In this sense, the gaps in the (i) column are always much higher than the gaps in the (l) column, which proves that the performance of the SAM approach greatly depends on the quality of the routing algorithm it employs when solving the homogeneous case at each round. A similar effect can be observed in Fig. 6.5.

		Homogeneous Case		Heterogeneous Case				
Instance	Q	CWS	BKS (1)	Fleet Rule (%)	CWS	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
P-n40-k5	140	518.37	461.73	150–125	473.78	426.83	(1,1,3)	-7.56 %
				125–125	493.58	441.87	(1,1,3)	-4.30 %
				125–80	496.56	461.73	(1,1,3)	0.00 %
				90–90	546.07	461.73	(1,1,3)	0.00 %
				90–80	546.07	462.93	(1,1,3)	0.26 %
B-n41-k6	100	898.09	834.92	150–125	794.97	782.00	*(1,1,3)	-6.34 %
				125–125	819.33	812.64	(1,1,4)	-2.67 %
				125–80	812.64	812.64	(1,1,4)	-2.67 %
				90–90	NA	836.79	(1,1,4)	0.22 %
				90–80	898.09	833.66	(1,1,5)*	-0.15 %
B-n45-k5	100	757.16	754.22	150–125	655.55	655.55	(1,1,3)	-13.08 %
				125–125	712.36	702.11	(1,1,3)	-6.91 %
				125–80	711.56	711.56	(1,1,3)	-5.66 %
				90–90	791.20	788.00	(1,1,4)*	4.48 %
				90–80	791.20	788.00	(1,1,4)*	4.48 %
A-n45-k6	100	1,006.45	944.88	150–125	898.69	876.87	(1,1,4)	-7.20 %
				125–125	911.64	911.64	(1,1,4)	-3.52 %
				125–80	NA	930.36	(1,1,4)	-1.54 %
				90–90	1,006.45	974.69	(1,1,5)*	3.15 %
				90–80	1,006.45	974.69	(1,1,5)*	3.15 %
A-n45-k7	100	1,199.98	1,146.71	150–125	1,060.38	1,036.77	*(1,1,4)	-9.59 %
				125–125	1,125.22	1,045.12	*(1,1,4)	-8.86 %
				125–80	1,166.37	1,121.88	(1,1,5)	-2.17 %
				90–90	1,199.98	1,147.00	(1,1,5)	0.03 %
				90–80	1,199.98	1,147.00	(1,1,5)	0.03 %
B-n50-k7	100	748.80	744.23	150–125	686.75	666.55	*(1,1,4)	-10.42 %
				125–125	698.21	666.65	*(1,1,4)	-10.42 %
				125–80	720.97	687.11	*(1,0,5)	-7.68 %
				90–90	748.80	744.23	(1,1,5)	0.00 %
				90–80	748.80	744.23	(1,1,5)	0.00 %

Table 6.12: Experimental results for small-size instances with different fleet configurations.

		Homogeneous Case		Heterogeneous Case				
Instance	Q	CWS	BKS (1)	Fleet Rule (%)	CWS	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
E-n51-k5	160	584.64	524.61	150–125	533.75	515.43	(1,1,3)	-1.75 %
				125–125	589.02	520.94	(1,1,3)	-0.70 %
				125–80	577.11	520.94	(1,1,3)	-0.70 %
				90–90	584.64	539.69	(1,1,4)*	2.87 %
				90–80	584.64	540.15	(1,1,4)*	2.96 %
P-n55-k15	70	978.07	944.56	150–125	937.83	905.06	(1,1,13)	-4.18 %
				125–125	965.11	926.23	(1,1,13)	-1.94 %
				125–80	967.71	938.85	(1,1,14)*	-0.60 %
				90–90	NA	952.02	(1,1,14)*	0.79 %
				90–80	NA	953.74	(1,1,14)*	0.97 %
P-n76-k5	280	698.51	635.04	150–125	678.09	621.77	(1,1,3)	-2.09 %
				125–125	699.71	631.47	(1,1,3)	-0.56 %
				125–80	691.03	631.47	(1,1,3)	-0.56 %
				90–90	703.20	645.74	(1,1,4)*	1.68 %
				90–80	703.20	645.74	(1,1,4)*	1.68 %
E-n76-k14	100	1,054.60	1,026.71	150–125	994.00	982.91	(1,1,12)	-4.27 %
				125–125	1,012.52	988.72	(1,1,12)	-3.70 %
				125–80	1,063.43	1,013.14	(1,1,12)	-1.32 %
				90–90	1,073.43	1,033.96	(1,1,13)*	0.71 %
				90–80	1,073.43	1,033.96	(1,1,13)*	0.71 %
B-n78-k10	100	1,264.56	1,229.27	150–125	1,142.94	1,133.37	*(1,1,7)	-7.80 %
				125–125	1,185.83	1,177.46	*(1,1,7)	-4.22 %
				125–80	1,238.49	1,201.46	(1,1,8)	-2.26 %
				90–90	1,264.56	1,242.38	(1,1,8)	1.07 %
				90–80	1,264.56	1,242.38	(1,1,8)	1.07 %

Table 6.13: Experimental results for medium-size instances with different fleet configurations.

		Homogeneous Case		Heterogeneous Case				
Instance	Q	CWS	BKS (1)	Fleet Rule (%)	CWS	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
A-n80-k10	100	1,860.94	1,766.50	150–125	1,768.52	1,639.56	*(1,1,7)	-7.19 %
				125–125	1,771.62	1,682.35	*(1,1,7)	-4.76 %
				125–80	1,856.38	1,731.49	(1,1,8)	-1.98 %
				90–90	1,863.74	1,779.49	(1,1,8)	0.74 %
				90–80	1,863.74	1,780.15	(1,1,8)	0.77 %
M-n101-k10	200	833.51	819.81	150–125	788.41	777.32	(1,1,8)	-5.18 %
				125–125	824.67	799.34	(1,1,8)	-2.50 %
				125–80	824.41	812.88	(1,1,8)	-0.85 %
				90–90	833.51	821.11	(1,1,8)	0.16 %
				90–80	833.51	821.11	(1,1,8)	0.16 %
M-n121-k7	200	1,068.14	1,045.16	150–125	1,093.12	1,011.11	(1,1,5)	-3.26 %
				125–125	1,100.01	1,011.11	(1,1,5)	-3.26 %
				125–80	1,059.95	1,030.12	(1,1,5)	-1.44 %
				90–90	1,079.37	1,052.32	(1,1,6)*	0.69 %
				90–80	1,079.37	1,052.32	(1,1,6)*	0.69 %
F-n135-k7	2,210	1,219.32	1,170.65	150–125	1,225.20	1,015.36	*(1,1,4)	-13.27 %
				125–125	1,225.55	1,086.76	(1,1,5)	-7.17 %
				125–80	1,240.95	1,131.19	(1,1,5)	-3.37 %
				90–90	1,227.48	1,191.89	(1,1,5)	1.81 %
				90–80	NA	1,191.89	(1,1,5)	1.81 %

Table 6.14: Experimental results for large-size instances with different fleet configurations.

Combining SAM with the SR-GCWS algorithm seems to provide an efficient approach for solving heterogeneous problems. In fact, even for the 90-90 and 90-80 fleet configurations, the gaps between the heterogeneous OBS and the homogeneous BKS are quite small for most instances.

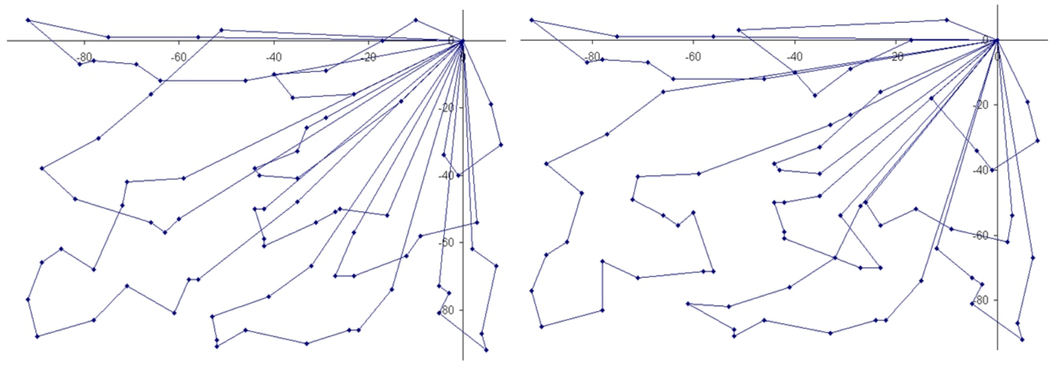


Figure 6.4: A-n80-k10 BKS (left, 10 routes) vs. heterogeneous 150-125 OBS (right, 9 routes).

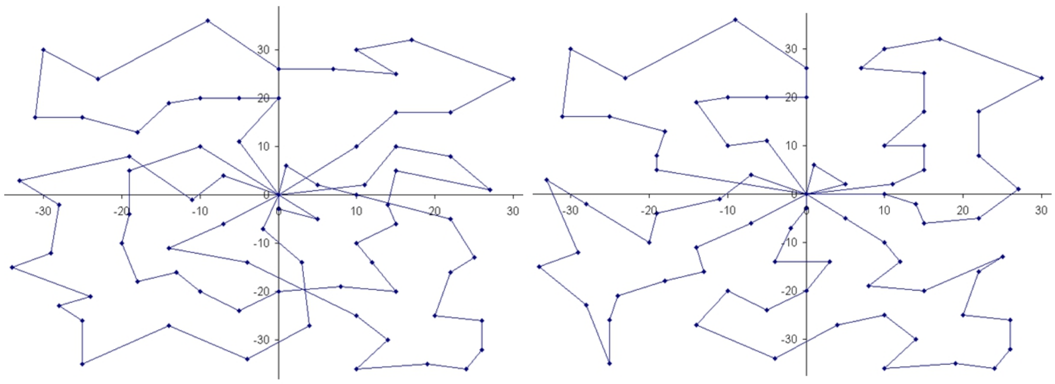


Figure 6.5: P-n76-k5 SAM-CWS (left) vs. SAM-SRGCWS (right).

The effect that different fleet configurations might have with respect to the routing costs is quantified in obtained results. This ‘what-if’ analysis might be particularly useful in those scenarios characterized by certain degree of flexibility during the vehicle-selection process. This could be the case, for example, when the company can rent one or two vehicles, or when it owns extra vehicles of different capacities. Despite its relevance for real-life applications, there is a lack of sensitivity-analysis studies in the HVRP literature and this work aims at providing some insight in the issue.

Fig. 6.6 shows a 3D scatterplot representing the average gap associated with each of the 6 fleet configurations considered in this article. In other words, for each fleet rule, the fifteen gaps with respect to the homogeneous BKS —one per base instance— have been averaged. Additionally, Fig. 6.7 shows an ANOVA output for the differences among average gaps associated with each fleet configuration. The corresponding p-value is almost zero, which means that there are, in fact, significant differences among these average gaps. As it can be derived from Fig. 6.7, average gaps associated with fleet rules 90–90 and 90–80 are positive but quite moderated, i.e., changing two 'standard' vehicles by two other vehicles with a somewhat smaller capacity does not seem to affect the expected routing costs in a noticeable way. In fact, Fig. 6.7 shows that these differences are not statistically significant. On the contrary, it can be observed in both figures that the average gaps associated with fleet rules 150–125 and 125–125 are not only negative but also significantly different from the homogeneous case 100–100. In other words, important reductions in average routing costs can be achieved by simply employing two vehicles with somewhat larger capacities. In summary, it seems reasonable to state that using a homogeneous fleet of vehicles is not a good business strategy, and that significant reductions in expected routing costs can be attained by introducing some degree of flexibility in the fleet configuration.

Finally, Fig. 6.8 shows a multiple box-plot of gaps. That is, for each fleet rule a box-plot is constructed from the fifteen gaps between the OBS and the homogeneous BKS. The multiple box-plot contributes to reinforce the idea that large negative gaps (up to 13%) can be attained when using a pair of vehicles with larger-than-standard capacities. Likewise, using two vehicles with smaller-than-standard capacities has the contrary effect, although the gaps seem not to be so notable —in part due to the asymmetry in the design of the fleet rules, which tries to avoid severe feasibility issues. Notice also how the variability in the gaps is much higher for the 150–125 and 125–125 fleet configurations, i.e., increasing the capacity of two vehicles in the homogeneous fleet will induce negative gaps, but the size of these gaps can vary in a sensible manner depending upon the specific instance being considered.

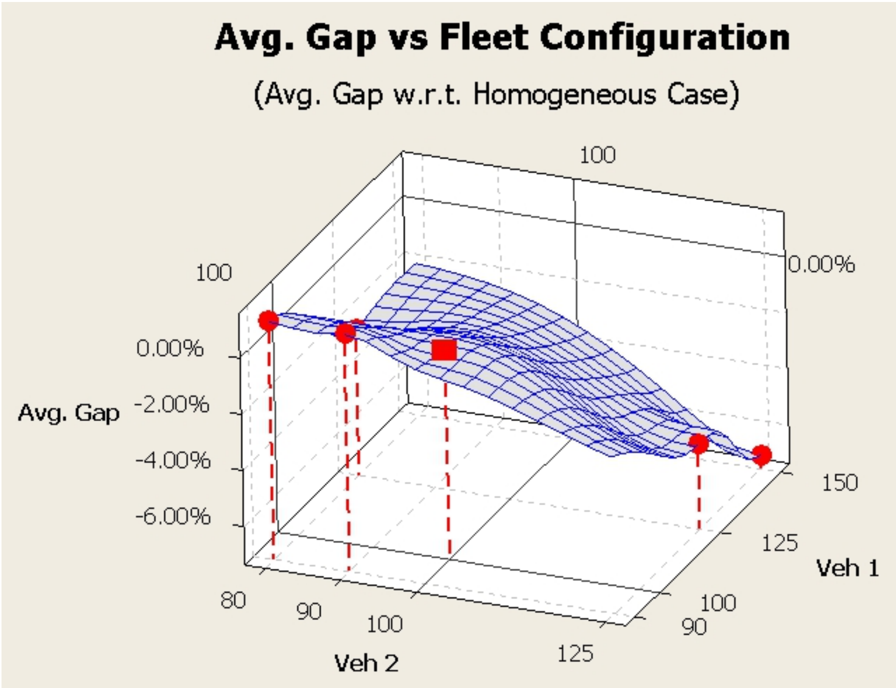


Figure 6.6: Surface Plot of Average Gap vs. Fleet Configuration.

```
                                     Individual 95% CIs For Mean Based on
                                     Pooled StDev
Level      N      Mean    StDev   -------+---------+---------+---------+--
150-125   15  -0.06886  0.03589   (---*---)
125-125   15  -0.04366  0.02852           (--*---)
125-80    15  -0.02187  0.02063                  (---*--)
100-100   15   0.00000  0.00000                        (---*---)
90-90     15   0.01085  0.01388                            (---*--)
90-80     15   0.01113  0.01383                            (---*--)
                                  -------+---------+---------+---------+--
                                      -0.060    -0.030     0.000     0.030

Pooled StDev = 0.02203
```

Figure 6.7: ANOVA output for Average Gap vs. Fleet Configuration.

6.7 Chapter Conclusions

So far, we have appreciated the potential of biased randomization of classical heuristic. They can be adapted to many specific VRP such as the HVRP and HVRPM. Biased randomized versions of the CWS have been used to solve theoretical and real-life data benchmarks considering different combinations of constraints. We present a real vehicle

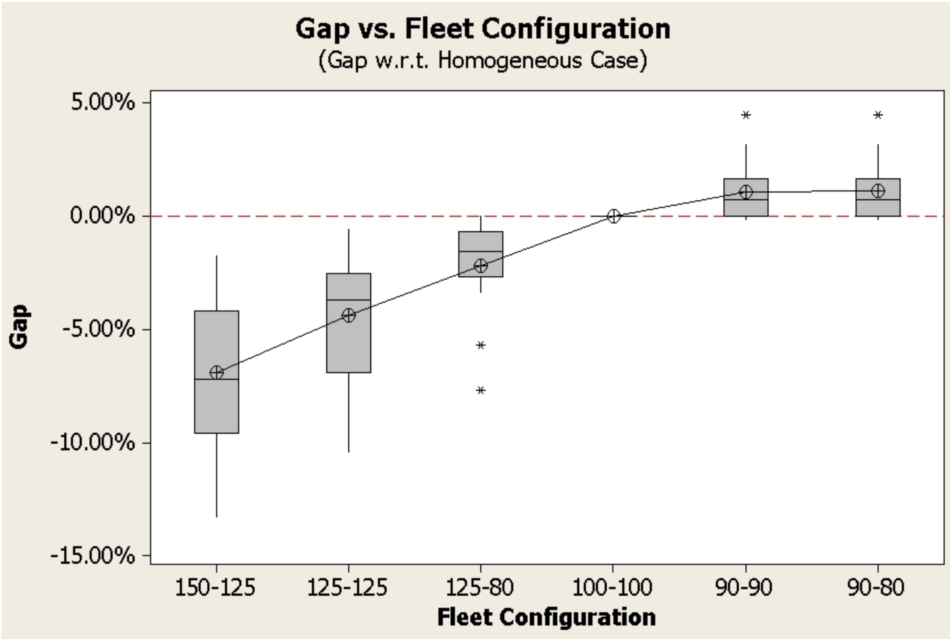


Figure 6.8: Multiple Boxplot of Gap vs. Fleet Configuration.

routing problem of a distribution company in the Northeast of Spain. The company distributes products daily to around 400 stores. One of the main differences of this application with respect to other VRP studies is the presence of a HVRPM, in which some are allowed to perform multiple trips on a single day. We use a biased-randomized heuristic approach combined with three local search processes for solving a real-life VRP. One of the advantages of this method is its easy implementation with no complex fine-tuning required. This makes it very suitable for companies. The results we obtained reduced the company distribution costs significantly with little computational effort, as solutions were obtained in just few seconds with two objective functions.

In the last part of this chapter, a Successive Approximations Method (SAM) for solving the HVRP is presented. The main idea behind SAM is to transform (decompose) the challenge of solving a HVRP into the challenge of solving a series of related Homogeneous VRPs (CVRPs). This decomposition approach allows solving complex HVRP variants —including time windows, stochastic demands, two-dimensional loading, asymmetric costs, multi-depot, etc.— by simply combining SAM with any efficient algorithm already developed for the corresponding CVRP variant. We have generalized some classical CVRP benchmarks in order to perform a sensitivity analysis on the fleet composition. In other words, we have computed the variations in the distribution (distance-based) costs due to variations in the configuration of the vehicles fleet. In fact, the computational results show how it is possible to obtain pronounced differences in average and individual routing costs by varying the loading capacity of just two vehicles in an initially homogeneous fleet. This information can be extremely valuable to decision-makers, since it allows them to estimate variations in average routing costs due to small adjustments in the configuration of their fleet.

The VRPs inspired in real-life situations still represent a challenge for the research community. There is a wide set of constraints combinations in enterprise scenarios as we have seen in the two real applications presented. However, to create a tailored adaptation for an specific VRP is still a hard task. In fact, it takes time for each adaptation even when we consider deterministic scenarios. On the next chapter, we will study tailored approaches for other real types of VRPs such the AVRP and HAVRP.

7

Heterogeneous and Asymmetric VRPs

Parts of this chapter have been taken from the co-authored publications:

Cáceres-Cruz, Riera, Buil, Juan, and Herrero (2013) in Proceedings of ICORES; Cáceres-Cruz, Riera, Buil, and Juan (2013b) in Proceedings of ICAOR; Herrero, Rodríguez, Cáceres-Cruz, and Juan (2014), Int. J. of Advanced Operations Management.

In this chapter, we will present a randomized tailored-purpose approach for realistic variants of the VRP (AVRP and HAVRP). There is a more frequent interest on addressing real cases. The Rich VRP ($RVRP$) is a generalized variant of the VRP where several constraints, aspects or objectives functions are considered at the same time. So the challenge for researchers is to solve the larger set of problems with a single approach. On the group of constraints considered for the RVRP could be multi-depot, periodic visits to clients, open routes, multi-products, time windows, etc. (Drexl, 2012). Mostly, these real case studies has considered the heterogeneous capacity of vehicles inside of the combinatorial problem addressing other constraints. As we do, in some studies like Bolduc et al. (2006); Irnich (2008); Oppen et al. (2010); Prescott-Gagnon et al. (2010); Prins (2002); Rieck and Zimmermann (2010); Vallejo et al. (2012) the variable and fixed costs are ignored when are combined with other routing features. Notice that the problem to solve is still $NP-Hard$. Notice that most of these just include the heterogeneous capacity of vehicles and not the related costs to each vehicle. There are cases where the companies count with their own heterogeneous fleet of vehicles, so there are not renting extra costs; and also it just consider the high-quality-customer perspective so the variable costs are ignored. Therefore, there is not a single way to include the HVRP feature in a RVRP.

In the VRP is also common to handle a cost matrix between pair of locations. This matrix can be evaluated in time, speed, money cost, and/or fuel consumption. In any case, it is usually a function based on the distance. There is the possibility of calculating the Euclidean distance between each pair locations. However, this distance may not correspond to the real distance between two locations which are connected by a transport network or highway. The real distance of the shortest path that connects two points in a road network is always equal or higher than the Euclidean distance. Therefore, it depends on the location of the nodes in the territory and the structure of the road network that communicates them. For this purpose, it is also important to consider an asymmetric distance matrix (Rodríguez and Ruiz, 2012). However, the combination of the commented two restrictions, Heterogeneous Fleet and Asymmetric cost matrix, is not frequent in the literature. Although in realistic scenarios like inside and between cities, it is more appropriate to consider a distribution planning with asymmetric costs due to congestion issues and to the structure of the transportation network. In conclusion it is a real life scenario of a Rich Vehicle Routing Problem.

7.1 Literature Review

In the literature, few variants of the $AVRP$ have been studied. Many techniques have been focused on solving the symmetric $CVRP$, some of which can be adapted to solve the asymmetric case. In Laporte et al. (1986) presents an exact algorithm. In Fischetti et al. (1994) present a branch-and-bound algorithm and its practical application to a real case of pharmaceutical distribution in a city of Italy. In Vigo (1996), it discusses the extension to the $AVRP$ of two of the most important and successful techniques: savings algorithm of Clarke and Wright (1964), and the optimization method of Fisher and Jaikumar (1981). The author states that the solutions found using the proposed asymmetric version of the CWS quickly evolves to worse values as the number of customers increases, in addition to the inconvenience of the parameter combination for the parametric saving function. Other studies are presented in Alonso et al. (2007); Azi et al. (2010a,b); Battarra et al. (2009); Hernandez et al. (2011); Salhi and Petch (2007).

More recently, there are two promising techniques that have been shown to work well in both cases of symmetrical and asymmetrical $CVRP$. The first is the generic approach proposed by Pisinger and Ropke (2007) which is the result of an unified heuristic for several variants of VRP using the Adaptive LNS ($ALNS$). The second is a Memetic Algorithm described in Nagata (2007).

So far, we could not find previous studies related to the HAVRP. Even considering the realistic condition of the HAVRP for urban transportation. The most approximated ones are presented in Marmion et al. (2010); Pessoa et al. (2008). In the first study, the authors analyze the sensitivity of two classical neighbourhood methods for the HAVRP. Thus, they simulate a heterogeneous fleet assigning different variable costs to each vehicle but the capacity remains unchanged. On the second work, the authors developed a set of robust Branch-Cut-and-Price algorithms for several VRPs. Some promising experiments are presented but with an unjustified change on the capacity of fleets. The original fleet has a capacity of 1000, then they execute the same experiments but with other general capacity values (500, 250 and 150).

7.2 Proposed Approach

The algorithm we propose is based on a randomized version of the Clarke and Wright (1964) Savings heuristic (CWS). It uses the concept of savings associated with each arc for merging routes. At each step, the arc with the greatest savings is selected if and only if the two corresponding routes can be combined into a new feasible route and if the selected arc is composed of nodes that are directly connected with the depot. We address the AVRP and HAVRP without considering an extensive asymmetric saving list —i.e., a list including two directed arcs for each pair of customers. Instead we consider a weighted savings list considering just one arc for each pair of customers. Also, we consider the direction of the resulting route after each merging.

Fig. 7.1 shows a flowchart diagram offering a high-level view of our algorithm. Our approach starts solving the problem as proposed in the CWS heuristic —i.e.: computing a dummy solution assigning one round-trip route from the depot to each customer.

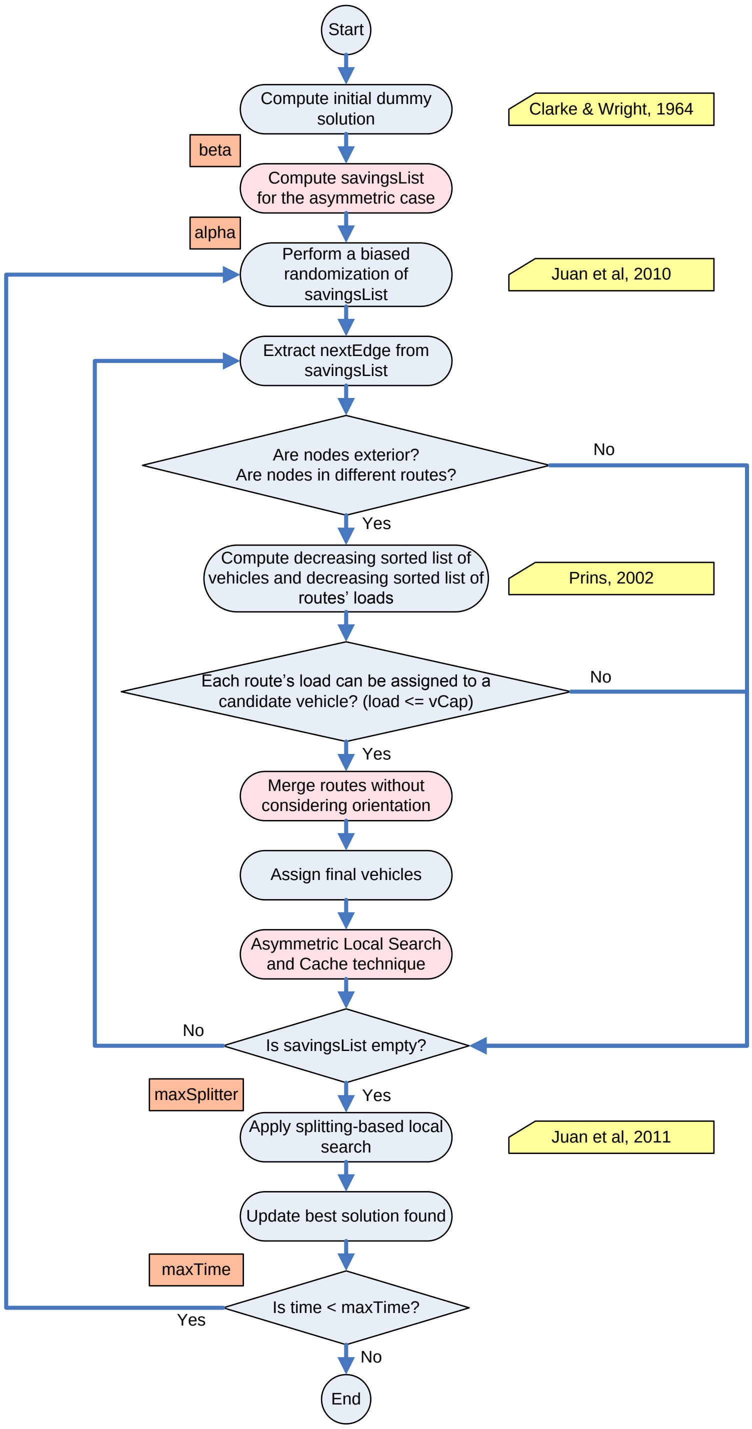


Figure 7.1: Overview of our HAVRP approach.

Then the algorithm computes the weighted savings list using an auxiliary parameter (*beta*). At this point the CWS heuristics is combined with Monte-Carlo Simulation (MCS). We use a pseudo-geometric distribution to assign a biased randomization probability to each edge not used in the dummy solution (*alpha*). Moreover, this selection probability is coherent with the weighted saving value associated with each edge, i.e., edges with higher savings will be more likely to be selected from the list than those with lower savings. Therefore, each combination of edges has a chance of being selected and merged with previously built routes. Then, a multi-start process is initiated and controlled by a time parameter ($maxTime$). At each iteration of this process, different edges are selected using the aforementioned biased probability distribution. This allows obtaining different outputs at each iteration. After merging, we improve the merged route applying two promising local search processes. At the end, we apply a general local search to the whole solution which is explained in the next section.

The validation of the capacity constraint in a heterogeneous fleet is addressed as an assignment problem. For this, an effective method based on CWS is proposed in Prins (2002). The list of vehicles and the list of routes are sorted decreasingly by capacity and accumulated demands respectively; after that, a temporary assignment between the two lists is searched. If a successful match —including all previously routes plus the new merged one— is found, then the capacity constraint is satisfied and the temporary assignment becomes final. Otherwise, the merge becomes unfeasible. If a situation arises in which the number of routes is greater than the number of vehicles, then new fictitious vehicles are assigned to the remaining routes. Notice that this vehicle assignment validation is made for each possible saving, increasing the computational operations. The author also imposes an assumption that the largest demand cannot exceed the capacity of the smallest vehicle.

One important contribution of our approach is the fact that we consider a weighted savings list merging two routes without taking into account directions at this initial stage. See an example in Fig. 7.2. The application of a local search will help to define the best direction. The weighted saving associated with an arc connecting customers i and j is defined as:

$$\hat{S}_{ij} = \beta * max\{S_{ij}, S_{ji}\} + (1 - \beta) * min\{S_{ij}, S_{ji}\}$$

where $\beta \in [0.5, 1]$ and $S_{ij} = c_{0i} + c_{0j} - c_{ij}$.

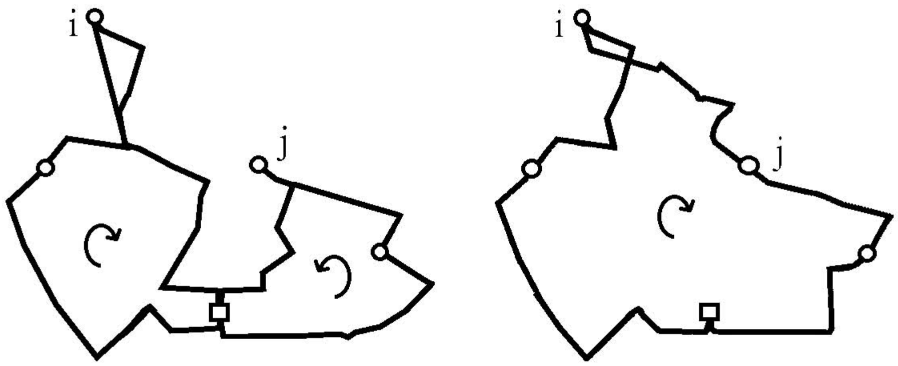


Figure 7.2: Merging example.

The disregard of orientation is important given that an asymmetric savings list could avoid choosing some arcs which do not match the orientation established, i.e., that reduces the solution space. This worsens the general solution; even some obtained solutions are using a greater number of vehicles.

Given that the orientations are not considered, an original local search for the asymmetric context was created exploring the near solution space with few steps. It represents another important contribution of our approach.

7.2.1 Asymmetric Local Searches

Once a merged route is obtained, two local searches are applied in order to explore the solution space with few steps. The first local search procedure is the so called Reversing Routes local search. This procedure intends to find an improvement in the order and orientation of the nodes. Given a merged route, we first try to sort the nodes in a more efficient way. If a route is composed by more than four nodes, then we take each four nodes —i.e., (i, j, k, l)— and try to determine if a swapping of two middle-nodes could improve the cost —i.e., (i, k, j, l). After that, we try to reverse the order in which nodes are traversed.

A second local search, originally described in Juan et al. (2011e), is focused on checking if a given set of nodes already exists in a memory but with a better order of the nodes. The basic idea of this learning mechanism is to store in a cache memory the best-know order to travel among the nodes that constitute one route. This cache is constantly updated whenever a better order with a lower cost is found for a given set of nodes. At the same time, the routes contained in this cache are re-used whenever possible to improve newly merged routes. Notice that this procedure does not search a new vehicle assignment. The previously assigned vehicle to each route remains unchanged during this process.

Finally, once all edges in the saving list have been considered, the resulting solution is improved through a Splitting local search method proposed in Juan et al. (2011e). The current solution is divided into disjoint subsets of routes together with their previously assigned vehicles; then, each of these subsets are solved applying the same methodology described before during a given number of iterations (*maxSplitter*). This tries to apply a "divide and conquer" approach since smaller instances could be easier to solve. So a new set of routes could be created on each partition with the previously assigned vehicles.

7.3 Computational Results

The most commonly-used methodology to compare the performance of different algorithms for solving VRPs consists in running these algorithms over a set of well—defined benchmark instances. In the case of the CVRP or the AVRP, several benchmark sets are available through open-access websites, so that researchers worldwide can use them. Usually, these data sets contain complete information, including not just the instance inputs and the best-known value for the objective function, but also a complete description of the corresponding solution —i.e., the specific composition of each route in the best-known solution. In the case of the HAVRP, however, there is not a commonly-accepted set of instances to test algorithms, since the HAVRP have been not quite considered on the community.

For the AVRP, some researchers have used a set of real instances related to Fischetti et al. (1994) which are available on demand. In our case, we have selected 20 public AVRP instances from `http://soa.iti.es/files/Instances_CVRP.7z` generated and analyzed by Rodríguez and Ruiz (2012). These instances have been generated with a realistic perspective and mathematical justification. The selection was made at random among the set of medium- and large-size instances (in terms of number of nodes). They have 50 or 100 customers and are designed to employ an homogeneous fleet from 2 to

7 vehicles. These instances consider great demand and vehicle capacity, and random location of the nodes within intra-city area.

Therefore, they has a higher number of stops, and are assuming asymmetric transportation in city distribution, so it challenges routing algorithms. The depot may be in the center of the area, those with 'C' in the second letter of the name, or a random position, those with 'A'. They are based on real problems. The intra-city instances were chosen given that they represent a higher asymmetry degree, Rodríguez and Ruiz (2012) conclude that these instances affects in a statistically significant way the CPU time needed by some algorithms and deteriorates the quality of the solutions obtained. For more information, the reader can visit `http://soa.iti.es/files/Instances_CVRP_explanation.txt`.

A very important factor, not related with the AVRP instances, is the state-of-the-art algorithms. We have selected the following $AVRP$ methods in order to compare with:

- General heuristic of Pisinger and Ropke (2007). It is a unified heuristic that works for several variants of routing problems and that uses an Adaptive Large Neighborhood Search (ALNS). It is a very capable and robust method.

- Memetic algorithm of Nagata (2007) (MA). Similar to ALNS, MA is a very powerful and recent $AVRP$ metaheuristic.

The previous algorithms have been selected by their performance and recognition. We have strived for a balance between simple classical techniques and current and state-of-the-art methods. Algorithms NA and ALNS were run from the original code which was kindly provided by their respective authors. No code modification was carried out and the methods were run according to their recommendations.

For the HAVRP, as commented before some preliminary experiments are based on a set of real instances related to AVRP of Fischetti et al. (1994). Likely, Marmion et al. (2010) simulates the heterogeneous fleet over a range of values for testing some operators on different algorithms. However, the proposed studies have only considered the effect of variable cost on vehicles selection by ignoring the different capacities, i.e., the vehicles have the same capacity. Also Pessoa et al. (2008) have used this benchmarks modifying the capacity of original fleets and then running the experiments with homogeneous fleets. The original fleet has a capacity of 1000, then they execute the same experiments but with other general capacity values (500, 250 and 150).

Therefore, we propose to use exactly the same nodes of AVRP, including their asymmetric costs and demands, and the same number of vehicles. We then consider a heterogeneous fleet composed of *standard* vehicles —i.e., vehicles with the capacity defined in the AVRP instances— and $non-standard$ vehicles with modified capacities. In our opinion, this is a natural way to adapt the homogeneous-capacity instances, since it allows the decision-maker to answer sensitivity-analysis questions such as: "How would my routing costs be changed if we could employ one or two trucks with a different capacity?". Thus, in order to test our approach, a total of twenty classical AVRP instances were selected and adapted as *base* HAVRP instances. Finally, the design of experiments proposed in the previous section will be repeated with the selected AVRP instances.

7.3.1 AVRP

Aiming to validate our algorithm, we first present the results of a homogeneous case of AVRP. For this, we compare regarding the Memetic algorithm of Nagata (2007) and the Adaptive Large Neighbourhood Search ($ALNS$) of Pisinger and Ropke (2007). Finally, we developed some experiments for the HAVRP. The algorithm described in this study has been implemented as a Java application. At the core of this implementation, we included the SSJ library provided in L'Ecuyer et al. (2002) and, in particular, the LFSR113 pseudo-random number generator. An Intel QuadCore i5 at 3.2 GHz and 4 GB RAM was used to perform all tests, which were run over Windows XP.

For the 20 AVRP instances, we have used 10 random seeds (10 replicas), an elapsed time of 1 minute ($maxTime$) for each seed, and 60 iterations for splitting technique (*maxSplitter*). In order to perform a biased randomization of the weighted savings list, a quasi-geometric distribution with parameter $\alpha \in \{0.5, 0.1\}$ was used; and the value chosen for the weighted saving was $\beta = 0.6$. Nagata algorithm was executed with a parameter setting: $N_{pop} = 100$, $N_{ch} = 30$, 10 trials and 2 parents. Also ten runs with elapsed time of 1 minute were executed for each instance. For the ALNS, only one run was executed for each instance without time limit. Both algorithms were run from the original code which was kindly provided by their respective authors.

Instance	n	M	MA (1)	ALNS (2)	Gap (1-2)	time (s)	OBS (3)	Gap (1-3)
G-A-CAA0501	50	2	370.26	370.26	0.00 %	17.83	370.26	0.00 %
G-A-CAA0502	50	3	414.44	414.44	0.00 %	13.30	414.44	0.00 %
G-A-CAA0503	50	4	444.69	444.69	0.00 %	10.48	444.69	0.00 %
G-A-CAA0504	50	2	362.01	362.01	0.00 %	17.95	362.01	0.00 %
G-A-CAA0505	50	3	395.78	395.78	0.00 %	15.59	398.47	0.68 %
G-A-CAA1001	100	5	661.88	664.53	0.40 %	43.03	675.31	2.03 %
G-A-CAA1002	100	5	621.06	622.67	0.26 %	39.36	625.82	0.77 %
G-A-CAA1003	100	5	627.29	627.29	0.00 %	42.23	627.29	0.00 %
G-A-CAA1004	100	6	681.89	681.89	0.00 %	34.84	686.25	0.64 %
G-A-CAA1005	100	7	810.97	810.97	0.00 %	29.03	820.56	1.18 %
G-C-CAA0501	50	2	376.62	376.62	0.00 %	17.70	376.62	0.00 %
G-C-CAA0502	50	3	372.48	372.48	0.00 %	13.31	372.48	0.00 %
G-C-CAA0503	50	4	404.30	404.30	0.00 %	10.36	404.30	0.00 %
G-C-CAA0504	50	2	361.74	361.74	0.00 %	17.84	361.74	0.00 %
G-C-CAA0505	50	3	386.73	386.73	0.00 %	13.80	386.73	0.00 %
G-C-CAA1001	100	5	596.54	596.86	0.05 %	40.83	600.35	0.64 %
G-C-CAA1002	100	5	578.15	578.15	0.00 %	38.61	583.39	0.90 %
G-C-CAA1003	100	5	561.08	561.08	0.00 %	41.13	566.10	0.89 %
G-C-CAA1004	100	6	660.81	660.81	0.00 %	35.19	664.14	0.50 %
G-C-CAA1005	100	7	652.08	652.18	0.02 %	28.63	652.42	0.05 %
Average					**0.04 %**			**0.41 %**

Table 7.1: Comparison of results for AVRP instances.

No code modification was carried out and the methods were run according to their recommendations.

The results of these tests are summarized in Table 7.1, which contains the following information for each instance: name of instance; number of nodes; number of vehicles; the best solution of 10 replicas of Nagata algorithm (MA), (1); the ALNS solution, (2); gap, expressed as a percentage value, between columns (1) and (2); the time used for ALNS in seconds; our best solution found, OBS (3); and gap between columns (1) and (3).

Notice that our approach seems to be quite competitive, showing gaps quasi-lower 2% for all instances, with respect to Nagata (2007) which obtains the best results. Our approach also found the same solution for 10 of the 20 instances.

		Homogeneous Case		Heterogeneous Case			
Instance	n	M	OBS (1)	Fleet Rule (%)	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
G-A-CAA0501	50	2	370.26	150–125	368.83	(1,1,0)	-0.39 %
				125–125	368.83	(1,1,0)	-0.39 %
				125–80	378.11	(1,1,0)	2.12 %
				90–90	384.20	(1,1,1)*	3.77 %
				90–80	388.65	(1,1,1)*	4.97 %
G-A-CAA0502	50	3	414.44	150–125	372.18	*(1,1,0)	-10.20 %
				125–125	383.22	*(1,1,0)	-7.53 %
				125–80	398.93	(1,1,1)	-3.74 %
				90–90	414.44	(1,1,1)	0.00 %
				90–80	414.44	(1,1,1)	0.00 %
G-A-CAA0503	50	4	444.69	150–125	404.26	*(1,1,1)	-9.09 %
				125–125	426.96	(1,1,2)	-3.99 %
				125–80	432.41	(1,1,2)	-2.76 %
				90–90	452.60	(1,1,2)	1.78 %
				90–80	459.58	(1,1,2)	3.35 %
G-A-CAA0504	50	2	362.01	150–125	363.54	(1,1,0)	0.42 %
				125–125	359.15	(1,1,0)	-0.79 %
				125–80	360.21	(1,1,0)	-0.50 %
				90–90	377.80	(1,1,1)*	4.36 %
				90–80	379.38	(1,1,1)*	4.80 %
G-A-CAA0505	50	3	398.47	150–125	378.35	*(1,1,0)	-5.05 %
				125–125	380.18	*(1,1,0)	-4.59 %
				125–80	382.43	*(1,1,0)	-4.03 %
				90–90	404.16	(1,1,1)	1.43 %
				90–80	402.23	(1,1,1)	0.94 %

Table 7.2: Experimental results for small-size instances with different fleet configurations.

7.3.2 HAVRP

Finally, in order to test our approach, these twenty AVRP instances were adapted as ‘base’ HAVRP instances. For each base instance, six different fleet typologies were defined —see section “Experimental Design” in the previous chapter of HVRP for more details. Thus, 120 different instances were considered in total.

Tables 7.2, 7.3, 7.4 and 7.5 contain, for each base instance, the following information: name of instance; number of nodes; number of vehicles; the best known solution for the homogeneous case, BKS (1); different fleet rules for the heterogeneous case, each of them defining a new routing instance; our best solution found for the heterogeneous case,

		Homogeneous Case		Heterogeneous Case			
Instance	n	M	OBS (1)	Fleet Rule (%)	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
G-C-CAA0501	50	2	376.62	150–125	367.70	(1,1,0)	-2.37 %
				125–125	367.70	(1,1,0)	-2.37 %
				125–80	367.70	(1,1,0)	-2.37 %
				90–90	384.93	(1,1,1)*	2.21 %
				90–80	384.93	(1,1,1)*	2.21 %
G-C-CAA0502	50	3	372.48	150–125	358.01	*(1,1,0)	-3.88 %
				125–125	359.32	*(1,1,0)	-3.53 %
				125–80	372.48	(1,1,1)	0.00 %
				90–90	372.48	(1,1,1)	0.00 %
				90–80	372.48	(1,1,1)	0.00 %
G-C-CAA0503	50	4	404.30	150–125	379.88	*(1,1,1)	-6.04 %
				125–125	397.43	(1,1,2)	-1.70 %
				125–80	398.02	(1,1,2)	-1.55 %
				90–90	405.12	(1,1,2)	0.20 %
				90–80	414.64	(1,1,3)*	2.56 %
G-C-CAA0504	50	2	361.74	150–125	356.35	(1,1,0)	-1.49 %
				125–125	357.22	(1,1,0)	-1.25 %
				125–80	358.82	(1,1,0)	-0.81 %
				90–90	381.99	(1,1,1)*	5.60 %
				90–80	381.99	(1,1,1)*	5.60 %
G-C-CAA0505	50	3	386.73	150–125	374.05	*(1,1,0)	-3.28 %
				125–125	374.05	*(1,1,0)	-3.28 %
				125–80	374.05	*(1,1,0)	-3.28 %
				90–90	386.73	(1,1,1)	0.00 %
				90–80	386.73	(1,1,1)	0.00 %

Table 7.3: Experimental results for small-size instances with different fleet configurations (*continuation*).

		Homogeneous Case		Heterogeneous Case			
Instance	n	M	OBS (1)	Fleet Rule (%)	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
G-A-CAA1001	100	5	675.31	150–125	634.14	*(1,1,2)	-6.10 %
				125–125	634.75	*(1,1,2)	-6.01 %
				125–80	649.35	*(1,1,2)	-3.84 %
				90–90	683.22	(1,1,3)	1.17 %
				90–80	683.22	(1,1,3)	1.17 %
G-A-CAA1002	100	5	625.82	150–125	583.82	*(1,1,2)	-6.71 %
				125–125	603.27	*(1,1,2)	-3.60 %
				125–80	612.55	(1,1,3)	-2.12 %
				90–90	626.28	(1,1,3)	0.07 %
				90–80	626.28	(1,1,3)	0.07 %
G-A-CAA1003	100	5	627.29	150–125	605.83	*(1,1,2)	-3.42 %
				125–125	620.88	*(1,1,2)	-1.02 %
				125–80	622.52	*(1,1,2)	-0.76 %
				90–90	643.89	(1,1,3)	2.65 %
				90–80	643.08	(1,1,3)	2.52 %
G-A-CAA1004	100	6	686.25	150–125	654.07	*(1,1,3)	-4.69 %
				125–125	659.95	*(1,1,3)	-3.83 %
				125–80	664.56	*(1,1,3)	-3.16 %
				90–90	689.19	(1,1,4)	0.43 %
				90–80	694.97	(1,1,4)	1.27 %
G-A-CAA1005	100	7	820.56	150–125	776.32	(1,1,5)	-5.39 %
				125–125	799.07	(1,1,5)	-2.62 %
				125–80	814.77	(1,1,5)	-0.71 %
				90–90	846.25	(1,1,6)*	3.13 %
				90–80	847.80	(1,1,6)*	3.32 %

Table 7.4: Experimental results for medium-size instances with different fleet configurations.

		Homogeneous Case		Heterogeneous Case			
Instance	n	M	OBS (1)	Fleet Rule (%)	OBS (2)	(m_1, m_2, m_s)	Gap (1-2)
G-C-CAA1001	100	5	600.35	150–125	585.12	*(1,1,2)	-2.54 %
				125–125	586.22	*(1,1,2)	-2.35 %
				125–80	588.74	*(1,1,2)	-1.93 %
				90–90	604.70	(1,1,3)	0.72 %
				90–80	604.24	(1,1,3)	0.65 %
G-C-CAA1002	100	5	583.39	150–125	562.81	*(1,1,2)	-3.53 %
				125–125	562.66	*(1,1,2)	-3.55 %
				125–80	579.05	(1,1,3)	-0.74 %
				90–90	588.20	(1,1,3)	0.82 %
				90–80	587.20	(1,1,3)	0.65 %
G-C-CAA1003	100	5	566.10	150–125	543.78	*(1,1,2)	-3.94 %
				125–125	549.96	*(1,1,2)	-2.85 %
				125–80	552.63	*(1,1,2)	-2.38 %
				90–90	565.01	(1,1,3)	-0.19 %
				90–80	567.90	(1,1,3)	0.32 %
G-C-CAA1004	100	6	664.14	150–125	633.03	*(1,1,3)	-4.68 %
				125–125	635.76	*(1,1,3)	-4.27 %
				125–80	647.01	*(1,1,3)	-2.58 %
				90–90	667.83	(1,1,4)	0.56 %
				90–80	672.17	(1,1,4)	1.21 %
G-C-CAA1005	100	7	652.42	150–125	635.25	(1,1,5)	-2.63 %
				125–125	640.44	(1,1,5)	-1.84 %
				125–80	648.60	(1,1,5)	-0.59 %
				90–90	668.16	(1,1,6)*	2.41 %
				90–80	672.15	(1,1,6)*	3.02 %

Table 7.5: Experimental results for medium-size instances with different fleet configurations (*continuation*).

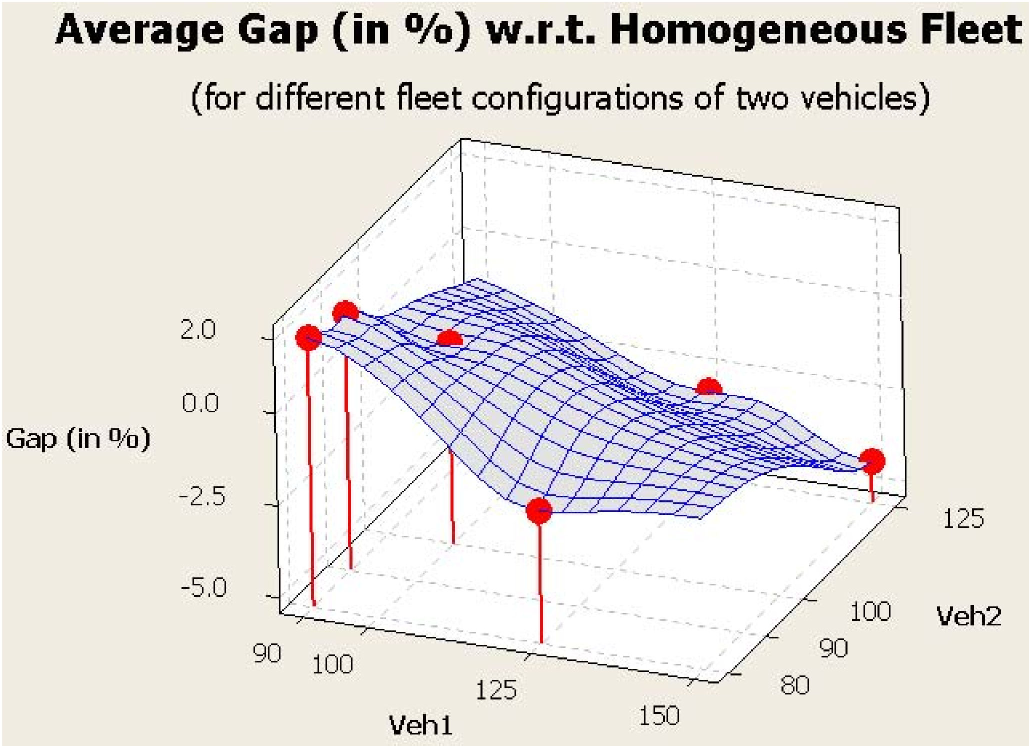


Figure 7.3: Surface Plot of Average Gap vs. Fleet Configuration.

OBS (3); the obtained number of vehicles for the fleet configuration; and percentage gap between the BKS for the homogeneous case and the OBS for the heterogeneous case. Instances are distributed in both tables according to their sizes.

Observe that star (*) highlights different number of vehicles. For example, (1,1,1)* of the fifth row remarks that this heterogeneous solution is using one more vehicle than the homogeneous solution. It uses one vehicle of 90% of capacity, one vehicle of 80% and one *standard* vehicle. Instead, *(1,1,0) of the sixth row remarks that this solution is using one less *standard* vehicle.

Fig. 7.3 shows a 3D scatterplot representing the average gap associated with each of the 6 fleet configurations considered in this article. In other words, for each fleet rule, the twenty gaps with respect to the homogeneous OBS —one per base instance— have been averaged. From these results, it can be noticed the following:

- Just by employing two large vehicles (fleet 150–125) instead of two *standard* vehicles (fleet 100–100), it is possible to obtain noticeable costs reductions that can go up to 10% in some instances (e.g., G-A-CAA0502).

- Likewise, when using two small vehicles (fleet 90–80) instead of two 'standard' vehicles, costs can suffer an increase of about 5% for some instances (e.g., G-C-CAA0504).

Therefore, it can be concluded that routing costs can be in fact quite sensitive to small variations in the fleet configuration. This justifies the necessity for employing new approaches in real-life routing applications, i.e., algorithms which are able to deal with both asymmetric costs as well as heterogeneous fleets.

7.4 Real Case II: HAVRP with Extra Constraints

With the analysis based on Baldacci et al. (2008); Marmion et al. (2010); Pessoa et al. (2008), we have identified standard individual benchmarks for the ACVRP and HVRP. But this is not the case for the combination of these two. As a case of study, we used the information of a food distribution company located in Barcelona, Spain. The company has provided us with the delivery address of their customers in six independent days along with their demands for those days. The transportation limits are defined inside of the city borders (urban distribution).

The main interest of the company is to apply the proposed approach to bigger datasets using a web information tool. For this reason, the company just compile the information during a short period (as a sample) in order to produce a preliminary result. In addition, the compiling process represented an important investment of resources considering the size of the company. Therefore on a daily basis, this company receives requests from around 50 customers. Everyday, this information serves as input to manually design the company's routing planning.

According to the size of the company it is not possible to employ a person specialized in mathematical software in order to apply exact methods. Therefore they prefer to have an approximated solution algorithm embed in a web tool which could be used to give automatic solution in little time. There is a specific constraint: each vehicle must visit all customers of a route in a maximum period of 180 minutes. This route length restriction must to include the travelling time and the service time. The service time is the period of time that the vehicle needs to unload/load for delivering product. So far, the company uses two types of vehicles, which are described in Table 7.6. The columns of this table show the capacity (Q_k) and quantity (m_k) of available vehicles for each

k	Q_k	m_k
1	20	2
2	30	2

Table 7.6: Composition of the current company fleet.

type (k). Actually the company used four vehicles, but they needed to determine if it is possible to reduce the total routing costs and also execute the same deliveries with fewer routes.

We have used a map-location service, like Google Maps to generate the asymmetric cost matrix between every pair of nodes (51 x 51 maximum cells). Even when this kind of routing considers all possible streets of the city, the cost matrix will only represent the best traveling time between each two nodes.

The main features of given six data instances are summarized in Table 7.7. On the first column, we present the identification of each instance that represents a day. The second column shows the number of customers with demands. Third column is the total demand. And the last column represents the total service time of all the nodes on the instance.

As commented before, the company provides us with the historic data of some of their service times and routes. But some fields were incomplete. So we have randomly generated the respective values for the instances, using simulation theory (Monte-Carlo Simulation) and the provided data. Then, we have defined that the service time for each client follows a triangular distribution with min = 1, max = 12 and mode = 3 minutes. This distribution is often used to represent time in general simulation models. However, the routes used differ among all days. Notice that the company did not save exact information of all their routes, even within a whole day. Likely they do not apply any specific routing method. A person in charge, who tries to assign routes to all drivers, designs the routing planning.

7.4.1 Proposed Approach

For this problem, the previously method was simplified. The savings construction is modified for being applied to the HAVRP, because the inversed edges are also considered in the set of options (multiplying the original quantity on the symmetric version by

Instance (day)	n	Total Requested Demand	Total Service Time
A	40	53	163
B	50	75	213
C	40	60	163
D	39	54	159
E	40	57	162
F	18	28	75

Table 7.7: General features of real instances.

two), i.e., for two different nodes i and j: $Sav(i,j) = c_{i0} + c_{0j} - c_{ij}$ and also $Sav(j,i) = c_{0i} + c_{j0} - c_{ji}$. Therefore, all savings will be competing to be taken in the biased randomized process, and those with higher savings will define the orientation of routes. Likely the route construction process will consider the direction of savings edges. Once a route takes a direction then all considered candidate routes to be merged with the first one must follow the same direction. Finally, we have used only one local search, the Cache memory, for improving routing cost.

7.4.2 Computational Results

Our algorithm was implemented as a Java application and used to run the six instances described above on an Intel Xeon E5603 at 1.60 Ghz and 8 GB of RAM. For each instance, a single run with a total maximum time of 500 seconds was employed. The limitation in computing time is due to the fact that we wanted to obtain results in a 'reasonable' amount of time. We employ the Random Number Generator (RNG) library for Stochastic Simulation developed by researchers of the Montreal University (`http://www.iro.umontreal.ca/~simardr/ssj/`).

Table 7.8 shows the results obtained in experiments. The first column shows the instance id; the second, the number of routes defined in the solution; the third column, the total travelling times of routes; the fourth column, the total routing costs considering the travelling times plus the service times of the instance; and the last column, the computational time needed to find the best solution.

The travelling costs on instances B and E represent the higher values obtained. Both of them travelling costs are bigger than the previously commented restriction of 180 minutes. However, this restriction is applied to the route duration and also it

Instance (day)	M	Total Traveling Cost (min)	Total Routing Cost (min)	Time (sec)
A	2	173	336	1.14
B	3	189	402	114.76
C	2	170	333	137.52
D	2	172	331	275.90
E	2	186	348	253.42
F	2	116	191	0.25
Average	2.17	167.67	323.50	130.50

Table 7.8: Results of Best Solutions after 500 seconds running.

considers the service time on each node. On these two instances, the average total routing cost of routes has to be considered. For this, the total routing cost is divided by the number of routes on the solution producing 134 and 174 minutes respectively. Notice that even when the running time is set to a maximum limit of 500 seconds, the average time for finding the best solutions is less than 131 seconds.

In order to validate the solution quality of our approach, we have compared our results against an approximated value of the current total routing costs. As we said before, the company does not have the exact values of routing costs. However, they tend to use all four vehicles as an attempt to reduce delivery times, in an intuitive way. Therefore we have forced our algorithm to use four vehicles in order to produce a near value of current company solutions. The output represents the best solution found in 500 seconds. We delivered the forced four-route solution to the company in order to validate it with the real planning, and we obtained a positive confirmation. Table 7.9 presents the traveling times for each scenario and the gap between these two solutions.

The difference between the approximated company solutions and our approach results is around 13%. In the next two images, we have illustrated both routing solutions of the approximated planning (Fig. 7.4), and the new proposed solution (Fig. 7.5) for the instance B, where the number of routes was reduced to 3. Notice that the average number of routes of our approach is around 2 which represents a considerable reduction of the amount of routes.

Instance (day)	Best Costs using 4 routes (min) (2)	Best Costs (min) (1)	Gap (2-1)
A	192	173	-9.90%
B	205	189	-7.80%
C	206	170	-17.48%
D	190	172	-9.47%
E	211	186	-11.85%
F	153	116	-24.18%
Average	192.83	167.67	-13.45%

Table 7.9: Comparison with extreme case using whole fleet (four vehicles).

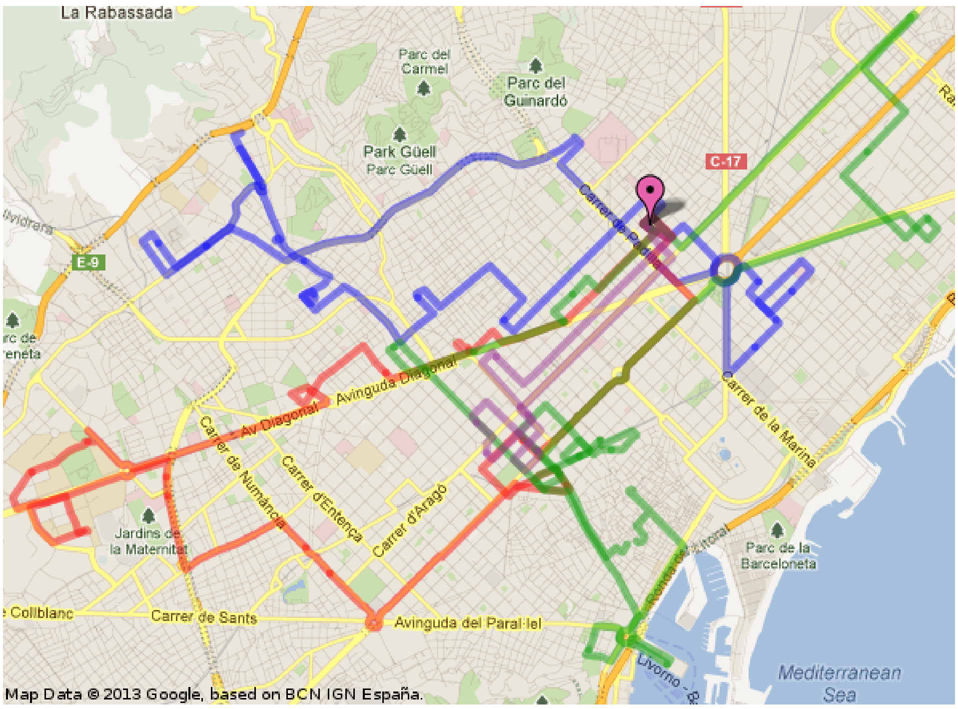


Figure 7.4: Approximated routing planning of the company for instance B, using Google Maps.

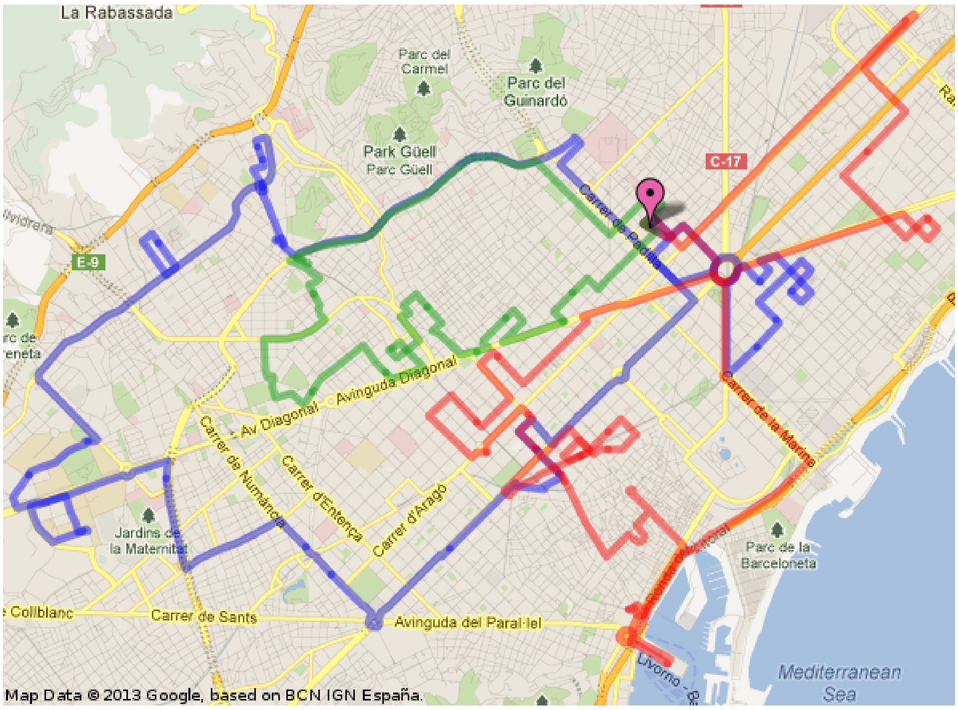


Figure 7.5: Designed routes in the proposed solution for instance B, using Google Maps.

7.4.3 Sub-case: New Extra Constraints

After previous results, the distribution company wants to continue considering other constraints like equally balanced loads between routes and optionally allow open routes. The first constraint tries to apply an equality criteria of route construction between drivers. While the second create some flexibility on the ending point of routes. For some enterprises these constraints could be interesting depending on the business nature. The company is mainly interested in building a set of alternative routing solutions. These solutions can include a subset of the previously specified restrictions. The restrictions can be separated as mandatory for all scenarios (asymmetric cost matrix, heterogeneous fleet of vehicles, service times at customers and limited routes length) and optional (open routes, and balanced loads). These last constraints create new scenarios for routing planning which are the main contributions of this study. In fact, the company is especially interested in the open routes option because their drivers can take delivery vehicles with them. So the time for going to the parking place and going to the depot point (on the next day) is not counted for the delivery process. Therefore it creates some flexibility on selecting the ending point of routes (Li et al., 2007b). For other part, the balanced loads constraint represents an equally working condition between drivers.

We did so small changes in the decision steps of the algorithm to implement the new two features: an initial distinction regarding the open routes requirement is made. If it is the case, we set to 0 the cost of all edges going to the depot. The purpose of this is to ignore returning edges in the route construction process including the dummy solution. The consideration of returning edges will also affect the savings concept as it will be explained next. The savings construction is modified for being applied to both contexts the asymmetric and open routes contexts. First, the inversed edges must be also considered in the set of eligible options (multiplying the original quantity on the symmetric version by two), i.e., for two different nodes i and j: $Sav(i,j) = c_{i0} + c_{0j} - c_{ij}$ as well as for $Sav(j,i)$. Then the commented asymmetric savings concept for the open routes case will be $Sav(i,j) = c_{0j} - c_{ij}$. The edge for going to the depot is excluded from the merging or construction of routes. Therefore, all savings will be competing to be taken in the biased randomized process, and those with higher savings will define the orientation of routes. Once a saving edge is selected and successfully used to merge to given routes, the opposite edge must be also removed from the savings edge list, in order to save computational time. Likely the routes construction process will consider the direction of savings edges. Once a route takes a direction then all considered candidate routes to be merged with the first one must follow the same direction. In Fig. 7.6, a simplified example is depicted in order to give an idea of the route construction process under the given routing constraints. In this directed graph, we have two open routes and two possible savings edges to be considered (A and B). Then it is easy to appreciate that the savings value related to B is better than A. So the new route will be made considering saving edge B since this is more probably to be selected in the biased-randomized process. Notice that resulting routes will tend to join routes where the first customer of one route is near to the last visit of other route.

Second, for considering the balanced loads in routes, we add another validation aspect in the merging step of the CWS process. Once the inputs are read, a maximum load limit per route is estimated using the total requested demand on the instance as well as a number of desirable routes indicated as a new parameter. This last parameter can be set to two in order to try to find the minimum number of routes with balanced

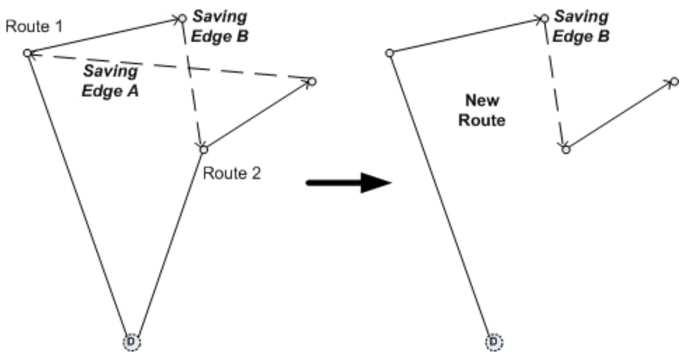


Figure 7.6: Example of saving edge merging decision in an open routes context.

Scenario	M	Distance Cost (min)	Total Cost (min)	% of Used Capacity	Load per Route	Time (sec)
Current	4.00	192.83	348.67	54.50%	13.63	NA
Best	2.17	167.67	323.50	85.63%	25.17	130.50
Open	2.83	144.83	300.67	70.07%	18.94	174.98
Balanced 3 routes	3.17	182.00	337.83	65.00%	17.13	129.81
Open-Balanced 3 routes	3.67	147.50	303.33	59.25%	15.60	248.65
Balanced 2 routes	2.17	168.83	324.67	85.63%	25.17	162.34
Open-Balanced 2 routes	3.00	144.67	300.50	68.13%	18.17	232.76

Table 7.10: Averages results on different solution scenarios combining constraints.

loads, as we did. This load limit is then adjusted with a percentage range in order to allow a flexible criterion in the route construction. This value will serve as a basic limit for checking capacity when two routes are merged (see CWS heuristic).

So with this new version, we repeat the 500 seconds running for each instance. Table 7.10 presents the average information for comparing several scenarios: (a) Current company solutions; (b) previously generated Best found solutions of Table 7.9; (c) solutions allowing only Open routes; (d) solutions only balancing the total load to 3 routes; (e) solutions balancing the total load to 3 routes and also allowing open routes; (f) solutions only balancing the total load to 2 routes; and (g) solutions balancing the total load to 2 routes and also allowing open routes. For each of these, we present the average number of routes, the average distance-time cost (minutes), the average total cost (minutes), average percentage of used capacity in assigned vehicles, average load per route, and average CPU time until the solution is found (seconds).

As it can be appreciated, the Best scenario generated in the first experiments reduces

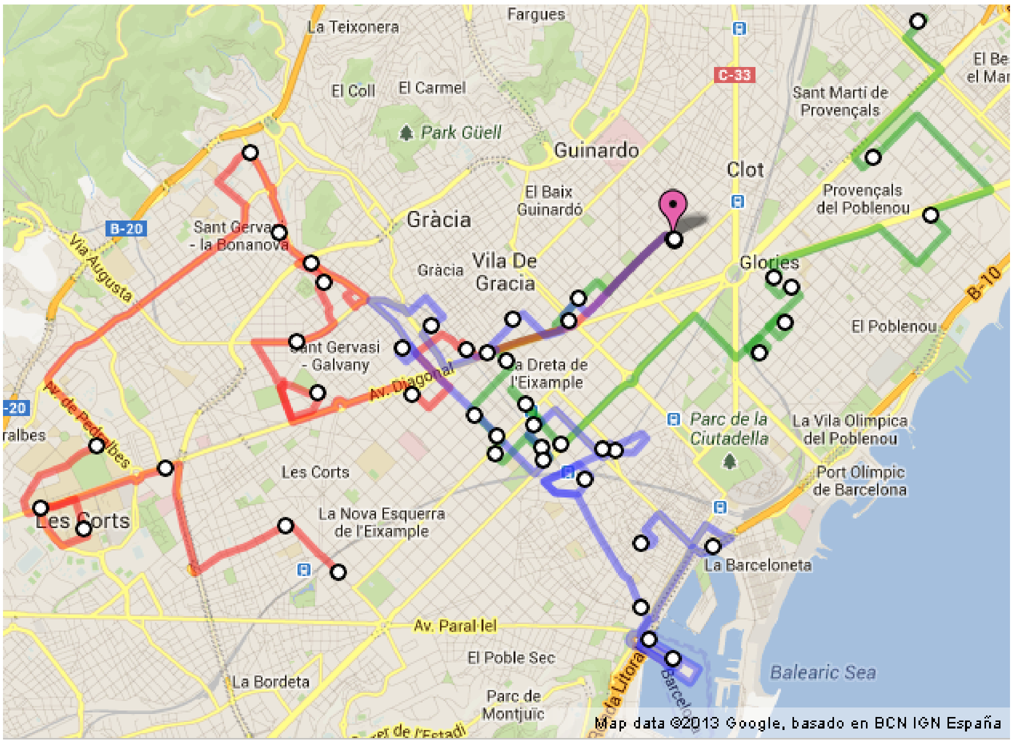


Figure 7.7: Open designed routes in the proposed solution for instance B, using Google Maps.

total costs as well as routes, where the used percentage of vehicle capacities is the higher obtained value. The Open scenario (cheapest) reduce even more total costs because the returning path to the depot is not being considered. However the average number of routes slightly increases. Although the balancing scenarios are focused on creating solutions with an equality criteria on route loads, the cost tends to increase. The algorithm finds better solutions when balancing to the smallest number of routes which is near to the Best scenario. For instance, when we mix the balance and open criteria, the best total cost is found with an average balance of loads. Notice that all generated solutions have better values for the percentage of used capacities of vehicles than the Current scenario. However, longer CPU times are needed to find solutions combining open and balancing constraints. In Fig. 7.7 and 7.8 it can be appreciated the routing planning for both the Open and Balanced to two routes scenarios.

Figure 7.8: Balanced designed routes in the proposed solution for instance B, using Google Maps.

7.5 Chapter Conclusions

In this chapter, a biased randomization of classical heuristic for solving a different branch of VRPs have been proposed. Biased randomized algorithms have been used to solve theoretical and real-life data benchmarks considering different combinations of constraints. The VRPs inspired in real-life situations still represent a challenge for the research community (Bochtis and Sörensen, 2009). Despite the fact that most real-life fleets of vehicles are heterogeneous and that real-life distances are frequently asymmetric —especially in urban transportation—, there is a lack of works considering both situations simultaneously. Accordingly, we have presented a hybrid algorithm for solving the HAVRP. This algorithm combines a randomized savings heuristic with three local search processes specifically adapted to the asymmetric nature of costs in real-life scenarios. A complete set of AVRP and HAVRP tests have been performed to illustrate the methodology and analyze its efficiency when compared with two state-of-the-art algorithms. The results show that our approach is able to produce competitive results for the AVRP while, at the same time, it is much simpler to implement and requires less parameters —and fine-tuning efforts— than current state-of-the-art algorithms. Moreover, since our methodology can also consider heterogeneous fleets, a set of benchmarks for the HAVRP have been developed and a sensitivity analysis on the fleet composition has been performed. This last experiment shows how decision-makers can benefit from our approach when deciding the actual composition of their heterogeneous fleets. Also we present a case study that support a food distribution company to: (a) realize the current situation with quantitative methods; and (b) improve their routing planning with a simple approach. We used Monte-Carlo Simulation to complete the missing data from the company, and obtain the information required for testing. On the next chapter, other biased-randomized heuristic example for a different VRP family is explained.

8

VRPs with Time Windows

Parts of this chapter have been taken from the co-authored publication:

Cáceres-Cruz, Riera, Juan, and Padrón (2013) in Proceedings of MAEB.

In last decades, optimization routing problems have been the target of many studies (Golden et al., 2008). The Vehicle Routing Problem with Time Windows (VRPTW) is probably one of the most developed research lines inside of the classical Vehicle Routing Problem (Potvin and Bengio, 1996; Potvin et al., 1996). On this problem, a set of vehicles must deliver the goods to a set of customers. Unlike the original problem, VRPTW must respect some delivery time windows on each customer and considering arrival, waiting and service times among others. The objective of this chapter is to adapt one of the popular VRPTW heuristics proposed in Solomon (1987). This heuristic is known as *Insertion* and basically consists on the iterative construction routes with the insertion of appropriate customers. Therefore our main idea is to apply the randomization concepts presented before in order to generate a new promising metaheuristic algorithm.

8.1 Definition

In VRPTW, the objective function is the same than CVRP but some delivery time windows must be considered. So the traveling time between a pair of customers (t_{ij}) is an important element on this study. The scheduling constraint is denoted by a predefined time interval, given as an earliest start time (e_i) and latest start time (l_i)

at the customer i. The vehicles leave the depot, at time e_0, at the earliest and must return to the depot by time l_0, at the latest. There is also a given service duration time (f_i) on each customer of a route for considering unloading time of goods. Therefore vehicles must arrive at the customers not later than the latest start time. If vehicles arrive earlier than the earliest start time, then a waiting occurs. So after the routing planning is defined, an effective delivery service at customer i begins at a given time (b_i) within the defined customer time window. In Eq. 8.1 and 8.2 we define the basic relations of time windows and sequential customers' visits using commented variables.

$$e_i \leq b_i \leq l_i, \qquad \forall i \in \Omega* \tag{8.1}$$

$$b_j = Max[e_j, b_i + f_i + t_{ij}], \quad \forall i \in \Omega, \forall j \in \Omega*, i \neq j \tag{8.2}$$

8.2 Literature Review

One of the most studied VRP is the VRPTW. Different approaches to the VRPTW have been explored during the last decades (Cordeau et al., 2001a). These approaches range from the use of pure optimization methods, such as linear programming, for solving small-size problems with relatively simple constraints to the use of heuristics and meta-heuristics that provide near-optimal solutions for medium and large-size problems. One of the most promising frameworks is presented in Cordeau et al. (2001b, 2004) which is based on a Tabu Search technique. Notice that this framework combines the time windows constraint with other routing restrictions. Another Tabu Search algorithm is parallelized in Badeau et al. (1997). A guided local search is proposed by Kilby et al. (1999). An interesting hybrid local search is developed by Bent and Van-Hentenryck (2004a). Some greedy approaches have been presented in Ioannou et al. (2001); Kontoravdis and Bard (1995). Comprehensible recent surveys of algorithms and metaheuristics for the VRPTW can be found in Bräysy and Gendreau (2005a,b). Also several variants of this problem have been studied: VRPTW minimizing route duration (Savelsbergh, 1992), dial-a-ride problems with time windows (Diana and Dessouky, 2004), VRPTW with a limited vehicle fleet (Lau et al., 2003), robust VRPTW (Agra et al., 2013), real waste collection with time windows (Kim et al., 2006), among others. Since other approaches could generate better results (Hu et al., 2013), they are also certainly more complex to implement and understand. Therefore the main advantage of the approach proposed on this study is its simplicity. Our method is focused to randomize a well-known heuristic.

Particularly, Solomon (1987) proposes six heuristics for the VRPTW. In between, we can find three sequential building heuristics based on the insertion of clients. These Insertion heuristics have been widely used in the research community (Berger and Barkaoui, 2004; Campbell and Savelsbergh, 2004b; Diana and Dessouky, 2004; Tan et al., 2001). In fact, Potvin and Rousseau (1993) propose a parallel version of the first insertion heuristic. Likely, the work of Ioannou et al. (2001) is based on Solomon's Insertion heuristic Framework for solving theoretical instances and a real-life case inspired in a Food Company.

8.3 Proposed Approach

Here we focus on the Insertion Solomon heuristic so called $I3$. On this, the author initializes every route construction using one sorting criteria of a set to be described later. After initializing a current route, the method uses two criteria, $Sc_1(i, u, j)$ and $Sc_2(i, u, j)$, to iteratively insert a new customer u into the current partial route, between two adjacent customers i and j on the route. One by one, until time windows and capacity constraints do not allow to add more clients. For each unrouted customer, we first compute its best feasible insertion place in the emerging route (Sc_1). Next, the best unrouted customer to be inserted in the route is selected as the one for which Sc_2 is optimum and feasible. When no more customers with feasible insertions can be found, the method starts a new route, unless it has already routed all customers. As the same author states: ""this class of heuristics is a generalization of the time-oriented, nearest-neighbour heuristic, in that we allow insertion of an unrouted customer in any feasible location between a pair of customers on the route, rather than only at the end of the route"." Formally, the value criteria that Solomon proposes for the I3 are presented next (Eq. 8.3 to 8.8). In summary, these criteria consist in a weighted addition of sub-elements where each represents an important routing-scheduling feature. Using four parameter values ($\mu; \alpha_1; \alpha_2; \alpha_3$), each feature is then related in the next expression of $Sc_1(i, u, j)$.

$$Sc_1(i, u, j) = \alpha_1 \cdot Sc_{11}(i, u, j) + \alpha_2 \cdot Sc_{12}(i, u, j) + \alpha_3 \cdot Sc_{13}(i, u, j) \tag{8.3}$$

subject to:

$$Sc_{11}(i,u,j) = d_{iu} + d_{uj} - \mu \cdot d_{ij}, \quad \mu \geq 0 \tag{8.4}$$

$$Sc_{12}(i,u,j) = b_{j_u} - b_j; \tag{8.5}$$

$$Sc_{13}(i,u,j) = l_u - b_u; \tag{8.6}$$

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \qquad \alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0 \tag{8.7}$$

$$Sc_2(i,u,j) = Sc_1(i,u,j) \tag{8.8}$$

Where b_{ju} is the new start time for service at customer j, given than u is on the route.

Our algorithm is implemented as described next (see Pseudo-code 6). We propose to randomize in two points of the algorithm the original I3 of (Solomon, 1987). First, we apply a uniform randomization over the selection of the sorting criteria for the list of customers (explained in the next section). On each iteration, the combined effect of sorting with different criteria will create an intensive and promising search guide inside of the solution space. Second, a geometric (biased) distribution is used to pick up the next client over the sorted list. Thus, the clients at the top of the list will be more likely to be selected than others. This kind of double randomization has been previously applied in González-Martín et al. (2012) with good results.

Algorithm 6 General pseudocode for RandI3.

procedure RANDI3($inputs, \mu, \alpha_1, \alpha_2, \alpha_3, \beta$)
  $nodes \leftarrow computeSolomonInitialCriteria(\mu, \alpha_1, \alpha_2, \alpha_3)$
  while *stopping criterion is not satisfied* **do** ▷ time or iterations
    $route \leftarrow createNewRoute()$
    while *there are unrouted clients on the list* **do**
      $sortCriteria \leftarrow selectSortCriteria(nodes)$ ▷ Use a uniform randomization for selecting 1 of 4 sorting criteria of customers list
      $unroutedClientsList \leftarrow sortClients(nodes, sortCriteria)$
      $client \leftarrow selectNextClient(unroutedClientsList, \beta)$ ▷ Use a biased randomization for selecting next customer to be included in the route, Sc_2
      $unroutedClientsList \leftarrow removeClient(unroutedClientsList, client)$
      $positionCandidates \leftarrow computePositionsInRoute(route, client, \mu, \alpha_1, \alpha_2, \alpha_3)$ ▷ Sc_1, Considering capacity and time windows
      if *positionCandidates is empty* **then**
        $solution \leftarrow addRoute(solution, route)$
        $route \leftarrow createNewRoute(client)$
      else
        $positionCandidates \leftarrow sortPositions(positionCandidates)$
        $route \leftarrow insertClientInBestPosition(route, client)$
      end if
    end while
  end while
  return *solution*
end procedure

8.4 Computational Results

Our algorithm RandI3 was implemented as a Java application and used to run instances on an Intel Xeon E5603 at 1.60 Ghz and 8 GB RAM. The implementation uses some state-of-the-art pseudo-random number generator. In particular, some classes from the SSJ library (L'ecuyer and Buist, 2005) were implemented. For preliminary experiments, we use a 100-customers test-bed also proposed by Solomon (1987). All instances are represented by Euclidean distance, and the speed of all vehicles is assumed to be equivalent to the travel unit.

As in the original work, we have used the same four initialization sorting criteria and the parameters values for Solomon criteria values. The parameters values are $(\mu; \alpha_1; \alpha_2; \alpha_3) = \{(1; 0.5; 0.5; 0), (1; 0.4; 0.4; 0.2), (1; 0; 1; 0)\}$. The uniformly selected initialization sorting criteria: (a) the farthest unrouted customer, (b) the unrouted customer with the earliest deadline, (c) the unrouted customer with the minimum equally weighted combination of direct route-time and distance, and (d) actual heuristic criterion value. For each set of values $(\mu; \alpha_1; \alpha_2; \alpha_3)$, 900 iterations per each instance were executed —i.e., 2,700 total iterations per each instance— which selects the best one. The biased random selection of customer is done using a Geometric distribution with $\beta = 0.1$.

Table shows the routes and costs in the Best Known Solution (BKS) from Tan et al. (2001); the routes and costs using the original I3; the routes and costs obtained using our approach RandI3, and finally the gaps between our approach and commented benchmarks. The I3 solutions were obtained with a Java implementation following indications on original article (Solomon, 1987). In general, the solutions are obtained in less than 5 minutes per each instance.

Our approach outperforms the original version of the heuristics with an average improvement of around 14% [column 'Gap (2-3)']. Notice that some BKS are found. However, there is still a positive average gap of almost 10% with the BKS [column 'Gap (1-3)'] that can be improved.

8.5 Future lines

The first point of randomization proposed on our approach could be complemented with a learning approach. As we have appreciated in previous chapters, the selection of a sorting criteria could be naturally improved if we use a biased randomization instead of a uniform selection. The proposed learning approach consist on evaluate the quality of solutions generated by each criteria. Then this information could be used for selecting the criteria with a preference criteria. A biased randomization of this ranking of successful criteria where the best ones would tend to be at the top, will provide a learning approach over the proposed approach.

	BKS		I3		RandI3			
Instance	Routes	Cost (1)	Routes	Cost (2)	Routes	Cost (3)	Gap (2-3)	Gap (1-3)
c101-100	10	829	10	855	10	829	-3.04%	0.00%
c102-100	10	827	11	1376	10	971	-29.43%	17.41%
c103-100	10	828.06	11	1162	10	969	-16.61%	17.02%
c104-100	10	824.78	11	1290	11	958	-25.74%	16.15%
c105-100	10	829	10	855	10	829	-3.04%	0.00%
c106-100	10	827	10	910	10	861	-5.38%	4.11%
c107-100	10	829	10	1027	10	830	-19.18%	0.12%
c108-100	10	827	10	993	10	865	-12.89%	4.59%
c109-100	10	829	10	1063	10	911	-14.30%	9.89%
Average				**1059**		**891.44**	**-14.40%**	**7.70%**
c201-100	3	590	3	590	3	590	0.00%	0.00%
c202-100	3	590	4	840	3	638	-24.05%	8.14%
c203-100	3	591.55	4	1109	3	727	-34.45%	22.90%
c204-100	3	590.6	4	1158	3	707	-38.95%	19.71%
c205-100	3	589	3	657	3	612	-6.85%	3.90%
c206-100	3	588	3	660	3	634	-3.94%	7.82%
c207-100	3	588	3	727	3	617	-15.13%	4.93%
c208-100	3	588	3	662	3	635	-4.08%	7.99%
Average				**800.38**		**645**	**-15.93%**	**9.42%**

Table 8.1: Preliminary results.

8.6 Chapter Conclusions

Again, we have applied a biased randomization of classical heuristic for solving a different branch of VRPs. Biased randomized algorithms have became useful and powerful tools to solve theoretical and real problems. The preliminary results show that our approach is able to produce good results for the VRPTW using a classical heuristic. Although the proposed approach is quite promising, it needs to combine with other approaches for addressing more difficult problems. In next chapters, some tailored approaches for the stochastic scenarios will be studied.

9

Simheuristics

Parts of this chapter have been taken from the co-authored publication:

Juan, Faulin, Jorba, Cáceres-Cruz, and Marques (2013a), Annals of Operations Research.

As we explain in chapter 5, the Combinatorial Optimization Problems (COPs) represent a wide set of real-complex situations. Inside of this huge group, we can find the stochastic COP where the non-deterministic variables are included. In this kind of problem a random element is considered inside of the possible decision actions proper of the COP. On these problems, a random variable is then considered related to the uncertainty of real-life scenarios. Therefore the probability theory is used for assigning a probability distribution representation to internal variables or parameters (Law and McComas, 2002). After more than 20 years, the simulation-based optimization field is still a promising research line. Several studies have been done on this matter for different purposes (Glover et al., 1996, 1999). In fact, the complexity of COPs used to be also related to the size of the problems and not only to the relation and representation of variables (Azadivar, 1999). So large-scale problems are one of the main targets to be optimized. For this complex large-scale problems, some parallel and distributed computing techniques can also be applied.

Several studies have combined simulation and optimization approaches to find original resolution procedures to complex real problems. The supply chain process has been a popular target for this type of techniques. Likely, the work of Eskandari et al.

(2010) is focused on channel coordination on the supply chain process. This study highlights the sensitive influence of stochastic demands in supplier and retailer perspectives. Then a decision support tool based on simulation-optimization is proposed. The authors state that unlike traditional mathematical techniques, the use of simulation-optimization modelling helps to deal with more realistic-complex scenarios. For instance, the scheduling problem in complex assembly lines is studied in Angelidis et al. (2012). The authors proposed a decentralized heuristic based on simulation. Also the particular inventory problem has been addressed using simulation-optimization techniques. Alizadeh et al. (2011) use models with deteriorating items, stochastic lead times, and Poisson demands. The authors are focused on minimizing long-run total expected costs allowing shortages. On this, three stochastic parameters are included in their simulation model: item life time, demands, and lead time. Some other examples of the application of simulation-based optimization can be found in: scheduling (Kim and Kim, 1994), supply-chain (Ding et al., 2004, 2006; Truong and Azadivar, 2003; Zhang and Li, 2004), telecommunication networks (Cabrera et al., 2009; Khandani et al., 2005; Lin and Shroff, 2006; Xiang et al., 1999), city logistics (Barceló et al., 2007; Teklu et al., 2007), among others. The main goal of this chapter is to present a hybrid scheme which combines biased-randomized classical heuristics with Monte-Carlo Simulation, called *Simheuristic* (Juan and Rabe, 2013). As it will be discussed later, this hybrid scheme represents an efficient, relatively simple, parallelizable, and flexible way to deal with several COPs in different fields, even when considering realistic and $non-trivial$ constraints as well as uncertainty values. In the last part of this chapter, the distribution and parallelization of the Simheuristic methodology is discussed.

9.1 Background

The potential of simulation technology based on mathematical basis have been widely proven (Carson and Maria, 1997). In fact, the stochastic behaviour in real systems used to be addressed using simulation. A stochastic system is a set of dynamic-interdependent components where some values of its variables change randomly. As real systems, these require to be also optimized in order to provide better quality of solutions. Therefore simulation-based optimization is a research field that emerges from the combination of optimization and simulation (Deng, 2007). In Fig. 9.1, we can appreciate the basic interaction of these two research lines. From the simulation, it is analyzed by the optimization procedure. Then this process is repeated until a certain stopping condition is satisfied (Glover et al., 1996). Plus, the global economy competition has promoted a great interest in large-scale problems for different types of business. However the natural complexity of this type of systems has delayed the creation of new methodologies. There are many challenges surrounding the optimization of stochastic systems. Simulation-based optimization and even more the proposed Simheuristics can provide some useful answers.

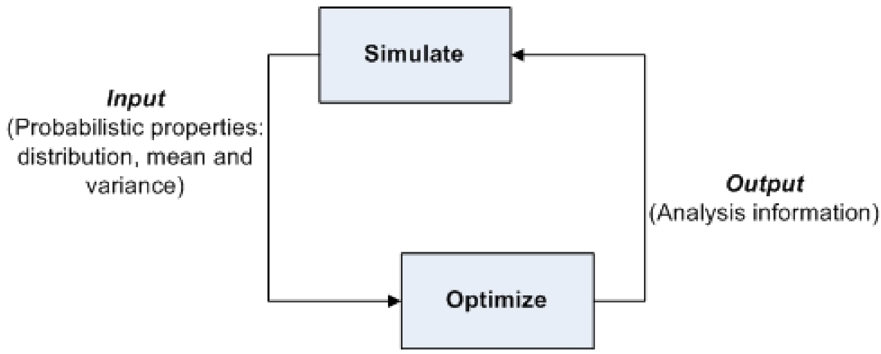


Figure 9.1: General process of Simulated-based Optimization methods.

On this type of stochastic COP, the corresponding objective function is a measurement of an experimental simulation. Due to the complexity of the simulation, the objective function may be expensive to evaluate. Moreover, the evaluation difficulty of the objective function can complicate the optimization process itself (Gosavi, 2003). In fact, this type of research is also related to discrete-event simulation. The reader can find comprehensible surveys on the subject of simulation-based optimization methods, such as Andradóttir (1998); Fu (1994, 2002); Fu et al. (2005). In general, the design of the experiments with stochastic variables needs at least the next basic components:

1. Selection of the random behaviour of a specific variable in the COP which can follows an uniform or non-uniform distribution. This distribution must represent the natural generation of values inside of the random variable. The non-uniform distributions (Geometric, Triangular, LogNormal, etc.) used to represent quite proper the conduct of real-life variables than the uniform selections.

2. Once the probability distribution is defined, several parameters must be setted. There are two universal parameters for this type of approach. The first parameter

is the Expected Value of the random variable. This value can be generated from using the corresponding mean or average. After many trials, the average value of a random variable can be found with the sum of all values between the number of trials.

3. The second parameter is the standard deviation which completes the basic information about the random variable. This value represent the variation or dispersion from the expected value. Then a low variation indicates that the generated values are close to the mean; while a high variance increases the range size of possible generated values, as can be appreciated in Fig. 9.2. On this, all curves have zero as average, and the range of possible values increase from the top to the lowest curve.

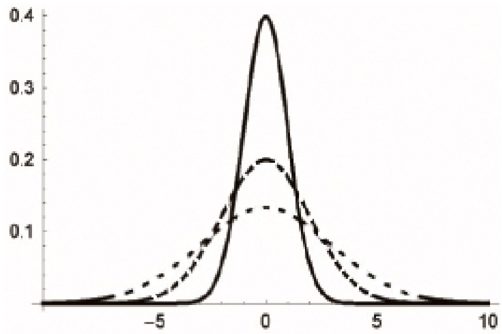


Figure 9.2: Variance examples in a normal distribution (full line, low variance; segmented line, medium variance; dotted line, high variance).

Heuristic methods have proven to be widely useful in many real-world applications (Cordeau et al., 2002; Gendreau et al., 2008; Laporte et al., 2000). In general, they are fast and easy to understand and implement. On simulation-based optimization, the three most popular heuristic methods are: genetic algorithms, tabu search and simulated annealing. In the line of VRP, some Simheuristics have been proposed by Faulín et al. (2008); Hu et al. (2008); Koskosidis et al. (1992) showing good results. We propose a methodology called Simheuristic which consists in the combination of biased-randomized heuristics and Monte-Carlo simulation for addressing complex real-life problems with uncertainty variables. Previous works has proven that this methodology can be easily applied in many research areas like vehicle routing. For instance, the Stochastic Vehicle Routing Problem (SVRP) is a family of well-known vehicle routing problems characterized by the randomness of at least one of their parameters or structural variables (Bastian and Rinnooy-Kan, 1992). This uncertainty is usually modelled by means of suitable random variables which, in most cases, are assumed to be independent. A related problem having only one route is the Stochastic Travelling Salesman Problem (Balaprakash et al., 2010). The Vehicle Routing Problem with Stochastic Demands (VRPSD) is among the most popular routing problems within the SVRP family. There are two other classical problems belonging to that family: the Vehicle Routing Problem with Stochastic Customers (VRPSC) (Bent and Van-Hentenryck, 2004b; Jézéquel, 1985) which was solved by Gendreau et al. (1996b) using an adapted Tabu Search, and the Vehicle Routing Problem with Stochastic Times (VRPST) (Verweij et al., 2003), but their applications are rather limited in comparison with the VRPSD. A good review of all the cases for the SVRP is done by Gendreau et al. (1996a).

Using Simheuristics, the interaction presented in Fig. 9.1 can be translated to a simple routing problem where the random values are integrated at the end of the optimization process. In Fig. 9.3, the routing values (costs) are preliminary defined using a randomized CWS algorithm. Then a simulation of random demands is executed (under a some specific conditions). Notice that this simulation can affect the previous results. So the idea is to define how this routing costs have changed under certain conditions. The creation of this relation depends on the studied problem and the proposed algorithm. In fact, this basic model is used and explained by Juan et al. (2011d).

9.2 Building a Simheuristic

The key aspect for creating a Simheuristic is focused on promoting the interaction between the simulation and the heuristic. On this way, the sequential number of decision steps in the general optimization process harnesses the fast times of the heuristic for producing added-value information. Plus, classical heuristics for solving COPs employs an iterative process in order to construct a feasible —and hopefully good— solution. So this added-value information can improve the decision-making process. The heuristic process can be executed with a set of promising values assigned to the stochastic variables. Perhaps 'simple' problems will require few interaction points with the simulation for the generation of new values. However this is not restrictive. Other problems

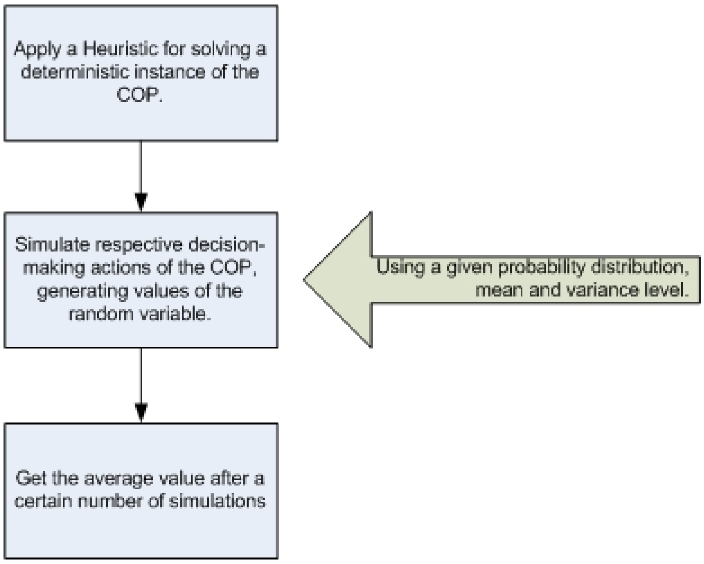


Figure 9.3: Simheuristic example with a single simulation point after the heuristic procedure.

may require several points inside of the process for generate proper information for the decision-making process to be optimized. Therefore the optimization procedure can be considered as repetitive calls to the biased-randomized heuristics for creating a general purpose procedures. Thus, it is possible to identify the following steps when creating a new Simheuristic algorithm by means of biased randomization:

1. Given a COP, select a biased-randomized heuristic inside of a multi-start-like approach for generating fast and useful information for the general optimization process.

2. Once the base heuristic is selected, the most proper generation point of values must be defined in the general optimization procedure. Maybe it is just necessary one generation simulation at the beginning of the heuristic process. Most complex problems will probably require several simulation outputs in order to recreate the most proper behavior of stochastic variables inside of optimization process.

3. Define a set of scenarios to be studied where each one is related to a probability distribution, mean, and level of uncertainty. There are three basic stochastic levels that could be considered in order to represent the most appropriated variance of the variable: low (25%), medium (50%) or high (75%).

4. Optionally, the simulated values can be related to a specific set of policies or characteristics of the COP. This will allow a post biased-randomization of obtained results. The built solution can be associated to the specific used characteristic and compared with others in order to create a rank of characteristics in a sorted set. The main advantage of this process is to generate a set of alternatives scenarios for finding the best solution in a given uncertainty level.

Plus, the general optimization procedure described above is able to quickly generate several feasible solutions with different characteristics and under specific probabilistic conditions. Therefore, a list containing the top 'best-found' solutions —each of them having different properties— can be saved and considered by the decision maker.

9.3 Benefits

As said in chapter 5, the desirable features of a metaheuristic, described by Cordeau et al. (2002), are the main evaluation aspects —i.e., accuracy, speed, simplicity, and flexibility. In general, the two first features are quite popular for measuring the performance of a solution method. The quality of solutions used to be represented by the numerical cost obtained in a given period of execution time. However, the simplicity aspect is an important factor that is focused on an easy implementation and parameterization. Finally, the flexibility is focused on the adaptation of a given method to be modified for a different problem or constraint set. The natural adaptation to different realistic scenarios is a feature quite demanded between the solution methods.

Having in mind these measured attributes, we list the main benefits of Simheuristic over other related approaches:

- The ever-increasing complexity of systems can be considered, like the real-natural representation of variants in mathematical models (e.g., stochasticity). Complex relations and real variables can be modelled in a comprehensible way.

- The use of different probabilistic properties (e.g., uncertainty levels) in stochastic variables offers a more natural and efficient way to select the most proper solution in different realistic scenarios. This offers a well-known starting point to conditionate the execution of any Simheuristic (parameterization).

- The generation of internal added-value information from simulation allows to intensify the search in the solution space in the promising regions. In fact, it can produce a set of solutions with different properties in order to offer different solution-scenarios to the decision maker.

- Being based on well-tested heuristics, they are relatively simple and easy to implement methods, which can be adapted to account for new constraints (flexibility). Plus, the general performance of heuristics used to be quite fast.

- The natural and easy parallelization of this general process combined with multi-start-like approaches and different probabilistic properties.

9.4 Cooperative and Distributed Approaches

Many small- and medium-enterprises (SMEs) could employ this type of methodology (Simheuristic) for solver complex real problems. Furthermore, the performance of these methods can be improved with some Parallel and Distributed Computing Systems (PDCS) in order to save time and money on implementation projects. Usually, SMEs in the logistics business lack technical expertise and high-tech computational resources. It is not likely that they can afford buying expensive software or powerful computer systems to solve their complex routing problems in real time. In such scenarios, two alternative PDCS approaches are possible: (a) to use thirdparty resources on demand, i.e., a cloud system; or (b) to employ idle computing capabilities of SME's desktop computers. As described in Armbrust et al. (2010), cloud computing systems are data centers that make available they hardware and software to the general public in a pay-as-you-go manner. Amazon's Elastic Compute Cloud (`http://aws.amazon.com/ec2`) and Microsoft's Azure Services Platform (`www.microsoft.com/windowsazure`) are examples of this kind of systems. Cloud users have complete control on rented resources presented as virtual machines. However, many SMEs may not like this model for data-privacy issues, i.e., they might want to avoid running and storing sensitive and confidential information of their business in servers located in an external company. In addition, pricing may be too expensive for some SMEs, although this factor might be less important than the previous one since current prices of these cloud services are quite affordable.

The second option, to use idle enterprise resources, is based on the aggregation of unused resources from existing computers in a SME in order to concurrently execute thousands of clones or instances of an algorithm. This way, pseudo-optimal solutions for large and complex real-life problems might be obtained in nearly real time at an inexpensive monetary cost. This approach has some similarities with the so-called “volunteer computing” or “contributory computing” model of distributed computing. In this model, computer owners donate their computing resources to some scientific or academic projects. In effect, a standard SME owns a number of commodity computers distributed among its different departments and/or facilities. Most of these personal computers offer more computing capabilities than required to complete their daily activities, which in most cases involve using word processors, spreadsheets, e-mail, etc. Moreover, they happen to be underutilized or idle during nightly hours. Thus, it makes sense to spare resources from each computer and aggregate those resources into a computational environment where hundreds or even thousands of instances of a parallelizable algorithm, like the one presented here, can be run simultaneously. As Fig. 9.4 shows, resources from a SME may be federated with resources from other SMEs, therefore resulting in an even larger PDCS. To avoid interferences with the current tasks executed in each computer, contributed resources could be provided through the use of virtual machines. Therefore, whenever a user in a SME needs to solve a computationally intensive problem, it sends a query to the Directory-of-resources service, which keeps updated information about available computing resources in the federated network.

Resources might be provided as virtual machines running over real computers or by a middleware. Once the Directory service has provided the user with a list of available resources, it can submit the task (a VRP instance, for example) to be executed (solved) by them. As more computational resources become available, more agents (algorithm’s instances) will be concurrently executed, thus increasing the chances of finding pseudo-optimal solutions in a reduced time-period.

The idea of aggregating computational resources from different machines in a network has been successfully explored in several works and real-life applications. In particular, the Volunteer Computing platforms (Anderson, 2004; Marques et al., 2007) aggregate computing capacities from the edges of the Internet. Those platforms offer tools

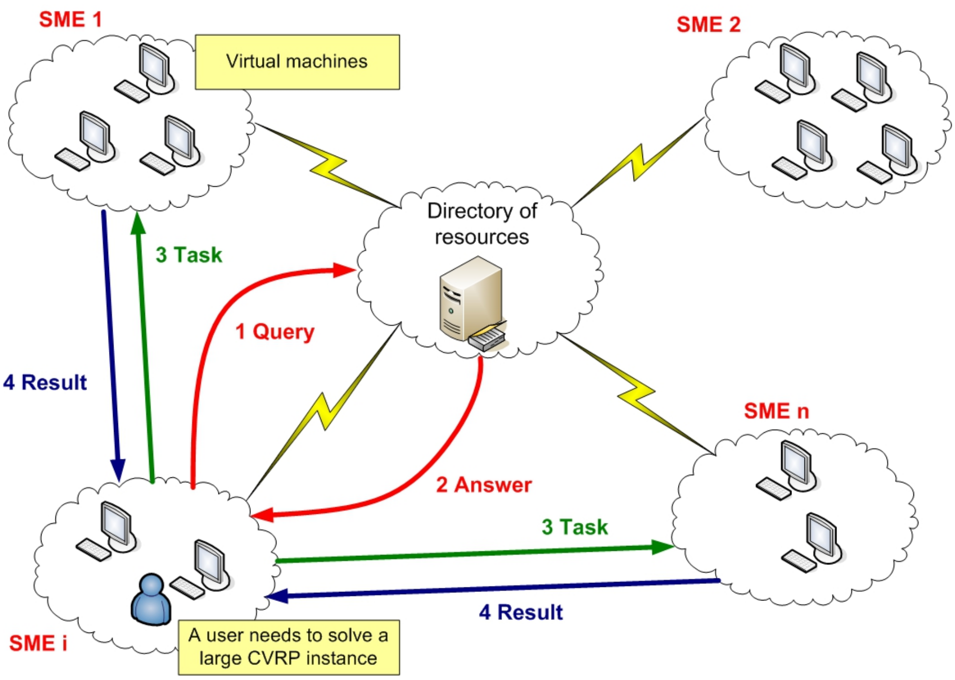


Figure 9.4: A distributed computing approach.

to create adhoc communities that perform massive computation by aggregating the resources of their participants. Amongst others, networks such as Seti@HOME (`http://setiathome.berkeley.edu`), Distributed.net (`http://www.distributed.net`) or Prime.net (`http://www.mersenne.org/prime.htm`) are examples of those communities. In particular, the Berkeley Open Infrastructure for Network Computing (BOINC) is a middleware that offers the functionalities to build up a volunteer computing network (Anderson, 2004). Each client (computing node) is linked to one or more servers (application specific entry nodes). When tasks are submitted for execution to the community they are replicated for redundancy and distributed amongst clients. Results are collected and validated before being delivered to the final user. Finally, it is interesting to notice that these large-scale volunteer-computing systems open interesting challenges to the Operations Research/Computer Science community. For instance, in order to be efficient, these systems need to consider some issues related to the Reliability and Availability (R&A) levels of their nodes and the services they offer. These systems are usually characterized by extremely dynamic and heterogeneous environments, where nodes offering different computer capabilities and features can enter or leave freely.

This dynamism and heterogeneity introduce uncertainty which, in turn, makes it difficult to develop accurate models to predict the temporal evolution of the RA levels in distributed environments. In addition to that, most of the applications to be executed in these contributory systems have different components with different roles that have to be scheduled in a way that satisfies the overall reliability.

9.4.1 Related work of PDCS for VRP

As in most other COPs, instances of interest in the VRP arena are becoming larger in size as well as in complexity in terms of constraints and objective functions, including multi-objective and non-smooth functions (Crainic, 2008; Talbi, 2009, 2012). In particular, most researchers are focused on specific versions of metaheuristics applied to different VRP variants, such as the CVRP, the VRPTW, the VRPSD, etc. Some of these VRP versions might present dynamic (time-varying) conditions or multiple scenarios which require a high computational efficiency without decreasing solution quality. Normally, parallel and distributed methods in VRP are used based on: (a) how the global search is conducted, i.e., either by a unique process or by a coordinated set of processes; (b) the type of communication and synchronization patterns during the global search, which might require different amounts of data exchange; and (c) whether or not the synchronization steps are rigid. Typically, another point of interest is the set of initial parameters of the search, which can be used to find one particular solution from a set of solutions with different constraints or different objective functions, thus generating multiple analysis scenarios. Furthermore, algorithm parallelization can be done in different ways depending on the problem and the hardware/software computing platform being employed.

Several parallel and distributed computing approaches have been already applied to different VRP variants. Generally speaking, one common resource is to use a parallel/distributed 'master-slave' approach, where the master (coordinator) processor can take a sequential-based search and dispatch intensive computations to a set of slave computation processors or workers (Fig. 9.5).

Alternatively, the master processor can also take a combination of initial parameters/ constraints or a set of alternative scenarios, and distribute those scenarios among the slave processors for a concurrent execution. Information sharing at global level can be then used as a way to improve the local searches/scenarios. Some of the simplest

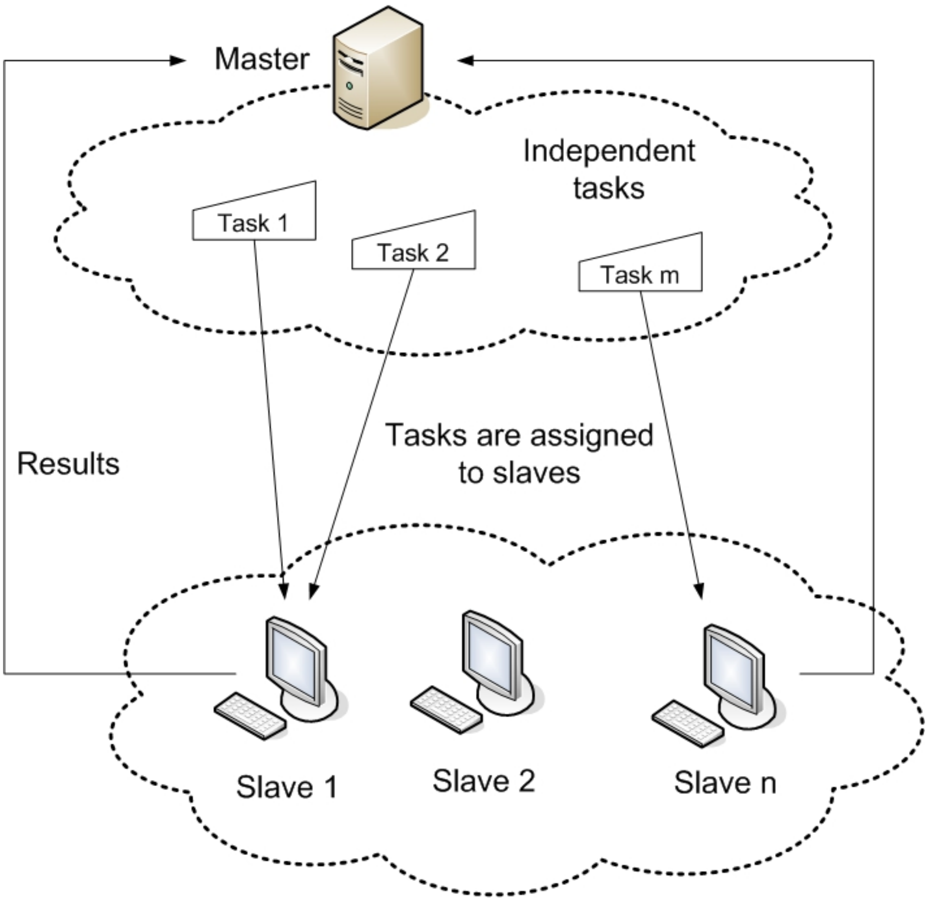


Figure 9.5: A typical master-slave schema in a distributed system.

applications of parallel and distributed models do not involve shared information but rather attempt to use as much parallel computation as possible. Jeevan-Madhu and Saxena (1998) review some initial attempts of applying parallel techniques to solve different VRPTW instances. For example, one of the approaches they describe consists on the use of parallel techniques to find minimum-cost routes between pairs of nodes, so that both the time employed to serve all locations and the sum of waiting times are minimized. All these parallelization techniques are based on: (a) the idea of subdividing the direct acyclic graph associated with the VRP instance into several subgraphs, which are then assigned to the available processors in each interaction; and (b) the use of composition operations to allow sharing global information in each step. They use Parallel-shared RAM (PRAM) memory models of computation to evaluate algorithms of high complexity. Intuitively, this conceptual model corresponds to the programmers' view of a particular kind of parallel computers (one with a shared-memory multiprocessor), but it ignores lower-level architectural constraints and some other important details, such as memory access contention and overheads, synchronization overheads, interconnection network throughput, connectivity, speed limits and link bandwidths, etc. Those parameters, in fact, limit the performance obtained by parallel or distributed implementations and are not appropriately considered in most parallel approaches to VRPs.

Protonotarios et al. (2000) propose an approach based on Genetic Algorithms (GAs) to the VRP with time windows and stochastic demands. These authors use HPC techniques to score each chromosome's fitness in a parallel way. Hence, it is possible to consider larger problems. They try to reduce the amount of communication between processes by replicating the genetic part in each process and maintaining, during the evolution, the evaluation process spliced between processors. This helps to balance computation efforts as each processor has a similar number of chromosomes. Synchronization phases are then carried out to communicate the scores of all the population. They have developed an experimental test-bed based on the use of multi-computers in a LAN environment, as well as some shared-memory trials, which obtained the best-known results due to the shared-chromosomes population scores. In a similar way, Berger and Barkaoui (2004) propose solving the VRPTW by employing a hybrid strategy based on the use of GAs in a master-slave structure, which is implemented using the message-passing paradigm. In their approach, parallel slaves evolve into two populations to concurrently try to minimize total traveled distance and temporal constraint violations. Rego (2001) uses a network of multi-computers, with message passing implementation to explore the parallelization of a Tabu Search strategy for the VRP with capacity and distance restrictions. The parallel Tabu Search algorithm follows a master-slave model, where each slave executes a complete Tabu Search algorithm with a different set of parameters, starting with the initial solution provided by the Clarke and Wright (1964) heuristic. Then, the algorithm collects the best-known local solution from each slave and retransmits it to all slaves for the next iteration. Ghiani et al. (2003) review different parallel strategies related to both Tabu Search and dynamic/stochastic VRPs, where an initial effort is needed to obtain a starting near-optimal solution and then recalculations are done based on dynamic demands. Their paper experiments with some masterslave strategies running over an affordable network of computers (cluster computing). Their approach uses data-domain decomposition. Le-Bouthillier and Crainic (2005) propose a functional parallelism where different hybrid metaheuristics are executed concurrently. Particularly, these authors use an evolutionary algorithm and a Tabu Search with a central point of communication called the solution warehouse, where the partial solutions are stored. The metaheuristics processes do not have inter-communications, thus making the cooperation design simple and allowing them to test different metaheuristics without changing the main collaboration design.

Other recent approaches are focused in less studied variants of VRP. For example, Mitra (2007) has researched VRPs with split deliveries/pickups allowed in each location. To solve this problem, he proposes a parallel-clustering technique, which employs a fixed number of clusters equal to the minimum number of vehicles to fulfill the demands. He uses different steps to assign elements to the clusters, thus minimizing the distances among the elements in each cluster. Once the clusters are consolidated, the only remaining step is to schedule the vehicle routes within each cluster independently, which can be done through a general route-construction heuristic. This approach allows for data-parallel domain decomposition and, later, for a parallel route-construction. Subramanian et al. (2010) examine the VRP with Simultaneous Pickup and Delivery (VRPSPD). In their work, a parallel algorithm is used to start a multi-heuristics local search in a master-slave structure. Some of the experiments are performed with clusters of multi-core processors (in a HPC environment), which allows for analyzing the algorithm scalability as more CPU cores are added. In fact, in a hybrid multiprocessor/multi-computer environment using Message Passing Interface (MPI) to communicate master and slaves processes, a scale of 256 cores is used as a test-bed. One particularly interesting aspect of this paper is that the authors have tested and studied some of the performance bottlenecks in their implementation, some of which are related to communication overheads in the message-passing parallel paradigm, and its implications in algorithmic efficiency with their hardware platform.

In addition to the preceding parallel approaches, advances in hardware parallel architectures (Kirk and Wen-mei, 2010) have created new opportunities to study some of the unexplored computational areas for VRPs and other combinatorial optimization problems. In particular, computational paradigms like shared memory can now utilize a new form of computation available by combining multiple computation-dedicated cores or CPU units. In some cases, several cores are present in multi-core generic purpose CPUs, currently between 2 and 12 cores per chip die. Multi-core solutions can also be found in current Graphic Processing Units (GPUs). These GPUs are based on simplified core architectures making available hundreds of cores for computation in new programming models like Nvidia CUDA and OpenCL (Sanders and Kandrot, 2010). Notice, however, that not all the metaheuristics and combinatorial optimization problems will adapt well to these new parallelization models. For example, when using multi-cores with multiple communicating agents or tasks several synchronization-contention problems must be addressed in order to avoid performance bottlenecks. In other cases, such as in GPU multi-cores, the restrictions of local memory available (only some small number of KBytes per core in current hardware models), and the many systematic levels of memory, create a massive and inefficient movement of computation data during the algorithm execution. Thus, it might be very difficult to make efficient implementations for some algorithms or, in most cases, to be able to provide performance gains in comparison to their counterpart serial implementations. However, these new parallel architectures are a promising new computational background to be explored that may lead to a new generation of combinatorial-optimization algorithms.

As said before, DPCS offer the possibility of accelerating computations. Several surveys could be found in Crainic (2008); Crainic and Toulouse (2003); Talbi (2012). The method consists in a combination of search efforts of different sub-methods. However, classical parallel approaches, based on functional or data decomposition, do not significantly modify the search trajectories of metaheuristics. Thus, they cannot improve the quality of the solution, nor do they enhance the robustness of the search when faced with different problem instances than those which were originally calibrated and applied. Consequently, in recent years, multi-search (or multi-thread) metaheuristics, with varying degrees of cooperation, have increasingly been used for difficult combinatorial problems and have been shown to both speed up the search and dramatically improve the robustness and the quality of the solutions obtained (Le-Bouthillier and Crainic, 2005). The complexity of this research line could rise high because the problem for the researcher is centered in determining the information to be exchanged, the exchange points in the algorithm, the moment where it happens (synchronized or asynchronized), and how each agent or thread uses this information. On this research line, Crainic has developed a large path of achievements. As the authors of Crainic and Toulouse (2003) state:

> "the first goal is to solve larger problem instances in reasonable computing times. In appropriate settings, such as co-operative multi-thread strategies, parallel meta-heuristics also prove to be much more robust than sequential versions in dealing with differences in problem types and characteristics. They also require less extensive, and expensive, parameter calibration efforts".

In the paper Le-Bouthillier and Crainic (2005), the authors have proposed a new co-operative parallel metaheuristic for the VRP with time windows. It is based on the solution warehouse strategy, in which several independent search threads cooperate by asynchronously exchanging information on the best solutions identified. The logic within each process consists in the implementation of a different metaheuristic —i.e., an Evolutionary Algorithm or a TS procedure, without any particular calibration of parameters and methods. In addition, construction and improvement heuristics were also included to generate an initial set of solutions in order to perform post-optimization. The proposed metaheuristic displays good performance in terms of solution quality and computational effort. It was tested with a set of instances in the range of 200 and 1000 customers. The authors stated that the cooperative framework is simple to implement and expand to other problems.

The article of Crainic et al. (2009a,b) proposes a self-adaptive meta-heuristic, called Integrative Concurrent Evolutionary Method (ICEM). ICEM is focused on the decomposition of a given VRP along subgroups of attributes and the concurrent evolution of heterogeneous populations. The concurrent evolution is based on the cooperative metaheuristic paradigm, which proposes the parallel execution of the methods using some degree of communication. They apply their method to a Rich VRP model that includes duration and capacity constraints as well as time windows, multiple periods and multiple depots. They proposed future tests and the creation of benchmarks for this type of problems. On the line of using LNS, Bartodziej et al. (2010) propose a parallelizable framework to address the VRP with pickup and delivery and time windows. They have tests different scenarios with LNS sub-heuristics using (Li and Lim, 2003) instances of 200 customers. For instance, Yu and Zhen-Yang (2011) propose a coarse-grained parallel Ant Colony Optimization algorithm. In the community is quite natural to propose a parallel variant of a promising technique. For example, a parallel version of the Unified Tabu Search approach (Cordeau and Laporte, 2003; Cordeau et al., 1997, 2001b, 2004) has been proposed later by Cordeau and Maischberger (2012).

9.5 Chapter Conclusions

In this chapter, we have described an emerging approach based on the combination of MCS and randomized classical heuristics. As complex scenarios, approaches for stochastic VRPs can consider to include some other promising techniques, like MCS. Nowadays, the combination of complementary techniques is getting quite popular in the research community —e.g., Matheuristics (Doerner and Schmid, 2010). The uncertainty modelling feature of MCS mixed with efficient and fast VRP heuristics can create interesting approaches for real-life problems. Even more, the advantages of Simheuristics can increase using distributed and parallel techniques. The role of parallel and distributed computing systems for solving combinatorial optimization problems and, in particular, vehicle routing problems, has been discussed. A literature review shows that the use of parallel strategies is a well-established and increasingly relevant topic in combinatorial optimization. Potential applications of distributed computing to solve large-size VRPs with real-life constraints have also been pointed out. In next two chapters, we will present the application of this methodology for solving the VRPSD and the IRPSD (see Fig. 9.6).

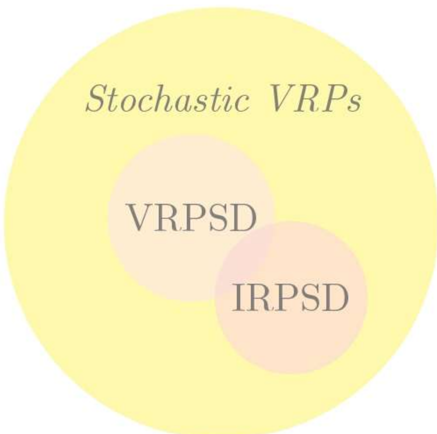


Figure 9.6: VRPs studied in this dissertation using Simheuristics.

10

VRPs with Stochastic Demands

Parts of this chapter have been taken from the co-authored publication:

Juan, Faulin, Jorba, Cáceres-Cruz, and Marques (2013a), Annals of Operations Research.

In a broad sense, which includes also variants and extensions, Vehicle Routing Problems (VRPs) comprise a popular family of combinatorial-optimization problems which is a natural area of application for PDCS. This is especially the case when considering complex scenarios given by large-size instances, real-life constraints (e.g., time windows, maximum route length, service priorities, etc.), dynamic conditions, intangible costs (e.g., environmental costs due to pollution), or uncertainty conditions (e.g., stochastic or fuzzy demands). VRPs constitute a relevant topic for current researchers and practitioners. In fact, according to Eksioglu et al. (2009), the number of VRP-related articles published in refereed journals has experienced an exponential growth in the last 50 years. One of the most challenging vehicle routing problems is the VRP with Stochastic Demands (VRPSD). The VRPSD is a NP-hard problem in which a set of customers with random or stochastic demands must be served by a fleet of homogeneous vehicles departing from a depot, which initially holds all available resources. There are some tangible costs associated with the distribution of these resources from the depot to the customers. In particular, it is usual for the model to explicitly consider costs due to moving a vehicle from one node, customer or depot, to another. These costs are often related to the total distance travelled, but they can also include other factors such as number of vehicles employed, service times for each customer, etc. The classical goal here consists of determining the optimal solution (set of routes) that minimizes those tangible costs subject to the following constraints: (i) all routes begin and end at the depot; (ii) each vehicle has a maximum load capacity, which is considered to be the same for all vehicles; (iii) all (stochastic) customer demands must be satisfied; (iv) each customer is supplied by a single vehicle; and (v) a vehicle cannot stop twice at the same customer without incurring penalty costs. The main difference between the classical CVRP and the VRPSD is that in the first one all customer demands are known beforehand, while in the second one the actual demand of each customer has a stochastic nature, i.e., its probability distribution is known beforehand, but its exact value is revealed only when the vehicle reaches the customer. For the CVRP, a large set of efficient optimization methods, heuristics, and metaheuristics have been already developed (Golden et al., 2008; Laporte, 2007). However, this is not yet the case for the VRPSD, which is a more complex problem due to the uncertainty introduced by the random behaviour of customer demands. Therefore, as suggested by Novoa and Storer (2009), there is a real necessity for developing more efficient and flexible approaches for the VRPSD. On one hand, these approaches should be efficient in the sense that they should provide optimal or near-optimal solutions to small and medium VRPSD instances in reasonable computing time. On the other hand, they should be flexible in the sense that no further assumptions need to be made concerning the random variables used to model customer demands, e.g., these variables should not be assumed to be discrete neither to follow any particular distribution.

To the best of our knowledge, most of the existing approaches to the VRPSD do not satisfy the aforementioned efficiency and flexibility requirements. Therefore, one of the major contributions of this chapter is the application of an efficient and flexible methodology that combines Monte-Carlo simulation and parallel-computing to provide real-time solutions to the VRPSD (*Simheuristics*).

10.1 Definition

Consider a complete network constituted by $n+1$ nodes, $\Omega = \{0, 1, \ldots, n\}$, where *node* 0 symbolizes the central depot and $\Omega* = \Omega/\{0\}$ is the set of nodes or vertices representing the n customers. The costs associated with travelling from node i to node j are denoted by $c_{ij}, \forall i, j \in \Omega$, where the following assumptions hold true: (i) $c_{ij} = c_{ji}$ (i.e., costs are usually assumed to be symmetric, although this assumption could be relaxed if necessary); (ii) $c_{ii} = 0$, and (iii) $c_{ij} \leq c_{iu} + c_{uj}, \forall u \in \Omega$ (i.e., the triangle inequality is satisfied). These costs are usually expressed in terms of travelled distances, travelling and service times or a combination of both distances and times. Let the maximum capacity of each vehicle be $VMC \gg Max_{i \in \Omega*}, \{D_i\}$, where $\{D_i\} \geq 0$ ($\forall i \in \Omega*$) are the independent random variables that describe customer demands (it is assumed that the depot has zero demand). This capacity constraint implies that the random demand value will never be larger than the VMC, which allows us an adequate performance of our procedure. For each customer, the exact value of its demand is not known in advance; it is revealed when the vehicle visits the node. No further assumptions are made on these random variables other than that they follow a well-known theoretical or empirical probability distribution, either discrete or continuous, with existing mean denoted by $E[D_i]$. In this context, the classical goal is to find a feasible solution (set of routes) that minimizes the expected delivery costs while satisfying all customer demands and vehicle capacity constraints. Even when these are the most typical restrictions, other constraints and factors are sometimes considered, e.g., maximum number of vehicles, maximum allowable costs for a route, costs associated with each delivery, time windows for visiting each customer, solution attractiveness or balance, environmental costs, and other externalities.

10.2 Literature review

The study of the VRPSD is within the current popularity of introducing randomness into combinatorial problems as a way of describing new real problems in which most of the information and data cannot be known beforehand. This tendency can be observed in Van-Hentenryck and Bent (2009), which provides an interesting review of many traditional combinatorial problems with stochastic parameters. Thus, those authors studied Stochastic Scheduling, Stochastic Reservations and Stochastic Routing in order to make decisions on line, i.e., to re-optimize solutions when their initial conditions have changed and, therefore, are no longer optimal. This type of analysis has designed the Online VRP in which re-optimization is needed apart from a previous situation.

This set of routing problems seems to be well analyzed with the use of stochastic hypothesis in their definitions Bent and Van-Hentenryck (2007) providing more reality in their formulation. Another routing field in which randomness has also been developed is the resolution of inventory routing problems where the product usage is stochastic (Hemmelmayr et al., 2010). Bianchi et al. (2009) have written an interesting survey of the appropriate metaheuristics to solve a wide class of combinatorial optimization problems under uncertainty. This survey is a good reference for obtaining an appropriate list of articles regarding the use of metaheuristics in VRPSD and other related problems.

The random behaviour of customer demands could cause an expected feasible solution to become infeasible if the final demand of any route exceeds the actual vehicle capacity. This situation is referred to as "route failure", and when it occurs some corrective actions must be introduced to obtain a new feasible solution. For example, after a route failure, the associated vehicle might be forced to return to the depot in order to reload and resume the distribution at the last visited customer. Of course, it is also possible to consider preventive vehicle reloads even before the actual route failure occurs, e.g., when the expected demand of the next customer exceeds the current load of the vehicle. Some authors have already focused on modelling the costs associated with these route failures (Tan et al., 2007). Our methodology proposes the construction of routes in which the associated expected demand will be somewhat lower than the vehicle capacity. Particularly, the idea is to keep a certain amount of surplus vehicle capacity (safety stock or buffer) while designing the routes so that if the final routes' demands exceed their expected values up to a certain limit, they can be satisfied without incurring a route failure. The idea itself is not new in the literature. Sungur et al. (2008), for instance, built a robust solution approach for the VRPSD using adequate management of the remaining vehicle capacity compared to a uniform and non-uniform distribution of that slack over all the considered vehicles. However, while their goal is to find a robust solution "that optimizes the worst case value over all data uncertainty", our goal is to find robust solutions with optimal or pseudo-optimal total expected costs for a given uncertainty scenario. Moreover, we plan to do that in 'real-time' by developing a simple, flexible, efficient and parameter-free algorithm that can benefit from current trends in parallel and distributed computing. Precisely, the focus on the parallel and distributed computing approach is one of the main differences between this work and the paper of Juan et al. (2011d). Another fundamental difference resides in the core algorithm. On one hand, the algorithm proposed in the aforementioned reference uses a two-stage approach where the deterministic and the simulation stages are employed in a sequential way, i.e., the simulation is only executed once the deterministic stage has finished. This implies that the simulation stage is only applied to the best-found deterministic solution. On the other hand, the algorithm presented in this study integrates the simulation inside the deterministic stage, which implies that the simulation process will now be executed each time a 'promising' solution for the deterministic problem is generated. As expected, the numerical tests performed show that the integrated approach provides better results than the sequential one. Finally, by incorporating the parallel computing approach, computing times are significantly reduced to provide 'real-time' solutions.

10.3 Proposed Approach

As introduced before, our approach deals with uncertainty in the customer demands by considering a safety stock in the vehicle load, i.e., a certain percentage of the vehicle maximum capacity is not accounted for when designing the routes. Instead, this percentage is reserved to deal with potential emergency situations caused by unexpected demands. Using safety stocks not only contributes to reduce variable costs due to route failures but, related to that, it also increases the reliability or robustness of the planned routes, i.e., as safety stock levels increase, the probability of suffering a route failure diminishes. Notice, however, that employing safety stocks also increases fixed costs associated with aprioristic routing design, since more vehicles and more routes are needed when larger buffers are considered. Therefore, when minimizing the total expected cost a trade-off exists between its two components, fixed costs and expected variable costs. Thus, the challenge relies in the selection of the appropriate buffer size. As Fig. 10.1 shows, given a VRPSD instance, our approach considers different levels of this buffer size and then solves the resulting scenarios in parallel by assigning each to a different processing unit. This is performed by employing a modified version of the algorithm introduced in Juan et al. (2011d). As it will be explained later with more detail, in this modified version a Monte-Carlo simulation stage is integrated inside the multi-start process, which allows estimating the variable costs associated with each candidate solution. Thus, among the multiple solutions generated for each scenario, the ones with lowest total expected costs are stored as the best-found result associated with the corresponding safety-stocks level. Once the concurrent execution of the different scenarios ends, the corresponding solutions are compared to each other and the one with the lowest total expected costs is selected as the best-found routing plan.

Once a general overview of the multiple-scenario approach has been given, it is time to explain how the SR-GCWS-CS algorithm (Juan et al., 2011e), a randomized algorithm which was originally designed to solve deterministic CVRP instances, has been modified to deal with VRPSD instances. Notice, however, that even when a general overview of the algorithm will be given next, it is not the goal of this study to explain the SR-GCWS-CS algorithm in detail since it has been extensively described and tested in the aforementioned reference. Instead, this section focuses on how a Monte-Carlo Simulation stage has been integrated into the multi-start constructive process defined in the original algorithm in order to obtain, for each generated solution, estimates of its expected variable costs.

During its multi-start construction stage, the SR-GCWS-CS algorithm introduces a biased random behaviour within the CWS heuristic in order to generate alternative starting solutions satisfying the problem constraints. Each of these feasible solutions consist of a set of round-trip routes from the depot that, altogether, satisfy all demands of the nodes by visiting and serving all of them exactly once. While the classical CWS heuristic always chooses the edge with the largest savings value at each step, the SR-GCWS-CS uses a pseudo-geometric distribution to assign a selection probability to each edge in the savings list. Therefore, for each potential edge its probability of being selected is coherent with its savings value, i.e., edges with greater savings will be more likely to be selected from the list than those with smaller savings. By iterating this solution-construction process, different randomized CWS solutions, some of them outperforming the original CWS solution, can be obtained in just a few milliseconds for most small- and mid-size instances. Each time a new randomized CWS solution is generated it is compared against the original CWS solution. If the new randomized solution outperforms the CWS one, then a local search process is applied to the new solution in order to further improve it. This local search process uses: (a) a cache or memory-based stage that allows to quickly substitute specific routes in the current solution by previously found routes covering the same set of customers in a less costly

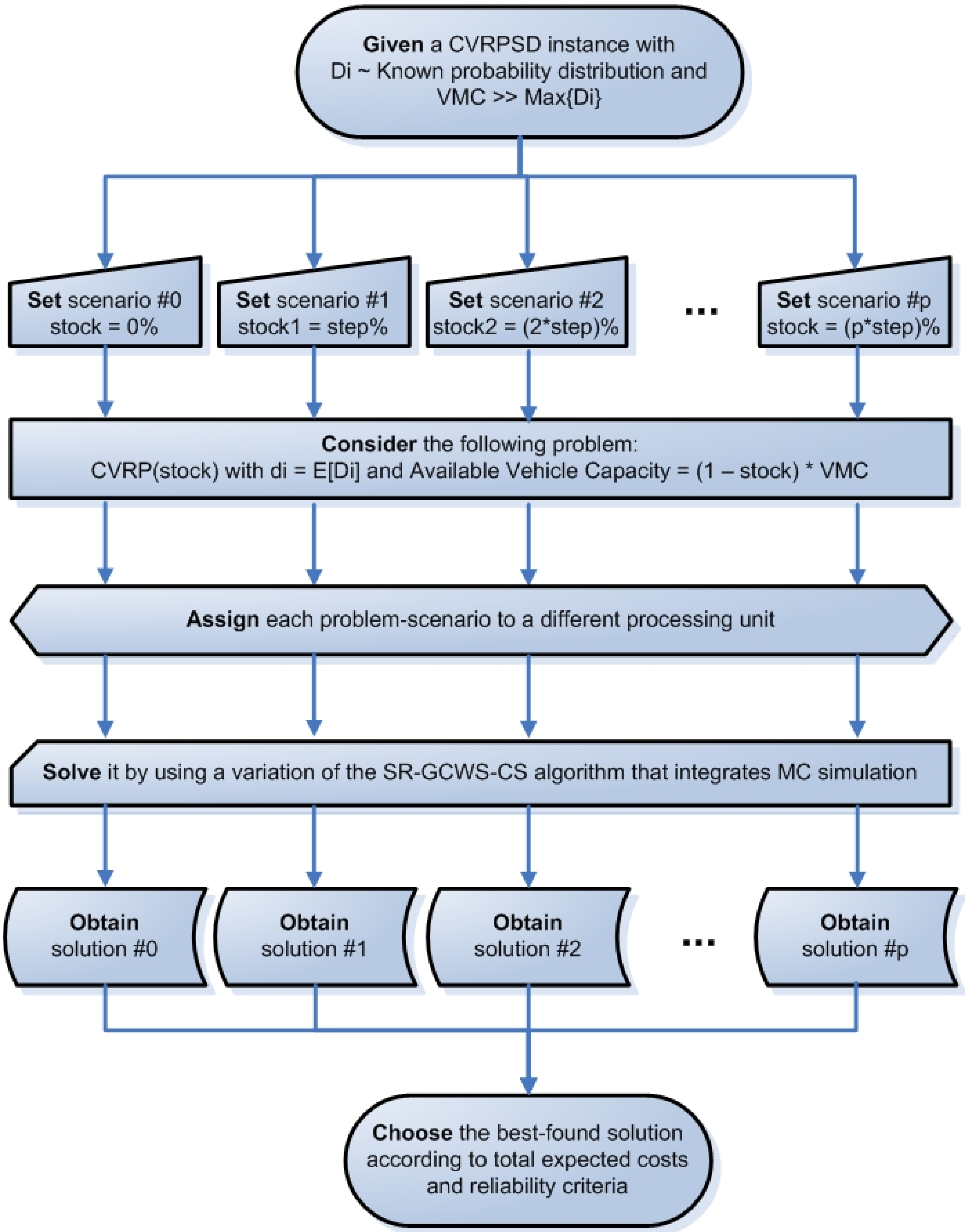


Figure 10.1: A multiple scenario approach based on the safety stocks level.

order; and (b) a splitting or divide-and-conquer stage that allows to reduce the combinatorial complexity of the instance being solved. Pseudo-code 7 shows the logic flow of this main procedure. The algorithm receives as input, the nodes to be served, the set of constraints, the costs matrix, and the algorithm parameters, including the random number generator (RNG), the number of best solutions to save (*nSols*), and the number of first- and second-level iterations to run (*nIter* and *nIterPerSplit*, respectively). Then, the savings matrix is calculated and a savings list is constructed and sorted. The resulting list contains the potential edges to be selected sorted by their associated savings. Later, an initial solution is obtained applying the CWS heuristics.

Algorithm 7 Main procedure of the modified SR-GCWS-CS algorithm.

procedure SR-GCWS-CS($vrpNodes, vrpConsts, algParam, cMatrix$) ▷ $vrpNodes$ includes customers' coordinates and demands; $vrpConsts$ includes vehicle available capacity constraint (considering safety stocks); $algParam$ includes rng, nSols, nIter and nIterSplitting; and $vrpSol$ represents a given solution for the deterministic VRP.
    $savList \leftarrow makeSavingsList(vrpNodes, cMatrix)$
    $cwsSol \leftarrow constructCWSSol(vrpNodes, cMatrix, savList, vrpConsts)$
    while *stopping criterion is not satisfied* **do** ▷ It depends on nIter
        $vrpSol \leftarrow constructRandomSol(vrpNodes, cMatrix, savList, vrpConsts, rng)$
        $vrpSol \leftarrow improveSolWithRoutesCache(vrpSol, cMatrix)$
        if *vrpSol outperforms cwsSol* **then**
            $vrpSol \leftarrow sppliting(vrpSol, cMatrix, savList, vrpConsts, algParam, rCache)$
            $calcExpectedCosts(vrpNodes, vrpConsts, vrpSol)$
            $bestSols \leftarrow updateBestSolsList(vrpSol, bestSols, nSols)$
        end if
    end while
    return $bestSols$
end procedure

The costs associated with this solution will be used as an upper bound limit for the costs of what we will consider a good solution. It is at this point when we start the first-level iterative process to generate new solutions outperforming the CWS. At each first-level iteration, a new solution is constructed by using the randomized CWS heuristic (Pseudo-codes 8 and 9); then this new randomized solution is processed by the cache procedure, which uses cache best results from previous iterations to improve, if possible, the current randomized solution. If the resulting solution outperforms the CWS heuristic, it is considered a promising solution and it is then processed by the splitting procedure; this splitting procedure tries to improve it by first considering different subsets of routes (i.e., by reducing the problem dimension), and then applying a second-level iterative process over each of these subsets.

Algorithm 8 Randomized CWS procedure to generate a random initial solution.

```
procedure CONSTRUCTRANDOMSOL(nodes, cMatrix, sList, rng)
    effList ← copyList(sList)
    sol ← constructInitialSol(nodes, cMatrix)
    while effList contains edges do            ▷ It depends on nIter
        e ← selectEdgeAtRandom(effList, rng)
        iNode ← getOrigin(e)
        jNode ← getEnd(e)
        iR ← getRoute(iNode, sol)
        jR ← getRoute(jNode, sol)
        if all CWS route − merging conditions are satisfied then        ▷ see constraints
            sol ← mergeRoutesUsingEdge(e, iR, jR, sol)        ▷ see CWS heuristic
        end if
        deleteEdgeFromList(e, effList)
    end while
    return sol
end procedure
```

At the end of each first-level iteration, the resulting solution goes through a Monte-Carlo simulation procedure which provides estimates of its associated expected variable costs (see Pseudo-code 10). These estimates are obtained by iteratively sampling the random variables characterizing customer demands in each route. This way, whenever a random route failure occurs, we account for its associated costs, i.e., the ones due to performing an extra trip to the depot to reload the vehicle before resuming the delivery of goods among the remaining customers. After several iterations, estimates for the expected variable costs are obtained by averaging route-failure costs. Finally, the total

Algorithm 9 Randomized edge-selection procedure.

```
procedure SELECTEDGEATRANDOM(list, rng)
    beta ← generateRandomNumber(rng, a, b)        ▷ e.g.: a=0.05 and b=0.25
    randomValue ← generateRandomNumber(rng, 0, 1)
    pos ← floor(log(randomValue/log(1 − beta)))        ▷ random from a geometric dist.
    pos ← pos mod listSize        ▷ random position from the list.
    return getEdgeAtPosition(pos)
end procedure
```

expected cost of the resulting solution is used to determine if it should be stored or not in a sorted array of best solutions found so far.

It is important to notice here that the SR-GCWS-CS algorithm is a probabilistic process. This means that it will provide slightly different results each time it is run with a different seed of the random number generator. Therefore, as it will be described in the experimental section, it is possible to concurrently launch different instances of the algorithm, each one using a different initial seed, by using a multi-thread approach in a multi-core CPU to speed up further the local search process.

10.4 Computational Results

In the CVRP literature, there exists a classical set of very well-known benchmarks commonly used to test their algorithm. However, as noticed by Bianchi et al. (2006), there are no commonly used benchmarks in the VRPSD literature and, therefore, each paper presents a different set of randomly generated benchmarks which, in our opinion, reveals the immatureness of the VRPSD knowledge area when compared with the CVRP area. Unfortunately, most authors only provide details regarding the parameters used to randomly generate their instances, but they do not provide the exact coordinates of the nodes, which are necessary to calculate the travelling costs. Similarly, the exact parameters of the distributions that model customer demands are not usually provided. This situation makes it extremely difficult to compare the performance of different approaches. Consequently, we decided to employ a natural generalization of several classical CVRP instances by using random demands instead of constant ones. This approach has at least three advantages: (1) all data details, including nodes coordinates

Algorithm 10 MCS procedure to obtain variable costs and reliability estimates.

```
procedure calcExpectedCosts(vrpNodes, vrpConstraints, vrpSol)
    solExpectedCosts ← 0                          ▷ 1. Reset solution expected costs.
    for each route r in vrpSol do                 ▷ 2. For each route r in the given solution...
        rExpectedCosts ← 0
        for iter = 1 to (iter − nIter) do
            rCosts ← getCosts(r)                  ▷ fixed costs for r
            rAccumDemand ← 0
            for each customer c in r do
                newDemand ← generateRandomDemand(c)
                rAccumDemand ← newDemand
                if rAccumDemand > vehicleCapacity then
                    rCosts ← rCosts + roundTripCosts(c, depot)
                    rAccumDemand ← newDemand
                end if
            end for
            rExpectedCosts ← rExpectedCosts + rCosts
        end for
        rExpectedCosts ← rExpectedCosts/nIter
        solExpectedCosts ← solExpectedCosts + rExpectedCosts
    end for
    return solExpectedCosts                       ▷ 3. Return expected costs for the given solution
end procedure
```

and random demands, are given, so that other authors can use the same data sets for verifying and benchmarking purposes; (2) we are using a well-known set of instances which includes a diversity of clustered and disperse problems of different sizes; and (3) our CVRPSD results for each instance can be compared with the corresponding CVRP best known solution (BKS) and, ideally, our results should converge to the CVRP BKS as variances in customers' demands tend to zero. In order to test our methodology, we generalized a set of 55 classical CVRP instances, which details (in terms of nodes coordinates, deterministic demands, and vehicle capacity), can be found at `http://www.branchandcut.org`. So, for each instance, while we decided to keep all node coordinates and vehicle capacities, we changed d_i, the deterministic demands of client i ($\forall i \in \Omega *$) to stochastic demands D_i with $E[D_i] = d_i$. In other words, we considered the demand of each client as a random variable following a well-known probability distribution with a given mean and a given variance (e.g., $Var[D_i] = 0.25 \cdot d_i$ as 'low' variance). To illustrate this, we selected an Exponential distribution for modelling demands, although any other distribution with a known mean could have been used instead. In fact, in a real-world problem historical data would be used to model each client's demands by a different probability distribution, which can be naturally supported by our simulation-based approach.

A multi-thread version of the previously described algorithm was implemented in Java and executed under Windows 7 Professional on a 2 GHz E5504 IntelR Quad-Core XeonR CPU with 4 GB. Each thread was an instance of our algorithm using a different stream of the LFSR113 pseudo-random number generator (L'ecuyer and Buist, 2005). In our tests, four threads —one per core— were running on the aforementioned CPU. All threads shared a common memory (a cache of routes). Notice that no fine-tuning process was carried out, since one of our goals was to prove that our algorithm is robust and can provide efficient solutions to any VRPSD problem without any initial adjustments. As discussed in the previous section, another major goal of our approach was to allow 'realtime' decision-making by parallelizing the execution of multiple scenarios, the ones defined by considering different safety-stocks levels. In order to test this goal, a limit of 10 seconds was set as the maximum computing time allowed per scenario. Each scenario was defined by using a different level of safety-stocks during the design stage. Thus, safety-stocks were ranging from 0% to 20% of the vehicle real capacity.

Tables 10.1 and 10.2 show the complete results obtained for all 55 classical instances we generalized and tested. First column in each table contains the name of each instance, which includes the number of nodes and also the number of routes of the 'standard' solution, e.g., B-n78-k10 is an instance of class B with 78 nodes and able to be solved with a 10-route solution. Columns 2 to 4 are related to solutions obtained by our algorithm when a 100% of the vehicle maximum capacity is considered during the design stage. Notice that this strategy always provides pseudo-optimal solutions in terms of fixed costs (column 2), since they can be directly compared with the CVRP BKS. However, since no safety stock is used, there is a chance that these solutions can suffer from route failures. In turn, route failures might imply high expected variable costs (estimated in column 3 by Monte-Carlo simulation), thus increasing the total expected costs, which is estimated in column 4. Here is where using safety stocks can be of value: by not necessarily using all vehicle maximum capacity during the design stage, some route failures can be avoided. Hopefully, this might lead to new solutions with slightly higher fixed costs but also with lower expected variable costs. At the end, these alternative solutions might present lower total expected costs, which are the ones to be minimized. On one hand, columns 5 to 9 show the results obtained with the algorithm presented in Juan et al. (2011d), which applies simulation once the local search process that solves the deterministic VRP has finished. On the other hand, columns 10 to 14 show the results obtained with the algorithm proposed in this study, in which simulation is integrated, and applied several times, throughout the local search process. Notice that fixed costs in columns 7 and 12 are always higher or equal to those in column 2. However, total expected costs in columns 9 and 13 are always lower or equal to those in column 4. Notice also that sometimes the best-found strategy (for this set of benchmarks) is to use a 100% of the vehicle maximum capacity (i.e., no safety stocks at all) when designing the routes (columns 5 and 10).

However, in other occasions it pays off to design the routes using a safety stock, e.g., for the P-n101-k4, the best-found solution has been obtained by using only 85% of the vehicle maximum capacity, even when that solution contains five routes, one more than strictly necessary as denoted by its "k4" term. Finally, the respective gaps between total expected costs are shown in columns 9 and 14. Notice that, even when in most cases this gap is small, sometimes it can be above 3%, which means that using the correct safety-stocks level can sensibly reduce the total expected costs when

	Without using safety stocks (used capacity = 100%)			Using safety stocks (used capacity = p% of vehicle capacity)										
				Simulation applied after the local search					Simulation integrated inside the local search					
Instance name	Fixed costs	Expected variable costs	Expected total costs (1)	Used capacity	M	Fixed costs	Expected total costs (2)	Gap (1-2)	Used capacity	M	Fixed costs	Expected total costs (3)	Gap (1-3)	Gap (2-3)
A-n32-k5	787.1	231.7	1,018.80	95%	5	797.5	1,006.70	-1.2%	100%	5	787.1	993.2	-2.5%	-1.3%
A-n33-k5	662.1	179.9	842	97%	5	676.1	830.1	-1.4%	100%	5	674.1	815.4	-3.2%	-1.8%
A-n33-k6	742.7	176.5	919.1	100%	6	742.7	919.1	0.00%	100%	6	744.6	912.6	-0.7%	-0.7%
A-n37-k5	672.5	127.1	799.6	100%	5	672.5	799.6	0.00%	99%	5	672.5	795	-0.6%	-0.6%
A-n38-k5	733.9	181.4	915.3	96%	6	753.2	886.1	-3.2%	97%	6	753.2	885.1	-3.3%	-0.1%
A-n39-k6	833.2	216.9	1,050.10	98%	6	835.3	1,029.90	-1.9%	100%	6	842.9	1,010.60	-3.8%	-1.9%
A-n45-k6	952.2	257.7	1,209.90	96%	7	972.7	1,190.60	-1.6%	100%	7	972.2	1,184.30	-2.1%	-0.5%
A-n45-k7	1,147.40	396	1,543.40	96%	7	1,155.80	1,527.80	-1.0%	98%	7	1,155.20	1,502.00	-2.7%	-1.7%
A-n55-k9	1,074.50	343.7	1,418.20	100%	9	1,074.50	1,418.20	0.00%	100%	9	1,086.40	1,408.40	-0.7%	-0.7%
A-n60-k9	1,360.60	472.7	1,833.30	99%	9	1,361.30	1,820.80	-0.7%	100%	9	1,360.60	1,795.70	-2.1%	-1.4%
A-n61-k9	1,040.30	339.2	1,379.50	98%	10	1,058.40	1,340.80	-2.8%	97%	10	1,065.10	1,330.60	-3.5%	-0.8%
A-n63-k9	1,633.70	573	2,206.70	100%	9	1,633.70	2,206.70	0.00%	100%	10	1,649.60	2,203.70	-0.1%	-0.1%
A-n65-k9	1,184.70	394.7	1,579.40	94%	10	1,241.70	1,564.10	-1.0%	100%	9	1,185.90	1,555.30	-1.5%	-0.6%
A-n80-k10	1,776.20	609.2	2,385.30	90%	11	1,867.40	2,328.40	-2.4%	90%	11	1,867.40	2,328.40	-2.4%	0.00%
B-n31-k5	676.1	189.8	865.9	95%	5	681	862	-0.5%	100%	5	684.7	855.7	-1.2%	-0.7%
B-n35-k5	958.9	296.3	1,255.20	100%	5	958.9	1,255.20	0.00%	100%	5	958.9	1,255.50	0.00%	0.00%
B-n39-k5	553.2	165.7	718.9	99%	5	557.4	701.8	-2.4%	96%	5	563.2	695.9	-3.2%	-0.9%
B-n41-k6	835.8	279.7	1,115.50	91%	7	907.5	1,108.00	-0.7%	97%	7	899.7	1,103.20	-1.1%	-0.4%
B-n45-k5	754	174.8	928.8	96%	6	763.8	908	-2.2%	91%	6	764.3	904.6	-2.6%	-0.4%
B-n50-k7	744.2	227.5	971.8	92%	7	756.8	949.3	-2.3%	91%	7	757.8	945.8	-2.7%	-0.4%
B-n52-k7	754.5	224	978.5	99%	7	756.7	953.1	-2.6%	99%	7	770	944.4	-3.5%	-0.9%
B-n56-k7	716.4	215.1	931.5	86%	8	765.7	928.1	-0.4%	98%	7	728.6	920	-1.2%	-0.9%
B-n57-k9	1,602.30	623.8	2,226.00	98%	9	1,619.60	2,199.70	-1.2%	98%	9	1,619.60	2,199.70	-1.2%	0.00%
B-n64-k9	868.3	312.9	1,181.20	94%	10	903.3	1,180.00	-0.1%	100%	10	916.3	1,179.60	-0.1%	0.00%
B-n67-k10	1,042.30	402.6	1,444.80	93%	11	1,105.30	1,409.10	-2.5%	97%	11	1,099.10	1,404.50	-2.8%	-0.3%
B-n68-k9	1,300.20	487.6	1,787.80	96%	9	1,308.20	1,770.60	-1.0%	97%	9	1,313.60	1,754.70	-1.9%	-0.9%
B-n78-k10	1,250.60	432.9	1,683.50	95%	11	1,305.70	1,668.00	-0.9%	100%	10	1,254.80	1,659.60	-1.4%	-0.5%
Average	987.3	316	1,303.30	96%	8	1,008.60	1,287.50	-1.3%	98%	8	1,005.50	1,279.40	-1.9%	-0.7%

Table 10.1: Results for instances A and B using an exponential with $E[D_i] = d_i$ (using 10 seconds per scenario).

compared with the best-found solution without using safety stocks. Finally, notice also that column 15 shows the gap between the two discussed stochastic approaches, i.e., applying simulation only once after the local search and integrating simulation throughout the local search. According to this column it seems clear that the integrated approach presented in this study provides always equal or slightly superior results to the one which applies simulation only after the deterministic VRP has been solved.

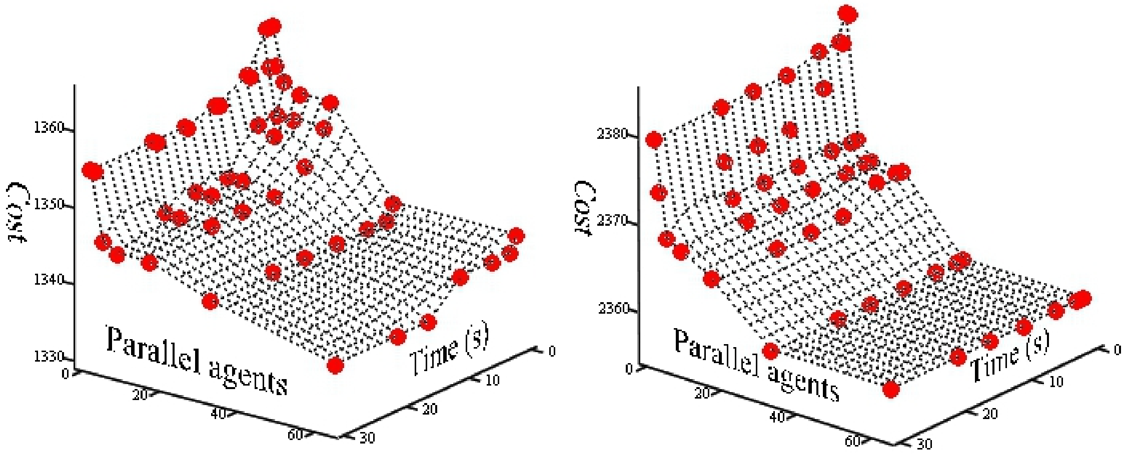
In order to analyze how expected total costs provided by the algorithm depend upon the variables 'computing time' and 'parallelization level', a final experiment was designed. The experiment consisted in choosing some of the largest VRPSD instances considered in this study, and then running them in a cluster of computers under different scenarios. The instances selected were the following ones: E-n76-k14, A-n80-k10, P-n101-k4, M-n101-k10, M-n121-k7, and F-n135-k7. Each of the aforementioned scenarios

	Without using safety stocks (used capacity = 100%)			Using safety stocks (used capacity = p% of vehicle capacity)										
				Simulation applied after the local search					Simulation integrated inside the local search					
Instance name	Fixed costs	Expected variable costs	Expected total costs (1)	Used capacity	M	Fixed costs	Expected total costs (2)	Gap (1-2)	Used capacity	M	Fixed costs	Expected total costs (3)	Gap (1-3)	Gap (2-3)
E-n22-k4	375.3	107.7	482.9	97%	4	383.9	479.1	-0.8%	100%	4	383.5	468.3	-3.0%	-2.2%
E-n30-k3	505	96.3	601.3	91%	4	505	599.7	-0.3%	98%	4	506.1	589.8	-1.9%	-1.6%
E-n33-k4	837.7	276	1,113.70	94%	4	851.2	1,106.50	-0.6%	94%	4	851.2	1,085.40	-2.5%	-1.9%
E-n51-k5	524.6	86.8	611.5	100%	5	524.6	611.5	0.00%	100%	5	524.6	611.5	0.00%	0.00%
E-n76-k7	692.7	113.2	805.9	92%	7	702.3	792.5	-1.7%	100%	7	698.9	790	-2.0%	-0.3%
E-n76-k10	841.3	229.8	1,071.10	93%	11	870.1	1,044.50	-2.5%	93%	11	870.1	1,044.50	-2.5%	0.00%
E-n76-k14	982.7	307.6	1,290.30	100%	15	982.7	1,290.30	0.00%	100%	15	982.7	1,290.30	0.00%	0.00%
F-n45-k4	727.7	121.5	849.2	99%	5	730	826.8	-2.6%	99%	5	730	824.9	-2.9%	-0.2%
F-n72-k4	244.1	41.6	285.7	100%	4	244.1	285.7	0.00%	99%	4	248.7	283.8	-0.7%	-0.7%
F-n135-k7	1,183.80	325.4	1,509.30	100%	7	1,183.80	1,509.30	0.00%	100%	7	1,183.80	1,509.30	0.00%	0.00%
M-n101-k10	819.6	221.7	1,041.30	100%	10	819.6	1,041.30	0.00%	100%	10	825.6	1,034.70	-0.6%	-0.6%
M-n121-k7	1,047.60	296	1,343.70	100%	7	1,047.60	1,343.70	0.00%	100%	7	1,047.60	1,343.70	0.00%	0.00%
P-n19-k2	212.7	40.9	253.6	94%	3	220.6	253.1	-0.2%	99%	3	220.6	252.2	-0.5%	-0.4%
P-n20-k2	217.4	42.7	260.1	98%	2	218.3	258.9	-0.5%	100%	2	218.3	257.5	-1.0%	-0.5%
P-n22-k2	217.9	41.9	259.8	100%	2	217.9	259.8	0.00%	100%	2	217.9	255	-1.8%	-1.8%
P-n22-k8	588.8	216	804.8	99%	9	589.4	801.7	-0.4%	100%	9	588.8	787.3	-2.2%	-1.8%
P-n40-k5	461.7	76.7	538.4	100%	5	461.7	538.4	0.00%	99%	5	466.3	537.9	-0.1%	-0.1%
P-n50-k8	632.7	180.4	813.1	91%	9	652.4	794.5	-2.3%	100%	9	641.9	790.5	-2.8%	-0.5%
P-n50-k10	700.7	212.5	913.1	99%	10	700.7	908.9	-0.5%	100%	10	704.9	903.7	-1.0%	-0.6%
P-n51-k10	741.5	219.3	960.8	100%	10	741.5	960.8	0.00%	100%	10	741.5	958.2	-0.3%	-0.3%
P-n55-k7	574.5	120.6	695.1	91%	7	588.1	674.3	-3.0%	91%	7	588.1	672.2	-3.3%	-0.3%
P-n55-k15	952.1	355.5	1,307.60	98%	16	965.5	1,301.50	-0.5%	99%	16	955	1,267.50	-3.1%	-2.6%
P-n60-k10	756.3	215.2	971.6	96%	10	763.2	955.2	-1.7%	100%	10	756.9	947	-2.5%	-0.9%
P-n65-k10	807	206.7	1,013.60	97%	10	812.9	1,011.70	-0.2%	97%	11	817.7	1,005.10	-0.8%	-0.7%
P-n70-k10	839.1	208.6	1,047.70	97%	11	858.8	1,043.80	-0.4%	99%	11	848.2	1,043.30	-0.4%	0.00%
P-n76-k4	615.5	60.9	676.4	87%	5	628.9	662.3	-2.1%	87%	5	628.9	662.3	-2.1%	0.00%
P-n76-k5	642.7	88.1	730.8	87%	6	664.6	716.6	-1.9%	97%	6	663	716.3	-2.0%	0.00%
P-n101-k4	718.8	46.1	764.9	85%	5	729.8	754	-1.4%	85%	5	729.8	754	-1.4%	0.00%
Average	659.3	162.7	822	96%	7	666.4	815.2	-0.8%	98%	7	665.7	810.2	-1.5%	-0.6%

Table 10.2: Results for instances E, F, M and P using an exponential with $E[D_i] = d_i$ (using 10 seconds per scenario).

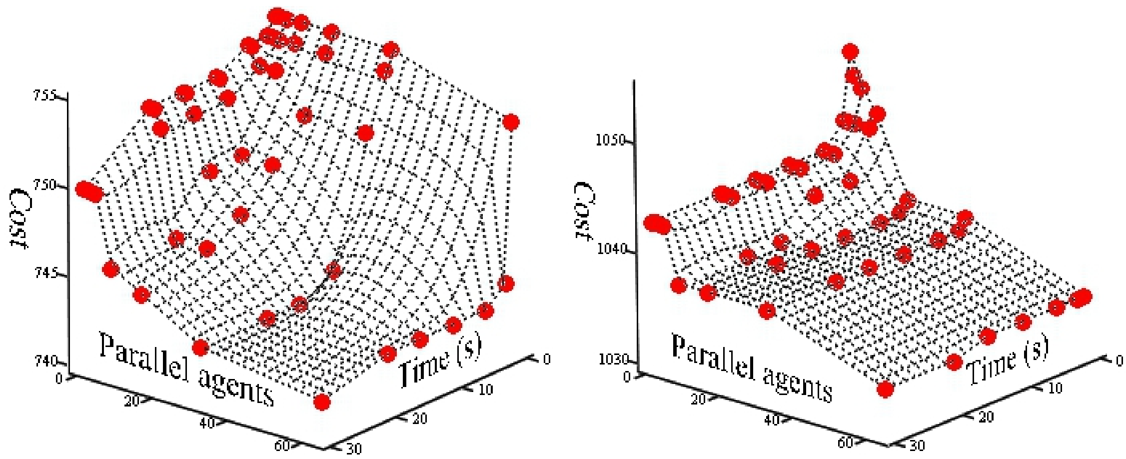
is defined by a particular combination of the parameters 'computing time' (1, 2, 5, 10, 15, 20, and 30 seconds), and 'number of agents' (algorithm instances) running in parallel (1, 2, 4, 8, 16, 32, and 64 parallel instances). We carried out this experiment using a heterogeneous cluster environment composed of 16 multi-core nodes (with Intel Xeon Series 54xx/55xx and AMD Quad Opteron 23xx CPUs). Notice that the described execution testbed is just one possible example of a PDCS, but a similar testbed could be proposed by using alternative computing nodes, i.e., several cores in a multi-core CPU environment, networked PCs with mono-core CPUs, or even nodes in a volunteer computing or cloud computing environment.

Fig. 10.2, 10.3 and 10.4 show the resulting 3D scatterplots representing expected total costs versus computing time (in seconds) and number of agents running in parallel. As it can be noticed in every scatterplot, costs always diminish very fast as the number of parallel agents increases, even when considering just one or two seconds of computing time. Of course, noticeable cost reductions can also be attained by considering longer execution times of 30 seconds, even for just one or two agents. However, the interesting thing to notice here is that, being based on a probabilistic algorithm, our approach largely benefits from using a PDCS environment. In fact, according to the obtained results, it seems that near-optimal solutions can be obtained for small- and medium-sized VRPSD instances in just a few seconds when several instances of the algorithm are concurrently executed in a relatively affordable PDCS. This might be due to the diversification obtained by using different initial randomized solutions, which increases the chances of starting the search process in the vicinity of a pseudo-optimal solution even without a long 'warm-up' period.



E-n76-k14 A-n80-k10

Figure 10.2: Scatterplots for expected total costs vs. time and number of agents of E-n76-k14 and A-n80-k10 instances.



P-n101-k4 M-n101-k10

Figure 10.3: Scatterplots for expected total costs vs. time and number of agents of P-n101-k4 and M-n101-k10 instances.

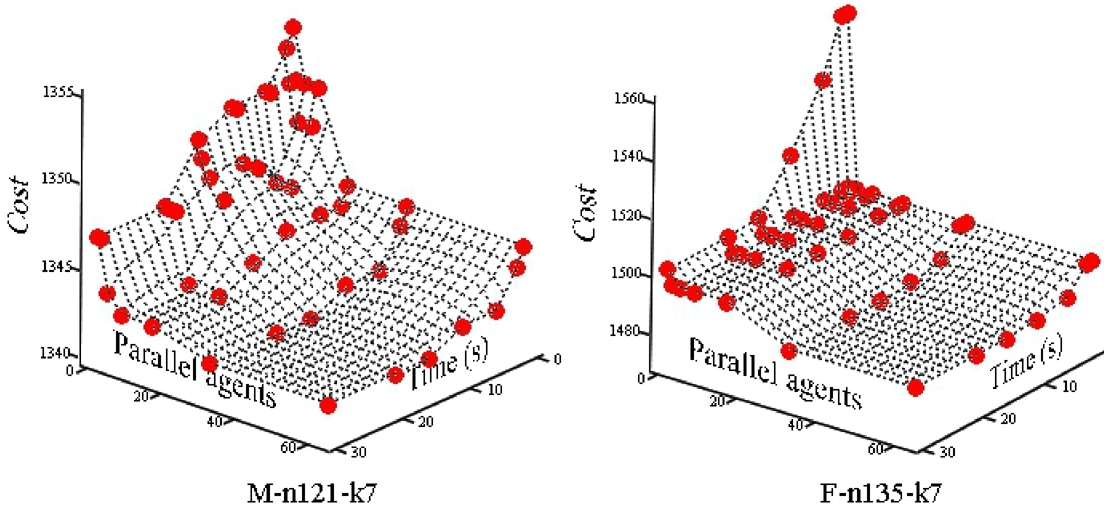


Figure 10.4: Scatterplots for expected total costs vs. time and number of agents of M-n121-k7 and F-n135-k7.

10.5 Chapter Conclusions

So far, we have have appreciated the application of the proposed Simheuristic methodology for solving the VRPSD. A multiple-scenario approach is designed. This approach combines parallel computation, Monte-Carlo simulation, a well-tested randomized heuristic, and the use of safety stocks to offer a flexible as well as efficient algorithm. Parallelization techniques are used at two different levels: first, a parallel-execution environment is designed to deal with the multiple-scenario analysis; secondly, several concurrent threads sharing a common memory or, alternatively, several concurrent processes are considered during the algorithm execution for each scenario. Among the special characteristics of this approach it is important to highlight that it has provided 'real-time' competitive solutions to most small- and medium-tested instances, that it does not need any complex fine-tuning process, and that it does assume any particular probability distribution for modelling customers' demands. Furthermore, the proposed Simheuristic approach in this chapter points out the potentials of parallelization techniques. The use of DPCS can quite improve the computational execution of optimization procedures. In the next chapter, we will apply the Simheuristic idea to a different stochastic problem called Inventory Routing Problem with Stochastic Demands.

11

Inventory Routing Problem with Stochastic Demands

> Parts of this chapter have been taken from the co-authored publication: Cáceres-Cruz, Juan, Grasman, Bektas, and Faulin (2012b) in Proceedings of WSC.

Transportation and inventory decisions are traditionally made sequentially, which lacks collaboration between the participants and does not allow the close cooperation that optimizes supply chain performance. Today, one of the most important concepts in supply chain management is replacing sequential decision making with global decision making, where all parties in the supply chain determine the best policy for the entire system; whereas in sequentially optimized supply chains, each party determines its own course of action independent of the benefit to the entire membership.

Inventory and transportation systems are good examples of sequential decision making. However, driven by business practices such as vendor managed inventory (VMI), integrated inventory and transportation systems have received much recent attention (Kaipia et al., 2002). VMI is a supply chain centralized control initiative where the supplier is authorized to manage inventories of the retailers and to make decisions such as when and how much inventory to ship to the retailer. VMI is seen as an effective means of managing inventory through the strategic use of Internet technologies, leverages advanced technology and trading-partner relationships to enable the flow of information and inventory throughout the entire supply chain. Despite the potential benefits, and probably due to its complexity, only a relatively small number of works have analytically approached the issue of integrating inventory and transportation (vehicle routing) decisions. This issue is known in the literature as the Inventory Routing Problem or IRP (Campbell et al., 1998). Therefore, model formulations with exact or approximate solution procedures are still needed to assist with the widespread adoption of VMI and use of synchronized inventory and transportation systems.

In this chapter, we consider a single-period IRP consisting of multiple retailer centers with stochastic demands and a single distribution depot. Since final demands at the retailer centers are assumed to be random variables, potential stock-outs are considered in our model. In a decentralized version of this problem, each retailer would utilize the inventory policy that either minimizes its own expected costs or achieves a prescribed service level. Supply requests would then be transferred to the distribution depot, so that it can design the corresponding delivery routes. On the contrary, in the centralized version that we are addressing, no assumption is made about the inventory policy at an individual retailer. The distribution depot will analyze the inventory position of the retailers and make joint inventory and routing decisions aiming at minimizing the total cost to the system.

Another aspect to notice is that most of the existing literature has considered the IRP as a long-term, multi-period problem (Campbell et al., 2001). This is especially the case when the final demands at the retailer centers are assumed to be deterministic. However, we consider that it is worthy to also study the single-period problem, particularly in those scenarios characterized by: (a) information and communication tools, which are able to efficiently monitor and report retailers' stocks levels at the end of each period, and (b) random demands with a high variability, which make it difficult to forecast future inventory levels. Under those conditions —which seem quite common among real-life IRP applications—, long-run planning could be a much more inefficient policy than just solving the problem with updated data at the end of each period. In Fig. 11.1 we present an example to illustrate a simple solution of the studied context.

Each RC owns an inventory, which is managed by the central depot. For each RC, the inventory level at the end of a period depends on the initial stock level and also on the end-clients' demands during that period. These end-clients' demands are stochastic in nature. In our approach, we will assume that, for each RC, it has been

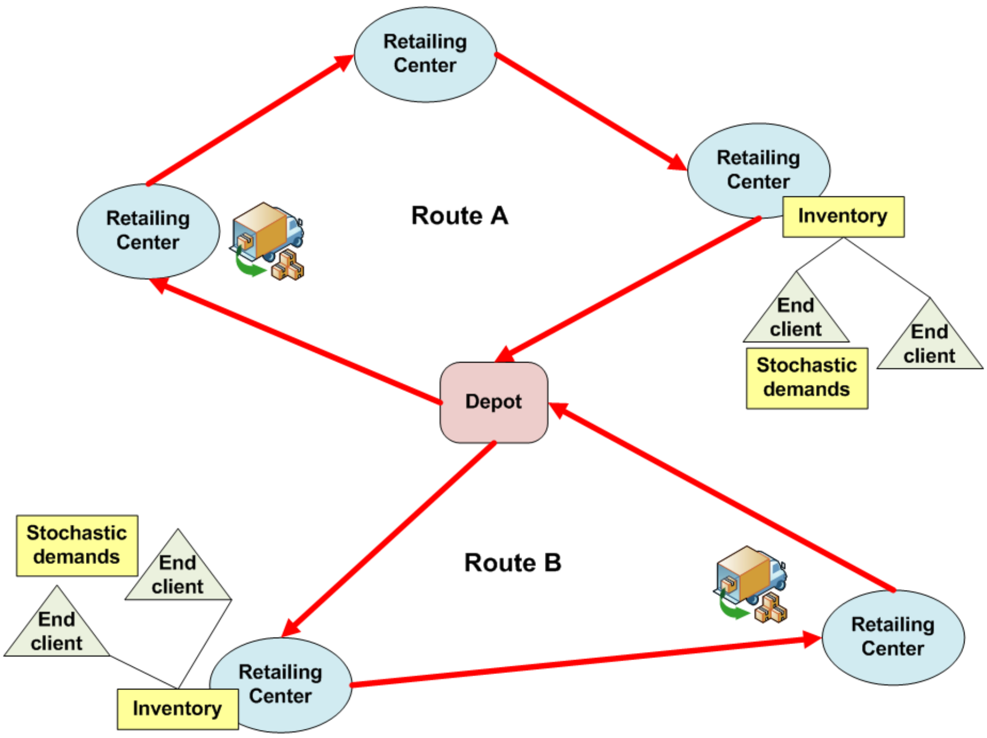


Figure 11.1: Scheme of the IRP with stochastic demands.

possible to use historical data to model end-clients' demands through a theoretical or empirical probability distribution. Notice that no particular assumption is made on the type of distribution used to model these demands —as long as it has an associated mean value. Therefore, at the end of each period there may be costs associated with inventory holding and inventory stock-outs. These costs should be incorporated to the decision-making process and integrated with (added to) the distribution or routing costs, which are usually based on travelling distances and/or times. At the end of each period, inventory levels are registered by the RC and updated in the central depot, so that a new routing strategy is defined for the new period taking into account the new data. Our goal is to minimize total expected costs —distribution costs plus expected inventory-related costs— in each single-period scenario. As explained earlier, we focus on the single-period scenario since we assume stochastic demands with high variability, which makes it difficult to forecast the evolution of stocks with time.

Accordingly, in this chapter we describe a biased-randomized hybrid algorithm for solving the single-period IRP with stochastic demands and stock-outs. Our approach is hybrid in the sense that it combines Monte-Carlo simulation with a multi-start biased randomization of a classical routing heuristic (Simheuristic). First, the algorithm considers a discrete set of different potential inventory policies for each retailer center, and estimates through simulation the inventory costs associated with each retailer-policy combination. Then, the algorithm considers the routing plan with the highest possible cost, i.e., the one in which all retailers are filled up to their maximum inventory levels. Using this 'worst-case scenario' as a reference, a fast heuristic is employed to estimate the marginal savings in routing costs associated with each retailer-policy combination —i.e., varying just the policy at that retailer while keeping the remaining policies unaltered. That way, for each retailer it is possible to rank its potential inventory policies according to their (estimated) total costs, i.e., both inventory and routing costs. Once the inventory policies have been ranked by total cost in a list, a multi-start process is used to iteratively construct a set of promising solutions for the IRP. At each iteration of this multi-start process, a new set of policies is selected by performing a biased randomization on the ranked list. This biased randomization is driven by the use of some non-symmetric (biased) probability distribution. Thus, assuming the less costly policies are located at the top of the ranked list, the higher the position of a policy the higher its probability of being selected during the random-selection process.

The approach presented in this chapter has similarities with some previous work, especially with those considering stochastic demands, stock-outs, and rollout periods. Probably the most closely related works are that of Bertazzi et al. (2013); Hvattum et al. (2009). However, our approach shows some significant differences with them: (a) we consider several replenishment policies —personalized for each retail center— instead of just an order-to-level policy; (b) we use a hybrid algorithm combining simulation with a metaheuristic, which allows us to obtain 'good' solutions to large-size instances in a reasonable time; (c) we promote the use of biased randomization of heuristics (e.g., using biased probability distributions) as a more efficient method than using non-biased randomization (e.g., using the uniform distribution); and (d) we propose a completely described set of instances (not a randomly generated one), which can be employed by other researchers as well-defined benchmarks.

11.1 Definition

The single-period Stochastic IRP that is considered in this chapter can be described as follows: consider a Capacitated Vehicle Routing Problem (CVRP) with n intermediate customers or retail centers (RC), plus the depot (node 0). Using a more formal description, the IRP is defined on a complete and undirected graph $G = (\Omega, A)$, where $\Omega* = \{1, 2, \ldots, n\}$ is the set of RC nodes, $\Omega = \Omega * \cup \{0\}$ (depot), and $A = \{(i, j) \in \Omega x \Omega / i < j\}$ is the set of arcs connecting those nodes. The parameters of the problem can be summarized as follows:

- For each $RC_i \in \Omega*$, both the current inventory level $L_i = 0$, as well as the maximum allowable inventory level $\hat{L}_i > 0$, are known.
- For each $RC_i \in \Omega*$, i has to serve several customers for whom the aggregated demand is a random variable, $D_i = 0$ following a known probability distribution with $E[D_i] = d_i > 0$.
- A fleet of k homogeneous vehicles is used to perform the routing, each vehicle with a maximum capacity $Q > 0$ (also known). It will be assumed that $Q = D_i \forall i \in \Omega$.
- For each $(i, j) \in A$, the cost of travelling from node i to node j, $c_{ij} > 0$, is known. Moreover, a symmetric cost matrix is assumed, i.e., $c_{ij} = c_{ji}$.

A decentralized-policy would dictate that inventory decisions would be made first by each RC in order to minimize their own expected inventory costs. Once desired inventory levels are fixed, routing decisions would be made by the depot in order to serve the individual orders. In other words, each RC would choose to have an inventory level, L_i^* (decision variable), which minimizes its expected inventory costs —i.e., without considering routing costs. Therefore, the quantity of product it would request to the depot is given by $q_i = L_i^* - L_i$ if $L_i^* > L_i$ and 0 otherwise. Once these q_i values are for each RC, they would be considered as parameter inputs for the associated vehicle routing problem.

The centralized approach assumed in this study aims at jointly determining the amount of inventory each $RC_i \in \Omega*$ is supplied with, which we denote by and the routing plan in order to minimize the total expected costs of combining routing and inventory decisions, i.e., the sum of the routing costs and the expected inventory costs. The latter is calculated by summing $f(q_i) \forall i \in \Omega*$, each of which is the inventory cost for the RC_i written as a function of the decision variables q_i. Without loss of generality, this study assumes the following structure for the inventory cost function:

$$f(q_i) = \begin{cases} \lambda \cdot s_i & if\ s_i \geq 0 \\ 2 \cdot c_{0i} & if\ s_i < 0 \end{cases} \tag{11.1}$$

where $\lambda \geq 0$ represents the cost of holding a unit of product in stock at the end of the period (assumed to be known) and s_i represents the total surplus at the end of the period, i.e.: $s_i = L_i + q_i - D_i, \forall i \in \Omega*$. Notice that if a stock-out occurs in RC_i, then the inventory cost is modelled as the cost of sending a new vehicle from the depot to i (round-trip).

It is possible to formulate this problem as a mixed-integer stochastic program, a class of formulations known to be difficult to solve. Even when demands q_i are known, the model becomes that of a VRP, a NP-Hard problem Augerat et al. (1995), and the fact that these are decision variables in our model add another layer of complexity. It is for this reason that we develop a hybrid solution algorithm combining simulation and heuristics. This algorithm is described in the following section.

11.2 Literature review

In the past years, several approaches have been proposed for different variants of the Inventory Routing Problem. Probably the main factors to take into account when classifying the different works are: (a) whether they consider deterministic or stochastic demands; (b) whether they consider single- or multiple-periods (including an infinite horizon); (c) whether they allow inventory shortages or not; (d) whether they consider single- or multiple-products; (e) whether they use the same refill policy for all nodes or personalized replenishment policies for each node; and (f) whether they use exact or approximate methods to solve the problem. We have divided our literature review according to the first —and probably most relevant— criteria, i.e., whether the demands are considered to have a deterministic or a stochastic nature.

11.2.1 IRP with deterministic demands

Regarding the IRP with deterministic demands, Chien et al. (1989) discuss the importance of considering both the inventory allocation and the vehicle routing when making logistical decisions. These authors formulate the integrated problem as a mixed integer program and develop an approach based on Lagrangian relaxation to obtain upper bounds for several randomly-generated instances with up to 30 nodes. In their paper, Anily and Federgruen (1990) address an integrated inventory-routing problem with infinite horizon and deterministic demands, and describe a class of heuristics which allow them to obtain 'good' solutions for problems varying from 100 to 10,000 nodes. They assume that the depot cannot keep inventories itself. In Anily and Federgruen (1993), the authors extend their previous analysis to the case where central inventories might be kept at the depot. Kohli and Park (1994) examine joint order policies for multiple products over a planning horizon where each product has an independent price. Bramel and Simchi-Levi (1995) develop a location-based heuristic for solving general routing problems, including the IRP with deterministic demands. In their approach, however, inventory shortages are not allowed. One of the first works using the term "Inventory Routing Problem" is that of Campbell et al. (1998). In this chapter, the authors give a general description of the problem, analyze the one- and two-customer cases, and propose two solution approaches based on integer and dynamic programming, respectively. Chan et al. (1998) study the IRP with infinite horizon and deterministic demands. They propose an algorithm based on fixed partition policies, which create a partition of the set retailers into a number of regions such that each region is served separately and independently from all other regions. In Campbell et al. (2001), the authors undertake some practical issues related to problems with deterministic demands and long-term planning horizons. They propose an integer programming model and discuss the quality of the solution it generates over commonly used rules-of-thumb. Bertazzi et al. (2002) study a multi-period IRP with deterministic demands. They propose a two-stage constructive heuristic algorithm, which is tested against a set of randomly generated instances with up to 50 customers and 30 periods. One of the first Tabu Search (TS) algorithms for solving the IRP is that from Cousineau-Ouimet (2002). However, the detailed pseudo-code of the TS algorithm is not provided. The author also generated some well-documented instances for the multi-period and deterministic problem "...to overcome the lack of appropriate case studies in the literature". In Campbell and Savelsbergh (2004c), a two-phase approach is presented. The approach first generates a delivery schedule using integer programming and then generates the routing plan using heuristics. Interestingly, the authors support the use of GRASP-like randomization (non-biased in nature) "as a powerful tool to improve the performance of insertion heuristics". Campbell and Savelsbergh (2004a) also studied the impact of incorporating complex constraints on insertion heuristics for routing and scheduling problems. In fact, Campbell and Savelsbergh (2004d) considered optimizing the maximum volume deliverable to the customers in a given instant and under time windows constraints. The authors have considered four main serving policies: as early as possible, as late as possible, as greedy as possible, and always serve to the customer with the maximum usage rate.

11.2.2 IRP with stochastic demands

One of the first works on the IRP with stochastic demands is due to Federgruen and Zipkin (1984a). They address the single-period combined problem of "allocating a scarce resource available at some central depot among several locations, each experiencing a random demand pattern, and planning deliveries using a fleet of vehicles". Transportation, holding, and shortage costs are considered, and the authors define this problem as "an extension of the standard vehicle routing problem". These authors provide several examples of potential applications, including: (a) deliveries of fuel oil to automotive service stations; (b) periodic replenishment of gas tanks at customer locations; or (c) coordinated allocation and supply to various locations of a perishable product such as blood. They propose a mathematical model and design a modified interchange heuristic as well as an exact algorithm to solve some randomly generated instances with up to 75 nodes. Replenishment policies with a reorder point s and an order level S are called (s, S) policies. In Federgruen and Zipkin (1984b), the authors prove the fast convergence to 'good' solutions, under standard assumptions of (s, S) policies, using a policy-iteration algorithm. They have tested their approach with 4 sets of 192 instances varying the mean and variance of demand distribution —one for each set. Good results are obtained in a reasonable computing time. Trudeau and Dror (1992) address a multi-period version of the IRP with stochastic demands in which stock-outs are also allowed. In their computational experiments, they consider 12 weekly periods and 2,077 customers, making use of simulation to randomly generate their demands. It is assumed that each customer's random demand is not reported until the customer is visited by a vehicle, which leads to the possibility of route failures. Barnes-Schuster and Bassok (1997) address an infinite-horizon scenario in which demands are stochastic. From the computational results, they conclude that direct shipping (one truck delivering one retailer and then returning to the depot) can be a simple yet effective strategy whenever the capacity of the truck is close to the customer's average demand. In Bard et al. (1998), the authors study the IRP with satellite facilities (depots geographically scattered throughout the service area, which permit drivers to refill their vehicles with commodity during a shift). Interestingly, the authors use a randomized version of the classical Clarke and Wright (1964) Savings (CWS) heuristic to solve routing instances with up to 500 nodes in about two hours. They show that this randomized heuristic outperforms other algorithms, including a GRASP. While the randomness process they propose is based on a uniform (non-biased) distribution, in this chapter, we make use of a biased distribution to randomize the aforementioned routing heuristic. Reiman et al. (1999) consider an IRP with stochastic demands and one vehicle covering a region composed of several customers, i.e., they assume that a previous process has been carried out in which customers have been assigned to different regions, each region covered by a single vehicle. Then they compare three strategies, the first one based on the direct shipping; the second one based on the pre-specified tour (i.e., a Traveling Salesman

Problem) and the third one based on dynamic choice between the TSP and direct shipping options, concluding that the direct shipping strategy is the preferred strategy in most situations. In Kleywegt et al. (2002) the authors address the IRP with stochastic demands over an infinite horizon. They formulate the aforementioned problem as a discrete time Markov decision process and propose several approximation methods to solve it. Berman and Larson (2001) focus on the problem associated with the distribution of industrial gases to replenish customer tanks. They model customers' demands as random processes and propose four dynamic-programming algorithms for solving the associated problem. In Kleywegt et al. (2004), the authors extend their previous work with a Markov process considering multiple resources for improving safety and reduce contamination between products on transportation and storage phases. Then, approximate methods are proposed to solve instances with up to 20 nodes in reasonable computing times. This is one of the few articles that provides complete information (e.g., nodes coordinates) on the tested instances. The work of Jaillet et al. (2002) has noticeable similarities with our IRP model: these authors support the convenience of considering a short-time rolling horizon framework when dealing with the stochastic IRP. In their words:

> "For typical IRPs, a customer's consumption rate is difficult to predict with certainty and can only be represented at best by a random variable with known probability distribution... Planning the entire annual distribution scheme in advance would, however, be unreliable and prone to many needed adjustments".

Also, they assume the possibility of stock-outs; in particular "...as soon as the [customer's] tank becomes empty, an immediate, and thus costly, special delivery is made". However, they also assume that the central depot cannot monitor the inventory levels and that these are only revealed after each customer is visited by a truck. Thus, delivery strategies based upon end-of-period inventory levels are not considered in their work, while they are a fundamental part of our approach. They also propose several replenishment strategies for a finite horizon. Gaur and Fisher (2004) describe an application to a real-life IRP with stochastic demands and heterogeneous fleet. In order to simplify their problem, they split the set of customers into several disjoint regions. Originally, only partitions with at most two customers are allowed, but once the partitions are created, they try to improve them by using a heuristic. However, they do not really consider inventory costs, since in their case the relevant costs are the routing ones. Adelman (2004) uses linear programming and Markov processes for solving the IRP with stochastic demands and infinite horizon. However, the largest instances included in the experimental section, which have been randomly generated, contain no more than 40 customers. In Yu et al. (2006), the authors analyze the multi-period stochastic IRP with split delivery. Their approach aims at transforming the stochastic model into a deterministic one and then using Lagrangian relaxation to decompose this latter model into inventory and rouging subproblems. Using this approach, they solve some randomly generated instances with up to 100 nodes in reasonable computing times. Another real-life IRP application is presented by Custódio and Oliveira (2006). After discussing the different strategies commonly employed in the literature (fixed partition, direct shipping, and ratios of integers), they propose a classical heuristic for solving the case study. In Jarugumilli et al. (2006), the authors make use of a modified version of the A* algorithm to solve the stochastic IRP with a single vehicle. Hvattum et al. (2009) address the stochastic IRP with infinite horizon as a Markov process. They formulate a scenario tree in order to examine a finite horizon as a good approximation to the infinite horizon model. Again, since solving the Markov process is unpractical for all but the smallest instances, they employ a GRASP heuristic, which can be considered as a non-biased randomized algorithm. Finally, Bertazzi et al. (2013) undertake a stochastic IRP with stock-out and finite horizon. They assume an order-to-level policy, i.e., "the quantity sent to each retailer is such that its inventory level reaches the maximum level whenever the retailer is served". They present a dynamic programming model and propose a hybrid roll-out algorithm. In order to validate their approach, they use a randomly generated set of instances with up to 50 nodes and 6 periods.

11.3 Proposed Approach

Our approach focuses on solving the single-period IRP with stochastic demands and possible stock-outs. As explained before, we consider a rolling horizon with just one period ahead, and we assume that update information on current inventory levels is obtained at the end of each period. Notice that these end-of-period inventory levels might be very difficult to forecast, especially when the probability distributions modelling the random demands are characterized by high variances. Thus, we believe that under these realistic conditions it might make sense to follow a plan-one-step-ahead policy, i.e., plan just one period ahead and then update the current inventory levels before planning again.

In this context, we propose a hybrid approach that combines Monte-Carlo Simulation (MCS) with an efficient CVRP heuristic (see Fig. 11.2). MCS can be defined as a set of techniques that make use of random numbers and statistical distributions to solve certain stochastic and deterministic problems (Law, 2007). When properly combined with heuristic techniques, MCS has proven to be extremely useful for solving stochastic vehicle routing problems (Juan et al., 2011d,e). Our approach is also based on the SR-GCWS randomized algorithm proposed by Juan et al. (2010) for solving the CVRP. This algorithm makes use of a pseudo-geometric distribution to induce a biased randomization process into the CWS heuristic (Clarke and Wright, 1964). The algorithm also employs memory-based local search and a divide-and-conquer strategy.

Our approach starts by considering a discrete number of p centralized refill policies for each intermediate customer (retail center). For instance, given a retail center we could consider the following *natural* policies: (a) no refill; (b) refill up to one quarter of its capacity ($\frac{1}{4}$-refill policy); (c) refill up to half of its capacity ($\frac{1}{2}$-refill policy); (d) refill up to three quarters of its capacity ($\frac{3}{4}$-refill policy); (e) full refill; and (f) refill up to the optimal inventory level —this policy is related to a decentralized strategy in which each retailer center decides its refill level without considering routing costs. Note that our methodology could consider more intermediate policies if necessary, which makes it quite flexible. Of course, considering more intermediate policies —i.e., by using a higher granularity level— could lead to slightly better solutions, but will also increase the computational effort. For each retailer-policy combination, MCS is used to obtain estimates of the inventory costs associated with it —including both surplus and shortage costs. Then, the 'worst possible' routing scenario is considered (i.e., serving all retailers up to their capacity limits), and the associated routing costs are quickly estimated using the savings heuristic proposed by Clarke and Wright (1964). Next, marginal savings in routing costs associated with each retailer-policy combination are estimated using the same routing heuristic. In other words, for each retailer-policy combination, we compute the new routing costs due to using the new retailer-policy combination while

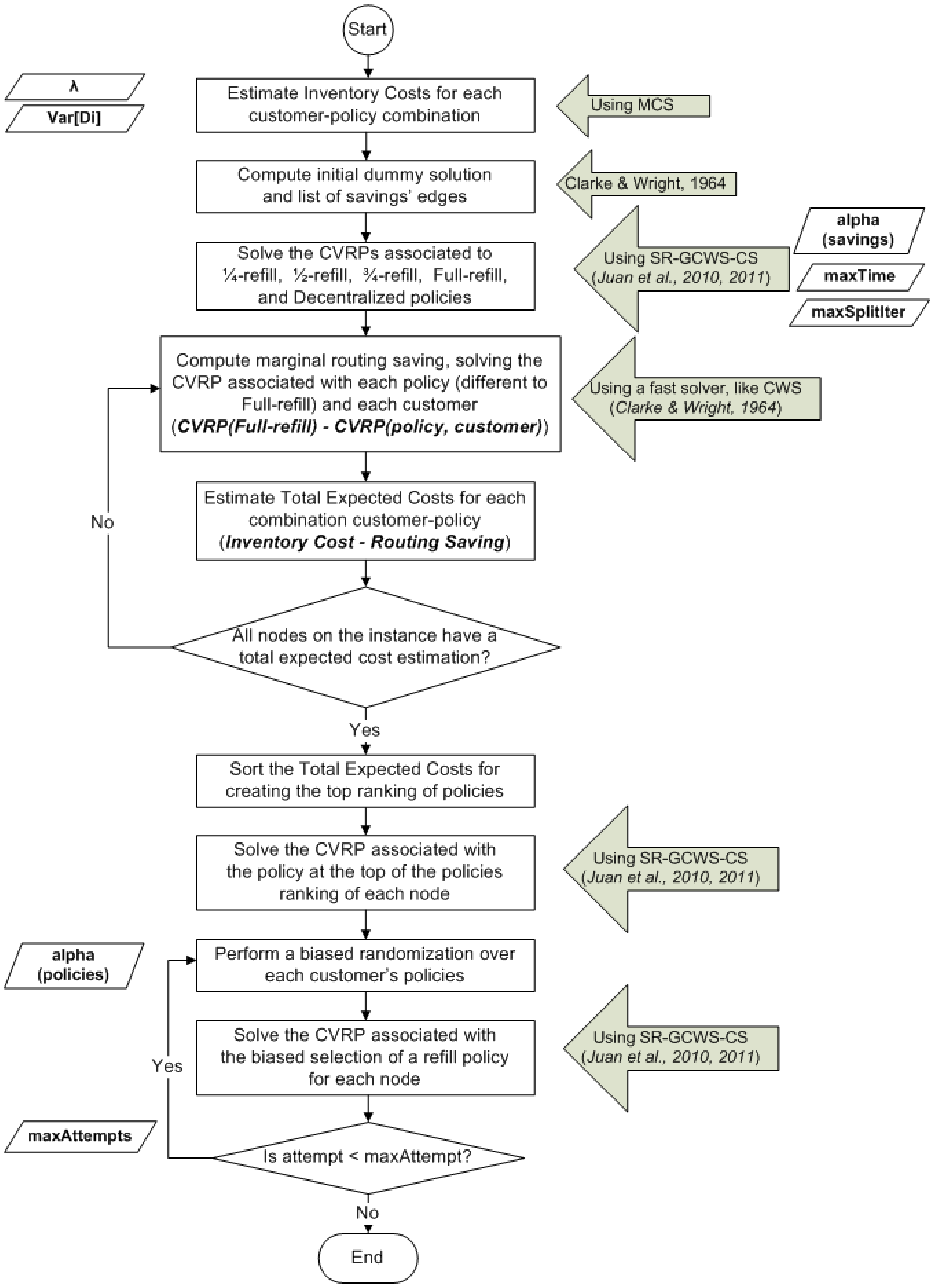


Figure 11.2: Flowchart scheme of our approach.

i_{th}-retailer Policy	Inventory Costs (estimated, MCS)	Marginal Savings (estimated, heuristic)	Total Costs (estimated)	Final Rank
1	550	0	550	7
2	600	-150	450	3
...	...	...	...	...
pth	450	-25	425	1

Table 11.1: Example of policy ranking for the i_{th} retailer.

keeping the 'fill-up-to-the-top' policy for the remaining retailers. The difference between the 'worse-case' routing cost and the new routing cost determines the marginal savings in routing associated with the current retailer-policy combination. For each of these combinations, its marginal savings can be then combined with its inventory costs in order to obtain an estimate of its total costs. These total costs are used to sort the service policies related to each retailer (see Table 11.1).

Logically, for each RC those service policies with a lower total costs will be preferred to those others with a higher total costs. Once the inventory policies have been ranked for each customer, a multi-start process is used to iteratively construct a set of promising solutions for the IRP. At each iteration of this multi-start process, a new set of policies is selected by performing a biased randomization on the list of service policies —i.e., as closer to the top of the list a policy is, the more probable is that it can be selected.

As any other approximate approach, this method does not guarantee to obtain an optimal solution, but it can produce feasible and 'good' solutions to the stochastic IRP in a reasonable amount of time, which is not a trivial task if one considers the complexity of the problem.

In order to facilitate the actual implementation of the proposed methodology, we provide a pseudo-code version of our algorithm (Pseudo-code 11 and 12), which is described next as a five-step procedure. First (lines 3 to 7 of Pseudo-code 11), for each retailer, the expected inventory costs associated with each eligible policy is estimated throughout Monte-Carlo simulation —using the corresponding probability distributions which model end-customers' demands. Here, both potential surplus and shortages (stock-outs) are considered for each of the refill policies described in the previous section. In the second step of the procedure (lines 9 to 14), we consider the worst-case scenario from a distribution point of view, i.e., all retailers receive a full refill. In this scenario, the CWS heuristic is used to obtain a 'good' solution for the associated CVRP. This solution will provide an estimate of the total distribution costs under the full-refill policy. In the third step (lines 16 to 27), we estimate for each retailer the routing “marginal savings”, i.e., the reduction in distribution costs associated with each non-full-refill policy. In order to do this, the CWS heuristic is used to solve a large set of CVRPs. A fast heuristic is employed here since this step implies solving one CVRP for each customer-policy combination, i.e., for each retailer and for each non-full-refill policy. Once these marginal costs have been estimated, for each RC, an approximated value for the total costs associated with each eligible policy can be obtained by simply adding up estimated routing and inventory costs (Table 11.1). Thus, for each retailer, the associated eligible policies can be sorted from lower to higher total costs, consequently defining a priority policy rank. In the fourth step (lines 3 to 11 of Pseudo-code 12), the ‘top’ policy for each RC (i.e., the one showing the lowest total cost) is selected, and a pseudo-optimal solution is obtained for the corresponding CVRP by using the SR-GCWS-CS algorithm (Juan et al., 2011e). Finally, in the fifth step (lines 13 to 24), a multi-start process is started. At each iteration of this multi-start process, a new policy is randomly selected for each retailer and, in a similar manner as in the previous step, a new pseudo-optimal solution is obtained for the corresponding CVRP. Notice, however, that the random selection process is not uniform but biased, i.e., a biased distribution like the Geometric one is used instead of a symmetric distribution. By using a biased distribution, we aim at giving greater probability of being selected to those policies that are located at the top positions of each retailer’s rank of policies. Thus, we are continuously generating different promising scenarios by randomly selecting refill policies which are likely to provide a good balance between routing and inventory costs. Of course, during the multi-start process, the best solution found so far is recorded. Using a multi-start approach makes it difficult for the algorithm to get trapped in a local minimum.

Algorithm 11 Pseudo-code for the proposed hybrid algorithm.

procedure SOLVE-IRPSD($nodes, vehicles$)

  for *each retailing center i* **do** ▷ 1. Use MCS to estimate expected inventory costs for each center-policy.
    for *each policy k* **do** ▷ Simulation
      $EIC_{i,k} \leftarrow Expected\ Inventory\ Cost\ for\ center\ i\ using\ policy\ k$
    end for
  end for

  for *each retailing center i* **do** ▷ 2. Solve VRP with full-refill policy for all centers (routing worst-case)
    $q_{i,full} \leftarrow \hat{L}_i - L_i$ ▷ Demand requested by center i with a full-refill policy.
    $q \leftarrow add\ q_{i,full}\ to\ the\ list\ of\ demands\ q$
  end for
  $\pi \leftarrow CWS - heuristic(q)$ ▷ Solve VRP using the fast CWS heuristic
  $DC_{full} \leftarrow Distribution\ (routing)\ costs\ associated\ with\ \pi$

  for *each retailing center i* **do** ▷ 3. Sort the list of policies associated with each center by using both inventory costs and marginal savings in routing.
    for *each policy* $k (k \neq full\ refill)$ **do**
      $q_{i,k} \leftarrow Max\{0, L_{i,k} - L_i\}$ ▷ Compute new demand for center i with policy k
      $q' \leftarrow update\ q\ by\ temporarily\ substituting\ q_{i,full}\ for\ q_{i,k}$
      $\pi' \leftarrow CWS - heuristic(q')$
      $DC_{i,k} \leftarrow Distribution\ (routing)\ costs\ from\ \pi'$
      $MS_{i,k} \leftarrow DC_{full} - DC_{i,k}$ ▷ Compute Marginal Savings in routing
      $CC_{i,k} \leftarrow EIC_{i,k} - MS_{i,k}$ ▷ Combined (inventory + routing) costs
      $P_i \leftarrow update\ and\ sort\ list\ of\ policies\ of\ center\ i\ according\ to\ CC_{i,k}$
      $q \leftarrow recover\ original\ list\ by\ substituting\ q_{i,k}\ for\ q_{i,full}$
    end for
  end for
end procedure

Algorithm 12 Pseudo-code for the proposed hybrid algorithm (*continuation*).

procedure SOLVE-IRPSD($nodes, vehicles$)

$q \leftarrow reset\ list\ of\ demands\ q$ ▷ 4. Use a metaheuristic to solve VRP considering the 'top' policy at each center.
for *each retailing center i* **do**
$EIC_{i,k} \leftarrow select\ thepolicy\ at\ the\ top\ of\ P_i$
$EIC_{i,k} \leftarrow Max\{0, L_{i,k'} - L_i\}$ ▷ Compute demand for center i with policy k'
$EIC_{i,k} \leftarrow add\ q_{i,k'}\ to\ the\ list\ of\ demandsq$
end for
$\pi_{top} \leftarrow SRGCWSCS_metaheuristic(q)$ ▷ Solve VRP using the SR-GCWS-CS
$DC_{top} \leftarrow Distribution\ (routing)\ Costs\ associated\ with\ \pi_{top}$
$TEC_{top} \leftarrow DC_{top} + Sum\{EIC_{i,top}\}$ ▷ Total Expected Costs using top policies

while *ending condition not meet* **do** ▷ 5. Use a metaheuristic to solve VRP considering randomly selected policies at each center.
$q \leftarrow reset\ list\ of\ demandsq$
for *each retailing center i* **do**
$k' \leftarrow randomly\ select\ a\ policy\ from\ P_i\ using\ a\ geometric\ distribution$
$q_{i,k'} \leftarrow Max\{0, L_{i,k'} - L_i\}$
$q \leftarrow add\ q_{i,k'}\ to\ the\ list\ of\ demands\ q$
end for
$\pi_{rand} \leftarrow SRGCWSCS_metaheuristic(q)$
$DC_{rand} \leftarrow Distribution\ (routing)\ Costs\ associated\ with\ \pi_{rand}$
$TEC_{rand} \leftarrow DC_{rand} + Sum\{EIC_{i,rand}\}$ ▷ Total Expected Costs using random policies
$\pi_{best} \leftarrow update\ best\ solution\ considering \pi_{top}\ and\ \pi_{rand}$
end while
return π_{best}
end procedure

11.4 Experimental Design

In the CVRP literature, there exists a classical set of well-known benchmarks commonly used to test new CVRP algorithms. As noticed in the previous "Literature Review" section, this might be also true for some deterministic versions of the IRP. However, this is not the case for the single-period IRP with stochastic demands and stock-outs, the one discussed in this study. In fact, for this and other IRP versions it is a usual practice that each paper presents a different set of randomly generated benchmarks —i.e., without providing the exact values obtained during the randomization process, which makes it impossible to reproduce the exact results and which makes it difficult to perform direct and fair comparisons among different approaches. Moreover, in some other cases the proposed set of instances is no longer available, as it happens with the expired link presented in Campbell et al. (1998).

For those reasons, and with the goal of providing complete information about the set of benchmarks employed so that other researchers can use them, we have developed our own set of data by generalizing the well-known datasets A and B from the CVRP literature (Augerat et al., 1995). These datasets consist of 27 small- and medium-size test instances. The full test set used in this study is available at `http://www.branchandcut.org/VRP`. A natural generalization has been carried out by using random, instead of deterministic, demands. So, for each instance, while we decided to keep all node coordinates and vehicle capacities, we changed d_i, the deterministic demand of retailer i ($\forall i = 1, 2, \ldots, n$) to the probabilistic demand D_i with $E[D_i] = d_i$. Since we use MCS, these random demands can follow any probability distribution as far as it has a mean. In particular, for the numerical experiments of this study we will assume that D_i will follow a LogNormal distribution with $E[D_i] = d_i$. The LogNormal distribution has been chosen because it should be preferred over the Normal distribution when modelling positive demands. Notice, however, that our simulation-based approach also supports any other distribution, such as the Weibull or the Gamma ones. We will also consider three different levels of variance, i.e., $Var[D_i] = 0.25 \cdot d_i$ ('low' variance scenario), $Var[D_i] = 0.50 \cdot d_i$ ('medium' variance scenario), and $Var[D_i] = 0.75 \cdot d_i$ ('high' variance scenario).

Regarding the inventory part of the problem, the following assumptions are made in order to define a numerical example to experiment with —notice that these are not model assumptions, but just assumptions to define a numerical example so that we can illustrate the potential of our approach with some benchmarks:

- For each RC i, its maximum inventory capacity is defined as $\hat{L}_i = 2 \cdot d_i$. As it usually happens in real-life, retailers with higher expected demands will also have higher inventory capacities.

- In correspondence with the distribution policies considered, the quantity that can be delivered to each RC, q_i, can only take a discrete number of values, i.e., according to the policies described above, q_i can only take values in the set $0, 0.5 \cdot d_i, d_i, 1.5 \cdot d_i, 2 \cdot d_i$.

- Trying to imitate a realistic scenario, in which it is likely that different retailers will present different starting stock levels, initial inventory level at retailer i, L_i, is assigned according to the following expression:

$$L_i = \begin{cases} 0 & if\ i\ is\ odd\ and\ multiple\ of\ 3\ (e.g.,\ L_3,\ L_9,\ L_{15},\ldots) \\ d_i/2 & if\ i\ is\ odd\ and\ not\ multiple\ of\ 3\ (e.g.,\ L_1,\ L_5,\ L_7,\ldots) \\ d_i & if\ i\ is\ even\ and\ multiple\ of\ 4\ (e.g.,\ L_4,\ L_8,\ L_{12},\ldots) \\ (3 \cdot d_i)/2 & if\ i\ is\ even\ and\ not\ multiple\ of\ 4\ (e.g.,\ L_2,\ L_6,\ L_{10},\ldots) \end{cases} \tag{11.2}$$

Notice that instead of using these pre-defined values as initial inventory levels, we could simply have selected these initial values at random. But then, it could not be possible to reproduce the computational experiment with exactly the same data.

Finally, regarding the inventory costs, these must be of a similar order of magnitude as the routing costs. Thus, it might make sense not to serve some retailers under certain conditions, e.g., high inventory levels and low stock-out costs. In order to attain this goal, we have used in our experiments the previously explained expression for defining the inventory costs associated with each retailer. On this we propose to use a parameter (λ) in order to relate inventory and routing costs. Notice that λ represents the cost per unit of stock at the end of the period. Also, notice that whenever a stock-out occurs, a 'penalty' cost incurred since a new vehicle must be sent from the depot to the retailer to solve the shortage issue. In our numerical experiments, we have used values of $\lambda \in \{0.01, 0.05\}$. These values were chosen inside a reasonable range such that it might not be worth serving some of the retailers. In particular, we might decide not to serve retailers with a low probability of suffering a stock-out —e.g., those in which the expected demand is much lower than the current inventory levels—, and also those with low penalty costs in case they suffer from a stock-out —e.g., those closer to the depot. Of course, different values of λ could be considered instead as far as the resulting inventory costs have the same order of magnitude as the routing costs —otherwise the problem would basically reduce to a routing one or to an inventory one. Notice that we are defining an inventory cost function for computational purposes only. Of course other functions or specific values for the parameters are possible

11.5 Computational Results

Our algorithm was implemented as a Java application and used to run the 27 instances described above on an Intel Xeon E5603 at 1.60 Ghz and 8 GB of RAM. For each instance, a single run with a total maximum time of 15 minutes was employed for each value of $\lambda \in \{0.01, 0.05\}$ and three different levels of variance (low, medium, and high). The limitation in computing time is due to the fact that we wanted to obtain results in a 'reasonable' amount of time. The selection of the λ values is due to the fact that we wanted inventory costs to be of a similar order of magnitude as the routing costs. The selection of biased-randomized policies is done with an alpha random value between 0.8 and 0.99. Tables 11.2, 11.3, 11.4, 11.5 show the summary results obtained in our experiments for the following policies:

a) *No-refill* policy, i.e., no retailer was served in advance, and only those retailers suffering a stock-out were served with a direct vehicle from the depot —notice that this is an extreme and very expensive policy.

b) $\frac{1}{4}$-refill policy, i.e., all retailers are served up to one quarter of its maximum capacity —those retailers which already have that inventory level are not served.

c) $\frac{1}{2}$-refill policy, i.e., all retailers are served up to half of its maximum capacity —those retailers which already have that inventory level are not served.

d) $\frac{3}{4}$-refill policy, i.e., all retailers are served up to three quarters of its maximum capacity —those retailers which already have that inventory level are not served.

e) *Full-refill* policy, i.e., all retailers are served up to its maximum capacity.

f) *Decentralized* or k*th* policy, i.e., all retailers are served up to the level which optimizes its inventory costs —without considering routing costs.

g) *Top* individual inventory-routing policy, i.e., each retailers is served according to the 'best' or top policy in its sorted priority list of policies —notice that this top policy could imply that the retailers does not need to be served if its inventory level is appropriate enough.

h) *Biased-randomized* policy, i.e., each retailers is served according to a policy which has been biased-randomly selected from its sorted policies list.

Additionally, Tables 11.3 and 11.5 also show the percentage gaps between the solution obtained using each policy and our best solution —i.e., the one obtained with the biased-randomized process— considering each variance level. Positive gaps imply that the total cost obtained with the biased-randomized process is lower (and therefore better) than the total cost obtained with the alternative method. In the Tables 11.6 to 11.17, the detailed results for each instance are depicted.

Variance Level	NO-REFILL POLICY (1) Inventory	Total	$\frac{1}{4}$-REFILL POLICY (2) M	Routing	Inventory	Total	$\frac{1}{2}$-REFILL POLICY (3) M	Routing	Inventory	Total	$\frac{3}{4}$-REFILL POLICY (4) M	Routing	Inventory	Total	FULL-REFILL POLICY (5) M	Routing	Inventory	Total
Low	2,400.26	2,400.26	4	670.41	758.64	1,429.05	6.4	895.61	25.57	921.18	8.1	1,068.38	6.56	1,074.94	8.7	1,124.77	6.58	1,131.34
Medium	2,462.30	2,462.30	4	670.41	780.47	1,450.88	6.4	895.61	71.58	967.19	8.1	1,068.38	9.19	1,077.57	8.7	1,124.77	8.28	1,133.04
High	2,508.62	2,508.62	4	670.41	804.25	1,474.66	6.4	895.61	109.65	1,005.26	8.1	1,068.38	13.02	1,081.40	8.7	1,124.77	10.95	1,135.72

Table 11.2: Average results of 27 instances for no-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and full-refill policies ($\lambda = 0.01$, max. computation time = 15 minutes).

Variance Level	DECENTRALIZED POLICY (6) % served	M	Routing	Inventory	Total	TOP POLICY (7) % served	M	Routing	Inventory	Total	BIASED RANDOMIZED POLICY (8) % served	M	Routing	Inventory	Total	Gaps (1-8)	(2-8)	(3-8)	(4-8)	(5-8)	(6-8)	(7-8)
Low	78%	5.6	808.05	4.09	812.14	64%	4.1	623.46	43.86	667.32	63%	4	610.99	45.66	656.65	265.53%	117.63%	40.28%	63.70%	72.29%	23.64%	1.62%
Medium	80%	5.7	818.43	6.16	824.59	70%	4.6	670.39	93.19	763.58	70%	4.5	666.76	92.58	759.34	224.27%	91.07%	27.37%	41.91%	49.21%	8.59%	0.56%
High	81%	5.7	825.7	9.34	835.05	73%	4.7	696.47	111.28	807.75	73%	4.7	695.04	107.02	802.06	212.77%	83.86%	25.34%	34.83%	41.60%	4.11%	0.71%

Table 11.3: Average results of 27 instances for decentralized, top, and biased-randomized policies plus summary gaps ($\lambda = 0.01$, max. computation time = 15 minutes).

Variance Level	NO-REFILL POLICY (1) Inventory	Total	$\frac{1}{4}$-REFILL POLICY (2) M	Routing	Inventory	Total	$\frac{1}{2}$-REFILL POLICY (3) M	Routing	Inventory	Total	$\frac{3}{4}$-REFILL POLICY (4) M	Routing	Inventory	Total	FULL-REFILL POLICY (5) M	Routing	Inventory	Total
Low	2,403.97	2,403.97	4	670.41	855.49	1,525.89	6.4	895.61	47.24	942.85	8.1	1,068.38	32.54	1,100.91	8.7	1,124.77	34.74	1,159.50
Medium	2,466.04	2,466.04	4	670.41	944.78	1,615.18	6.4	895.61	97.02	992.63	8.1	1,068.38	35.45	1,103.82	8.7	1,124.77	36.75	1,161.52
High	2,512.40	2,512.40	4	670.41	1,012.35	1,682.75	6.4	895.61	137.2	1,032.81	8.1	1,068.38	39.6	1,107.98	8.7	1,124.77	39.88	1,164.64

Table 11.4: Average results of 27 instances for no-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and full-refill policies ($\lambda = 0.05$, max. computation time = 15 minutes).

Variance Level	DECENTRALIZED POLICY (6) % served	M	Routing	Inventory	Total	TOP POLICY (7) % served	M	Routing	Inventory	Total	BIASED RANDOMIZED POLICY (8) % served	M	Routing	Inventory	Total	Gaps (1-8)	(2-8)	(3-8)	(4-8)	(5-8)	(6-8)	(7-8)
Low	77%	5.4	770.95	18.38	789.32	63%	4.1	617.49	53.71	671.20	63%	4	612.19	53.99	666.18	260.86%	129.05%	41.53%	65.26%	74.05%	18.49%	0.75%
Medium	78%	5.4	779.32	25.79	805.11	69%	4.5	668.42	105.37	773.79	70%	4.5	665.95	103.56	769.51	220.47%	109.90%	29.00%	43.45%	50.94%	4.63%	0.56%
High	80%	5.4	788.12	24.10	812.21	72%	4.7	692.78	125.91	818.69	72%	4.7	690.59	121.98	812.57	209.19%	107.09%	27.10%	36.35%	43.33%	-0.04%	0.75%

Table 11.5: Average results of 27 instances for decentralized, top, and biased-randomized policies plus summary gaps ($\lambda = 0.05$, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2062.51	2062.51	3	563.64	713.87	1277.51	4	745.95	20.71	766.67	6	899.89	4.34	904.23	6	954.98	4.72	959.7
A-n33-k5	33	1477.83	1477.83	3	505.86	490.97	996.83	4	622.53	14.08	636.61	6	720.32	4.21	724.53	6	754.64	3.17	757.8
A-n33-k6	33	1319.9	1319.9	3	546.36	134.5	680.86	5	702.23	19.54	721.77	7	797.2	5.37	802.57	7	812.31	5.72	818.03
A-n37-k5	37	1251.49	1251.49	3	563.32	120.03	683.36	4	623.81	3.07	626.88	5	694.71	10.83	705.53	5	706.1	5.66	711.76
A-n38-k5	38	1736.26	1736.26	3	550.46	489.16	1039.62	5	700.63	10.37	710.99	6	771.38	5.24	776.62	6	783.47	5.53	789.01
A-n39-k6	39	1687.92	1687.92	3	630.6	481.28	1111.88	5	751.95	11.81	763.76	7	860.12	5.38	865.51	7	879.78	4.93	884.7
A-n45-k6	45	2125.85	2125.85	3	669.56	614.73	1284.29	6	903.12	26.49	929.61	7	1018.08	5.88	1023.96	8	1072.01	4.74	1,076.75
A-n45-k7	45	3007.86	3007.86	4	769.5	844.11	1613.61	6	1055.75	26.3	1082.04	8	1305.94	3.71	1309.65	8	1344.19	5.1	1,349.29
A-n55-k9	55	2382.59	2382.59	5	761.39	894.34	1655.73	8	993.22	25.97	1019.18	11	1232.09	6.2	1238.29	11	1302.96	8.66	1,311.63
A-n60-k9	60	3074.06	3074.06	5	885.27	898.66	1783.93	8	1252.69	39.12	1291.81	10	1509.37	6.06	1515.42	11	1649.71	8.9	1,658.61
A-n61-k9	61	2143.27	2143.27	5	734.03	654.49	1388.53	8	928.54	15.72	944.26	10	1079.62	7.62	1087.24	11	1106.76	7.28	1,114.04
A-n63-k9	63	4045.73	4045.73	5	1037.03	1550.49	2587.52	8	1478.71	43.53	1522.24	10	1750.36	5	1755.36	11	1883.2	5.01	1,888.21
A-n65-k9	65	2338.99	2338.99	5	817.05	408.78	1225.83	8	1075.08	34.26	1109.34	10	1266.74	9.37	1276.11	11	1311.92	9.83	1,321.75
A-n80-k10	80	5098.77	5098.77	5	1152.13	1658.02	2810.15	9	1603.11	62.67	1665.78	11	1927.03	16.48	1943.51	12	2071.27	6.74	2,078.01
B-n31-k5	31	1748.79	1748.79	3	454.28	616.76	1071.04	4	595.89	20.9	616.79	5	701.21	4.07	705.28	6	785.8	4.27	790.07
B-n35-k5	35	2256.98	2256.98	3	587.88	856.73	1444.62	4	805.3	20.17	825.47	5	1037.95	5.22	1043.17	6	1092.62	5.49	1,098.11
B-n39-k5	39	1589.71	1589.71	3	373.15	616.71	989.85	4	515.9	23.38	539.28	5	588.24	5.88	594.12	6	602.77	6.05	608.82
B-n41-k6	41	1879.36	1879.36	3	520.57	573.63	1094.2	6	794.61	18.04	812.65	7	861.97	5.44	867.42	8	922.87	5.87	928.74
B-n45-k5	45	1688.66	1688.66	3	542.83	667.36	1210.2	5	686.97	24.24	711.21	6	810.96	5.98	816.95	7	861.82	2.52	864.34
B-n50-k7	50	1945.08	1945.08	4	496.33	652.22	1148.54	6	636.84	17.6	654.44	7	824.86	3.98	828.84	8	856.4	6.34	862.74
B-n52-k7	52	2293.3	2293.3	4	545.07	950.17	1495.24	6	661.59	22.3	683.89	8	873.56	4.26	877.82	8	900.86	7.06	907.92
B-n56-k7	56	1890.19	1890.19	4	522.63	594.23	1116.85	6	609.26	20.22	629.48	8	661.98	6.56	668.54	8	679.13	7.09	686.22
B-n57-k9	57	4374.52	4374.52	5	969.42	1541.68	2511.09	8	1480.43	47.66	1528.09	10	1760.09	5.94	1766.04	10	1851.48	8.36	1,859.84
B-n64-k9	64	2260.68	2260.68	5	577.2	832.53	1409.74	9	800.84	29.08	829.92	11	1001.4	9.09	1010.49	12	1037.36	9.54	1,046.90
B-n67-k10	67	2541.93	2541.93	5	714.24	840.68	1554.91	9	957.25	25.85	983.1	12	1165.23	9.24	1174.47	12	1259.61	10.09	1,269.70
B-n68-k9	68	3456.38	3456.38	5	785.45	1048.38	1833.83	8	1120.88	41.03	1161.91	10	1384.54	6.77	1391.3	11	1475.73	9.59	1,485.31
B-n78-k10	78	3128.27	3128.27	5	825.79	738.8	1564.59	9	1078.42	26.22	1104.64	11	1341.39	8.99	1350.38	12	1408.94	9.36	1,418.30

Table 11.6: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.01$, low variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	84%	4	727.58	2.71	730.28	59%	3	600.08	24.12	624.21	59%	3	600.08	24.12	624.21	230.42%	104.66%	22.82%	44.86%	53.75%	16.99%	0.00%
A-n33-k5	79%	4	623.91	2.4	626.31	58%	3	425.1	39.46	464.56	58%	2	379.74	42.46	422.2	250.03%	136.10%	50.78%	71.61%	79.49%	48.34%	10.03%
A-n33-k6	79%	4	621.84	3.03	624.87	58%	3	504.88	27.2	532.08	58%	3	491.88	29.14	521.03	153.33%	30.68%	38.53%	54.04%	57.00%	19.93%	2.12%
A-n37-k5	84%	4	554.74	3.7	558.44	62%	2	444.54	23.97	468.5	62%	2	437.12	23.97	461.09	171.42%	48.21%	35.96%	53.02%	54.37%	21.11%	1.61%
A-n38-k5	76%	4	606.33	3.14	609.47	71%	3	515.49	22.51	538	71%	3	515.49	22.51	538	222.73%	93.24%	32.16%	44.35%	46.66%	13.29%	0.00%
A-n39-k6	77%	5	680.59	3.12	683.71	56%	3	491.57	47.61	539.18	59%	3	489.58	42.58	532.16	217.18%	108.94%	43.52%	62.64%	66.25%	28.48%	1.32%
A-n45-k6	78%	5	817.93	3.22	821.15	64%	4	650.59	31.58	682.17	64%	4	650.59	31.58	682.17	211.63%	88.27%	36.27%	50.10%	57.84%	20.37%	0.00%
A-n45-k7	80%	5	1002.76	4.02	1,006.77	58%	4	683.32	52.95	736.27	58%	4	683.32	52.95	736.27	308.53%	119.16%	46.96%	77.88%	83.26%	36.74%	0.00%
A-n55-k9	75%	8	986.14	4.49	990.63	58%	5	699.57	59.72	759.29	58%	5	699.57	59.72	759.29	213.79%	118.06%	34.23%	63.08%	72.74%	30.47%	0.00%
A-n60-k9	78%	7	1129.48	4.82	1,134.30	62%	4	766.22	75.2	841.42	62%	4	766.22	75.2	841.42	265.34%	112.01%	53.53%	80.10%	97.12%	34.81%	0.00%
A-n61-k9	75%	6	746.07	4.75	750.81	59%	4	546.39	57.41	603.8	59%	4	546.39	57.41	603.8	254.96%	129.96%	56.39%	80.07%	84.50%	24.35%	0.00%
A-n63-k9	78%	7	1297.66	5.03	1,302.68	65%	5	951.54	72.16	1,023.70	65%	5	951.54	72.16	1,023.70	295.21%	152.76%	48.70%	71.47%	84.45%	27.25%	0.00%
A-n65-k9	77%	7	931.4	5.51	936.91	63%	5	681.05	61.42	742.47	60%	5	672.41	65.27	737.69	217.07%	66.17%	50.38%	72.99%	79.17%	27.01%	0.65%
A-n80-k10	80%	8	1476.63	6.06	1,482.69	58%	5	1015.04	148.77	1,163.81	58%	5	989.05	152.59	1,141.64	346.62%	146.15%	45.91%	70.24%	82.02%	29.87%	1.94%
B-n31-k5	77%	4	576.97	2.25	579.22	74%	3	500.47	12.66	513.13	71%	3	475.04	18.29	493.33	254.49%	117.11%	25.03%	42.96%	60.15%	17.41%	4.01%
B-n35-k5	80%	4	755.01	3.38	758.39	69%	4	713.53	18.87	732.4	63%	3	620.17	35.21	655.38	244.38%	120.42%	25.95%	59.17%	67.55%	15.72%	11.75%
B-n39-k5	74%	4	461.32	3.9	465.22	64%	3	357.73	18.35	376.08	67%	3	359.92	15.3	375.22	323.67%	163.81%	43.72%	58.34%	62.26%	23.99%	0.23%
B-n41-k6	76%	5	637.21	3.03	640.24	61%	4	536.08	19.09	555.17	59%	4	531.12	22.38	553.5	239.54%	97.69%	46.82%	56.72%	67.79%	15.67%	0.30%
B-n45-k5	80%	4	627.37	4.2	631.57	64%	4	454.48	30.56	485.05	67%	3	429.69	33.53	463.22	264.55%	161.25%	53.53%	76.36%	86.59%	36.34%	4.71%
B-n50-k7	82%	5	546.66	3.36	550.02	62%	4	479.66	33.68	513.34	62%	4	476.75	33.68	510.43	281.07%	125.02%	28.21%	62.38%	69.02%	7.76%	0.57%
B-n52-k7	81%	5	618.38	4.07	622.45	69%	4	459.22	37.79	497.01	69%	4	459.22	37.79	497.01	361.42%	200.85%	37.60%	76.62%	82.68%	25.24%	0.00%
B-n56-k7	77%	5	489.17	4.07	493.24	68%	4	434.08	20.74	454.82	68%	4	427.24	20.14	447.38	322.50%	149.64%	40.70%	49.43%	53.39%	10.25%	1.66%
B-n57-k9	79%	7	1312.75	4.4	1,317.14	72%	6	1130.46	39.62	1,170.09	67%	5	1052.23	59.07	1,111.30	293.64%	125.96%	37.51%	58.92%	67.36%	18.52%	5.29%
B-n64-k9	78%	7	745.89	5.23	751.12	69%	6	604.24	28.71	632.95	69%	6	604.24	27.43	631.67	257.89%	123.18%	31.39%	59.97%	65.74%	18.91%	0.20%
B-n67-k10	76%	8	869.73	5.63	875.36	66%	6	698.07	52.95	751.01	67%	6	698.09	51.25	749.33	239.23%	107.51%	31.20%	56.74%	69.45%	16.82%	0.22%
B-n68-k9	78%	7	986.85	5.49	992.35	66%	5	735.55	67.35	802.9	66%	5	735.55	67.35	802.9	330.49%	128.40%	44.71%	73.28%	84.99%	23.59%	0.00%
B-n78-k10	77%	8	980.29	5.44	985.73	62%	6	754.48	59.72	814.2	62%	6	754.48	59.72	814.2	284.21%	92.16%	35.67%	65.85%	74.20%	21.07%	0.00%

Table 11.7: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.01$, low variance level, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2111.37	2111.37	3	563.64	792.48	1356.12	4	745.95	60.47	806.43	6	899.89	6.53	906.42	6	954.98	6.91	961.89
A-n33-k5	33	1513.39	1513.39	3	505.86	546.95	1052.81	4	622.53	41.75	664.28	6	720.32	4.53	724.85	6	754.64	3.49	758.13
A-n33-k6	33	1353.18	1353.18	3	546.36	161.81	708.16	5	702.23	48.51	750.74	7	797.2	5.85	803.04	7	812.31	6.2	818.51
A-n37-k5	37	1281.17	1281.17	3	563.32	378.69	942.01	4	623.81	37.54	661.35	5	694.71	17.14	711.85	5	706.1	9.64	715.74
A-n38-k5	38	1771.99	1771.99	3	550.46	562.06	1112.52	5	700.63	45.87	746.5	6	771.38	7.75	779.14	6	783.47	7.29	790.76
A-n39-k6	39	1726.17	1726.17	3	630.6	546.98	1177.58	5	751.95	53.68	805.63	7	860.12	6.62	866.74	7	879.78	6.16	885.94
A-n45-k6	45	2191.14	2191.14	3	669.56	677.42	1346.97	6	903.12	73.41	976.53	7	1018.08	6.54	1024.62	8	1072.01	5.04	1,077.05
A-n45-k7	45	3065.13	3065.13	4	769.5	968.67	1738.17	6	1055.75	73.65	1129.4	8	1305.94	9.22	1315.16	8	1344.19	6.39	1,350.58
A-n55-k9	55	2432.56	2432.56	5	761.39	868.8	1630.19	8	993.22	76.62	1069.83	11	1232.09	7.55	1239.64	11	1302.96	8.69	1,311.65
A-n60-k9	60	3150.31	3150.31	5	885.27	1309.12	2194.39	8	1252.69	107.76	1360.45	10	1509.37	5.97	1515.34	11	1649.71	11.88	1,661.59
A-n61-k9	61	2201.76	2201.76	5	734.03	728.76	1462.8	8	928.54	62.19	990.73	10	1079.62	7.07	1086.69	11	1106.76	7.32	1,114.08
A-n63-k9	63	4129.64	4129.64	5	1037.03	935.35	1972.38	8	1478.71	134.67	1613.38	10	1750.36	5.83	1756.19	11	1883.2	5.84	1,889.04
A-n65-k9	65	2425.94	2425.94	5	817.05	524.89	1341.95	8	1075.08	83.56	1158.63	10	1266.74	10.35	1277.09	11	1311.92	12.42	1,324.34
A-n80-k10	80	5230.78	5230.78	5	1152.13	1832.89	2985.01	9	1603.11	142.74	1745.85	11	1927.03	27.58	1954.6	12	2071.27	9.17	2,080.44
B-n31-k5	31	1787.34	1787.34	3	454.28	678.69	1132.97	4	595.89	32.25	628.14	5	701.21	5.25	706.45	6	785.8	4.46	790.26
B-n35-k5	35	2312.58	2312.58	3	587.88	576.15	1164.03	4	805.3	91.85	897.15	5	1037.95	4.14	1042.09	6	1092.62	8.4	1,101.02
B-n39-k5	39	1625.14	1625.14	3	373.15	669.31	1042.46	4	515.9	67.83	583.73	5	588.24	12.34	600.58	6	602.77	11.05	613.82
B-n41-k6	41	1947.19	1947.19	3	520.57	649.92	1170.49	6	794.61	49.03	843.64	7	861.97	5.44	867.42	8	922.87	5.87	928.74
B-n45-k5	45	1739.36	1739.36	3	542.83	737.92	1280.75	5	686.97	39.01	725.98	6	810.96	6.06	817.02	7	861.82	2.68	864.5
B-n50-k7	50	2000.88	2000.88	4	496.33	723.99	1220.31	6	636.84	35.56	672.4	7	824.86	6.73	831.59	8	856.4	6.97	863.38
B-n52-k7	52	2360.84	2360.84	4	545.07	571.21	1116.28	6	661.59	81.54	743.13	8	873.56	10.78	884.34	8	900.86	9.79	910.65
B-n56-k7	56	1941.31	1941.31	4	522.63	672.77	1195.39	6	609.26	42.81	652.07	8	661.98	8.69	670.67	8	679.13	9.22	688.35
B-n57-k9	57	4492.34	4492.34	5	969.42	1693.48	2662.9	8	1480.43	116.43	1596.86	10	1760.09	8.92	1769.02	10	1851.48	10.1	1,861.58
B-n64-k9	64	2321.79	2321.79	5	577.2	913.71	1490.92	9	800.84	78.23	879.07	11	1001.4	11.66	1013.06	12	1037.36	11.21	1,048.57
B-n67-k10	67	2605.11	2605.11	5	714.24	398.56	1112.8	9	957.25	75.82	1033.08	12	1165.23	12.22	1177.45	12	1259.61	12.68	1,272.29
B-n68-k9	68	3562.55	3562.55	5	785.45	1150.62	1936.06	8	1120.88	107.99	1228.87	10	1384.54	15.11	1399.64	11	1475.73	12.19	1,487.91
B-n78-k10	78	3201.06	3201.06	5	825.79	801.51	1627.3	9	1078.42	71.75	1150.17	11	1341.39	12.38	1353.77	11	1408.94	12.49	1,421.43

Table 11.8: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.01$, medium variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	88%	4	747.16	4.92	752.08	75%	4	676.44	38.43	714.88	78%	3	648.75	63.86	712.61	196.29%	90.30%	13.17%	27.20%	34.98%	5.54%	0.32%
A-n33-k5	79%	4	623.91	2.77	626.68	58%	3	451.03	118.02	569.05	64%	3	455.02	103.42	558.45	171.00%	88.52%	18.95%	29.80%	35.76%	12.22%	1.90%
A-n33-k6	79%	4	621.84	3.51	625.35	67%	3	522.73	55.38	578.11	67%	3	522.73	55.38	578.11	134.07%	22.50%	29.86%	38.91%	41.58%	8.17%	0.00%
A-n37-k5	84%	4	554.74	7.68	562.42	62%	3	441.57	65.04	506.61	62%	3	441.57	61.73	503.31	154.55%	87.16%	31.40%	41.43%	42.21%	11.75%	0.66%
A-n38-k5	76%	4	622.76	4.93	627.69	76%	3	539.21	33.23	572.43	76%	3	539.21	33.23	572.43	209.55%	94.35%	30.41%	36.11%	38.14%	9.65%	0.00%
A-n39-k6	79%	5	707.12	4.43	711.55	64%	4	521.62	87.65	609.27	64%	4	521.62	87.65	609.27	183.32%	93.28%	32.23%	42.26%	45.41%	16.79%	0.00%
A-n45-k6	80%	5	822.01	3.68	825.69	78%	4	725.46	43.81	769.27	78%	4	725.46	43.81	769.27	184.83%	75.10%	26.94%	33.19%	40.01%	7.33%	0.00%
A-n45-k7	80%	5	1002.76	5.81	1,008.57	67%	4	752.03	112.82	864.85	67%	4	752.03	112.82	864.85	254.41%	100.98%	30.59%	52.07%	56.16%	16.62%	0.00%
A-n55-k9	76%	8	998.25	4.58	1,002.82	65%	6	759.81	133.54	893.35	65%	6	759.81	133.54	893.35	172.30%	82.48%	19.76%	38.76%	46.82%	12.25%	0.00%
A-n60-k9	82%	7	1146.77	7.87	1,154.64	65%	5	810.01	224.4	1,034.41	65%	5	810.01	224.4	1,034.41	204.55%	112.14%	31.52%	46.49%	60.63%	11.62%	0.00%
A-n61-k9	79%	7	786.26	5.04	791.3	67%	5	631.13	96.93	728.05	67%	5	631.13	96.93	728.05	202.42%	100.92%	36.08%	49.26%	53.02%	8.69%	0.00%
A-n63-k9	78%	7	1344.95	6.71	1,351.66	71%	6	1098.34	129.43	1,227.77	71%	6	1098.34	129.43	1,227.77	236.35%	60.65%	31.41%	43.04%	53.86%	10.09%	0.00%
A-n65-k9	77%	7	931.4	8.15	939.54	66%	5	700.19	160.67	860.86	66%	5	700.19	160.67	860.86	181.80%	55.88%	34.59%	48.35%	53.84%	9.14%	0.00%
A-n80-k10	81%	8	1539.74	11.78	1,551.52	63%	5	1056.08	401.31	1,457.39	63%	5	1056.08	401.31	1,457.39	258.92%	104.82%	19.79%	34.12%	42.75%	6.46%	0.00%
B-n31-k5	77%	4	576.97	2.46	579.43	77%	3	489.35	35.78	525.13	77%	3	503.72	11.22	514.94	247.10%	120.02%	21.98%	37.19%	53.47%	12.52%	1.98%
B-n35-k5	80%	4	761.76	6.31	768.07	71%	4	714.15	63.27	777.42	69%	3	613.83	102.35	716.18	222.91%	62.53%	25.27%	45.51%	53.74%	7.25%	8.55%
B-n39-k5	74%	4	461.32	8.91	470.23	67%	3	358.25	54.36	412.6	72%	3	363.6	45.5	409.1	297.24%	154.82%	42.69%	46.80%	50.04%	14.94%	0.86%
B-n41-k6	76%	5	637.21	3.03	640.24	71%	4	590.28	6.79	597.07	71%	4	590.28	6.79	597.07	226.12%	96.04%	41.30%	45.28%	55.55%	7.23%	0.00%
B-n45-k5	84%	4	637.25	8.32	645.58	71%	4	503.25	61.47	564.72	73%	4	507.01	53.61	560.62	210.26%	128.45%	29.50%	45.74%	54.20%	15.16%	0.73%
B-n50-k7	82%	5	548.19	4.1	552.29	68%	5	526.19	53.89	580.08	70%	5	526.19	39.49	565.67	253.72%	115.73%	18.87%	47.01%	52.63%	-2.37%	2.55%
B-n52-k7	85%	6	630.71	6.86	637.57	81%	5	572.56	48.49	621.06	81%	5	572.56	48.49	621.06	280.13%	79.74%	19.66%	42.39%	46.63%	2.66%	0.00%
B-n56-k7	82%	5	491.67	6.32	497.99	71%	4	435.36	52.23	487.59	71%	4	435.36	52.23	487.59	298.15%	145.17%	33.73%	37.55%	41.18%	2.13%	0.00%
B-n57-k9	79%	7	1308.43	6.15	1,314.58	75%	6	1151.63	81.32	1,232.95	75%	6	1151.63	81.32	1,232.95	264.36%	115.98%	29.52%	43.48%	50.99%	6.62%	0.00%
B-n64-k9	78%	7	745.89	6.97	752.86	75%	6	619.57	53.37	672.93	75%	6	622.06	45.98	668.04	247.55%	123.18%	31.59%	51.65%	56.96%	12.70%	0.73%
B-n67-k10	78%	8	870.01	8.25	878.27	72%	7	712.89	124.75	837.64	72%	7	712.89	124.75	837.64	211.01%	32.85%	23.33%	40.57%	51.89%	4.85%	0.00%
B-n68-k9	79%	7	987.41	8.14	995.55	71%	6	832.1	120.29	952.39	71%	6	832.1	120.29	952.39	274.07%	103.29%	29.03%	46.96%	56.23%	4.53%	0.00%
B-n78-k10	81%	8	991.15	8.73	999.88	72%	7	909.39	59.37	968.75	72%	7	909.39	59.37	968.75	230.43%	67.98%	18.73%	39.74%	46.73%	3.21%	0.00%

Table 11.9: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.01$, medium variance level, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2152.34	2152.34	3	563.64	804.97	1368.61	4	745.95	93.41	839.37	6	899.89	9.55	909.45	6	954.98	9.93	964.92
A-n33-k5	33	1538.96	1538.96	3	505.86	592.95	1098.81	4	622.53	52.45	674.98	6	720.32	5.25	725.57	6	754.64	4.21	758.84
A-n33-k6	33	1378.76	1378.76	3	546.36	185.68	732.03	5	702.23	72.58	774.81	7	797.2	6.46	803.66	7	812.31	6.81	819.12
A-n37-k5	37	1304.8	1304.8	3	563.32	173.77	737.09	4	623.81	53.74	677.55	5	694.71	23.4	718.1	5	706.1	14.77	720.87
A-n38-k5	38	1797.33	1797.33	3	550.46	615.61	1166.07	5	700.63	69.15	769.78	6	771.38	10.03	781.41	6	783.47	8.67	792.15
A-n39-k6	39	1755.43	1755.43	3	630.6	596.19	1226.79	5	751.95	75.97	827.91	7	860.12	8.17	868.29	7	879.78	7.66	887.43
A-n45-k6	45	2239.81	2239.81	3	669.56	723.35	1392.91	6	903.12	80.15	983.26	7	1018.08	6.53	1024.61	8	1072.01	5.52	1,077.53
A-n45-k7	45	3110.7	3110.7	4	769.5	1023.23	1792.73	6	1055.75	108.57	1164.32	8	1305.94	7.15	1313.09	8	1344.19	7.95	1,352.15
A-n55-k9	55	2468.55	2468.55	5	761.39	919.05	1680.44	8	993.22	120.4	1113.62	11	1232.09	7.67	1239.76	11	1302.96	8.8	1,311.77
A-n60-k9	60	3204.53	3204.53	5	885.27	1395.96	2281.24	8	1252.69	168.47	1421.16	10	1509.37	18.09	1527.46	11	1649.71	16.38	1,666.09
A-n61-k9	61	2248.2	2248.2	5	734.03	784.74	1518.77	8	928.54	54.21	982.75	10	1079.62	8.76	1088.38	11	1106.76	7.41	1,114.17
A-n63-k9	63	4194.46	4194.46	5	1037.03	1421.91	2458.94	8	1478.71	212.15	1690.86	10	1750.36	8.47	1758.83	11	1883.2	8.48	1,891.68
A-n65-k9	65	2490.84	2490.84	5	817.05	589.98	1407.03	8	1075.08	121.25	1196.32	10	1266.74	18.05	1284.79	11	1311.92	16.42	1,328.33
A-n80-k10	80	5324.59	5324.59	5	1152.13	1806.92	2959.05	9	1603.11	275.58	1878.69	11	1927.03	44.93	1971.96	12	2071.27	17.67	2,088.94
B-n31-k5	31	1816.47	1816.47	3	454.28	92.93	547.21	4	595.89	84.82	680.72	5	701.21	5.46	706.66	6	785.8	5.01	790.82
B-n35-k5	35	2353.23	2353.23	3	587.88	625.21	1213.09	4	805.3	139.79	945.09	5	1037.95	15.37	1053.31	6	1092.62	12.77	1,105.39
B-n39-k5	39	1650.19	1650.19	3	373.15	114.5	487.64	4	515.9	88.99	604.89	5	588.24	20.73	608.98	6	602.77	16.77	619.54
B-n41-k6	41	2000.67	2000.67	3	520.57	703.45	1224.03	6	794.61	71.65	866.26	7	861.97	5.44	867.42	8	922.87	5.87	928.74
B-n45-k5	45	1778.7	1778.7	3	542.83	555.31	1098.14	5	686.97	104.41	791.38	6	810.96	8.91	819.88	7	861.82	3.21	865.03
B-n50-k7	50	2042.68	2042.68	4	496.33	780.46	1276.78	6	636.84	84.69	721.52	7	824.86	9.06	833.92	8	856.4	8.63	865.03
B-n52-k7	52	2409.51	2409.51	4	545.07	593.71	1138.77	6	661.59	121.63	783.22	8	873.56	16.14	889.7	8	900.86	12.7	913.57
B-n56-k7	56	1980.32	1980.32	4	522.63	729.42	1252.05	6	609.26	79.58	688.84	8	661.98	11.35	673.33	8	679.13	11.54	690.67
B-n57-k9	57	4573.64	4573.64	5	969.42	1807.99	2777.4	8	1480.43	206.74	1687.18	10	1760.09	13.66	1773.75	10	1851.48	13.25	1,864.73
B-n64-k9	64	2364.5	2364.5	5	577.2	973.69	1550.89	9	800.84	57.18	858.02	11	1001.4	15.79	1017.19	12	1037.36	13.82	1,051.18
B-n67-k10	67	2651.23	2651.23	5	714.24	1025.71	1739.94	9	957.25	97.88	1055.14	12	1165.23	12.62	1177.85	12	1259.61	16	1,275.61
B-n68-k9	68	3643.56	3643.56	5	785.45	1229.42	2014.87	8	1120.88	157.59	1278.47	10	1384.54	14.86	1399.39	11	1475.73	16.1	1,491.83
B-n78-k10	78	3258.74	3258.74	5	825.79	848.63	1674.41	9	1078.42	107.53	1185.95	11	1341.39	19.71	1361.1	12	1408.94	19.29	1,428.23

Table 11.10: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.01$, high variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	88%	4	747.16	7.94	755.1	78%	4	683.14	36.74	719.88	78%	3	635.85	52.17	688.02	212.83%	98.92%	22.00%	32.18%	40.25%	9.75%	4.63%
A-n33-k5	79%	4	623.91	3.51	627.42	61%	3	459.12	166.48	625.59	61%	3	459.12	131.55	590.67	160.55%	86.03%	14.27%	22.84%	28.47%	6.22%	5.91%
A-n33-k6	82%	4	622.37	4.15	626.52	73%	3	546	66.84	612.84	73%	3	546	66.84	612.84	124.98%	19.45%	26.43%	31.14%	33.66%	2.23%	0.00%
A-n37-k5	86%	4	565.84	12.85	578.69	62%	3	441.57	100.48	542.05	62%	3	443.75	93.14	536.89	143.03%	37.29%	26.20%	33.75%	34.27%	7.79%	0.96%
A-n38-k5	79%	4	628.41	6.39	634.8	76%	3	546.68	49.18	595.86	76%	3	546.68	49.18	595.86	201.64%	95.70%	29.19%	31.14%	32.94%	6.54%	0.00%
A-n39-k6	79%	5	726.11	5.99	732.11	67%	4	539.84	119.54	659.38	64%	4	527.4	115.38	642.78	173.10%	90.86%	28.80%	35.08%	38.06%	13.90%	2.58%
A-n45-k6	80%	5	822.01	4.53	826.54	78%	4	735.79	36.45	772.24	78%	4	735.79	36.45	772.24	190.04%	80.37%	27.33%	32.68%	39.53%	7.03%	0.00%
A-n45-k7	82%	5	1002.76	7.76	1,010.52	69%	5	855.28	105	960.28	71%	5	858.99	94.7	953.69	226.18%	87.98%	22.09%	37.69%	41.78%	5.96%	0.69%
A-n55-k9	76%	8	998.25	4.69	1,002.94	69%	6	854.66	98.47	953.13	69%	6	834.03	118.77	952.8	159.08%	76.37%	16.88%	30.12%	37.67%	5.26%	0.03%
A-n60-k9	82%	7	1156.37	12.42	1,168.79	67%	5	830.46	300.2	1,130.67	67%	5	830.46	300.2	1,130.67	183.42%	101.76%	25.69%	35.09%	47.36%	3.37%	0.00%
A-n61-k9	79%	7	786.26	5.78	792.04	72%	6	697.48	69.15	766.63	72%	6	697.48	69.15	766.63	193.26%	98.11%	28.19%	41.97%	45.33%	3.31%	0.00%
A-n63-k9	78%	7	1344.96	10.5	1,355.46	75%	6	1164.52	111.52	1,276.04	75%	6	1164.52	111.52	1,276.04	228.71%	92.70%	32.51%	37.84%	48.25%	6.22%	0.00%
A-n65-k9	78%	7	942.94	12.17	955.1	68%	5	702.59	246.57	949.17	68%	5	702.59	232.86	935.45	166.27%	50.41%	27.89%	37.34%	42.00%	2.10%	1.47%
A-n80-k10	83%	8	1530.88	24.39	1,555.27	65%	6	1123.28	512.77	1,636.05	65%	6	1145.72	486.41	1,632.13	226.23%	81.30%	15.11%	20.82%	27.99%	-4.71%	0.24%
B-n31-k5	77%	4	576.97	3.02	579.99	77%	3	493.2	40.05	533.25	77%	3	486.78	40.15	526.93	244.73%	3.85%	29.19%	34.11%	50.08%	10.07%	1.20%
B-n35-k5	80%	4	761.76	10.68	772.44	74%	4	740.51	67.4	807.91	80%	4	742.05	52.54	794.59	196.16%	52.67%	18.94%	32.56%	39.11%	-2.79%	1.68%
B-n39-k5	77%	4	466.06	14.66	480.72	67%	3	358.25	78.24	436.49	67%	3	358.25	78.24	436.49	278.06%	11.72%	38.58%	39.52%	41.94%	10.13%	0.00%
B-n41-k6	76%	5	647.15	3.06	650.22	71%	4	590.28	9.16	599.44	71%	4	590.28	9.16	599.44	233.76%	104.19%	44.51%	44.70%	54.93%	8.47%	0.00%
B-n45-k5	84%	4	637.25	14.65	651.9	76%	4	521.61	91.82	613.43	76%	4	521.61	91.82	613.43	189.96%	79.02%	29.01%	33.65%	41.02%	6.27%	0.00%
B-n50-k7	84%	5	552.96	5.85	558.81	76%	5	539.22	49.27	588.49	84%	5	544.54	30.64	575.19	255.13%	121.98%	25.44%	44.98%	50.39%	-2.85%	2.31%
B-n52-k7	85%	6	643.45	9.82	653.27	83%	5	595.18	53.23	648.4	83%	5	595.18	53.23	648.4	271.61%	75.63%	20.79%	37.21%	40.89%	0.75%	0.00%
B-n56-k7	84%	5	579.54	8.73	588.27	75%	4	436.92	78.34	515.26	75%	4	436.92	78.34	515.26	284.33%	142.99%	33.69%	30.68%	34.04%	14.17%	0.00%
B-n57-k9	81%	7	1311.77	9.44	1,321.21	75%	6	1154.4	124.91	1,279.30	75%	6	1153.22	125.25	1,278.47	257.74%	117.24%	31.97%	38.74%	45.86%	3.34%	0.07%
B-n64-k9	80%	7	752.67	9.66	762.34	78%	6	633.89	73.69	707.58	78%	6	643.15	62.52	705.67	235.07%	119.78%	21.59%	44.14%	48.96%	8.03%	0.27%
B-n67-k10	81%	8	881.49	11.66	893.15	76%	7	735.19	138.82	874.01	76%	7	735.19	138.82	874.01	203.34%	99.08%	20.72%	34.76%	45.95%	2.19%	0.00%
B-n68-k9	82%	7	990.79	12.13	1,002.92	74%	6	855.33	157.98	1,013.31	76%	6	860.22	148.18	1,008.40	261.32%	99.81%	26.78%	38.77%	47.94%	-0.54%	0.49%
B-n78-k10	82%	8	993.88	15.82	1,009.71	79%	8	970.23	22.39	992.62	79%	8	970.23	22.39	992.62	228.30%	68.69%	19.48%	37.12%	43.89%	1.72%	0.00%

Table 11.11: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.01$, high variance level, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2064.12	2064.12	3	563.64	719.23	1282.87	4	745.95	31.55	777.5	6	899.89	19.84	919.74	6	954.98	21.74	976.73
A-n33-k5	33	1480.64	1480.64	3	505.86	498.11	1003.96	4	622.53	26.09	648.62	6	720.32	21.03	741.35	6	754.64	23.1	777.74
A-n33-k6	33	1322.59	1322.59	3	546.36	547.37	1093.73	5	702.23	35.53	737.76	7	797.2	26.26	823.45	7	812.31	28.01	840.32
A-n37-k5	37	1253.83	1253.83	3	563.32	332.96	896.29	4	623.81	29.21	653.02	5	694.71	26.86	721.57	5	706.1	22.68	728.77
A-n38-k5	38	1739.49	1739.49	3	550.46	497.44	1047.9	5	700.63	30.66	731.29	6	771.38	23.89	795.27	6	783.47	25.52	808.99
A-n39-k6	39	1690.64	1690.64	3	630.6	543.34	1173.94	5	751.95	35.85	787.8	7	860.12	25.77	885.89	7	879.78	27.54	907.32
A-n45-k6	45	2128.91	2128.91	3	669.56	852.82	1522.38	6	903.12	44.19	947.31	7	1018.08	29.03	1047.11	8	1072.01	30.76	1,102.77
A-n45-k7	45	3011.26	3011.26	4	769.5	854.06	1623.55	6	1055.75	43.97	1099.72	8	1305.94	30.76	1336.7	8	1344.19	33.4	1,377.59
A-n55-k9	55	2387.15	2387.15	5	761.39	906.3	1667.69	8	993.22	46.34	1039.56	11	1232.09	37.63	1269.72	11	1302.96	43.31	1,346.27
A-n60-k9	60	3077.79	3077.79	5	885.27	1207.98	2093.25	8	1252.69	61.53	1314.22	10	1509.37	39.62	1548.98	11	1649.71	43.26	1,692.97
A-n61-k9	61	2149.21	2149.21	5	734.03	670.28	1404.32	8	928.54	52.84	981.37	10	1079.62	43.4	1123.02	11	1106.76	45.76	1,152.51
A-n63-k9	63	4050.78	4050.78	5	1037.03	1565.56	2602.59	8	1478.71	68.7	1547.41	10	1750.36	41.56	1791.92	11	1883.2	45.49	1,928.69
A-n65-k9	65	2344.05	2344.05	5	817.05	807.34	1624.39	8	1075.08	59.67	1134.74	10	1266.74	43.12	1309.86	11	1311.92	46.2	1,358.12
A-n80-k10	80	5104.65	5104.65	5	1152.13	1836.81	2988.93	9	1603.11	88.41	1691.52	11	1927.03	52.33	1979.36	12	2071.27	49.92	2,121.19
B-n31-k5	31	1751.16	1751.16	3	454.28	623.31	1077.59	4	595.89	32.32	628.21	5	701.21	19.75	720.95	6	785.8	21.35	807.15
B-n35-k5	35	2259.29	2259.29	3	587.88	863.69	1451.57	4	805.3	46.61	851.91	5	1037.95	21.69	1059.63	6	1092.62	23.66	1,116.27
B-n39-k5	39	1592.72	1592.72	3	373.15	624.37	997.52	4	515.9	37.92	553.82	5	588.24	22.87	611.11	6	602.77	24.42	627.19
B-n41-k6	41	1882.63	1882.63	3	520.57	583.52	1104.09	6	794.61	34.84	829.45	7	861.97	27.22	889.2	8	922.87	29.35	952.22
B-n45-k5	45	1691.45	1691.45	3	542.83	674.87	1217.71	5	686.97	37.02	723.99	6	810.96	24.03	834.99	7	861.82	26.88	888.7
B-n50-k7	50	1948.46	1948.46	4	496.33	662.26	1158.59	6	636.84	39.18	676.02	7	824.86	29.81	854.67	8	856.4	31.68	888.09
B-n52-k7	52	2296.36	2296.36	4	545.07	958.77	1503.84	6	661.59	46.29	707.88	8	873.56	29.89	903.45	8	900.86	32.32	933.18
B-n56-k7	56	1893.65	1893.65	4	522.63	604.42	1127.04	6	609.26	37.88	647.14	8	661.98	30.19	692.17	8	679.13	32.84	711.97
B-n57-k9	57	4378.7	4378.7	5	969.42	1554.3	2523.72	8	1480.43	70.33	1550.76	10	1760.09	38.49	1798.58	10	1851.48	41.61	1,893.09
B-n64-k9	64	2265.89	2265.89	5	577.2	845.97	1423.17	9	800.84	53.59	854.43	11	1001.4	43.35	1044.75	12	1037.36	45.93	1,083.29
B-n67-k10	67	2546.82	2546.82	5	714.24	853.94	1568.18	9	957.25	53.77	1011.02	12	1165.23	43.34	1208.57	12	1259.61	47.69	1,307.30
B-n68-k9	68	3461.28	3461.28	5	785.45	1150.74	1936.19	8	1120.88	64.66	1185.54	10	1384.54	42.72	1427.25	11	1475.73	44.42	1,520.14
B-n78-k10	78	3133.73	3133.73	5	825.79	1258.37	2084.15	9	1078.42	66.64	1145.06	11	1341.39	44.02	1385.41	12	1408.94	49.02	1,457.96

Table 11.12: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.05$, low variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	81%	4	699.73	11.67	711.4	59%	3	600.08	31.38	631.46	59%	3	600.08	31.38	631.46	226.88%	103.16%	23.13%	45.65%	54.68%	12.66%	0.00%
A-n33-k5	79%	4	571.96	12	583.97	55%	2	393.93	50.05	443.97	55%	2	393.93	50.05	443.97	233.50%	126.13%	46.09%	66.98%	75.18%	31.53%	0.00%
A-n33-k6	70%	4	563.59	14.26	577.85	58%	3	504.88	30.68	535.55	58%	3	504.88	30.68	535.55	146.96%	104.22%	37.76%	53.76%	56.91%	7.90%	0.00%
A-n37-k5	81%	3	504.55	12.59	517.13	62%	2	437.12	32.26	469.38	62%	2	436.83	32.26	469.09	167.29%	91.07%	39.21%	53.82%	55.36%	10.24%	0.06%
A-n38-k5	76%	4	577.61	13.56	591.17	66%	3	510.51	34.08	544.58	66%	3	510.51	34.08	544.58	219.42%	92.42%	34.28%	46.03%	48.55%	8.55%	0.00%
A-n39-k6	74%	4	642.47	13.31	655.77	56%	3	491.57	53.39	544.96	56%	3	491.57	53.39	544.96	210.24%	115.42%	44.56%	62.56%	66.49%	20.33%	0.00%
A-n45-k6	78%	5	757.03	16.09	773.12	67%	4	667.04	36.31	703.35	67%	4	667.04	36.31	703.35	202.68%	116.45%	34.69%	48.88%	56.79%	9.92%	0.00%
A-n45-k7	80%	5	945.12	17.68	962.8	58%	4	683.32	60.69	744.01	58%	4	683.32	60.69	744.01	304.73%	118.22%	47.81%	79.66%	85.16%	29.41%	0.00%
A-n55-k9	75%	7	951.19	22.32	973.51	60%	5	705.08	65.93	771.01	60%	5	703.49	66.53	770.02	210.01%	116.58%	35.00%	64.90%	74.84%	26.43%	0.13%
A-n60-k9	78%	7	1092.53	22.54	1,115.07	62%	4	754.07	85.34	839.41	63%	4	745.48	88.02	833.5	269.26%	151.14%	57.68%	85.84%	103.12%	33.78%	0.71%
A-n61-k9	75%	6	746.07	23.69	769.76	57%	4	542.34	69.62	611.96	57%	4	536.78	69.62	606.4	254.42%	131.58%	61.84%	85.20%	90.06%	26.94%	0.92%
A-n63-k9	78%	7	1216.34	24.01	1,240.35	63%	5	949	85.05	1,034.06	62%	5	944.41	89.1	1,033.51	291.94%	151.82%	49.72%	73.38%	86.62%	20.01%	0.05%
A-n65-k9	77%	7	896.11	24.6	920.71	63%	5	689.08	70.3	759.38	65%	5	686.83	69.02	755.85	210.12%	114.91%	50.13%	73.30%	79.68%	21.81%	0.47%
A-n80-k10	80%	8	1432.12	26.96	1,459.08	54%	5	983.42	164.41	1,147.83	54%	5	983.42	164.41	1,147.83	344.72%	160.40%	47.37%	72.44%	84.80%	27.12%	0.00%
B-n31-k5	77%	3	505.23	11.24	516.47	74%	3	482.04	20.21	502.25	71%	3	471.42	22.16	493.58	254.79%	118.32%	27.28%	46.07%	63.53%	4.64%	1.76%
B-n35-k5	80%	4	750.89	13.11	763.99	69%	4	713.53	27.46	740.99	69%	3	625.31	30.85	656.16	244.32%	121.22%	29.83%	61.49%	70.12%	16.43%	12.93%
B-n39-k5	74%	4	434.6	13.65	448.25	62%	3	357.72	28.62	386.34	62%	3	357.72	24.66	382.38	316.53%	160.87%	44.83%	59.82%	64.02%	17.23%	1.04%
B-n41-k6	73%	4	605.99	14.99	620.98	61%	4	536.08	28.83	564.91	63%	4	535.55	28.56	564.11	233.73%	95.72%	47.04%	57.63%	68.80%	10.08%	0.14%
B-n45-k5	80%	4	618.25	15.07	633.32	64%	4	454.48	38.05	492.53	64%	4	454.48	38.05	492.53	243.42%	147.24%	46.99%	69.53%	80.44%	28.59%	0.00%
B-n50-k7	78%	5	543.25	16.7	559.95	62%	4	479.1	43.9	523	62%	4	476.75	43.9	520.65	274.23%	122.53%	29.84%	64.15%	70.57%	7.55%	0.45%
B-n52-k7	79%	5	604.96	17.33	622.29	63%	4	456.15	52.38	508.53	69%	4	465.48	40.96	506.44	353.43%	196.94%	39.77%	78.39%	84.26%	22.87%	0.41%
B-n56-k7	75%	5	462.89	17.56	480.45	64%	4	431.61	32.71	464.32	64%	4	431.61	32.71	464.32	307.83%	142.73%	39.37%	49.07%	53.34%	3.47%	0.00%
B-n57-k9	79%	6	1194.11	21.78	1,215.89	70%	5	1055.03	56.55	1,111.58	68%	5	1039	65.54	1,104.54	296.43%	128.49%	40.40%	62.84%	71.39%	10.08%	0.64%
B-n64-k9	77%	7	718.5	24.27	742.77	69%	6	604.24	44.16	648.4	69%	6	604.24	44.16	648.4	249.46%	119.49%	31.77%	61.13%	67.07%	14.55%	0.00%
B-n67-k10	76%	8	851.72	25.23	876.95	66%	6	674.09	68.43	742.52	66%	6	666.93	69.22	736.15	245.97%	113.02%	37.34%	64.17%	77.59%	19.13%	0.87%
B-n68-k9	78%	7	961.97	23.89	985.86	68%	5	761.41	68.27	829.69	68%	5	761.41	68.27	829.69	317.18%	133.36%	42.89%	72.02%	83.22%	18.82%	0.00%
B-n78-k10	76%	8	966.77	26.14	992.91	63%	6	755.27	71.12	826.39	64%	6	750.7	73.08	823.78	280.41%	153.00%	39.00%	68.18%	76.98%	20.53%	0.32%

Table 11.13: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.05$, low variance level, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2113	2113	3	563.64	797.86	1361.5	4	745.95	71.32	817.27	6	899.89	22.03	921.92	6	954.98	23.93	978.91
A-n33-k5	33	1516.22	1516.22	3	505.86	554.11	1059.97	4	622.53	53.77	676.3	6	720.32	21.35	741.67	6	754.64	23.43	778.06
A-n33-k6	33	1355.89	1355.89	3	546.36	590.63	1136.99	5	702.23	64.51	766.74	7	797.2	26.73	823.93	7	812.31	28.48	840.79
A-n37-k5	37	1283.53	1283.53	3	563.32	385.86	949.18	4	623.81	49.99	673.8	5	694.71	33.17	727.88	5	706.1	26.65	732.74
A-n38-k5	38	1775.24	1775.24	3	550.46	570.37	1120.83	5	700.63	59.7	760.32	6	771.38	26.4	797.78	6	783.47	27.27	810.74
A-n39-k6	39	1728.92	1728.92	3	630.6	610.57	1241.16	5	751.95	68.46	820.41	7	860.12	26.99	887.11	7	879.78	28.77	908.54
A-n45-k6	45	2194.23	2194.23	3	669.56	926.02	1595.58	6	903.12	91.13	994.24	7	1018.08	29.69	1047.77	8	1072.01	31.16	1,103.17
A-n45-k7	45	3068.56	3068.56	4	769.5	979.09	1748.59	6	1055.75	91.34	1147.09	8	1305.94	33.33	1339.27	8	1344.19	35.16	1,379.36
A-n55-k9	55	2437.15	2437.15	5	761.39	995.14	1756.53	8	993.22	97.01	1090.23	11	1232.09	37.65	1269.74	11	1302.96	43.33	1,346.29
A-n60-k9	60	3154.08	3154.08	5	885.27	1320.89	2206.16	8	1252.69	130.18	1382.87	10	1509.37	43.54	1552.91	11	1649.71	46.23	1,695.94
A-n61-k9	61	2207.73	2207.73	5	734.03	744.6	1478.64	8	928.54	88.55	1017.09	10	1079.62	44.48	1124.1	11	1106.76	45.95	1,152.71
A-n63-k9	63	4134.73	4134.73	5	1037.03	1728.71	2765.74	8	1478.71	159.86	1638.58	10	1750.36	43.17	1793.53	11	1883.2	47.1	1,930.30
A-n65-k9	65	2431.03	2431.03	5	817.05	893.53	1710.58	8	1075.08	108.98	1184.06	10	1266.74	46.6	1313.34	11	1311.92	48.79	1,360.70
A-n80-k10	80	5236.72	5236.72	5	1152.13	2024.88	3177.01	9	1603.11	206.47	1809.58	11	1927.03	63.41	1990.44	12	2071.27	55.58	2,126.85
B-n31-k5	31	1789.73	1789.73	3	454.28	685.27	1139.55	4	595.89	66.58	662.48	5	701.21	20.93	722.13	6	785.8	21.54	807.34
B-n35-k5	35	2314.9	2314.9	3	587.88	942.15	1530.04	4	805.3	104.04	909.34	5	1037.95	26.53	1064.48	6	1092.62	26.56	1,119.18
B-n39-k5	39	1628.16	1628.16	3	373.15	677	1050.15	4	515.9	80.44	596.34	5	588.24	29.32	617.56	6	602.77	29.41	632.18
B-n41-k6	41	1950.48	1950.48	3	520.57	659.84	1180.41	6	794.61	65.85	860.46	7	861.97	27.22	889.19	8	922.87	29.34	952.21
B-n45-k5	45	1742.17	1742.17	3	542.83	745.46	1288.29	5	686.97	80.77	767.74	6	810.96	28.09	839.05	7	861.82	30.94	892.76
B-n50-k7	50	2004.28	2004.28	4	496.33	734.07	1230.4	6	636.84	73.65	710.48	7	824.86	30.57	855.43	8	856.4	32.31	888.72
B-n52-k7	52	2363.92	2363.92	4	545.07	1021.86	1566.92	6	661.59	97.93	759.52	8	873.56	34.06	907.62	8	900.86	35.04	935.91
B-n56-k7	56	1944.8	1944.8	4	522.63	683	1205.63	6	609.26	70.18	679.44	8	661.98	32.32	694.29	8	679.13	34.97	714.1
B-n57-k9	57	4496.56	4496.56	5	969.42	1706.15	2675.57	8	1480.43	157.09	1637.53	10	1760.09	41.57	1801.67	10	1851.48	43.33	1,894.81
B-n64-k9	64	2327.02	2327.02	5	577.2	927.2	1504.4	9	800.84	102.76	903.6	11	1001.4	45.91	1047.3	12	1037.36	47.59	1,084.95
B-n67-k10	67	2610.04	2610.04	5	714.24	959.44	1673.68	9	957.25	99.73	1056.98	12	1165.23	46.31	1211.54	12	1259.61	50.27	1,309.88
B-n68-k9	68	3567.49	3567.49	5	785.45	1276.34	2061.78	8	1120.88	131.64	1252.52	10	1384.54	48.28	1432.81	11	1475.73	47.01	1,522.74
B-n78-k10	78	3206.57	3206.57	5	825.79	1368.93	2194.71	9	1078.42	147.61	1226.03	11	1341.39	47.4	1388.8	12	1408.94	52.14	1,461.08

Table 11.14: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.05$, medium variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	84%	4	699.83	13.96	713.8	75%	4	676.44	46.53	722.97	75%	3	630.2	69.01	699.21	202.20%	94.72%	16.88%	31.85%	40.00%	2.09%	3.40%
A-n33-k5	79%	4	571.96	12.45	584.41	55%	3	450.77	126.15	576.91	58%	3	428.64	135.93	564.57	168.56%	87.75%	19.79%	31.37%	37.81%	3.51%	2.19%
A-n33-k6	79%	4	588.6	15.03	603.63	64%	3	520.11	64.33	584.45	61%	3	505.97	73.73	579.7	133.90%	96.13%	32.27%	42.13%	45.04%	4.13%	0.82%
A-n37-k5	84%	3	528.11	16.83	544.93	65%	2	445.76	76.46	522.22	65%	3	444.01	70.27	514.28	149.58%	84.57%	31.02%	41.53%	42.48%	5.96%	1.55%
A-n38-k5	76%	4	577.61	15.47	593.07	74%	3	535.13	46.03	581.16	74%	3	535.13	46.03	581.16	205.47%	92.86%	30.83%	37.27%	39.50%	2.05%	0.00%
A-n39-k6	79%	4	659.67	15.93	675.6	64%	4	521.62	95.24	616.86	62%	3	513.19	102.24	615.43	180.93%	101.68%	33.31%	44.15%	47.63%	9.78%	0.23%
A-n45-k6	80%	5	759.13	16.74	775.87	78%	4	724.34	59.74	784.08	78%	4	724.34	59.74	784.08	179.85%	103.50%	26.80%	33.63%	40.70%	-1.05%	0.00%
A-n45-k7	80%	5	946.97	19.52	966.5	67%	4	752.03	122.36	874.39	67%	4	752.03	122.36	874.39	250.94%	99.98%	31.19%	53.17%	57.75%	10.53%	0.00%
A-n55-k9	75%	7	981.21	22.63	1,003.84	67%	6	761.48	154.34	915.83	67%	6	761.33	154.34	915.67	166.16%	91.83%	19.06%	38.67%	47.03%	9.63%	0.02%
A-n60-k9	82%	7	1094.05	26.08	1,120.13	65%	5	796.43	241.3	1,037.73	68%	5	834.58	186.42	1,021.00	208.92%	116.08%	35.44%	52.10%	66.11%	9.71%	1.64%
A-n61-k9	79%	6	770.62	24.19	794.81	67%	5	634.95	111.43	746.38	67%	5	622.33	114.01	736.34	199.83%	100.81%	38.13%	52.66%	56.55%	7.94%	1.36%
A-n63-k9	78%	7	1259.54	25.95	1,285.49	71%	6	1098.9	145.2	1,244.10	71%	6	1098.9	145.2	1,244.10	232.35%	122.31%	31.71%	44.16%	55.16%	3.33%	0.00%
A-n65-k9	77%	7	890.52	27.38	917.9	66%	5	700.19	167.12	867.31	71%	5	726.97	138.14	865.12	181.01%	97.73%	36.87%	51.81%	57.29%	6.10%	0.25%
A-n80-k10	80%	8	1434.19	32.85	1,467.04	61%	5	1044.87	408.34	1,453.22	61%	5	1044.87	408.34	1,453.22	260.35%	118.62%	24.52%	36.97%	46.35%	0.95%	0.00%
B-n31-k5	77%	4	574.87	11.56	586.43	77%	3	489.78	43.4	533.18	81%	3	491.35	32.12	523.48	241.89%	117.69%	26.55%	37.95%	54.23%	12.03%	1.85%
B-n35-k5	80%	4	750.89	16.09	766.98	71%	4	714.15	72.02	786.17	71%	4	714.15	72.02	786.17	194.45%	94.62%	15.67%	35.40%	42.36%	-2.44%	0.00%
B-n39-k5	74%	4	434.85	18.74	453.59	67%	3	358.25	62.32	420.57	67%	3	358.25	54.82	413.07	294.16%	154.23%	44.37%	49.51%	53.04%	9.81%	1.81%
B-n41-k6	76%	4	611.11	15.17	626.28	71%	4	590.28	18.37	608.65	71%	4	590.28	18.37	608.65	220.46%	93.94%	41.37%	46.09%	56.45%	2.90%	0.00%
B-n45-k5	67%	4	571.23	157.21	728.45	71%	4	503.25	70.4	573.64	71%	4	503.25	70.4	573.64	203.70%	124.58%	33.84%	46.27%	55.63%	26.99%	0.00%
B-n50-k7	82%	5	543.69	17.71	561.4	68%	5	526.19	64.31	590.49	68%	5	526.19	64.3	590.48	239.43%	108.37%	20.32%	44.87%	50.51%	-4.93%	0.00%
B-n52-k7	81%	5	607.39	20.29	627.68	77%	4	554.7	59.08	613.78	77%	4	554.7	59.08	613.78	285.14%	155.29%	23.75%	47.87%	52.48%	2.26%	0.00%
B-n56-k7	79%	5	482.97	20.22	503.18	68%	4	433.62	88.09	521.71	68%	4	433.62	88.09	521.71	272.78%	131.09%	30.23%	33.08%	36.88%	-3.55%	0.00%
B-n57-k9	79%	6	1198.8	23.61	1,222.40	75%	6	1151.63	96.69	1,248.33	75%	6	1151.63	96.69	1,248.33	260.21%	114.33%	31.18%	44.33%	51.79%	-2.08%	0.00%
B-n64-k9	78%	7	720.38	26.25	746.63	75%	6	618.72	80.55	699.26	75%	6	618.72	80.55	699.26	232.78%	115.14%	29.22%	49.77%	55.16%	6.77%	0.00%
B-n67-k10	78%	8	853.64	28.11	881.75	70%	7	711.96	128.01	839.97	70%	7	711.96	128.01	839.97	210.73%	99.25%	25.84%	44.24%	55.94%	4.97%	0.00%
B-n68-k9	78%	7	962.21	26.63	988.84	69%	6	833.61	120.13	953.73	69%	6	833.61	120.13	953.73	274.06%	116.18%	31.33%	50.23%	59.66%	3.68%	0.00%
B-n78-k10	78%	8	967.5	29.77	997.27	73%	7	898.11	77.02	975.13	73%	7	870.35	85.89	956.24	235.33%	129.51%	28.21%	45.23%	52.79%	4.29%	1.97%

Table 11.15: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.05$, medium variance level, max. computation time = 15 minutes).

		NO-REFILL POLICY (1)		1/4-REFILL POLICY (2)				1/2-REFILL POLICY (3)				3/4-REFILL POLICY (4)				FULL-REFILL POLICY (5)			
Instance	n	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total	M	Routing	Inventory	Total
A-n32-k5	32	2153.98	2153.98	3	563.64	862.51	1426.15	4	745.95	104.27	850.22	6	899.89	25.06	924.95	6	954.98	26.96	981.94
A-n33-k5	33	1541.81	1541.81	3	505.86	600.14	1105.99	4	622.53	75.51	698.04	6	720.32	22.07	742.39	6	754.64	24.14	778.78
A-n33-k6	33	1381.49	1381.49	3	546.36	628.03	1174.39	5	702.23	88.59	790.83	7	797.2	27.34	824.54	7	812.31	29.09	841.4
A-n37-k5	37	1307.18	1307.18	3	563.32	429.98	993.3	4	623.81	66.2	690.01	5	694.71	39.43	734.14	5	706.1	31.79	737.88
A-n38-k5	38	1800.61	1800.61	3	550.46	623.96	1174.42	5	700.63	82.99	783.62	6	771.38	28.67	800.06	6	783.47	28.66	812.13
A-n39-k6	39	1758.2	1758.2	3	630.6	661.01	1291.6	5	751.95	90.76	842.71	7	860.12	28.55	888.67	7	879.78	30.26	910.04
A-n45-k6	45	2242.92	2242.92	3	669.56	981.88	1651.43	6	903.12	128.74	1031.86	7	1018.08	31.04	1049.12	8	1072.01	32.01	1,104.01
A-n45-k7	45	3114.16	3114.16	4	769.5	1075.49	1844.99	6	1055.75	126.28	1182.03	8	1305.94	36.13	1342.07	8	1344.19	37.09	1,381.28
A-n55-k9	55	2473.17	2473.17	5	761.39	1063.27	1824.66	8	993.22	140.82	1134.03	11	1232.09	37.77	1269.86	11	1302.96	43.45	1,346.41
A-n60-k9	60	3208.33	3208.33	5	885.27	1407.78	2293.05	8	1252.69	190.92	1443.61	10	1509.37	49.48	1558.85	11	1649.71	50.73	1,700.44
A-n61-k9	61	2254.21	2254.21	5	734.03	800.63	1534.66	8	928.54	117.09	1045.63	10	1079.62	46.65	1126.27	11	1106.76	46.66	1,153.42
A-n63-k9	63	4199.59	4199.59	5	1037.03	1846.92	2883.95	8	1478.71	237.37	1716.08	10	1750.36	46.91	1797.27	11	1883.2	50.84	1,934.04
A-n65-k9	65	2495.98	2495.98	5	817.05	957.82	1774.87	8	1075.08	146.7	1221.78	10	1266.74	51.8	1318.54	11	1311.92	52.79	1,364.70
A-n80-k10	80	5330.57	5330.57	5	1152.13	2163.59	3315.72	9	1603.11	301.37	1904.48	11	1927.03	80.77	2007.8	12	2071.27	67.93	2,139.21
B-n31-k5	31	1818.87	1818.87	3	454.28	733.19	1187.47	4	595.89	96.27	692.16	5	701.21	22.2	723.4	6	785.8	22.09	807.89
B-n35-k5	35	2355.58	2355.58	3	587.88	998.5	1586.38	4	805.3	151.99	957.29	5	1037.95	31.83	1069.78	6	1092.62	30.93	1,123.55
B-n39-k5	39	1653.24	1653.24	3	373.15	711.29	1084.43	4	515.9	110.61	626.51	5	588.24	37.72	625.96	6	602.77	35.14	637.91
B-n41-k6	41	2003.98	2003.98	3	520.57	713.41	1233.98	6	794.61	88.49	883.1	7	861.97	27.22	889.2	8	922.87	29.35	952.22
B-n45-k5	45	1781.54	1781.54	3	542.83	799.78	1342.61	5	686.97	117.23	804.2	6	810.96	34.39	845.35	7	861.82	37.24	899.06
B-n50-k7	50	2046.12	2046.12	4	496.33	790.58	1286.91	6	636.84	102.72	739.55	7	824.86	32.9	857.76	8	856.4	33.97	890.37
B-n52-k7	52	2412.62	2412.62	4	545.07	1072.47	1617.54	6	661.59	138.03	799.62	8	873.56	39.41	912.97	8	900.86	37.96	938.82
B-n56-k7	56	1983.84	1983.84	4	522.63	739.7	1262.32	6	609.26	97.28	706.54	8	661.98	34.98	696.96	8	679.13	37.29	716.41
B-n57-k9	57	4577.88	4577.88	5	969.42	1820.7	2790.12	8	1480.43	229.46	1709.89	10	1760.09	46.2	1806.3	10	1851.48	46.49	1,897.97
B-n64-k9	64	2369.77	2369.77	5	577.2	987.23	1564.43	9	800.84	139.49	940.33	11	1001.4	50.04	1051.44	12	1037.36	50.21	1,087.57
B-n67-k10	67	2656.2	2656.2	5	714.24	1039.07	1753.31	9	957.25	137.88	1095.13	12	1165.23	50.01	1215.25	12	1259.61	53.6	1,313.21
B-n68-k9	68	3648.54	3648.54	5	785.45	1373.3	2158.75	8	1120.88	181.27	1302.15	10	1384.54	55.91	1440.44	11	1475.73	50.93	1,526.65
B-n78-k10	78	3264.31	3264.31	5	825.79	1451.15	2276.94	9	1078.42	215.95	1294.37	11	1341.39	54.74	1396.13	12	1408.94	59.12	1,468.06

Table 11.16: Results for No-refill, $\frac{1}{4}$-refill, $\frac{1}{2}$-refill, $\frac{3}{4}$-refill, and Full-refill policies ($\lambda = 0.05$, high variance level, max. computation time = 15 minutes).

Instance	DECENTRALIZED POLICY (6)					TOP POLICY (7)					BIASED RANDOMIZED POLICY (8)					Gaps						
	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	% served	M	Routing	Inventory	Total	(1)-(8)	(2)-(8)	(3)-(8)	(4)-(8)	(5)-(8)	(6)-(8)	(7)-(8)
A-n32-k5	88%	4	700	17.01	717	78%	4	683.14	44.97	728.11	78%	4	683.14	44.97	728.11	195.83%	95.87%	16.77%	27.04%	34.86%	-1.52%	0.00%
A-n33-k5	79%	4	582.83	13.33	596.16	61%	3	459.12	172.44	631.56	61%	3	459.12	145.57	604.68	154.98%	82.90%	15.44%	22.77%	28.79%	-1.41%	4.44%
A-n33-k6	82%	4	588.6	15.77	604.37	73%	3	546	75.33	621.33	73%	3	546	75.33	621.33	122.35%	89.01%	27.28%	32.71%	35.42%	-2.73%	0.00%
A-n37-k5	86%	3	539.2	22.08	561.28	68%	3	445.46	89.77	535.23	68%	3	445.46	89.77	535.23	144.23%	85.58%	28.92%	37.16%	37.86%	4.87%	0.00%
A-n38-k5	76%	4	577.61	17.04	594.65	76%	3	539.21	58.1	597.31	76%	3	539.21	58.1	597.31	201.45%	96.62%	31.19%	33.94%	35.97%	-0.45%	0.00%
A-n39-k6	79%	4	667.73	17.65	685.38	67%	4	539.84	130.6	670.44	67%	4	539.84	130.6	670.44	162.24%	92.65%	25.69%	32.55%	35.74%	2.23%	0.00%
A-n45-k6	80%	5	759.13	17.66	776.78	78%	4	731.18	55.35	786.54	80%	4	743.35	38.99	782.34	186.69%	111.09%	31.89%	34.10%	41.12%	-0.71%	0.54%
A-n45-k7	80%	5	944.91	21.58	966.49	69%	4	813.05	135.59	948.64	69%	4	810.83	135.59	946.41	229.05%	94.95%	24.90%	41.81%	45.95%	2.12%	0.24%
A-n55-k9	76%	8	993.55	22.9	1,016.44	69%	6	824.5	146.11	970.6	69%	6	824.5	146.11	970.6	154.81%	87.99%	16.84%	30.83%	38.72%	4.72%	0.00%
A-n60-k9	82%	7	1098.51	30.91	1,129.42	67%	5	830.46	312.96	1,143.42	68%	5	830.1	307.08	1,137.18	182.13%	101.64%	26.95%	37.08%	49.53%	-0.68%	0.55%
A-n61-k9	79%	6	770.62	25.17	795.79	70%	6	685.89	83.62	769.51	70%	6	685.89	83.62	769.51	192.94%	99.43%	35.88%	46.36%	49.89%	3.41%	0.00%
A-n63-k9	78%	7	1270.65	29.91	1,300.56	75%	6	1161.83	128.98	1,290.82	75%	6	1161.83	128.98	1,290.82	225.34%	123.42%	32.95%	39.24%	49.83%	0.75%	0.00%
A-n65-k9	78%	7	892.92	31.54	924.46	66%	5	697.74	277.23	974.96	66%	5	712.96	259.16	972.11	156.76%	82.58%	25.68%	35.64%	40.39%	-4.90%	0.29%
A-n80-k10	83%	8	1458.1	46.19	1,504.28	65%	6	1131.75	509.55	1,641.30	68%	6	1159.14	445.44	1,604.58	232.21%	106.64%	18.69%	25.13%	33.32%	-6.25%	2.29%
B-n31-k5	77%	4	574.87	12.11	586.99	77%	3	485.4	48.07	533.47	77%	3	484.32	48.07	532.39	241.64%	123.04%	30.01%	35.88%	51.75%	10.25%	0.20%
B-n35-k5	80%	4	750.89	20.46	771.35	74%	4	740.51	76.5	817.01	71%	3	652.41	145.44	797.85	195.24%	98.83%	19.98%	34.08%	40.82%	-3.32%	2.40%
B-n39-k5	77%	4	438.85	24.56	463.41	67%	3	358.25	93.61	451.86	69%	3	358.49	87.58	446.07	270.62%	143.11%	40.45%	40.33%	43.01%	3.89%	1.30%
B-n41-k6	76%	4	611.11	15.22	626.34	71%	4	590.28	20.77	611.05	71%	4	590.28	20.77	611.05	227.96%	101.94%	44.52%	45.52%	55.83%	2.50%	0.00%
B-n45-k5	84%	4	624.7	25.91	650.61	73%	4	518.21	101.21	619.42	73%	4	518.21	93.28	611.49	191.34%	119.56%	31.51%	38.24%	47.03%	6.40%	1.30%
B-n50-k7	82%	5	543.69	19.87	563.56	76%	5	539.22	61.87	601.09	76%	5	540.72	49.99	590.71	246.38%	117.86%	25.20%	45.21%	50.73%	-4.60%	1.76%
B-n52-k7	83%	5	620.13	23.52	643.65	79%	5	587.51	71.37	658.88	77%	5	570.9	76.81	647.71	272.49%	149.73%	23.45%	40.95%	44.95%	-0.63%	1.73%
B-n56-k7	84%	5	555.64	23.19	578.83	73%	4	435.36	85.86	521.21	77%	4	433.63	68.06	501.69	295.43%	151.62%	40.83%	38.92%	42.80%	15.38%	3.89%
B-n57-k9	81%	6	1201.43	27.21	1,228.63	75%	6	1154.4	140.51	1,294.90	75%	6	1154.4	140.51	1,294.90	253.53%	115.47%	32.05%	39.49%	46.57%	-5.12%	0.00%
B-n64-k9	80%	7	721.36	29.29	750.65	77%	6	632.41	101.11	733.52	77%	6	632.41	101.11	733.52	223.07%	113.28%	28.19%	43.34%	48.27%	2.34%	0.00%
B-n67-k10	81%	8	855.73	31.9	887.62	75%	7	751.8	140.1	891.9	75%	7	751.8	140.1	891.9	197.81%	96.58%	22.79%	36.25%	47.24%	-0.48%	0.00%
B-n68-k9	81%	7	964.7	30.96	995.67	71%	6	851	194.26	1,045.27	74%	6	845.39	188.73	1,034.12	252.81%	108.75%	25.92%	39.29%	47.63%	-3.72%	1.08%
B-n78-k10	81%	8	971.77	37.66	1,009.44	78%	8	971.6	43.81	1,015.42	78%	8	971.57	43.81	1,015.38	221.49%	124.24%	27.48%	37.50%	44.58%	-0.59%	0.00%

Table 11.17: Results for Decentralized, Top, and Biased-randomized policies plus summary gaps ($\lambda = 0.05$, high variance level, max. computation time = 15 minutes).

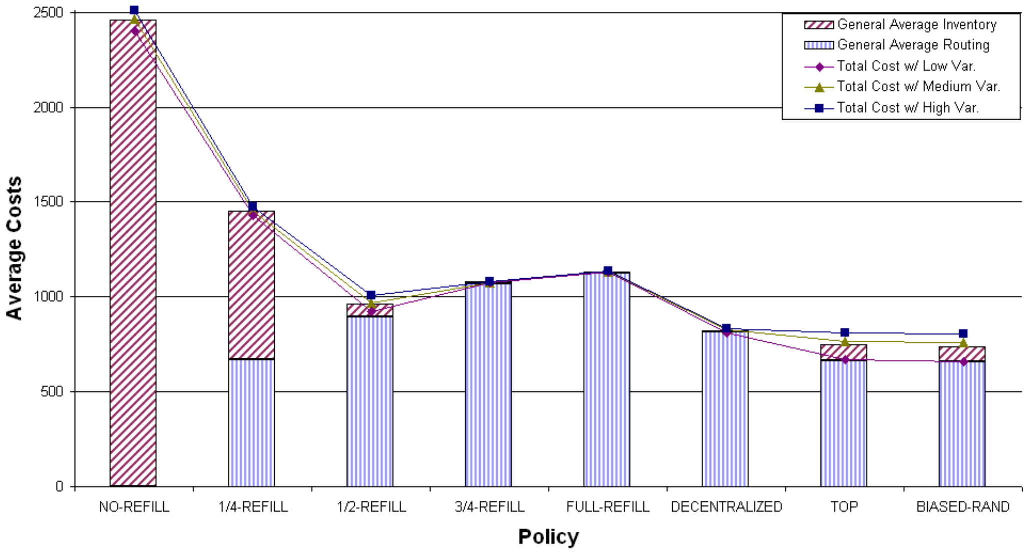


Figure 11.3: Comparison of routing and inventory costs for each refill policy ($\lambda = 0.01$).

From the average gaps in Tables 11.3 and 11.5 it can be derived that the best results are the ones obtained using our biased-randomization approach, i.e., using a different refill strategy for each node according to different factors such as: distance from the depot, current inventory level, expected demand, demand variability, etc. Also, notice that using the top strategy for each node —as proposed in an intermediate stage of our approach— provides a quite competitive solution for most instances. Even the solution with the lowest inventory costs (decentralized policy) has higher costs than the top and biased-randomized solutions. However, using non-personalized refill strategies — i.e., using the same refill strategy for all the customers as proposed in most existing articles— is a quite poor strategy, since it provides considerably higher costs.

Fig. 11.3 summarizes average routing and inventory costs associated with each policy. Notice that the two personalized (node-dependent) refill policies proposed in our approach are far superior to any other standard refill policy. Also, notice how these personalized policies tend to minimize both routing and inventory costs while minimizing total costs.

The average percentages of served retailers for each refill policy are depicted in Fig. 11.4. Notice that our customer-dependent policies show similar numbers in both statistics, i.e., about 65% of retailers will be served, implying an average number of

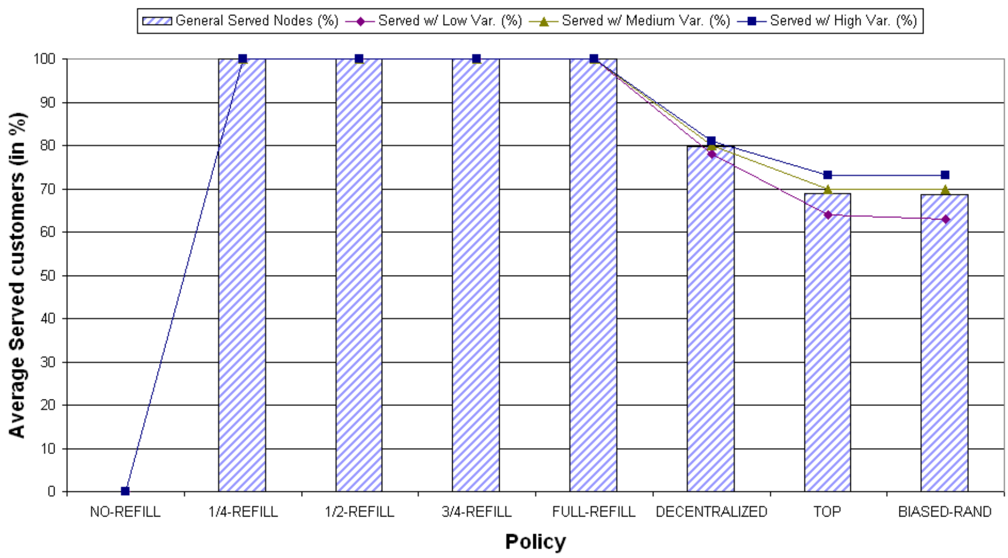


Figure 11.4: Comparison of the average number of served retailers for each refill policy ($\lambda = 0.01$).

routes close to 4. The average number of routes for each policy is depicted in Fig. 11.5. As the variance increase, the number of served nodes and the number of routes also rises in the Top and Biased-Randomized solutions.

Fig. 11.6 compares, for different configurations of the variance (uncertainty) level and the λ parameter, the average gaps between each policy and the best solution obtained with our methodology. Notice that our approach outperforms any other policy, either centralized or not. Also, observe that the quality of each policy seems to be quite robust against changes in the variance level as well as against changes in the λ parameter. Finally, notice that the decentralized policy —each retailer minimizing its inventory costs— can outperform other centralized (but more 'rigid') policies.

Finally, Fig. 11.7 illustrates four different solutions obtained with the four different refill policies proposed in our algorithm ('full', 'decentralized', 'top' and 'biased-randomization') for the B-n35-k5 instance. Squares ($\square$) represent customers receiving a full-refill. Diamonds ($\diamond$) show customers receiving a $\frac{3}{4}$-refill. Triangles ($\triangle$) represent customers receiving a $\frac{1}{2}$-refill. Circles ($\circ$) represent customers receiving a $\frac{1}{4}$-refill. Finally, stars ($*$) represent non-served customers. The first routing planning shows the worst case scenario using a 'full' refill policy for all nodes. The second solution proposes the application of a decentralized policy where the inventory cost is

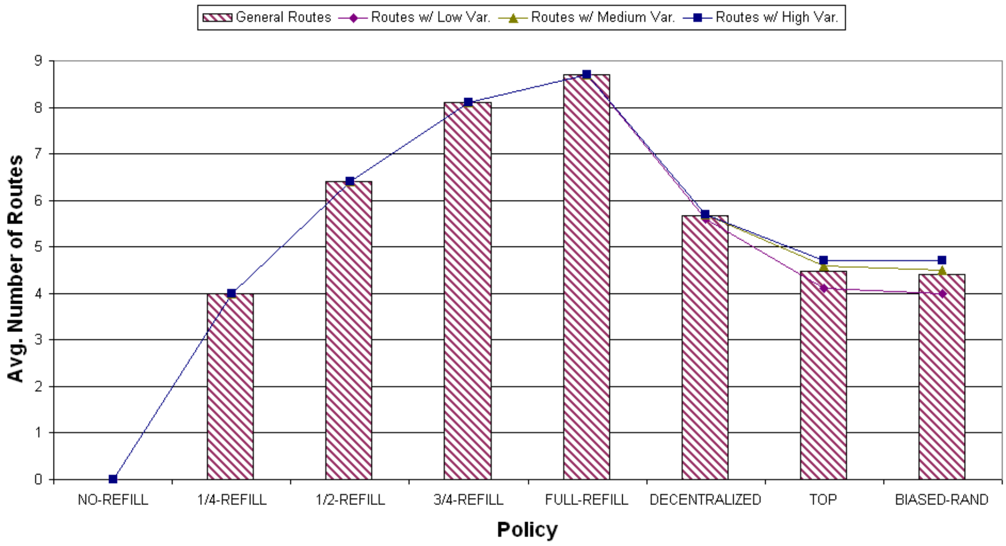


Figure 11.5: Comparison of the average number of routes for each refill policy ($\lambda = 0.01$).

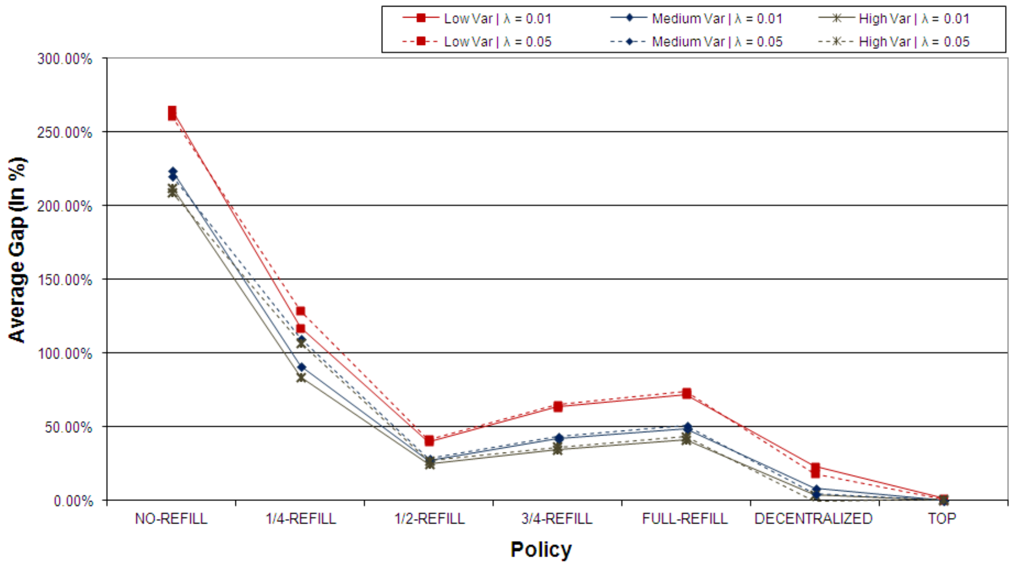


Figure 11.6: Average gaps between each proposed policy and our biased-randomized solution.

the only variable considered. On this, some nodes are non-served because this is the cheapest inventory option for them. The top policy solution proposes to visit some nodes with different inventory refill strategies and a routing configuration, while the asymmetrically-randomized policy applies some other inventory policies to some nodes. Thus the subset of served nodes between the top and biased-randomized solutions are distinct. Both solutions propose two alternative compositions thanks to the application of distinct refill policies to each node. The asymmetrically-randomized policy allows to find better and balanced configurations with individual policies for each node.

11.6 Chapter Conclusions

In this chapter, we have reviewed the second application example of Simheuristics. The IRPSD is a challenging research area because it introduces random behaviour into a problem combining two steps of supply chain management, inventory control and distribution planning. The proposed approach integrates Monte-Carlo simulation into different key phases of a heuristic approach. By doing so, it allows solving both the routing and the inventory problems in an integrated way. One of the main contributions of our methodology is that it can consider personalized refill policies for each retailer center, which contributes to significantly reduce total costs over other approaches using standard refill policies. Another important contribution is that our approach can be used with any probability distribution, which means that positive demands in retail centers are not assumed to follow a normal distribution —which is an unrealistic assumption usually employed in the existing literature. A set of benchmarks for the IRPSD were developed and a realistic expression to model inventory costs was also proposed. A complete set of tests has been performed to illustrate the methodology and analyze its efficiency as well as its potential benefits. So far, the uncertainty modelling feature of MCS mixed with a specific biased-randomized heuristic has created interesting approaches for the VRPSD and IRPSD. In the next block of chapters, we will study generic approaches for Rich VRPs and also propose a new methodology based on combining biased-randomized heuristics and constraint programming.

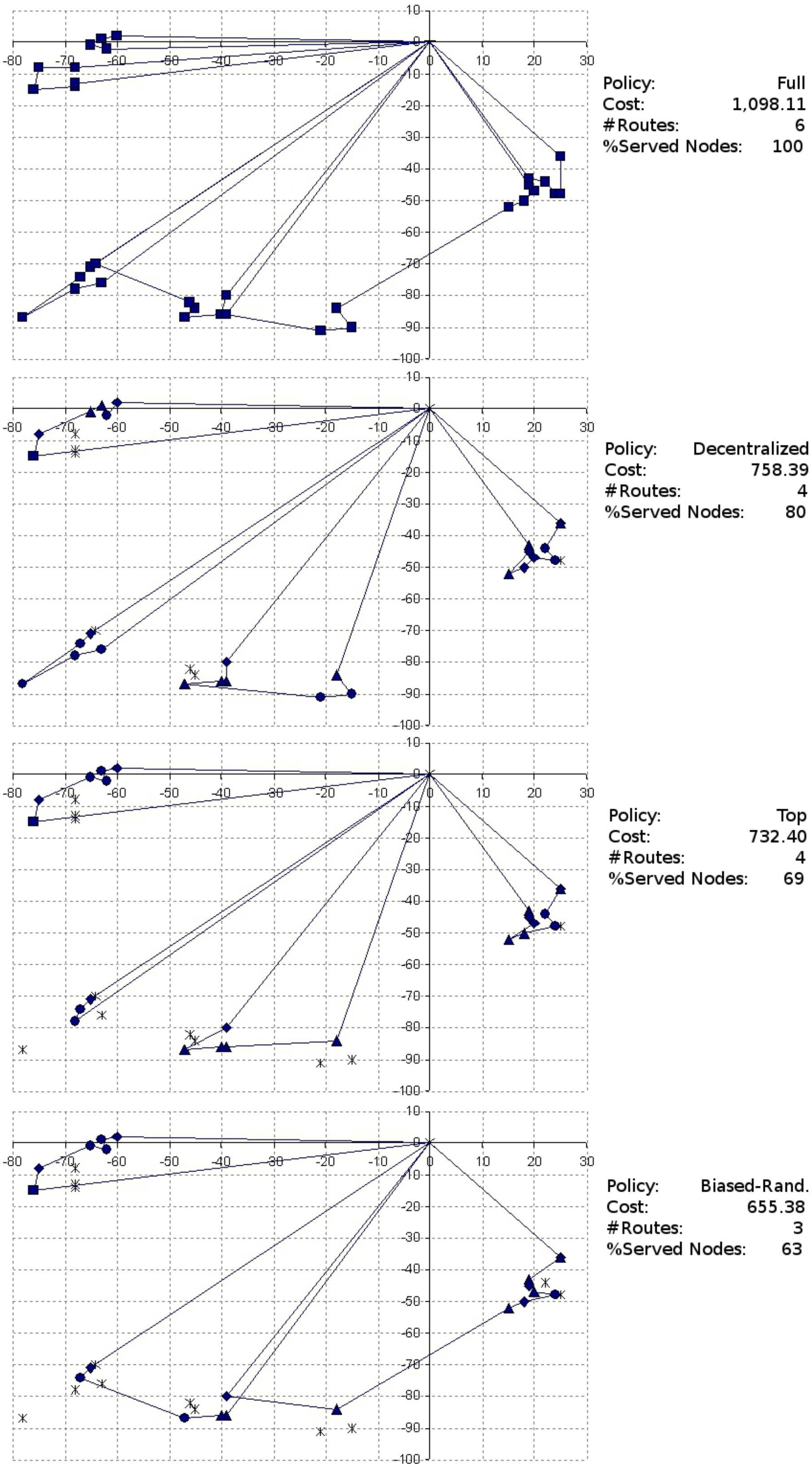


Figure 11.7: Solutions for the B-n35-k5 using a low variance level and different refill policies.

12

Generic Methodology for Rich VRPs

The design of software tools that can assist in the routing planning process is quite demanding (Drexl, 2012). The interest on this kind of support-decision tools has created a variety of algorithms (Partyka and Hall, 2012). In the Operation Research optimization field, few studies regarding the generic Rich VRP (RVRP) have been proposed. As we have commented before, the RVRP is a research line focused on the study of realistic routing planning problems. The challenge of RVRP is to consider several constraints at the same time where the main goal is to develop 'generic' techniques that can solve any given combination of constraints. In fact, commercial routing software usually offers a wide set of constraints but either some special adaptations for each client are done or the model does not exactly represent the real problem (Drexl, 2012). However in the academic literature, few studies have been suggested as generic approaches. For instance, Ropke and Pisinger (2006a,b) propose a heuristic based on LNS. Their approach is a unified heuristic with an adaptative layer. They are focused on the Backhauls VRP (BVRP) with time windows, pickup-and-delivery and multi-depots. They propose a model transformation of the BVRP to solve the simultaneous pickup-and-delivery. Nine data sets are used to test several configurations of the proposed heuristic, where more than 50% of the best known solutions for those instances are improved. Later, the same authors (Pisinger and Ropke, 2007) developed an Adaptative Large Neighbourhood Search (ALNS) framework for addressing the capacitated, time windows, multi-depot, split-deliveries and open routes constraints. They use several sets of instances with up to 1000 customers, and improve 183 best known solutions out of 486 benchmark tests. Other authors are also focused on the solution of Real VRPs. Likely Hasle and Kloster (2007); Hasle et al. (2005) present a generic approach centered on its modelling flexibility for addressing several routing constraints. The authors present a generic solver based on an unified algorithmic approach which is a combined operation of Local Search (LS) and Metaheuristics (Variable Neighbourhood Descent, VND; and Iterated Local Search, ILS). An initial solution is generated using the parallel version of CWS, then other methods are applied. They address the capacitated constraint, the distance limitation, the pickup-and-delivery, the fleet size and mix problem as well as the time windows. They present the possibility to extend it for multi-depot and site-dependent problems. Some authors promote the extension properties of optimization models to solve other problems. Irnich (2008) takes advantage of strong modelling capabilities and proposes an Unified Modelling and Heuristic Solution framework. The author highlights the potential of $k - edge$ exchange neighbourhoods. This approach is intended to support efficient local search procedures for addressing all standard types of VRPs. The author proposes to integrate the efficient search blocks into different metaheuristics. Some promising results are presented for VRPTW and MDVRPTW combining a VNS with LNS strategies.

Other highlighted generic Rich solvers have been proposed in the literature. First, Cordeau and Laporte (2003); Cordeau and Maischberger (2012); Cordeau et al. (1997, 2001b, 2004) propose an Unified Tabu Search (TS) approach for VRPs with time windows, multi-period, multi-depot and site-dependent. Several real and theoretical benchmarks have been used to test the performance of this approach. Some ILS approaches are proposed by Hashimoto et al. (2006, 2008); Ibaraki et al. (2005, 2008). In fact, Subramanian (2012) proposes a combination among ILS, Mixed Integer Programming (MIP) and Set Partitioning (SP) aspects for solving some Rich VRP variants. Second, Baldacci and Mingozzi (2009); Baldacci et al. (2010, 2011a,b) introduce an exact solution framework based on Set Partitioning (IPSP) modelling for solving several individual types of VRPs. A Column-and-Cut Generation algorithm is combined with the use of valid inequalities into the SP formulation. Some experiments are done with classical instances related to the CVRP, the VRPTW, the PDPTW, all types of HVRP, the MDVRP, and the PVRP. The results outperform all the other exact methods published so far and also solve several previously unsolved test instances. Last, another

Papers Involved	Applied Method	Number of Addressed Constraints
(Vidal et al., 2012b, 2013)	UHGS	19
(Penna et al., 2013; Subramanian, 2012; Subramanian et al., 2012)	ILS-MIP-SP	14
(Baldacci and Mingozzi, 2009; Baldacci et al., 2010, 2011a,b)	Exact method-based on SP	14
(Irnich, 2008)	LS-based metaheuristics	16
(Pisinger and Ropke, 2007; Ropke and Pisinger, 2006a,b)	ALNS Heuristic	14
(Hasle and Kloster, 2007)	VND-ILS	12
(Hashimoto et al., 2006, 2008; Ibaraki et al., 2005, 2008)	ILS	9
(Cordeau and Laporte, 2003; Cordeau et al., 1997, 2001b, 2004)	Unified TS	11

Table 12.1: State-of-the-art of Rich VRP methods.

approach is presented in Vidal et al. (2012b). This consists of a Unified Hybrid Genetic Search (UHGS) for several types of Rich VRP. The Framework uses efficient generic local search and genetic operators. The authors present interesting computational results using 39 benchmarks over 26 different Rich VRP. Furthermore, the authors apply their method combined with diversity management mechanisms to different large scale instances of Rich Time-constrained VRPs (Vidal et al., 2013). The proposed framework outperforms all current state-of-the-art approaches. The approach is addressed to any combination of periodic, multi-depot, site-dependent, and duration-constrained VRP with time windows. The used instances involve up to 1000 customers. The representation of giant-tour solution and local search of Prins (2004) has proven its efficiency in several other studies (Labadi et al., 2008; Ngueveu et al., 2010; Prins, 2009). Table 12.1 shows the summary of proposed approaches for addressing several VRP variants with the same logic core. The number of addressed constraints is derived from Table 4.2 presented in chapter 4.

In this chapter, we focus on the development of a general-purpose methodology for solving several variants of VRPs. The real-world routing planning demands for generic tools to be adapted to any problem without a great effort in the process. After defining a set of tailored solutions for different VRPs, we propose the combination of randomized heuristics and constraint programming as a flexible technique for addressing combinatorial optimization problems like the VRP. Some previous studies are discussed on the next section, and the design of a new generic methodology based in heuristics and CP is detailed.

12.1 Background

Several separated approaches have been used to solve specific VRP variants (Laporte et al., 2000; Toth and Vigo, 2001). However, a given methodology could be extended to solve other problems types. Regarding the current approaches, we have found that Constraint Programming is often used for solving routing problems. Some of the first studies can be found in De-Backer et al. (1997); Pesant et al. (1997, 1998); Shaw (1998). CP is a paradigm able to represent and solve several combinatorial problems (F. Rossi and Walsh, 2006). The main advantage of this approach is centered on its flexibility for addressing hard combinatorial problems. Depending on the enterprise technological environment, the main advantage of this approach is centered on the next features: fast coding development, easy maintenance and global efficient execution performance.

In general CP applications, the problems are represented using three components forming a 'model': *variables*, their associated natural *domains*, and finally the *constraints* relating them. *Constraints* represent logical relations among several *unknowns* (or *variables*), where each takes a value from the allowed *domain* of accepted values. *Domains* can be a range defined by minimum and maximum bounds or a discrete list of numbers. These problems, defined by *variables*, *domains* and *constraints* are known as Constraint Satisfaction Problems (CSPs) and are related to constraint propagation solving techniques. This feature makes it a very useful tool for modelling decision-making problems. Particularly, this natural representation helps to develop short and simple techniques easily to be adapted for changing addressed problems. For finding the best representation of a problem in CP, several models can be tested in a fast way by the programmer. The CP core is embedded in programming languages, such as Prolog. In that case, it is known as Constraint Logic Programming (CLP). Also it can be integrated in classical imperative languages like C/C+ and Java. All CLP languages join two basic elements: a) *logic* to define a set of possibilities to be explored using simple search methods as backtracking (incrementally find candidates as solution, where a candidate is abandoned as soon as it determines that it does not converge to a valid 'answer'); and b) *constraints* to simplify the search by eliminating non-desirable alternatives in advance by the use of consistency techniques.

Thus, CP combines reasoning and search; the proposed constraints are used to restrict and guide the search during the exploration of the solution space. Since CP gives a high importance to constraints (requirements) for solving problems, it is can be used to validate the satisfaction of all given constraints for a set of values (i.e. given built solutions).

For more details, a complete CP formulation for the VRP is presented in Guimarans (2012) which is based on Kilby and Shaw (2006). This formulation has been implemented into a 'CP-RVRP Library' (Riera et al., 2009) in order to overcome some limitations of the formulation and the emerging hybrid methodologies, such as the one we propose on this dissertation. It should be noticed that this CP formulation may be considered as a first step on the implementation of the CP-RVRP Library, able to cope with rich VRP variants and flexible enough to accept new constraints based on real applications. This library was first introduced by Riera et al. (2009).

The creation of hybrid methodologies based on CP for solving VRP variants has been tried in literature. The complementary effect of CP has been structured in studies like Backer et al. (2000); Kilby and Shaw (2006); Kilby et al. (2000), as we also propose. However, here we want to highlight that our approach can be applied to a wide range of routing combinatorial problems with few adaptation steps.

The VRPTW is the most studied VRP variant in the literature. Several papers propose to use CP techniques to solve the VRPTW, like Bent and Van-Hentenryck (2004a, 2006); Rousseau et al. (2002). In fact, Guimarans (2012) has also addressed the VRPTW using hybrid methodologies based on a 'CP Library'. Originally, this library was implemented using ECLiPSe (Apt and Wallace, 2006) and has considered the validation of capacity, length route limit, and time windows constraints. However, the library has been evolved in order to include more realistic constraints.

We have studied a large set of optimization routing papers and its considered restrictions. More than 30 involved routing conditions were found. We have structured and classified them for defining an unique point of comparison and framework. Then a summary of remarkable routing constraints is presented in Table 4.3 in chapter 4, while Table 12.2 highlights already implemented constraints in the 'CP-RVRP Library'. These restrictions were obtained from Table 4.2 —also in chapter 4— which shows the detailed relationship of each studied paper and the defined restrictions. For solving several routing problems, the key element of our approach is the combined interaction of modeling and validation centralization for a better maintenance of new constraints, as well as a randomized generation of solutions with $well - known$ heuristics. Therefore our proposal is based on the use of the commented validation CP library and biased-randomized classical heuristics.

12.2 Applying a CP Validation approach

For the Rich VRP, we propose a generic hybrid methodology based on the joined work of a randomized heuristic and the validation task of CP. As we discussed in previous chapters, the development of heuristics is wide popular in the VRP research community. In fact, there is a set of known classical heuristics (Golden et al., 2008; Laporte et al., 2000; Toth and Vigo, 2001). In general, once a heuristic is proposed then new adaptations or combinations with other methods emerge. Therefore we propose to extend lifetime of the heuristic with a randomization of its inner decision steps, and then combine it with a CP validator. In fact, the heuristic could work together with local search methods for improving the solution values.

The key aspect of our approach is to use the CP-RVRP library as a black box for evaluating complete or partial solutions generated by the selected heuristic. The full evaluation of CP will determine which of the generated solutions by the heuristic fulfil all the constraints. This may happen because the heuristic solution construction is based only in a partial set of constraints —e.g., CWS is based on vehicle capacity; I3 is based on both capacity and time windows—, while CP contains the whole model, validating then all the problem aspects. The biased-randomized process creates a promising set of solutions, and then CP checks the satisfaction of all desired restrictions. Notice that a specification for the communication is required in order to properly exchange information between the heuristic and the validator.

The challenge of this methodology is to find a balance point between: (a) a tailored biased-randomized heuristic (with or without the help of local search methods), and (b) the use of CP as a solutions validator. So as a starting point, we propose just to validate the solutions generated by a randomized classical heuristic for a specific combinatorial optimization problem. The biased-randomized heuristic helps to perform a diversified exploration of the solution space while the CP is focused only on the validation. So the integration point is important to determine the useful feedback the solutions generation process requires. From this, we can generate a good number of promising alternative

Code/Id	Constraint Description	Implemented on CP Library
CP	Multi-Products	
CD	Multi-Dimensional capacity	√
C	Vehicle Capacity	√
FO	Homogeneous Fleet of Vehicles	√
FE	Heterogeneous Fleet of Vehicles	√
VU	Unfixed Fleet of Vehicles	√
VF	Fixed Fleet of Vehicles	√
FC	Fixed Cost per Vehicle	√
VC	Variable Cost of Vehicle	√
MT	Multi-Trips	
DS	Vehicle Site Dependent	√
DR	Vehicle Road Dependent	
L	Duration Constraints/Lenght	√
D	Driver Shifts/Working Regulations	√
BR	Balanced Routes	√
CS	Symmetric Cost Matrix	√
CA	Asymmetric Cost Matrix	√
IR	Intra-route replenishments	
TD	Time Dependent/Dynamic/Stochastic times	
S	Stochastic Demands/Dynamic	
WT	Time Windows	√
WM	Multiple Time Windows	√
PD	Pick-up & Delivery	√
SP	Simultaneous Pick-up & Delivery	
B	Backhauls	√
MV	Multiple Visits/Splitted deliveries	
MP	Multi-Period/Periodic	
I	Inventory Levels Controls	
CC	Customer Capacity	
MD	Multi-Depot	
WD	Time Windows for the Depot	√
O	Different end locations/Open Routes	√
DA	Different start and end locations	√
DD	Departure from different locations	√
PC	Precedence constraints	√
MO	Multi-Objectives	√

Table 12.2: Rich VRP restrictions Implemented on CP library so far.

routing plans. We may also keep a memory of valid routes, and then find out a general plans to cover all customers, then we can build several routing solutions from the combinations of previously generated routes.

The main advantage of this methodology is that it avoids the waste of large developing of tailored evaluation methods. As a matter of fact, if a new constraint appears in the problem, CP allows a fast inclusion in the model and the methodology without needing to modify the search algorithm. Since generation of unfeasible solutions is common on VRP techniques, we prefer to invest that time exploring the solutions space in a natural and 'promising' way. Instead of applying complementary methods to evaluate solutions, CP —with an extremely fast computation— guarantees an immediate validation of given solutions. Furthermore, implemented components —both heuristics and CP restrictions— can be progressively built for addressing different types of routing problems. Depending on the type of routing problem, a constructive criteria will be used —e.g., CWS for only vehicle capacity, I3 heuristic for time windows, etc. In Fig. 12.1, the overview of the structure of the methodology is depicted. A heuristic is taken from a database of biased-randomized heuristics. Then a multi-start-like process is executed, where a CP integration is done. The integration can be focused on complete solutions at the end of the process (point 'B' in Fig. 12.1) or could be done inside of the generation process where partial solutions are handled (point 'A' in Fig. 12.1). Routing solutions are instantiations of the variables of a CP model. This model is built using a library with all the necessary routing constraints to be evaluated in each route. For solving any VRP variant, the library should include a large set of represented rules. The response of the CP solver is used in the solutions generation process in order to continue with it or start a new construction iteration. Notice that including the checking in point 'A' is much more efficient than in point 'B'. In Fig. 12.2, a solution space inspired in a tree-search —from CP operation— is depicted in order to give an intuitive view/approximation to 'early' feedbacks using both points. There are remarkable savings in the solution space exploration using integration point 'A', but these are out of the scope of this work.

Thus, it is possible to identify the following steps when creating a generic routing algorithm by using the CP-RVRP library-based methodology:

1. Given a routing problem, select a biased-randomized heuristic and include it

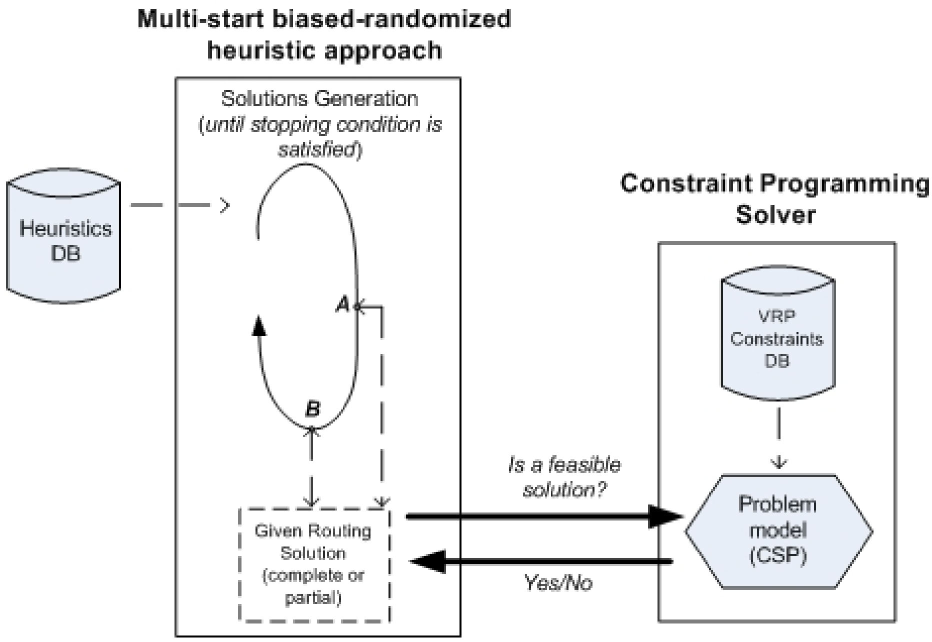


Figure 12.1: Relation of basic components in proposed hybrid methodology.

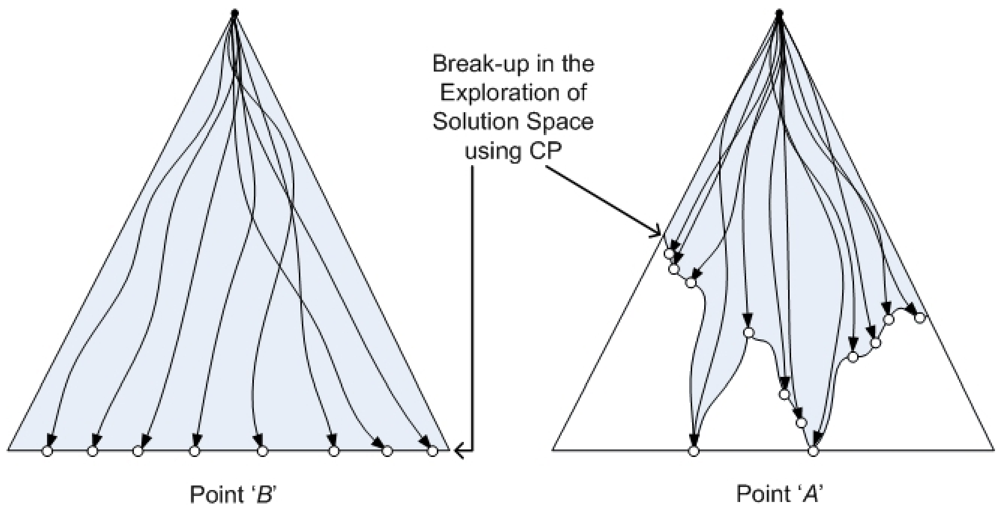


Figure 12.2: Exploration of solution space using commented points 'A' and 'B'.

inside of a multi-start-like approach for generating a diversified set of solutions in a fast iterated process.

2. Once the base heuristic has been selected, the most appropriate validation point of complete or partial solutions must be defined inside the optimization procedure.

The simplest way is to allocate it at the end of the solution building process, but it is the less efficient. Thus, an internal validation point of partial solutions also helps the generation process in a more efficient way with a proper feedback of solutions feasibility.

3. The constraints considered in the routing problem are included (if they existed previously) or implemented (if they are new) from/in the CP-RVRP Library. Therefore, a CP model is built.

4. After the generation of a total or partial solution, the CP validation results help the heuristic in the search process.

12.3 Benefits

Regarding the diversified set of approaches for generic Rich VRPs, the main benefits of combining randomized heuristics with CP techniques over other related approaches are:

- The idea can be applied to a wide set of combinatorial problems thanks to the versatile role of randomized heuristics combined with CP modelling power. Since it is focused on this CP modelling power, any combinatorial problems can be addressed.

- Given a specific combinatorial problem, the selection and randomization of an adequate heuristic can be executed in a few steps (see chapter 5). On the other hand, the validation model is built using the CP library for the corresponding problem.

- A set of randomized classical heuristics can be saved in a database. This allows to explore the solution space in different promising ways. The selection of a proper heuristic depends on the nature of the problem and the inner heuristic construction constraints —e.g., CWS is not useful for addressing VRP with time constraints, but I3 heuristic is.

- Depending on the integration level of CP within the heuristic, the proper feedback can save a remarkable number of computational steps. It can validate complete or partial solutions in order to identify the most promising solutions in the construction process.

- A distributed computation deployment of the methodology can be applied into network services. This may be useful for improving times and maintenance. The general CP validation process and heuristics execution can be located in different agents or computers.

12.4 Chapter Conclusions

In this chapter, a biased-randomized classical heuristics with Constraint Programming is proposed in order to solve several variants of VRPs with few (or no) adaptation steps. The key core of the approach is focused on the combination of promising solutions generation with biased randomization of classical heuristics and the flexibility of constraint programming techniques. The integration of these two methods can produce useful feedback in different points of the algorithm. Additionally, a set of implemented randomized heuristics can be stored in order to be used with the appropriate problems' instances.

In the next chapter, some example applications are presented in order to test the performance of the proposed hybrid approach. However, these are preliminary tests since our purpose is not to computationally compare with tailored methods because, as expected, they get better results quality by minimizing modelling flexibility. Therefore, the main objective is to produce generic solutions to some routing problems just to illustrate the use of proposed methodology. For this, we apply this methodology to two deterministic variants: DCVRP and HVRP (see Fig. 12.3).

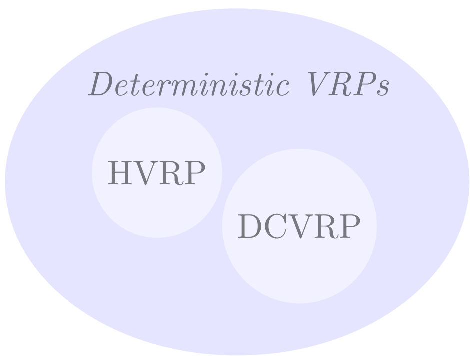


Figure 12.3: VRP Problems studied on this dissertation using randomized heuristics and CP.

13

Applying a Generic Methodology

The benefits of the general methodology proposed in the previous chapter have been tested with different VRPs in order to measure its performance. We have applied it in some basic VRP variants —like the DCVRP and HVRP instances— in order to preliminary test its usefulness. As explained in the previous chapter, first, we have chosen appropriate constructive heuristics —e.g. Clarke-and-Wright Savings. In general, classical heuristics are designed to address a specific combinatorial optimization problem. However, some may be used for multiple types of problems when embedded in the proposed methodology. Thus, we will use CWS over a wide set of benchmarks. For testing the proposed methodology, instances for each individual routing problem have been used. The idea is to appreciate the complementary potential of constraint programming to model and check any possible routing constraints at the same time that no extra development is needed. For this, we will use the CP-RVRP library introduced in the previous chapter.

13.1 From CVRP to DCVRP

As a first experiment, we have used a biased-randomized version of the CWS heuristic (Clarke and Wright, 1964) proposed by Juan et al. (2010) which targets the CVRP. This heuristic is based on the construction of routes using a savings concept. During the construction process, the capacity restriction is the only one validated —as it is the main target of the CVRP. Then we have integrated the previously commented CP-RVRP Library to the algorithm in order to check the solutions generated. Thus, any new extra constraints can be included with no extra effort. In the CVRP, the route distance is often validated (Distance-Constrained VRP, DCVRP). So we can use the algorithm to generate random solutions with different configurations, and validate the satisfaction of this extra restriction. After building a complete routing solution, we validate the satisfaction of all the constraints (corresponding to integration point 'B' in the overview of this methodology from the previous chapter).

13.1.1 Computational Results

The methodology has been implemented as a Java application. We run instances in an Intel Xeon E5603 at 1.60 Ghz and 8 GB RAM. For preliminary experiments, we use a test-bed of eight 'big' instances with a number of customers between 200 and 480, proposed by Golden et al. (1998). Each has a specific maximum distance limit. All instances are represented by Euclidian distances. First, we have tested the basic case, that is, checking all the solutions generated by the tailored procedure and no additional constraints. Thus, the heuristic procedure considers only vehicle capacity for each route. As expected, we got a 100% of valid solutions. So far, this helps to validate the proper connection and operations of components. Second, we have run the same experiment, but activating the maximum distance limit validation only in CP.

Table 13.1 presents the number of generated solutions and the number of positively validated by the algorithm after 60 seconds running each instance. The idea is to present the number of solutions generated by a constructive procedure and then see how many solutions get invalid by adding just one constraint. The first three columns are from the tailored CVRP heuristic, which produces CVRP feasible solutions. The second results (last three columns) are related to the heuristic generation and the CP validation. In these, the ratio of feasibility descends remarkably when a new routing condition is included. This first experiment also helps to see how sensitive is a tailored approach when the addressed problem gets just a little more constrained.

13.2 HVRP

As seen in Chapter 6, we have proposed an algorithm based on a biased-randomization of CWS combined with a vehicle assignment originally proposed by Prins (2002). The process starts from a dummy solution (the most expensive possible), then these basic

	Tailored CVRP Heuristic			Generic CVRP-DCVRP Heuristic		
Instance	Generated Solutions	Valid Solutions	Feasibility Ratio	Generated Solutions	Valid Solutions	Feasibility Ratio
$Kelly_1$	2870	2870	100.00%	2991	0	0.00%
$Kelly_2$	2037	2037	100.00%	2378	0	0.00%
$Kelly_3$	1456	1456	100.00%	1501	0	0.00%
$Kelly_4$	1120	1120	100.00%	1133	0	0.00%
$Kelly_5$	3467	3467	100.00%	3624	12	0.34%
$Kelly_6$	1779	1779	100.00%	2549	0	0.00%
$Kelly_7$	1902	1902	100.00%	2071	0	0.00%
$Kelly_8$	1327	1327	100.00%	1275	0	0.00%

Table 13.1: Results of CVRP-DCVRP algorithms after checking generated solutions with CP.

routes are merged using the savings list. For the HVRP, the vehicle assignment process consists in, first, sorting all given vehicles and routes demands —created so far— in a decreasing way. Then, vehicles are assigned to each route top-down. Given the case, if there are more clients than available vehicles, some fictitious vehicles will be needed. The heterogeneous fleet is a routing feature which appears in many real cases —as we have studied in two real cases in this thesis. So as the previous experiment, we have integrated the CP-RVRP library with a tailored algorithm in order to measure the number of feasible solutions. In this time, it is just focused on the Heterogeneous fixed fleet VRP (HVRP) —studied before.

13.2.1 Computational Results

As for the previous case, the algorithm was implemented as a Java application. We run instances in an Intel Xeon E5603 at 1.60 Ghz and 8 GB RAM. For testing HVRP approaches, a well-known dataset, wide used by the research community, proposed by Golden et al. (1984) and later modified by Taillard (1999) has been used. The number of customers in these instances, originally proposed by Christofides and Eilon (1969), is between 50 and 100. All instances are represented with Euclidian distances.

Table 13.2 presents the number of generated and validated solutions using the algorithm after 60 seconds of search for each instance. The first values are from the

Instance	Generated Solutions	Valid Solutions	Feasibility Ratio
GT_{13}	5154	7	0.14%
GT_{14}	4569	4569	100.00%
GT_{15}	5393	1934	35.86%
GT_{16}	5460	2675	48.99%
GT_{17}	5055	4637	91.73%
GT_{18}	4940	559	11.32%
GT_{19}	4837	4837	100.00%
GT_{20}	4623	3825	82.74%
Average	5003.88	2880.38	58.85%

Table 13.2: Results of Tailored HVRP algorithm after checking generated solutions with CP.

tailored HVRP heuristic. Notice that this methodology produces a certain number of unfeasible solutions. This is because there are some instances where the total demand to be delivered and the total vehicle capacity are very close. Thus CP detects if the solutions found fulfils or not the complete model. Of course, as expected, to complete the methodology some research should be made on how to move this checkings into the search loop to take advantage of the infeasibility detection before completing solutions. But this falls out of the scope of this thesis.

13.3 Future lines

After analysing these first results, we have detected that the use of CP support should be included in the inner steps of the constructive heuristic. At the moment, we evaluate only complete solutions, but it is interesting to study how a complete method as CP could help heuristics during the construction phase. This is something quite complex because if CP is included in the construction loop, it must check partial solutions, which requires some research to be done in the appropriate way. Once the integration is done, synergies will increase and the time spent creating unfeasible solutions should decrease dramatically. Until now we have been using only a few constraints from the library but many other VRPs can be modelled and solved.

This type of generic methodology can be applied to a wide set of optimization problems. The combination of different types of constraints represent a challenging target for the OR area. Without leaving the routing area, there are a number of tradeoffs that could be studied, like, for instance, the environmental impact of routing. Thus, 'green' aspects can be included in the VRP, as (Erdoğan and Miller-Hooks, 2012; McKinnon et al., 2012; Sbihi and Eglese, 2007) suggest. Likely there are some emerging problem definitions like the Pollution Routing Problem (Bektas and Laporte, 2011; Demir et al., 2012a,b). The particular feature of this VRP extension is that the objective function include an extra variable for the greenhouse emissions. In addition to the Green and Pollution VRP, there are some other variants focused in green logistics for reducing CO_2 emissions (Demir et al., 2011; Fagerholt et al., 2009; Figliozzi, 2011; Jabali et al., 2012).

13.4 Chapter Conclusions

In this chapter, we have proposed to combine randomized classical heuristics with constraint programming in order to solve several Rich VRPs. In real-life routing enterprises, there is a wide necessity of creating generic tools that allow to address any combination of constraints. There are some few works on this research line. For this, we have introductory proposed and tested a new methodology. Finally, some conclusions and comments related to this dissertation will be presented in the next chapter.

14

Conclusions and Future Work

This thesis dealt with several approaches for the Rich VRPs. These approaches are focused on three main axes: biased-randomized heuristics, integration of randomized heuristics with simulation, and also the combination of biased-randomized heuristics with constraint programming. One of the sub-objectives was to present the state-of-the-art of each VRP family addressed here. Some of them can be classified as deterministic contexts —i.e., HVRP, HVRP-V, HVRPM, AVRP, HAVRP, VRPTW, and DCVRP— while others can be considered as stochastic natures —i.e., VRPSD and IRPSD. In fact, few studies have addressed both types of problems. Thus an extensive literature review was carried out, focusing on describing the evolution of main contributions of previous works. By the substantial number of publications made on each VRP family, this optimization line is indeed an area of intense and continuous research in the fields of operational research and computer science.

On the deterministic context, two Rich VRPs inspired on real-life distribution companies were addressed with biased-randomized heuristics. First, an enterprise with almost 400 stores in Spain proposes to solve both cases of HVRP and HVRPM. Second, an intra-urban distribution company of around 50 customers in Barcelona propose an HAVRP with some extra constraints —like optionally open and/or balanced routes. On those case studies, we obtained a remarkable improvement on their routing planning which represent savings on their logistic activities. Also some promising results were generated on experiments using theoretical instances of HVRP, HVRP-V, AVRP, VRPTW, and also DCVRP. In fact, we have addressed a promising emerging family as it is the Heterogeneous Asymmetric VRP. In general, the biased-randomization of heuristics offers a natural understanding and implementation way to generate a diversified set of solutions for the decision-making process.

On the stochastic side, two popular problems (VRPSD and IRPSD) were addressed with the combination of biased-randomized heuristics and Monte-Carlo simulation (so called *Simheuristics*). Several routing solutions were created under specific assumptions of the random behavior of demands. First, for the VRPSD, a safety stock in trucks is proposed to face the stochasticity on customer demands on some simulation executions. In this study, the potential of parallel and distributed computing is highlighted for speeding up the computations without to represent a remarkable invest for enterprises. Second, for the single period IRPSD, an integrated approach is proposed to relate the inventory and routing costs on retail centers in a centralized vendor-managed inventory context. Also in one of the addressed real-life case studies, simulation techniques were used to generate information of service times in customers from incomplete data.

One step further have been done in order to design a generic approach to be applied for several variants of Rich VRPs. For this, a complementary library based on constraint programming (CP) was integrated to some biased randomized algorithms for addressing the DCVRP and HVRP. Some preliminary results were obtained to show the usefulness of this promising approach. However, a major integration of the CP inside of the heuristic will be required and then more experiments should be executed to prove its true potential.

14.1 Future Research

As we have appreciate in previous chapters, the biased-randomization of heuristics combined with other techniques (e.g., simulation, parallel and distributed computing, constraint programming) have been useful for addressing a large set of Rich VRPs. The research community is proposing even more hybrid approaches as a relevant direction (Doerner and Schmid, 2010). In fact, the randomization of heuristics can easily harness DPCS approaches for a better performance. This could be quite interesting for SME which can not afford specialized computer solutions. Some other works propose a decomposition or transformation of one proposed model problem into other equivalent in order to solve other Rich VRPs —see (Hasle and Kloster, 2007). This could be particularly useful on some cases where a general optimization model have been proposed to be transformed into other problem.

As for future work, the following lines of research are suggested: (i) since all proposed algorithms are based on a biased-randomized selection of elements inside of heuristics, other biased (non-symmetric) probabilistic distributions could be used to measure its performance and its impact on results (sensibility); (ii) due to the common nature of combinatorial optimization problems, the proposed algorithms could be applied to efficiently solve other problem scopes like arc routing (González-Martín et al., 2012), scheduling (Juan and Rabe, 2013; Montoya-Torres et al., 2012), flowshop (Juan et al., 2012b, 2013b), clustering (Muñoz-Villamizar et al., 2013) or green computing (Cabrera et al., 2013); (iii) the implementation and deployment of proposed methodologies in real enterprise environments in order to offer a day-to-day optimization routing planning —including more real constraints into it; (iv) explore other practical ways to apply some parallel and distributed techniques on proposed algorithms that allows to reuse the computing platform of an enterprise; (v) investigation of alternative forms of hybridization between heuristic and exact approaches for VRPs.

There are also some emerging research lines in the routing community that we could consider to adapt our approaches. The combination of routing and environmental aspects represent and promising and interesting trade-off to be studied. The ecological footprint and energy consumption are having an important place in national regulations impacting distribution planning (Ahn and Rakha, 2008; Dekker et al., 2012; Eglese and Black, 2010; Erdoğan and Miller-Hooks, 2012; Fagerholt et al., 2009; Srivastava, 2007). In (Erdoğan and Miller-Hooks, 2012; McKinnon et al., 2012; Sbihi and Eglese, 2007) some preliminary definitions and state-of-the-art for the so called “Green VRP”. While in (Bektas and Laporte, 2011; Demir et al., 2012a,b) the “Pollution VRP” is described in detail with some resolution approaches. Specific techniques are proposed in (Demir et al., 2012a; Erdoğan and Miller-Hooks, 2012; Jabali et al., 2012; Kuo, 2010; Lera-López et al., 2012). Some real applications are presented in (Bauer et al., 2009; Faulin et al., 2011; Figliozzi, 2011; Ubeda et al., 2011). Therefore there is still a long path for creating algorithm solutions for a broad green routing problem. Randomized and hybrid approaches offer a potential framework to address this type of problem.

15

Publications & Presentations derived from this Thesis

The generated publications so far related to this thesis are considered as part of the main contributions of this work. Thus in this chapter, we present the accepted publications, the $in - process - of - reviewing$ publications, some dissemination activities developed in last three years, and finally there are some extra contributions related to the objectives of this dissertation that it must be pointed out.

15.1 Publications

First, some partial parts of this thesis have been accepted for publication in the following articles in ISI-JCR or Elsevier-Scopus journals after a $peer - reviewing$ process:

- Juan, Faulin, Jorba, Cáceres-Cruz, and Marques (2013a): "Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands". Annals of Operations Research, 207: 43-65 (indexed in ISI SCI, 2012 IF = 1.029, Q2).

- Juan, Faulin, Cáceres-Cruz, Barrios, and Martínez (2014b): "A Successive Approximations Method for the Heterogeneous Vehicle Routing Problem: analyzing different fleet configurations". European J. of Industrial Engineering (indexed in ISI SCI, 2012 IF = 1.596, Q1).

- Grasas, Cáceres-Cruz, Lourenço, Juan, and Roca (2013): "Vehicle routing in a Spanish distribution company: Saving using a savings-based heuristic". OR Insight (indexed in UK ABS, Grade 1).

- Herrero, Rodríguez, Cáceres-Cruz, and Juan (2014): "Solving Vehicle Routing Problems with Asymmetric Costs and Heterogeneous Fleets". Int. J. of Advanced Operations Management (indexed in DBLP).

Second, there are some conference papers associated to ISI-WOS or Elsevier-Scopus journals which were accepted after a $peer - reviewing$ process:

- Cáceres-Cruz, Juan, Grasman, Bektas, and Faulin (2012b): "Combining Monte-Carlo Simulation with Heuristics for solving the Inventory Routing Problem with Stochastic Demands". In Proceedings of the Winter Simulation Conference (WSC), pp. 1–9. Berlin, Germany, December 9–12 (indexed in ISI Web of Science and Scopus, 2011 SJR = 0.372, Q2).

- Muñoz-Villamizar, Montoya-Torres, Juan, and Cáceres-Cruz (2013): "A Simulation-based Algorithm for the Integrated Location and Routing Problem in Urban Logistics". In Proceedings of the Winter Simulation Conference (WSC), Washington, USA, December 8–11 (indexed in ISI Web of Science and Scopus, 2011 SJR = 0.372, Q2).

- Cáceres-Cruz, Riera, Buil, Juan, and Herrero (2013): "Multi-start Approach for Solving an Asymmetric Heterogeneous Vehicle Routing Problem in a Real Urban Context". In Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems (ICORES), pp. 168–174, Barcelona, Spain. February 16–18 (indexed in Scopus, see Appendix Fig. 16.1).

Third, there is a research chapter book accepted after a $peer - reviewing$ process:

- Juan, Cáceres-Cruz, González-Martín, Riera, and Barrios (2014a): "Biased Randomization of Classical Heuristics". In: J. Wang (ed), Encyclopedia of Business Analytics and Optimization. IGI Global. USA.

Also there are other conference papers accepted after a $peer - reviewing$ process:

- Cáceres-Cruz, Grasas, Lourenço, Juan, and Roca (2012a): "Aplicación de un Algoritmo Randomizado a un Problema Real de Enrutamiento de Vehículos Heterogéneos". In Proceedings of the VIII Congreso Español sobre Metaheursticas, Algoritmos Evolutivos y Bioinspirados (MAEB), Albacete, Spain, February 8–10.
- Cáceres-Cruz, Riera, Juan, and Padrón (2013): "Hybrid Approach combining Insertion Heuristic and Biased Random Sampling for the Vehicle Routing Problem with Time Windows". In Proceedings of the IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB), Madrid, Spain, September 17–20.
- Cáceres-Cruz, Riera, Buil, and Juan (2013b): "Applying a Savings Algorithm for solving a Rich Vehicle Routing Problem in a Real Urban Context". In Proceedings of 5th International Conference on Applied Operational Research (ICAOR) - Lecture Notes in Management Science, vol. 5, pp. 84–92, July 29–31, Lisbon, Portugal.

Finally, other parts of this thesis have been submitted to a *peer−reviewing* process. For instance, the first paper in the list is in a second step of the process:

- Juan, Grasman, Cáceres-Cruz, and Bektas (?): "A Hybrid Algorithm for the Single-Period Stochastic Inventory Routing Problem with Stock-outs".
- Lourenço, Juan, Cáceres-Cruz, Grasas and Roca (?): "A Savings-based Randomized Heuristic for the Heterogeneous Fleet Multitrip Vehicle Routing Problem".
- García-García, Martínez-Juste and Cáceres-Cruz (?): "Using Genetic Algorithm-based Software on a Rich Vehicle Routing Problem: a Spanish Case Study".

15.2 Presentations

Some parts of this work have also been presented in several international Congress-Conferences-Workshops and published in the following activities:

- Juan, Faulín, Jorba, Cáceres-Cruz, and Marques (2011b), "A Simulation-based algorithm for solving the Vehicle Routing Problem with Stochastic Demands". In Proceedings of the 2011 ALIO/EURO Workshop, Porto, Portugal, May 4–6.

- González-Martín, Juan, Riera, and Cáceres-Cruz (2011), "A Hybrid Algorithm Combining Path Scanning and Biased Random Sampling for the Arc Routing Problem". In Proceedings of the 18th Knowledge Representation & Automated Reasoning Workshop (RCRA), Barcelona, Spain, July 17–18.

- Juan, Faulín, Cáceres-Cruz, and González-Martín (2011a), "Combining Randomized Heuristics, Monte Carlo Simulation and Parallel Computing to Solve the Stochastic Vehicle Routing Problem". Abstract in Proceedings of the International Conference on Optimization, Theory, Algorithms and Applications in Economics (OPT), Barcelona, Spain, October 24–28.

- Lourenço, Cáceres-Cruz, Grasas, Juan, and Roca (2012), "A Randomized Hybrid Algorithm based on Savings and Vehicle Assignment Policies for the Heterogeneous Vehicle Routing Problem". Abstract in Proceedings of the 1st EURO-VeRoLog Conference, Bologna, Italy, June 18–20.

- Cáceres-Cruz, Juan, Riera, and Lourenço (2012), "A Randomized Algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem". Abstract in Proceedings of 25th EURO Conference, Vilnius, Lithuania, July 8–11.

- Juan, Faulín, Agustín, and Cáceres-Cruz (2012a), "A Multi-Round Simulation Method which Analyzes Fleet Designs to Solve the Heterogeneous Vehicle Routing Problem". Abstract in Proceedings of the International Symposium on Combinatorial Optimization (CO), Oxford, UK, September 17–19.

- Cáceres-Cruz, Juan, Grasman, Bektas (2012), "A Hybrid Approach for the Inventory Routing Problem with Stochastic Demands". Abstract in Proceedings of the 2012 IN3-HAROSA International Workshop. Barcelona, Spain, June 13–15.

- Cáceres-Cruz, Juan, Riera, Lourenço (2012), "Hybrid Algorithms for solving the Rich VRP". Abstract in Proceedings of the 2012 IN3-HAROSA International Workshop for Junior Researchers, July 12–13. Barcelona, Spain.

- Cáceres-Cruz, Riera, Buil, Juan (2012), "Applying a Hybrid Approach to an Asymmetric Heterogeneous Vehicle Routing Problem". Abstract in Proceedings of the 2012 CYTED-HAROSA International Workshop. Valparaiso, Chile, November 12–13.

- Riera, Guimarans, Arias, Cáceres-Cruz, Juan (2012), "Solving the R^2VRP (Real Rich VRP)". Abstract in Proceedings of the 2012 CYTED-HAROSA International Workshop. Valparaiso, Chile, November 12–13.

- Cáceres-Cruz, Juan, Riera, Lourenço, Grasas, and Buil (2013a), "Rich and Real-life Vehicle Routing Problems: cases of study in Spain". Abstract in Proceedings of the 2nd EURO-VeRoLog Conference, Southampton, UK, July 7–10.

15.3 Other Contributions

During the thesis period, we have participated in several meetings in particular with three enterprises, as well as in some private sector conferences which allowed to gather the current situation of routing distribution in Spain. On this process, an exchange of information has took place between the responsible persons of routing in enterprises and the academic sector. All comments were oriented to point out the common concern to develop more efficient and generic tools. In fact, the academic sector could provide a remarkable assistance to design advanced solution techniques. Likely, some collaborations have been done with the next institutions for implement knowledge-transfer tools related with the used algorithms in this dissertation:

- "One Big Robot Company" for the creation of a web site routing game `http://www.onebigrobot.com/beta/uoc/viu_la_recerca/rutes/#`. This game illustrate the resolution power of advanced routing techniques. For this we used a simple version of CWS heuristic algorithm which minimizes the time taken by a fleet of vehicles with certain load capacity to serve a set of customers in a given area. The main idea of the algorithm is, from an initial very expensive solution, in small steps to improve it. In fact, this has been applied to solve some problems of SMEs in Spain –as we have appreciated on this thesis–, notably improving its logistics.

- "Pompeu Fabra University" for the creation of a web site `http://www.econ.upf.edu/~ramalhin/VRP-UPF/default.php` for public consumption of CWS heuristic algorithm. Dr. Helena R. Lourenço, as one its main promoter explains: "this will served to enterprise to show them how easy and effective could be to use advance techniques on their day-to-day routing planning".

Also there was a guidance in two UOC academic career projects during 2011: (a) that of Juan Ramon Pons called (in catalan) “Desenvolupament d’eines software per millorar la gestió d’inputs i outputs en problemes de Vehicle Routing i Scheduling”. This project search to integrate external geo-locational tools (like Google Maps) to advanced routing techniques. Also (b) that of León Monzón in the project “Optimization of SR-GCWS-CS algorithm using TSP process in petals of routes” which tries to improve the performance of the Cache-memory-local-search technique used on this dissertation. Additionally, the participation on the organization of several scientific events in order to create spaces where these routing optimization ideas could be discussed between experts and practicioners, like: 2011 IN3-HAROSA (Barcelona), 2012 IN3-HAROSA (Barcelona), 2012 IN3-HAROSA for Junior Researchers (Barcelona), 2013 ICSO-HAROSA (Barcelona).

References and Bibliography

D. Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004. 175

A. Agra, M. Christiansen, R. Figueiredo, L.M. Hvattum, M. Poss, and C. Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013. ISSN 0305-0548. doi: 10.1016/j.cor.2012.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0305054812002134`. Transport Scheduling. 122

K. Ahn and H. Rakha. The effects of route choice decisions on vehicle energy consumption and emissions. *Transportation Research Part D: Transport and Environment*, 13(3):151–167, 2008. ISSN 1361-9209. doi: 10.1016/j.trd.2008.01.005. URL `http://www.sciencedirect.com/science/article/pii/S1361920908000047`. 225

M. Alizadeh, H. Eskandari, S.M. Sajadifar, and C.D. Geiger. Analyzing a stochastic inventory system for deteriorating items with stochastic lead time using simulation modeling. In *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pages 1645–1657, 2011. doi: 10.1109/WSC.2011.6147881. 130

F. Alonso, M.J. Alvarez, and J.E. Beasley. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society*, 59(7): 963–976, 2007. 97

P. Amorim, S.N. Parragh, F. Sperandio, and B. Almada-Lobo. A rich vehicle routing problem dealing with perishable food: a case study. *TOP*, pages 1–20, 2012. ISSN 1134-5764. doi: 10.1007/s11750-012-0266-4. URL `http://dx.doi.org/10.1007/s11750-012-0266-4`. 33, 35, 39

D.P. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings Fifth IEEE/ACM International Workshop on Grid Computing, 2004*, pages 4–10. IEEE, 2004. 137, 138

S. Andradóttir. A review of simulation optimization techniques. In *Simulation Conference Proceedings, 1998. Winter*, volume 1, pages 151–158, 1998. doi: 10.1109/WSC.1998.744910. 131

E. Angelidis, D. Bohn, and O. Rose. A simulation-based optimization heuristic using self-organization for complex assembly lines. In *Proceedings of the Winter Simulation Conference*, WSC '12, pages 276:1–276:10. Winter Simulation Conference, 2012. URL `http://dl.acm.org/citation.cfm?id=2429759.2430131`. 130

S. Anily and A. Federgruen. One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36(1):92–114, 1990. 171

S. Anily and A. Federgruen. Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research*, 41(1):37–47, 1993. 171

K.R. Apt and M. Wallace. *Constraint logic programming using ECLiPSe*. Cambridge University Press, 2006. 209

REFERENCES AND BIBLIOGRAPHY

M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010. 136

P. Augerat, J.M. Belenguer, E. Benavent, A. Corberán, D. Naddef, and G. Rinaldi. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical report, Research Report 949-M, Université Joseph Fourier, Grenoble, France, 1995. 9, 84, 170, 182

F. Azadivar. Simulation optimization methodologies. In *Proceedings of the 31st Conference on Winter Cimulation: Simulation—a bridge to the future - Volume 1*, WSC 1999, pages 93–100, New York, NY, USA, 1999. ACM. ISBN 0-7803-5780-9. doi: 10.1145/324138.324168. URL `http://doi.acm.org/10.1145/324138.324168`. 129

N. Azi, M. Gendreau, and J.Y. Potvin. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. *European Journal of Operational Research*, 202(3):756–763, 2010a. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.06.034. URL `http://www.sciencedirect.com/science/article/pii/S0377221709004950`. 97

N. Azi, M. Gendreau, and J.Y. Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple trips. Technical report, Tech. rep., CIRRELT 2010-08, february 2010b. 97

B.D. Backer, V. Furnon, P. Shaw, P. Kilby, and P. Prosser. Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics*, 6:501–523, 2000. ISSN 1381-1231. doi: 10.1023/A:1009621410177. URL `http://dx.doi.org/10.1023/A%3A1009621410177`. 209

P. Badeau, F. Guertin, M. Gendreau, J.Y. Potvin, and E. Taillard. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109–122, 1997. ISSN 0968-090X. doi: 10.1016/S0968-090X(97)00005-3. URL `http://www.sciencedirect.com/science/article/pii/S0968090X97000053`. Parallel Computing in Transport Research. 122

P. Balaprakash, M. Birattari, T. Stützle, and M. Dorigo. Estimation-based metaheuristics for the probabilistic traveling salesman problem. *Computers & Operations Research*, 37(11):1939–1951, 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2009.12.005. URL `http://www.sciencedirect.com/science/article/pii/S030505480900330X`. Metaheuristics for Logistics and Vehicle Routing. 133

R. Baldacci and A. Mingozzi. A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120:347–380, 2009. ISSN 0025-5610. doi: 10.1007/s10107-008-0218-9. URL `http://dx.doi.org/10.1007/s10107-008-0218-9`. 31, 35, 39, 57, 206, 207

R. Baldacci, M. Battarra, and D. Vigo. Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 3–27. Springer US, 2008. ISBN 978-0-387-77777-1. doi: 10.1007/978-0-387-77778-8_1. URL `http://dx.doi.org/10.1007/978-0-387-77778-8_1`. 12, 56, 58, 110

R. Baldacci, M. Battarra, and D. Vigo. Valid inequalities for the fleet size and mix vehicle routing problem with fixed costs. *Networks*, 54(4):178–189, 2009. 32

R. Baldacci, E. Bartolini, A. Mingozzi, and R. Roberti. An exact solution framework for a broad class of vehicle routing problems. *Computational Management Science*, 7:229–268, 2010. ISSN 1619-697X. doi: 10.1007/s10287-009-0118-3. URL `http://dx.doi.org/10.1007/s10287-009-0118-3`. 31, 35, 39, 206, 207

R. Baldacci, E. Bartolini, A. Mingozzi, and A. Valletta. An exact algorithm for the period routing problem. *Oper. Res.*, 59(1):228–241, jan 2011a. ISSN 0030-364X. doi: 10.1287/opre.1100.0875. URL `http://dx.doi.org/10.1287/opre.1100.0875`. 31, 35, 39, 206, 207

R. Baldacci, A. Mingozzi, and R. Wolfler-Calvo. An exact method for the capacitated location-routing problem. *Oper. Res.*, 59(5):1284–1296, sep 2011b. ISSN 0030-364X. doi: 10.1287/opre.1110.0989. URL `http://dx.doi.org/10.1287/opre.1110.0989`. 31, 35, 39, 206, 207

J. Barceló, H. Grzybowska, and S. Pardo. Vehicle routing and scheduling models, simulation and city logistics. In V. Zeimpekis, C.D. Tarantilis, G.M. Giaglis, and I. Minis, editors, *Dynamic Fleet Management*, volume 38 of *Operations Research/Computer Science Interfaces Series*, pages 163–195. Springer US, 2007. ISBN 978-0-387-71721-0. doi: 10.1007/978-0-387-71722-7_8. URL `http://dx.doi.org/10.1007/978-0-387-71722-7_8`. 130

J.F. Bard, L. Huang, P. Jaillet, and M. Dror. A decomposition approach to the inventory routing problem with satellite facilities. *Transportation science*, 32(2):189–203, 1998. 173

D. Barnes-Schuster and Y. Bassok. Direct shipping and the dynamic single-depot/multi-retailer inventory system. *European Journal of Operational Research*, 101(3):509–518, 1997. ISSN 0377-2217. doi: 10.1016/S0377-2217(96)00188-9. URL `http://www.sciencedirect.com/science/article/pii/S0377221796001889`. 173

P. Bartodziej, U. Derigs, and U. Vogel. On the potentials of parallelizing large neighbourhood search for rich vehicle routing problems. *Learning and Intelligent Optimization*, pages 216–219, 2010. 144

R. Bastiaans. Reducing traffic jams, pollution and accidents. *Magazine for European Research*, 2000. URL `http://ec.europa.eu/research/rtdinfo/en/25/05.html`. 1

C. Bastian and A.H.G. Rinnooy-Kan. The stochastic vehicle routing problem revisited. *European Journal of Operational Research*, 56(3):407–412, 1992. ISSN 0377-2217. doi: 10.1016/0377-2217(92)90323-2. URL `http://www.sciencedirect.com/science/article/pii/0377221792903232`. 133

P. Bate. Inter-american development bank supports colombias national logistics policy. *BID News Release*, 2012. URL `http://www.iadb.org/en/news/news-releases/2012-04-27/new-trust-fund-for-cross-border-projects,9965.html`. 1

M. Battarra. Exact and heuristic algorithms for routing problems. *4OR*, 9:421–424, 2011. ISSN 1619-4500. doi: 10.1007/s10288-010-0141-9. URL `http://dx.doi.org/10.1007/s10288-010-0141-9`. 32, 35, 39

M. Battarra, M. Monaci, and D. Vigo. An adaptive guidance approach for the heuristic solution of a minimum multiple trip vehicle routing problem. *Computers & Operations Research*, 36(11):3041–3050, 2009. ISSN 0305-0548. doi: 10.1016/j.cor.2009.02.008. URL `http://www.sciencedirect.com/science/article/pii/S0305054809000410`. 32, 97

J. Bauer, T. Bektas, and T.G. Crainic. Minimizing greenhouse gas emissions in intermodal freight transport: an application to rail service design. *Journal of the Operational Research Society*, 61(3):530–542, 2009. 225

T. Bektas and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011. ISSN 0191–2615. doi: 10.1016/j.trb.2011.02.004. URL `http://www.sciencedirect.com/science/article/pii/S019126151100018X`. Supply chain disruption and risk management. 221, 225

P. Belfiore and H.T.Y. Yoshizaki. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in brazil. *European Journal of Operational Research*, 199(3):750–758, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2008.08.003. URL `http://www.sciencedirect.com/science/article/pii/S0377221708006978`. 57, 58

R. Bent and P. Van-Hentenryck. A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4):515–530, 2004a. 122, 209

REFERENCES AND BIBLIOGRAPHY

R. Bent and P. Van-Hentenryck. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893, 2006. ISSN 0305-0548. doi: 10.1016/j.cor.2004.08.001. URL `http://www.sciencedirect.com/science/article/pii/S0305054804001911`. Part Special Issue: Optimization Days 2003. 209

R. Bent and P. Van-Hentenryck. Waiting and relocation strategies in online stochastic vehicle routing. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1816–1821. Morgan Kaufmann Publishers Inc., 2007. 150

R.W. Bent and P. Van-Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004b. 133

J. Berger and M. Barkaoui. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research*, 31(12):2037–2053, 2004. ISSN 0305-0548. doi: 10.1016/S0305-0548(03)00163-1. URL `http://www.sciencedirect.com/science/article/pii/S0305054803001631`. 123, 141

O. Berman and R.C. Larson. Deliveries in an inventory/routing problem using stochastic dynamic programming. *Transportation Science*, 35(2):192–213, 2001. 174

L. Bertazzi, G. Paletta, and M.G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002. 172

L. Bertazzi, A. Bosco, F. Guerriero, and D. Laganà. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27(0):89–107, 2013. ISSN 0968-090X. doi: 10.1016/j.trc.2011.06.003. URL `http://www.sciencedirect.com/science/article/pii/S0968090X11000933`. Selected papers from the Seventh Triennial Symposium on Transportation Analysis (TRISTAN VII). 169, 175

A. Bettinelli, A. Ceselli, and G. Righini. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 19 (5):723–740, 2011. ISSN 0968-090X. doi: 10.1016/j.trc.2010.07.008. URL `http://www.sciencedirect.com/science/article/pii/S0968090X10001178`. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics). 35, 39

L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto. Hybrid metaheuristics for the vehicle routing problem with stochastic demands. *Journal of Mathematical Modelling and Algorithms*, 5(1):91–110, 2006. ISSN 1570-1166. doi: 10.1007/s10852-005-9033-y. URL `http://dx.doi.org/10.1007/s10852-005-9033-y`. 156

L. Bianchi, M. Dorigo, L.M. Gambardella, and W.J. Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8:239–287, 2009. ISSN 1567-7818. doi: 10.1007/s11047-008-9098-4. URL `http://dx.doi.org/10.1007/s11047-008-9098-4`. 150

D.D. Bochtis and C.G. Sörensen. The vehicle routing problem in field logistics part i. *Biosystems Engineering*, 104(4):447–457, 2009. ISSN 1537-5110. doi: http://dx.doi.org/10.1016/j.biosystemseng.2009.09.003. URL `http://www.sciencedirect.com/science/article/pii/S1537511009002712`. 120

M.C. Bolduc, J. Renaud, and B. Montreuil. Synchronized routing of seasonal products through a production/distribution network. *Central European Journal of Operations Research*, 14(2):209–228, June 2006. URL `http://ideas.repec.org/a/spr/cejnor/v14y2006i2p209-228.html`. 29, 35, 39, 95

J. Bramel and D. Simchi-Levi. A location based heuristic for general routing problems. *Operations Research*, 43(4):649–660, 1995. 171

J. Brandão. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 195(3):716–728, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2007.05.059. URL http://www.sciencedirect.com/science/article/pii/S0377221707010971. 57, 58

J. Brandão. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. *Computers & Operations Research*, 38(1):140–151, 2011. ISSN 0305-0548. doi: 10.1016/j.cor.2010.04.008. URL http://www.sciencedirect.com/science/article/pii/S0305054810000924. Project Management and Scheduling. 57, 58

J. Brandão and A. Mercer. The multi-trip vehicle routing problem. *Journal of the Operational research society*, pages 799–805, 1998. 58

O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science*, 39(1):104–118, 2005a. 122

O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation science*, 39(1):119–139, 2005b. 122

J.L. Bresina. Heuristic-biased stochastic sampling. In *Proceedings of the AAAI-96*, pages 271–278, 1996. 45, 46

G. Cabrera, A. Juan, H. Perez-Roses, J. Marques, and J. Faulin. Promoting green internet computing throughout simulation-optimization scheduling algorithms. In *Proceedings of the 2013 Winter Simulation Conference*, page 0, Washington, USA, December 2013. 225

V. Cabrera, F.J. Ros, and P.M. Ruiz. Simulation-based study of common issues in vanet routing protocols. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5, 2009. doi: 10.1109/VETECS.2009.5073851. 130

J. Cáceres-Cruz, A. Grasas, H. Lourenço, A. Juan, and M. Roca. Aplicación de un algoritmo randomizado a un problema real de enrutamiento de vehículos heterogéneos. In *Proceedings of the VIII Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)*, pages 767–773, Albacete, Spain, february 2012a. xvii, 229, 265

J. Cáceres-Cruz, A. Juan, S. Grasman, T. Bektas, and J. Faulin. Combining monte-carlo simulation with heuristics for solving the inventory routing problem with stochastic demands. In *Proceedings of the 2012 Winter Simulation Conference*, pages 1–9, Berlin, Germany, December 2012b. xvii, 165, 228, 272

J. Cáceres-Cruz, A.A. Juan, D. Riera, and H.R. Lourenço. A randomized algorithm for the heterogeneous fixed fleet vehicle routing problem. In *Proceedings of the 25th EURO Conference*, page 31, Vilnius, Lithuania, July 2012. 230

J. Cáceres-Cruz, A.A. Juan, D. Riera, H.R. Lourenço, A. Grasas, and R. Buil. Rich and real-life vehicle routing problems: cases of study in spain. In *Proceedings of the VeRoLog Conference 2013*, Southampton, UK, July 2013a. 231

J. Cáceres-Cruz, D. Riera, R. Buil, and A.A. Juan. Applying a savings algorithm for solving a rich vehicle routing problem in a real urban context. In *Proceedings of 5th International Conference on Applied Operational Research (ICAOR) - Lecture Notes in Management Science (LNMS)*, volume 5, pages 84–92, Lisbon, Portugal, July 2013b. xvii, 95, 229, 269

J. Cáceres-Cruz, D. Riera, R. Buil, A.A. Juan, and R. Herrero. Multi-start approach for solving an asymmetric heterogeneous vehicle routing problem in a real urban context. In *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems (ICORES)*, pages 168–174, Barcelona, Spain, February 2013. xvii, 95, 228, 262, 268

J. Cáceres-Cruz, D. Riera, A.A. Juan, and S. Padrón. Hybrid approach combining insertion heuristic and biased random sampling for the vehicle routing problem with time windows. In *Proceedings of the IX Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)*, page 8, Madrid, Spain, September 2013. xvii, 121, 229, 270

A.M. Campbell and M. Savelsbergh. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation science*, 38(3):369–378, 2004a. 172

A.M. Campbell and M. Savelsbergh. Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation science*, 38(3):369–378, 2004b. 123

A.M. Campbell and M.W.P. Savelsbergh. A decomposition approach for the inventory-routing problem. *Transportation Science*, 38(4):488–502, 2004c. 172

A.M. Campbell and M.W.P. Savelsbergh. Delivery volume optimization. *Transportation Science*, 38(2):210–223, 2004d. 172

Ann Campbell, Lloyd Clarke, Anton Kleywegt, and Martin Savelsbergh. The inventory routing problem. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 95–113. Kluwer Academic Publishers, Boston, MA, USA, 1998. 166, 171, 182

Ann M. Campbell, Lloyd W. Clarke, and Martin W. P. Savelsbergh. Inventory routing in practice. In P. Toth and D. Vigo, editors, *The vehicle routing problem*, pages 309–330. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. ISBN 0-89871-498-2. URL `http://dl.acm.org/citation.cfm?id=505847.505859`. 166, 172

Y. Carson and A. Maria. Simulation optimization: methods and applications. In *Proceedings of the 29th Conference on Winter simulation*, WSC '97, pages 118–126, Washington, DC, USA, 1997. IEEE Computer Society. ISBN 0-7803-4278-X. doi: 10.1145/268437.268460. URL `http://dx.doi.org/10.1145/268437.268460`. 130

S. Ceschia, L. Gaspero, and A. Schaerf. Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. *Journal of Scheduling*, 14(6):601–615, 2011. ISSN 1094-6136. doi: 10.1007/s10951-010-0213-x. URL `http://dx.doi.org/10.1007/s10951-010-0213-x`. 57, 58

A. Ceselli, G. Righini, and M. Salani. A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, 43(1):56–69, 2009. 31, 35, 39

L.M.A. Chan, A. Federgruen, and D. Simchi-Levi. Probabilistic analyses and practical algorithms for inventory-routing models. *Operations Research*, 46(1):96–106, 1998. 171

A.A. Chaves and L.A. Nogueira-Lorena. Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, 37(3):552–558, 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2008.09.011. URL `http://www.sciencedirect.com/science/article/pii/S0305054808001779`. Hybrid Metaheuristics. 47

T.W. Chien, A. Balakrishnan, and R.T. Wong. An integrated inventory allocation and vehicle routing problem. *Transportation Science*, 23(2):67–76, 1989. 171

E. Choi and D.W. Tcha. A column generation approach to the heterogeneous fleet vehicle routing problem. *Comput. Oper. Res.*, 34(7):2080–2095, July 2007. ISSN 0305-0548. doi: 10.1016/j.cor.2005.08.002. URL `http://dx.doi.org/10.1016/j.cor.2005.08.002`. 57, 58

N. Christofides and S. Eilon. An algorithm for the vehicle-dispatching problem. *OR*, pages 309–318, 1969. 9, 67, 219

N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, editors, *Combinatorial Optimization*, pages 315–338. John Wiley, NY, USA, 1979. 9

G. Clarke and J.W. Wright. Scheduling of vehicles from central depot to number of delivery points. *Operations Research*, 12(4):568–581, 1964. ISSN 0030-364X. 20, 47, 48, 59, 83, 96, 97, 141, 173, 176, 217

P. Constance. Inter-american development bank supports colombias national logistics policy. *BID News Release*, 2011. URL http://www.iadb.org/en/news/news-releases/2011-06-29/colombias-national-logistics-policy,9445.html. 1

J.F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594, 2003. ISSN 0191-2615. doi: 10.1016/S0191-2615(02)00045-0. URL http://www.sciencedirect.com/science/article/pii/S0191261502000450. 32, 39, 145, 206, 207

J.F. Cordeau and M. Maischberger. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033–2050, 2012. ISSN 0305-0548. doi: 10.1016/j.cor.2011.09.021. URL http://www.sciencedirect.com/science/article/pii/S0305054811002826. 145, 206

J.F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119, 1997. ISSN 1097-0037. doi: 10.1002/(SICI)1097-0037(199709)30:2⟨105::AID-NET5⟩3.0.CO;2-G. URL http://dx.doi.org/10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G. 32, 35, 39, 145, 206, 207

J.F. Cordeau, G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis. Vrp with time windows. In P. Toth and D. Vigo, editors, *the Vehicle Routing Problem*, volume 9, pages 157–193. SIAM - Society for Industrial and Applied Mathematics, Philadelphia, 2001a. 122

J.F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, pages 928–936, 2001b. 32, 35, 39, 122, 145, 206, 207

J.F. Cordeau, M. Gendreau, G. Laporte, J.Y. Potvin, and F. Semet. A guide to vehicle routing heuristics. *Journal of the Operational Research society*, pages 512–522, 2002. 8, 51, 132, 135

J.F. Cordeau, G. Laporte, and A. Mercier. Improved tabu search algorithm for the handling of route duration constraints in vehicle routing problems with time windows. *Journal of the Operational Research Society*, 55(5):542–546, 2004. 32, 35, 39, 122, 145, 206, 207

K. Cousineau-Ouimet. A tabu search heuristic for the inventory routing problem. In *Proceedings of 37th Annual ORSNZ Conference*, Auckland, New Zealand, nov. 2002. 172

T.G. Crainic. Parallel solution methods for vehicle routing problems. In Bruce Golden, S. Raghavan, and Edward Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 171–198. Springer US, 2008. ISBN 978-0-387-77777-1. doi: 10.1007/978-0-387-77778-8_8. URL http://dx.doi.org/10.1007/978-0-387-77778-8_8. 6, 139, 143

T.G. Crainic and M. Toulouse. Parallel strategies for meta-heuristics. In F. Glover and G.A. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 475–513. Springer US, 2003. ISBN 978-1-4020-7263-5. doi: 10.1007/0-306-48056-5_17. URL http://dx.doi.org/10.1007/0-306-48056-5_17. 6, 143, 144

T.G. Crainic, G.C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. A concurrent evolutionary approach for rich combinatorial optimization. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2017–2022. ACM, 2009a. 24, 39, 144

T.G. Crainic, G.C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. Multi-thread integrative cooperative optimization for rich combinatorial problems. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8. IEEE, 2009b. 24, 144

G.A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958. 62

A.L. Custódio and R.C. Oliveira. Redesigning distribution operations: a case study on integrating inventory management and vehicle routes design. *International Journal of Logistics Research and Applications: Research and Applications*, 9(2):169–187, 2006. doi: 10.1080/13675560600649982. URL http://www.tandfonline.com/doi/abs/10.1080/13675560600649982. 175

G. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. ISSN 0025-1909. 11

B. De-Backer, V. Furnon, P. Prosser, P. Kilby, and P. Shaw. Local search in constraint programming: Application to the vehicle routing problem. In *Proc. CP-97 Workshop Indust. Constraint-Directed Scheduling*, pages 1–15, Schloss Hagenberg, Austria, 1997. Citeseer. 208

R. Dekker, J. Bloemhof, and I. Mallidis. Operations research for green logistics: An overview of aspects, issues, contributions and challenges. *European Journal of Operational Research*, 219(3):671–679, 2012. ISSN 0377–2217. doi: 10.1016/j.ejor.2011.11.010. URL http://www.sciencedirect.com/science/article/pii/S0377221711009970. Operations Research in Health Care. EURO XXIII, 5-8 July 2009, Bonn The Past and Present of Optimization. EURO XXIV, 11-14 July 2010, Lisbon. 225

E. Demir, T. Bektas, and G. Laporte. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357, 2011. ISSN 1361–9209. doi: 10.1016/j.trd.2011.01.011. URL http://www.sciencedirect.com/science/article/pii/S136192091100023X. 221

E. Demir, T. Bektas, and G. Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012a. ISSN 0377–2217. doi: 10.1016/j.ejor.2012.06.044. URL http://www.sciencedirect.com/science/article/pii/S0377221712004997. 221, 225

E. Demir, T. Bektas, and G. Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 2012b. ISSN 0377–2217. 221, 225

G. Deng. *Simulation-Based Optimization*. PhD thesis, University of Wisconsin - Madison, 2007. 130

U. Derigs and T. Döhmer. Router: A fast and flexible local search algorithm for a class of rich vehicle routing problems. In H. Fleuren, D. Hertog, and P. Kort, editors, *Operations Research Proceedings 2004*, volume 2004 of *Operations Research Proceedings*, pages 144–149. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-24274-1. doi: 10.1007/3-540-27679-3_18. URL http://dx.doi.org/10.1007/3-540-27679-3_18. 30, 35, 39

U. Derigs, M. Pullmann, and U. Vogel. A short note on applying a simple ls/lns-based metaheuristic to the rollon-rolloff vehicle routing problem. *Computers & Operations Research*, 40(3):867–872, 2013. ISSN 0305-0548. doi: 10.1016/j.cor.2012.09.008. URL http://www.sciencedirect.com/science/article/pii/S0305054812002109. Transport Scheduling. 33, 35, 39

G. Desaulniers, J. Desrosiers, and M.M. Solomon. *Column generation*, volume 5. Springer, 2005. 18

M. Diana and M.M. Dessouky. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*, 38(6):539–557, 2004. ISSN 0191-2615. doi: 10.1016/j.trb.2003.07.001. URL http://www.sciencedirect.com/science/article/pii/S0191261503000912. 122, 123

H. Ding, L. Benyoucef, and X. Xie. A simulation-based optimization method for production-distribution network design. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4521–4526, 2004. doi: 10.1109/ICSMC.2004.1401244. 130

H. Ding, L. Benyoucef, and X. Xie. A simulation-based multi-objective genetic algorithm approach for networked enterprises optimization. *Engineering Applications of Artificial Intelligence*, 19(6):609–623, 2006. ISSN 0952-1976. doi: 10.1016/j.engappai.2005.12.008. URL http://www.sciencedirect.com/science/article/pii/S0952197606000376. Special Section on Innovative Production Machines and Systems (I*PROMS). 130

K.F. Doerner and V. Schmid. Survey: Matheuristics for rich vehicle routing problems. In M. Blesa, C. Blum, G. Raidl, A. Roli, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 6373 of *Lecture Notes in Computer Science*, pages 206–221. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16053-0. doi: 10.1007/978-3-642-16054-7_15. URL http://dx.doi.org/10.1007/978-3-642-16054-7_15. 31, 35, 39, 145, 224

M. Dorigo and L.M. Gambardella. Ant colonies for the travelling salesman problem. *BioSystems*, 43(2):73–81, 1997. 29

M. Dorigo and T. Stützle. Ant colony optimization: Overview and recent advances. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 227–263. Springer US, 2010. ISBN 978-1-4419-1663-1. doi: 10.1007/978-1-4419-1665-5_8. URL http://dx.doi.org/10.1007/978-1-4419-1665-5_8. 45

M. Drexl. Rich vehicle routing in theory and practice. *Logistics Research*, 5:47–63, 2012. ISSN 1865-035X. doi: 10.1007/s12159-012-0080-2. URL http://dx.doi.org/10.1007/s12159-012-0080-2. 3, 6, 23, 95, 205

R. Eglese and D. Black. Optimizing the routing of vehicles. In A. McKinnon, M. Browne, and A. Whiteing, editors, *Green logistics: Improving the environmental sustainability of logistics*, pages 215–228. Kogan Page, 2010. 225

B. Eksioglu, A. Volkan-Vural, and A. Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009. ISSN 0360-8352. doi: 10.1016/j.cie.2009.05.009. URL http://www.sciencedirect.com/science/article/pii/S0360835209001405. 147

S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012. ISSN 1366–5545. doi: 10.1016/j.tre.2011.08.001. URL http://www.sciencedirect.com/science/article/pii/S1366554511001062. Select Papers from the 19th International Symposium on Transportation and Traffic Theory. 221, 225

H. Eskandari, M. Darayi, and C.D. Geiger. Using simulation optimization as a decision support tool for supply chain coordination with contracts. In *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pages 1306–1317, 2010. doi: 10.1109/WSC.2010.5679062. 129

J. Euchi and H. Chabchoub. A hybrid tabu search to solve the heterogeneous fixed fleet vehicle routing problem. *Logistics Research*, 2:3–11, 2010. ISSN 1865-035X. doi: 10.1007/s12159-010-0028-3. URL http://dx.doi.org/10.1007/s12159-010-0028-3. 57, 58

P. Van-Beek F. Rossi and T. Walsh. *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*. Elsevier, 2006. 208

K. Fagerholt, G. Laporte, and I. Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3):523–529, 2009. 221, 225

J. Faulín and A.A. Juan. The algacea-1 method for the capacitated vehicle routing problem. *International Transactions in Operational Research*, 15(5):599–621, 2008. ISSN 1475-3995. doi: 10.1111/j.1475-3995.2008.00640.x. URL http://dx.doi.org/10.1111/j.1475-3995.2008.00640.x. 3

J. Faulín, M. Gilibert, A.A. Juan, X. Vilajosana, and R. Ruiz. Sr-1: A simulation-based algorithm for the capacitated vehicle routing problem. In *Winter Simulation Conference (WSC 2008)*, pages 2708–2716, 2008. doi: 10.1109/WSC.2008.4736388. 132

J. Faulin, F. Lera-López, and A.A Juan. Optimizing routes with safety and environmental criteria in transportation management in spain: A case study. *International Journal of Information Systems and Supply Chain Management (IJISSCM)*, 4(3):38–59, 2011. 225

J. Faulin, A.A. Juan, S. Grasman, and M. Fry. *Decision Making in Service Industries: A Practical Approach.* CRC Press - Taylor & Francis, Clermont, FL, USA, 2012. 44

A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984a. 172

A. Federgruen and P. Zipkin. An efficient algorithm for computing optimal (s, s) policies. *Operations Research*, 32(6):1268–1285, 1984b. 173

P. Festa and M.G.C. Resende. An annotated bibliography of grasp - part i: Algorithms. *International Transactions in Operational Research*, 16(1):1–24, 2009a. ISSN 1475-3995. doi: 10.1111/j.1475-3995.2009.00663.x. URL `http://dx.doi.org/10.1111/j.1475-3995.2009.00663.x`. 45

P. Festa and M.G.C. Resende. An annotated bibliography of grasp-part ii: Applications. *International Transactions in Operational Research*, 16(2):131–172, 2009b. ISSN 1475-3995. doi: 10.1111/j.1475-3995.2009.00664.x. URL `http://dx.doi.org/10.1111/j.1475-3995.2009.00664.x`. 45, 47

M.A. Figliozzi. The impacts of congestion on time-definitive urban freight distribution networks co2 emission levels: Results from a case study in portland, oregon. *Transportation Research Part C: Emerging Technologies*, 19(5):766–778, 2011. ISSN 0968-090X. doi: 10.1016/j.trc.2010.11.002. URL `http://www.sciencedirect.com/science/article/pii/S0968090X10001634`. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics). 221, 225

M. Fischetti, P. Toth, and D. Vigo. A branch-and-bound algorithm for the capacitated vehicle-routing problem on directed-graphs. *Operations Research*, 42(5):846–859, SEP-OCT 1994. ISSN 0030-364X. doi: 10.1287/opre.42.5.846. 9, 96, 101, 102

M.L. Fisher. Optimal solution of vehicle routing problems using minimum k-trees. *Operations research*, 42(4): 626–642, 1994. 9

M.L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle-routing. *Networks*, 11(2):109–124, 1981. ISSN 0028-3045. doi: 10.1002/net.3230110205. 96

M. Fleischmann. The vehicle routing problem with multiple use of vehicles. Technical report, Working paper, Fachbereich Wirtschaftswissenschaften, Universitt Hamburg, 1990. 58

M.C. Fu. Optimization via simulation: A review. *Annals of Operations Research*, 53(1):199–247, 1994. ISSN 0254-5330. doi: 10.1007/BF02136830. URL `http://dx.doi.org/10.1007/BF02136830`. 131

M.C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002. 131

M.C. Fu, F.W. Glover, and J. April. Simulation optimization: a review, new developments, and applications. In *Simulation Conference, 2005 Proceedings of the Winter*, page 13, 2005. doi: 10.1109/WSC.2005.1574242. 131

A. Fügenschuh. The vehicle routing problem with coupled time windows. *Central European Journal of Operations Research*, 14:157–176, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0166-5. URL `http://dx.doi.org/10.1007/s10100-006-0166-5`. 29, 35, 39

A. Funez. Inter-american development bank launches in panama the mesoamerican observatory on freight transport and logistics. *BID News Release*, 2012. URL `www.iadb.org/en/news/news-releases/2012-10-11/mesoamerican-observatory-freight-transport-and-logistic,10156.html`. 1

M.R. Garey and D.S. Johnson. "strong" np-completeness results: Motivation, examples, and implications. *J. ACM*, 25(3):499–508, jul 1978. ISSN 0004-5411. doi: 10.1145/322077.322090. URL `http://doi.acm.org/10.1145/322077.322090`. 14

M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979. ISBN 0716710447. 14

V. Gaur and M.L. Fisher. A periodic inventory routing problem at a supermarket chain. *Operations Research*, 52(6):813–822, 2004. 174

M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996a. ISSN 0377-2217. doi: 10.1016/0377-2217(95)00050-X. URL `http://www.sciencedirect.com/science/article/pii/037722179500050X`. 133

M. Gendreau, G. Laporte, and R. Séguin. A tabu search heuristic for the vehicle routing problem with stochastic demands and customers. *Operations Research*, 44(3):469–477, 1996b. 133

M. Gendreau, G. Laporte, C. Musaraganyi, and E.D.D. Taillard. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research*, 26(1153):1173, 1999. 57, 58

M. Gendreau, J.Y. Potvin, O. Bräumlaysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 143–169. Springer US, 2008. ISBN 978-0-387-77777-1. doi: 10.1007/978-0-387-77778-8_7. URL `http://dx.doi.org/10.1007/978-0-387-77778-8_7`. 19, 20, 132

G. Ghiani, F. Guerriero, G. Laporte, and R. Musmanno. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research*, 151(1):1–11, 2003. ISSN 0377-2217. doi: 10.1016/S0377-2217(02)00915-3. URL `http://www.sciencedirect.com/science/article/pii/S0377221702009153`. 141

F. Glover, J.P. Kelly, and M. Laguna. New advances and applications of combining simulation and optimization. In *Proceedings of the 28th Conference on Winter simulation*, WSC '96, pages 144–152, Washington, DC, USA, 1996. IEEE Computer Society. ISBN 0-7803-3383-7. doi: 10.1145/256562.256595. URL `http://dx.doi.org/10.1145/256562.256595`. 129, 131

F. Glover, J.P. Kelly, and M. Laguna. New advances for wedding optimization and simulation. In *Simulation Conference Proceedings, 1999 Winter*, volume 1, pages 255–260, 1999. doi: 10.1109/WSC.1999.823081. 129

A. Goel. A column generation heuristic for the general vehicle routing problem. In C. Blum and R. Battiti, editors, *Learning and Intelligent Optimization*, volume 6073 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13799-0. doi: 10.1007/978-3-642-13800-3_1. URL `http://dx.doi.org/10.1007/978-3-642-13800-3_1`. 31, 35, 39

A. Goel and V. Gruhn. Large neighborhood search for rich vrp with multiple pickup and delivery locations. In *Proceedings of the 18th Mini EURO Conference on VNS (MEC-VNS). ISBN*, pages 84–689, 2005. 24, 27, 35, 39

A. Goel and V. Gruhn. Solving a dynamic real-life vehicle routing problem. In H.D. Haasis, H. Kopfer, and J. Schönberger, editors, *Operations Research Proceedings 2005*, volume 2005 of *Operations Research Proceedings*, pages 367–372. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-32537-6. doi: 10.1007/3-540-32539-5_58. URL http://dx.doi.org/10.1007/3-540-32539-5_58. 24, 27, 35, 39

A. Goel and V. Gruhn. A general vehicle routing problem. *European Journal of Operational Research*, 191 (3):650–660, 2008. ISSN 0377-2217. doi: 10.1016/j.ejor.2006.12.065. URL http://www.sciencedirect.com/science/article/pii/S0377221707001233. 24, 27, 35, 39

B.L. Golden, J.S. Dearmon, and E.K. Baker. Computational experiments with algorithms for a class of routing problems. *Computers & Operations Research*, 10(1):47–59, 1983. ISSN 0305-0548. doi: 10.1016/0305-0548(83)90026-6. URL http://www.sciencedirect.com/science/article/pii/0305054883900266. 48

B.L. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984. 9, 20, 57, 58, 67, 219

B.L. Golden, E.A. Wasil, J.P. Kelly, and I.M. Chao. The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 33–56. Springer US, 1998. ISBN 978-0-7923-8161-7. doi: 10.1007/978-1-4615-5755-5_2. URL http://dx.doi.org/10.1007/978-1-4615-5755-5_2. 9, 218

B.L. Golden, S. Raghavan, and E.A. Wasil. *The vehicle routing problem: latest advances and new challenges*, volume 43. Springer, 2008. 2, 9, 14, 17, 58, 121, 148, 210

C.P. Gomes and B. Selman. Practical aspects of algorithm portfolio design. In *Proceedings of the Third ILOG International Users Meeting*, page 23. Citeseer, 1997. 48

J. González, J. Delgado, and H. Fraire. Construction of an optimal solution for a real-world routing-scheduling-loading problem. *Computación y Sistemas*, 13:398–408, 2010. 3

S. González-Martín, A. Juan, D. Riera, and J. Cáceres-Cruz. A hybrid algorithm combining path scanning and biased random sampling for the arc routing problem. In *Proceedings of the 18th RCRA Workshop*, pages 46–54, Barcelona, Spain, july 2011. 230

S. González-Martín, A.A Juan, D. Riera, Q. Castellà, R. Muñoz, and A. Pérez. Development and assessment of the sharp and randsharp algorithms for the arc routing problem. *AI Communications*, 25(2):173–189, 2012. 48, 124, 225

A. Gosavi. *Simulation-based optimization: parametric optimization techniques and reinforcement learning*, volume 25. Kluwer Academic Pub, 2003. 131

A. Grasas, J. Cáceres-Cruz, H.R. Lourenço, A.A. Juan, and M. Roca. Vehicle routing in a spanish distribution company: Saving using a savings-based heuristic. *OR Insight*, pages 1–12, 2013. ISSN 0953-5543. doi: 10.1057/ori.2013.2. xvii, 55, 227, 228, 264

C. Groër, B. Golden, and E. Wasil. A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2:79–101, 2010. ISSN 1867-2949. doi: 10.1007/s12532-010-0013-5. URL http://dx.doi.org/10.1007/s12532-010-0013-5. 32, 35, 39

D. Guimarans. *Hybrid Algorithms for Solving Routing Problems.* PhD thesis, Universitat Autonoma de Barcelona, 2012. 209

R. Hall. The 2006 vehicle routing survey. *OR/MS Today*, 33(3), june 2006. 23

P. Hansen, N. Mladenović, J. Brimberg, and J.A. Moreno-Pérez. Variable neighborhood search. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 61–86. Springer US, 2010. ISBN 978-1-4419-1663-1. doi: 10.1007/978-1-4419-1665-5_3. URL http://dx.doi.org/10.1007/978-1-4419-1665-5_3. 45

R.F. Hartl, G. Hasle, and G.K. Janssens. Special issue on rich vehicle routing problems. *Central European Journal of Operations Research*, 14:103–104, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0162-9. URL http://dx.doi.org/10.1007/s10100-006-0162-9. 28

H. Hashimoto, T. Ibaraki, S. Imahori, and M. Yagiura. The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics*, 154(16):2271–2290, 2006. ISSN 0166-218X. doi: 10.1016/j.dam.2006.04.009. URL http://www.sciencedirect.com/science/article/pii/S0166218X06001879. Discrete Algorithms and Optimization, in Honor of Professor Toshihide Ibaraki at His Retirement from Kyoto University, International Symposium on Discrete Algorithms and Optimization. 32, 39, 206, 207

H. Hashimoto, M. Yagiura, and T. Ibaraki. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization*, 5(2):434–456, 2008. ISSN 1572-5286. doi: 10.1016/j.disopt.2007.05.004. URL http://www.sciencedirect.com/science/article/pii/S1572528607000485. In Memory of George B. Dantzig. 32, 39, 206, 207

G. Hasle and O. Kloster. Industrial vehicle routing. In G. Hasle, K.A. Lie, and E. Quak, editors, *Geometric Modelling, Numerical Simulation, and Optimization*, pages 397–435. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-68782-5. doi: 10.1007/978-3-540-68783-2_12. URL http://dx.doi.org/10.1007/978-3-540-68783-2_12. 24, 28, 35, 39, 206, 207, 225

G. Hasle, O. Kloster, E.J. Nilssen, A. Riise, and M. Smedsrud. Scalable and instance robust metaheuristics for a rich vrp model. In *ROUTE 2005 - International workshop on vehicle routing and intermodal transportation*, University of Bologna, Bertinoro, Italy, june 2005. 28, 206

G. Hasle, A. Løkketangen, and S. Martello. Rich models in discrete optimization: Formulation and resolution (ecco xvi). *European Journal of Operational Research*, 175(3):1752–1753, 2006. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.06.003. URL http://www.sciencedirect.com/science/article/pii/S037722170500473X. 24

V. Hemmelmayr, K.F. Doerner, R.F. Hartl, and M.W.P. Savelsbergh. Vendor managed inventory for environments with stochastic product usage. *European Journal of Operational Research*, 202(3):686–695, 2010. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.06.003. URL http://www.sciencedirect.com/science/article/pii/S037722170900424X. 150

F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud. A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. Technical report, Tech. rep., Hal Archives Ouvertes, 2011. 97

R. Herrero, A. Rodríguez, J. Cáceres-Cruz, and A.A. Juan. Solving vehicle routing problems with asymmetric costs and heterogeneous fleets. *Int. J. of Advanced Operations Management*, pages 1–19, 2014. ISSN 1758–938X. xvii, 95, 228, 267

A. Hoff. *Heuristics for Rich Vehicle Routing Problem.* PhD thesis, Molde University College, 2006. 29

A. Hoff and A. Løkketangen. Creating lasso-solutions for the traveling salesman problem with pickup and delivery by tabu search. *Central European Journal of Operations Research*, 14:125–140, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0164-7. URL http://dx.doi.org/10.1007/s10100-006-0164-7. 28, 29, 35, 39

REFERENCES AND BIBLIOGRAPHY

A. Hoff and A. Løkketangen. A tabu search approach for milk collection in western norway using trucks and trailers. *Proc. Sixth Triennial Sympos. Transportation Anal.(TRISTAN VI), Phuket Island, Thailand*, 2007. 29

A. Hoff, I. Gribkovskaia, G. Laporte, and A. Løkketangen. Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 192(3):755–766, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2007.10.021. URL `http://www.sciencedirect.com/science/article/pii/S0377221707010235`. 29

A. Hoff, A.G. Lium, A. Løkketangen, and T.G. Crainic. A metaheuristic for stochastic service network design. *Journal of Heuristics*, 16:653–679, 2010. ISSN 1381-1231. doi: 10.1007/s10732-009-9112-8. URL `http://dx.doi.org/10.1007/s10732-009-9112-8`. 29, 81

W. Hu, H. Liang, C. Peng, B. Du, and Q. Hu. A hybrid chaos-particle swarm optimization algorithm for the vehicle routing problem with time window. *Entropy*, 15(4):1247–1270, 2013. 122

X.P. Hu, Y.X. Li, J.W. Guo, L. Sun, and A.Z. Zeng. A simulation optimization algorithm with heuristic transformation and its application to vehicle routing problems. *International Journal of Innovative Computing, Information and Control*, 4(5):1169–1181, 2008. 132

L.M. Hvattum, A. Løkketangen, and G. Laporte. Scenario tree-based heuristics for stochastic inventory-routing problems. *INFORMS Journal on Computing*, 21(2):268–285, 2009. 169, 175

T. Ibaraki, S. Imahori, M. Kubo, T. Masuda, T. Uno, and M. Yagiura. Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Science*, 39(2):206–232, may 2005. ISSN 1526-5447. doi: 10.1287/trsc.1030.0085. URL `http://dx.doi.org/10.1287/trsc.1030.0085`. 32, 39, 206, 207

T. Ibaraki, S. Imahori, K. Nonobe, K. Sobue, T. Uno, and M. Yagiura. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics*, 156(11):2050–2069, 2008. ISSN 0166-218X. doi: 10.1016/j.dam.2007.04.022. URL `http://www.sciencedirect.com/science/article/pii/S0166218X07001357`. In Memory of Leonid Khachiyan (1952–2005). 206, 207

Y. Ileri, M. Bazaraa, T. Gifford, G. Nemhauser, J. Sokol, and E. Wikum. An optimization approach for planning daily drayage operations. *Central European Journal of Operations Research*, 14:141–156, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0165-6. URL `http://dx.doi.org/10.1007/s10100-006-0165-6`. 29, 35, 39

A. Imran, S. Salhi, and N.A. Wassan. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 197(2):509–518, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2008.07.022. URL `http://www.sciencedirect.com/science/article/pii/S0377221708005687`. 57, 58

C. Ioannou, M. Kritikos, and G. Prastacos. A greedy look-ahead heuristic for the vehicle routing problem with time windows. *The Journal of the Operational Research Society*, 52(5):523–537, 2001. URL `http://search.proquest.com/docview/231376036?accountid=15299`. Copyright (c) 2001 Palgrave Publishers Ltd; ltima actualizacin - 2010-06-10; CODEN - OPRQAK; DOI - 73456528; 1390562; 12717; OPRQAK; OQT. 122, 123

S. Irnich. A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. *INFORMS Journal on Computing*, 20(2):270–287, 2008. doi: 10.1287/ijoc.1070.0239. URL `http://joc.journal.informs.org/content/20/2/270.abstract`. 30, 35, 39, 95, 206, 207

O. Jabali, T. Van-Woensel, and A.G. De-Kok. Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012. ISSN 1937–5956. doi: 10.1111/j.1937-5956.2012.01338.x. URL `http://dx.doi.org/10.1111/j.1937-5956.2012.01338.x`. 221, 225

P. Jaillet, J.F. Bard, L. Huang, and M. Dror. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science*, 36(3):292–300, 2002. 174

M. Janic. Sustainable transport in the european union: A review of the past research and future ideas. *Transport Reviews*, 26(1):81–104, 2006. 1

S. Jarugumilli, S.E. Grasman, and S. Ramakrishnan. A simulation framework for real-time management and control of inventory routing decisions. In *Simulation Conference, 2006. WSC 06. Proceedings of the Winter*, pages 1485–1492, Monterey, California, USA, dec. 2006. doi: 10.1109/WSC.2006.322917. 175

K. Jeevan-Madhu and S. Saxena. Parallel algorithms for vehicle routing problems. In *5th International Conference On High Performance Computing, 1998. HIPC '98*, pages 171–178, 1998. doi: 10.1109/HIPC.1998.737986. 140

A. Jézéquel. Probabilistic vehicle routing problems. Technical report, MSc dissertation. Department of Civil Engineering. Massachussets Institute of Technology, Cambridge, MA, 1985. 133

A.A. Juan and M. Rabe. Combining simulation with heuristics to solve stochastic routing and scheduling problems. In *Proceedings of the 15th ASIM Dedicated Conference*, Paderborn, Germany, october 2013. 130, 225

A.A. Juan, J. Faulin, R. Ruiz, B.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 J.W. Chinneck, B. Kristjansson, and M.J. Saltzman, editors, *Operations Research and Cyber-Infrastructure*, volume 47 of *Operations Research/Computer Science Interfaces*, pages 331–345. Springer US, 2009. ISBN 978-0-387-88842-2. doi: 10.1007/978-0-387-88843-9_17. URL http://dx.doi.org/10.1007/978-0-387-88843-9_17. 3

A.A. Juan, J. Faulin, R. Ruiz, B.B. Barrios, and S. Caballe. The sr-gcws hybrid algorithm for solving the capacitated vehicle routing problem. *Appl. Soft Comput.*, 10(1):215–224, 2010. URL http://dblp.uni-trier.de/db/journals/asc/asc10.html#JuanFRBC10. 3, 20, 47, 61, 81, 82, 84, 176, 217

A.A. Juan, F. Faulín, J. Cáceres-Cruz, and S. González-Martín. Combining randomized heuristics, monte carlo simulation and parallel computing to solve the stochastic vehicle routing problem. In *Proceedings of the International conference on Optimization, Theory, Algorithms and Applications in Economics (OPT 2011)*, pages 50–51, Barcelona, Spain, october 2011a. 230

A.A. Juan, F. Faulín, J. Jorba, J. Cáceres-Cruz, and J. Marques. A simulation-based algorithm for solving the vehicle routing problem with stochastic demands. In *Proceedings of the 2011 ALIO/EURO Workshop*, pages 131–134, Oporto, Portugal, may 2011b. 229

A.A. Juan, J. Faulin, A. Ferrer, H.R. Lourenço, and B.B. Barrios. Mirha: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *TOP*, pages 1–24, 2011c. ISSN 1134-5764. doi: 10.1007/s11750-011-0245-1. URL http://dx.doi.org/10.1007/s11750-011-0245-1. 46

A.A. Juan, J. Faulin, S. Grasman, D. Riera, J. Marull, and C. Mendez. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C: Emerging Technologies*, 19(5):751–765, 2011d. ISSN 0968-090X. doi: 10.1016/j.trc.2010.09.007. URL http://www.sciencedirect.com/science/article/pii/S0968090X10001439. Freight Transportation and Logistics (selected papers from ODYSSEUS 2009 - the 4th International Workshop on Freight Transportation and Logistics). 133, 151, 159, 176

A.A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip, and B.B. Barrios. On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics. *JORS*, 62(6):1085–1097, 2011e. URL http://dblp.uni-trier.de/db/journals/jors/jors62.html#JuanFJRMB11. 21, 59, 62, 81, 84, 100, 101, 152, 176, 179

A.A. Juan, J. Faulín, A. Agustín, and J. Cáceres-Cruz. A multi-round simulation method which analyzes fleet designs to solve the heterogeneous vehicle routing problem. In *Proceedings of the International Symposium on Combinatorial Optimization (CO2012)*, University of Oxford, UK, September 2012a. 230

A.A Juan, H.R. Lourenço, M. Mateo, Q. Castellà, and B.B. Barrios. Ils-esp an efficient, simple, and parameter-free algorithm for solving the permutation flow-shop problem. In *Proceedings of the 6th Workshop on Statistics, Mathematics and Computation*, 2012b. 48, 225

A.A. Juan, J. Faulin, J. Jorba, J. Cáceres-Cruz, and J.M. Marques. Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research*, 207(1):43–65, 2013a. ISSN 0254-5330. doi: 10.1007/s10479-011-0918-z. URL `http://dx.doi.org/10.1007/s10479-011-0918-z`. xvii, 52, 62, 129, 147, 227, 271

A.A. Juan, H.R. Lourenço, M. Mateo, R. Luo, and Q. Castellà. Using iterated local search for solving the flow-shop problem: parametrization, randomization and parallelization issues. *AI Communications*, 2013b. ISSN 0969-6016. 225

A.A. Juan, J. Cáceres-Cruz, S. González-Martín, D. Riera, and B.B. Barrios. Biased randomization of classical heuristics. In J. Wang, editor, *Encyclopedia of Business Analytics and Optimization*, pages 1–11. IGI Global, 2014a. xvii, 43, 228, 263

A.A. Juan, J. Faulin, J. Cáceres-Cruz, B.B. Barrios, and E. Martínez. A successive approximations method for the heterogeneous vehicle routing problem: analyzing different fleet configurations. *European J. of Industrial Engineering*, pages 1–19, 2014b. ISSN 1751–5254. xvii, 55, 227, 266

R. Kaipia, J. Holmström, and K. Tanskanen. Vmi: What are you losing if you let your customer place orders? *Production Planning & Control: The Management of Operations*, 13(1):17–25, 2002. doi: 10.1080/09537280110061539. URL `http://www.tandfonline.com/doi/abs/10.1080/09537280110061539`. 165

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948 vol.4, nov/dec 1995. doi: 10.1109/ICNN.1995.488968. 45

A.E. Khandani, E. Modiano, J. Abounadi, and L. Zheng. Cooperative routing in wireless networks. In B.K. Szymanski and B. Yener, editors, *Advances in Pervasive Computing and Networking*, pages 97–117. Springer US, 2005. ISBN 978-0-387-23042-9. doi: 10.1007/0-387-23466-7_5. URL `http://dx.doi.org/10.1007/0-387-23466-7_5`. 130

P. Kilby and P. Shaw. Chapter 23 - vehicle routing. In P. Van-Beek F. Rossi and T. Walsh, editors, *Handbook of Constraint Programming*, volume 2 of *Foundations of Artificial Intelligence*, pages 801–836. Elsevier, 2006. doi: 10.1016/S1574-6526(06)80027-1. URL `http://www.sciencedirect.com/science/article/pii/S1574652606800271`. 209

P. Kilby, P. Prosser, and P. Shaw. Guided local search for the vehicle routing problem with time windows. In S. Voβ, S. Martello, I.H. Osman, and C. Roucairol, editors, *Meta-Heuristics*, pages 473–486. Springer US, 1999. ISBN 978-1-4613-7646-0. doi: 10.1007/978-1-4615-5775-3_32. URL `http://dx.doi.org/10.1007/978-1-4615-5775-3_32`. 122

P. Kilby, P. Prosser, and P. Shaw. A comparison of traditional and constraint-based heuristic methods on vehicle routing problems with side constraints. *Constraints*, 5:389–414, 2000. ISSN 1383-7133. doi: 10.1023/A:1009808327381. URL `http://dx.doi.org/10.1023/A%3A1009808327381`. 209

B.I. Kim, S. Kim, and S. Sahoo. Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642, 2006. ISSN 0305-0548. doi: 10.1016/j.cor.2005.02.045. URL `http://www.sciencedirect.com/science/article/pii/S0305054805001322`. Part Special Issue: Recent Algorithmic Advances for Arc Routing Problems. 122

M.H. Kim and Y.D. Kim. Simulation-based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing Systems*, 13(2):85–93, 1994. ISSN 0278-6125. doi: 10.1016/0278-6125(94)90024-8. URL `http://www.sciencedirect.com/science/article/pii/0278612594900248`. 130

D.B. Kirk and W.H. Wen-mei. *Programming massively parallel processors: a hands-on approach*. Morgan Kaufmann, 2010. 142

A.J. Kleywegt, V.S. Nori, and M.W.P. Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation Science*, 36(1):94–118, 2002. 174

A.J. Kleywegt, V.S. Nori, and M.W.P. Savelsbergh. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42–70, 2004. 174

R. Kohli and H. Park. Coordinating buyer-seller transactions across multiple products. *Management Science*, 40(9):1145–1150, 1994. 171

G. Kontoravdis and J.F. Bard. A grasp for the vehicle routing problem with time windows. *ORSA journal on Computing*, 7(1):10–23, 1995. 122

Y.A. Koskosidis, W.B. Powell, and M.M. Solomon. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science*, 26(2):69–85, 1992. 132

C. Kumares and S. Labi, editors. *Transportation Decision Making: Principles of Project Evaluation and Programming*. Wiley, 2007. 1

Y. Kuo. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1):157–165, 2010. ISSN 0360-8352. doi: 10.1016/j.cie.2010.03.012. URL `http://www.sciencedirect.com/science/article/pii/S0360835210000835`. 225

N. Labadi, C. Prins, and M. Reghioui. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO-Operations Research*, 42(03):415–431, 2008. 207

R. Lahyani, M. Khemakhem, H. Chabchoub, and F. Semet. Design factors analysis for instances of rich vehicle routing problem. In *International Conference on Logistics*, 2011. doi: 10.1109/LOGISTIQUA.2011.5939428. 9

R. Lahyani, M. Khemakhem, F. Semet, and H. Chabchoub. Taxonomy for rich vehicle routing problems. In *1st International IEEE Conference on Logistics Operations Management*, 10 2012. 25, 34, 40

S. Lannez, C. Artigues, J.J. Damay, and M. Gendreau. Column generation heuristic for a rich arc routing problem. In *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*, volume 14, pages 130–141, 2010. 33

G. Laporte. What you should know about the vehicle routing problem. *Naval Research Logistics*, 54:811–819, 2007. 6, 17, 148

G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009. 6, 7, 9, 14, 47

G. Laporte, H. Mercure, and Y. Nobert. An exact algorithm for the asymmetrical capacitated vehicle-routing problem. *Networks*, 16(1):33–46, SPR 1986. ISSN 0028-3045. 96

G. Laporte, M. Gendreau, J.Y. Potvin, and F. Semet. Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4-5):285–300, 2000. ISSN 1475-3995. doi: 10.1111/j.1475-3995.2000.tb00200.x. URL `http://dx.doi.org/10.1111/j.1475-3995.2000.tb00200.x`. 12, 132, 208, 210

H.C. Lau, M. Sim, and K.M. Teo. Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*, 148(3):559–569, 2003. ISSN 0377-2217. doi: 10.1016/S0377-2217(02)00363-6. URL http://www.sciencedirect.com/science/article/pii/S0377221702003636. 122

A.M. Law. *Simulation modeling and analysis*, volume 4. McGraw-Hill: Boston, 2007. 176

A.M. Law and M.G. McComas. Simulation optimization: simulation-based optimization. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, WSC '02, pages 41–44. Winter Simulation Conference, 2002. ISBN 0-7803-7615-3. URL http://dl.acm.org/citation.cfm?id=1030453.1030461. 129

A. Le-Bouthillier and T.G. Crainic. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research*, 32(7):1685–1708, 2005. ISSN 0305-0548. doi: 10.1016/j.cor.2003.11.023. URL http://www.sciencedirect.com/science/article/pii/S0305054803003654. 142, 143, 144

P. L'ecuyer and E. Buist. Simulation in java with ssj. In *Winter Simulation Conference*, pages 611–620, 2005. doi: 10.1145/1162708.1162815. 125, 158

P. L'Ecuyer, L. Meliani, and J. Vaucher. Ssj: a framework for stochastic simulation in java. In *Proceedings of the 34th WSC: exploring new frontiers*, pages 234–242. Winter Simulation Conference, 2002. URL http://doi.acm.org/10.1145/1030453.1030488. 103

J.K. Lenstra and A.H.G. Rinnooy-Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981. ISSN 1097-0037. doi: 10.1002/net.3230110211. URL http://dx.doi.org/10.1002/net.3230110211. 14

F. Lera-López, J. Faulin, and M. Sánchez. Determinants of the willingness-to-pay for reducing the environmental impacts of road transportation. *Transportation Research Part D: Transport and Environment*, 17(3):215–220, 2012. ISSN 1361–9209. doi: 10.1016/j.trd.2011.11.002. URL http://www.sciencedirect.com/science/article/pii/S1361920911001490. 225

F. Li, B. Golden, and E. Wasil. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operation Research*, 34(9):2734–2742, 2007a. 9, 57, 58, 67, 70

F. Li, B. Golden, and E. Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. *Computers & Operations Research*, 34(10):2918–2930, 2007b. ISSN 0305-0548. doi: http://dx.doi.org/10.1016/j.cor.2005.11.018. URL http://www.sciencedirect.com/science/article/pii/S0305054805003515. 116

H. Li and A. Lim. A metaheuristic for the pickup and delivery problem with time windows. In *Tools with Artificial Intelligence, Proceedings of the 13th International Conference on*, pages 160 –167, nov 2001. ISBN 0-7695-1417-0. doi: 10.1109/ICTAI.2001.974461. 9, 28

H. Li and A. Lim. A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, 12(02):173–186, 2003. doi: 10.1142/S0218213003001186. URL http://www.worldscientific.com/doi/abs/10.1142/S0218213003001186. 144

X. Li, P. Tian, and Y.P. Aneja. An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):1111–1127, 2010. ISSN 1366-5545. doi: 10.1016/j.tre.2010.02.004. URL http://www.sciencedirect.com/science/article/pii/S1366554510000256. 57, 58

C.M.R.R. Lima, M.C. Goldbarg, and E.F.G. Goldbarg. A memetic algorithm for the heterogeneous fleet vehicle routing problem. *Electronic Notes in Discrete Mathematics*, 18(0):171–176, 2004. ISSN 1571-0653. doi: 10.1016/j.endm.2004.06.027. URL http://www.sciencedirect.com/science/article/pii/S1571065304010844. Latin-American Conference on Combinatorics, Graphs and Applications. 57, 58

C.K.Y. Lin and R.C.W. Kwok. Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data. *European Journal of Operational Research*, 175(3):1833–1849, 2006. ISSN 0377-2217. doi: 10.1016/j.ejor.2004.10.032. URL http://www.sciencedirect.com/science/article/pii/S0377221705004790. 58

X. Lin and N.B. Shroff. An optimization-based approach for qos routing in high-bandwidth networks. *IEEE ACM Trans Netw*, 14(6):1348–1361, 2006. ISSN 1063-6692. doi: 10.1109/TNET.2006.886290. URL http://dx.doi.org/10.1109/TNET.2006.886290. 130

S. Liu, W. Huang, and H. Ma. An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(3):434–445, 2009. ISSN 1366-5545. doi: 10.1016/j.tre.2008.10.003. URL http://www.sciencedirect.com/science/article/pii/S1366554508001324. 57, 58

E.M. Loiola, N.M. Maia de Abreu, P.O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.09.032. URL http://www.sciencedirect.com/science/article/pii/S0377221705008337. 47

A. Løkketangen and F. Glover. Solving zero-one mixed integer programming problems using tabu search. *European Journal of Operational Research*, 106(2–3):624–658, 1998. ISSN 0377-2217. doi: 10.1016/S0377-2217(97)00295-6. URL http://www.sciencedirect.com/science/article/pii/S0377221797002956. 45

H. Lourenço, J. Cáceres-Cruz, A. Grasas, A. Juan, and M. Roca. A randomized hybrid algorithm based on savings and vehicle assignment policies for the heterogeneous vehicle routing problem. In *Proceedings of the VeRoLog Conference 2012*, Bologna, Italy, June 2012. 230

H.R. Lourenço, O.C. Martin, and T. Stützle. Iterated local search: Framework and applications. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 363–397. Springer US, 2010. ISBN 978-1-4419-1663-1. doi: 10.1007/978-1-4419-1665-5_12. URL http://dx.doi.org/10.1007/978-1-4419-1665-5_12. 45

J.M.D Magalhães and J.P. Sousa. Dynamic vrp in pharmaceutical distribution - a case study. *Central European Journal of Operations Research*, 14:177–192, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0167-4. URL http://dx.doi.org/10.1007/s10100-006-0167-4. 29, 35, 39

M.E. Marmion, J. Humeau, L. Jourdan, and C. Dhaenens. Comparison of neighborhoods for the hff-avrp. In *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, pages 1–7, may 2010. doi: 10.1109/AICCSA.2010.5586979. 97, 102, 110

J.M. Marques, Z. Vilajosana, T. Daradoumis, and L. Navarro. Lacolla: Middleware for self-sufficient online collaboration. *Internet Computing, IEEE*, 11(2):56–64, 2007. 137

A. McKinnon, M. Browne, and A. Whiteing. *Green logistics: Improving the environmental sustainability of logistics*. Kogan Page, 2012. 221, 225

S. Mitra. A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of the Operational Research Society*, 59(11):1532–1546, 2007. 142

N. Mladenović, J. Brimberg, P. Hansen, and J.A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179(3):927–939, 2007. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.05.034. URL `http://www.sciencedirect.com/science/article/pii/S0377221706000750`. 47

J.R. Montoya-Torres, A.A. Juan, L.H. Huatuco, J. Faulin, and G.L. Rodríguez-Verján. *Hybrid algorithms for service, computing and manufacturing systems: routing and scheduling solutions*. IGI Global, Hershey, 2012. 44, 225

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, March 1957. 70

A. Muñoz-Villamizar, J.R. Montoya-Torres, A.A. Juan, and J. Cáceres-Cruz. A simulation-based algorithm for the integrated location and routing problem in urban logistics. In *Proceedings of the 2013 Winter Simulation Conference*, pages 1–12, Washington, USA, December 2013. xvii, 225, 228, 273

İ. Muter, Ş. Birbil, and G. Şahin. Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems. *INFORMS Journal on Computing*, 22(4):603–619, 2010. 47

Y. Nagata. Edge assembly crossover for the capacitated vehicle routing problem. In C. Cotta and J.I. Van-Hemert, editors, *Evolutionary Computation in Combinatorial Optimization, 7th European Conference, EvoCOP*, Lecture Notes in Computer Science, pages 142–153, Valencia, 2007. Springer. 97, 102, 103, 104

M. Nawaz, E.E. Enscore-Jr, and I. Ham. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11(1):91–95, 1983. ISSN 0305-0483. doi: 10.1016/0305-0483(83)90088-9. URL `http://www.sciencedirect.com/science/article/pii/0305048383900889`. 48

S.U. Ngueveu, C. Prins, and R. Wolfler-Calvo. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 37(11):1877–1885, 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2009.06.014. URL `http://www.sciencedirect.com/science/article/pii/S0305054809001725`. Metaheuristics for Logistics and Vehicle Routing. 207

A.G. Nikolaev and S.H. Jacobson. Simulated annealing. *Handbook of Metaheuristics*, pages 1–39, 2010. 45

C. Novoa and R. Storer. An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. *European Journal of Operational Research*, 196(2):509–515, 2009. ISSN 0377-2217. doi: 10.1016/j.ejor.2008.03.023. URL `http://www.sciencedirect.com/science/article/pii/S0377221708003172`. 148

A. Olivera and O. Viera. Adaptive memory programming for the vehicle routing problem with multiple trips. *Computers & Operations Research*, 34(1):28–47, 2007. ISSN 0305-0548. doi: 10.1016/j.cor.2005.02.044. URL `http://www.sciencedirect.com/science/article/pii/S0305054805001346`. 9, 58

J. Oppen, A. Løkketangen, and J. Desrosiers. Solving a rich vehicle routing and inventory problem using column generation. *Computers & Operations Research*, 37(7):1308–1317, 2010. ISSN 0305-0548. doi: 10.1016/j.cor.2009.09.014. URL `http://www.sciencedirect.com/science/article/pii/S0305054809002329`. Algorithmic and Computational Methods in Retrial Queues. 31, 35, 39, 95

J. Orozco. *A Microscopic Traffic Simulation Based Decision Support System for Real-Time Fleet Management*. PhD thesis, Universitat Politecnica de Catalunya, 2011. 2

J. Partyka and R. Hall. Software survey - vehicle routing: On the road to innovation. *OR/MS Today*, pages 38–45, february 2012. 23, 205

P. Pellegrini. Application of two nearest neighbor approaches to a rich vehicle routing problem. Technical report, Technical Report TR/IRIDIA/2005-15, IRIDIA, Université Libre de Bruxelles, Belgium, 2005. 27, 35, 39

P. Pellegrini, D. Favaretto, and E. Moretti. Multiple ant colony optimization for a rich vehicle routing problem: A case study. In B. Apolloni, R.J. Howlett, and L. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4693 of *Lecture Notes in Computer Science*, pages 627–634. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-74826-7. doi: 10.1007/978-3-540-74827-4_79. URL http://dx.doi.org/10.1007/978-3-540-74827-4_79. 24, 29, 35, 39

P.H.V. Penna, A. Subramanian, and L.S. Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232, 2013. ISSN 1381-1231. doi: 10.1007/s10732-011-9186-y. URL http://dx.doi.org/10.1007/s10732-011-9186-y. 57, 58, 207

G. Pesant, M. Gendreau, and J.M. Rousseau. Genius-cp: A generic single-vehicle routing algorithm. In G. Smolka, editor, *Principles and Practice of Constraint Programming-CP97*, volume 1330 of *Lecture Notes in Computer Science*, pages 420–434. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-63753-0. doi: 10.1007/BFb0017457. URL http://dx.doi.org/10.1007/BFb0017457. 208

G. Pesant, M. Gendreau, J.Y. Potvin, and J.M. Rousseau. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, 32(1):12–29, 1998. 208

A. Pessoa, M.P. de Aragão, and E. Uchoa. Robust branch-cut-and-price algorithms for vehicle routing problems. In B. Golden, S. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*, pages 297–325. Springer US, 2008. ISBN 978-0-387-77777-1. doi: 10.1007/978-0-387-77778-8_14. URL http://dx.doi.org/10.1007/978-0-387-77778-8_14. 97, 102, 110

R.J. Petch and S. Salhi. A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics*, 133(1-3):69–92, 2003. ISSN 0166-218X. doi: 10.1016/S0166-218X(03)00434-7. URL http://www.sciencedirect.com/science/article/pii/S0166218X03004347. International Symposium on Combinatorial Optimisation. 58

M. Pinedo. *Scheduling: theory, algorithms, and systems*. Springer Science+ Business Media, 2012. 47

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007. ISSN 0305-0548. doi: 10.1016/j.cor.2005.09.012. URL http://www.sciencedirect.com/science/article/pii/S0305054805003023. 28, 35, 39, 97, 102, 103, 205, 207

A. Poot, G. Kant, and A.P.M. Wagelmans. A savings based method for real-life vehicle routing problems. *Journal of the Operational Research Society*, 53(1):57–68, 2002. 59

J.Y. Potvin and S. Bengio. The vehicle routing problem with time windows part ii: genetic search. *INFORMS journal on Computing*, 8(2):165–172, 1996. 121

J.Y. Potvin and J.M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340, 1993. ISSN 0377-2217. doi: 10.1016/0377-2217(93)90221-8. URL http://www.sciencedirect.com/science/article/pii/0377221793902218. 123

J.Y. Potvin, T. Kervahut, B.L. Garcia, and J.M. Rousseau. The vehicle routing problem with time windows part i: tabu search. *INFORMS Journal on Computing*, 8(2):158–164, 1996. 121

E. Prescott-Gagnon, G. Desaulniers, and L.M. Rousseau. Heuristics for an oil delivery vehicle routing problem. *Les Cahiers du GERAD G-2010-63*, 10 2010. 32, 35, 39, 95

C. Prins. Efficient heuristics for the heterogeneous fleet multitrip vrp with application to a large-scale real case. *J. Math. Model. Algorithms*, pages 135–150, 2002. 9, 58, 59, 61, 65, 67, 68, 95, 99, 218

C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004. ISSN 0305-0548. doi: 10.1016/S0305-0548(03)00158-8. URL `http://www.sciencedirect.com/science/article/pii/S0305054803001588`. 34, 207

C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Eng. Appl. Artif. Intell.*, 22 (6):916–928, September 2009. ISSN 0952-1976. doi: 10.1016/j.engappai.2008.10.006. URL `http://dx.doi.org/10.1016/j.engappai.2008.10.006`. 57, 58, 207

J. Privé, J. Renaud, F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, 57(9):1045–1052, 2005. 55

M. Protonotarios, G. Mourkousis, I. Vyridis, and T. Varvarigou. Very large scale vehicle routing with time windows and stochastic demand using genetic algorithms with parallel fitness evaluation. In M. Bubak, H. Afsarmanesh, B. Hertzberger, and R. Williams, editors, *High Performance Computing and Networking*, volume 1823 of *Lecture Notes in Computer Science*, pages 467–476. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67553-2. doi: 10.1007/3-540-45492-6_47. URL `http://dx.doi.org/10.1007/3-540-45492-6_47`. 141

C.R. Reeves. Genetic algorithms. In M. Gendreau and J.Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science*, pages 109–139. Springer US, 2010. ISBN 978-1-4419-1663-1. doi: 10.1007/978-1-4419-1665-5_5. URL `http://dx.doi.org/10.1007/978-1-4419-1665-5_5`. 44

C. Rego. Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. *Parallel Computing*, 27(3):201–222, 2001. ISSN 0167-8191. doi: 10.1016/S0167-8191(00)00102-2. URL `http://www.sciencedirect.com/science/article/pii/S0167819100001022`. 141

M.I. Reiman, R. Rubio, and L.M. Wein. Heavy traffic analysis of the dynamic stochastic inventory-routing problem. *Transportation Science*, 33(4):361–380, 1999. 173

M. Reimann and H. Ulrich. Comparing backhauling strategies in vehicle routing using ant colony optimization. *Central European Journal of Operations Research*, 14:105–123, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0163-8. URL `http://dx.doi.org/10.1007/s10100-006-0163-8`. 28, 35, 39

J. Renaud and F.F. Boctor. A sweep-based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, 140(3):618–628, 2002. ISSN 0377-2217. doi: 10.1016/S0377-2217(01) 00237-5. URL `http://www.sciencedirect.com/science/article/pii/S0377221701002375`. 57, 58

M.G.C. Resende and C.C. Ribeiro. 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. ISBN 978-1-4419-1663-1. doi: 10.1007/978-1-4419-1665-5_10. URL `http://dx.doi.org/10.1007/978-1-4419-1665-5_10`. 45

J. Rieck and J. Zimmermann. A hybrid algorithm for vehicle routing of less-than-truckload carriers. In K. Sörensen, M. Sevaux, W. Habenicht, and M.J. Geiger, editors, *Metaheuristics in the Service Industry*, volume 624 of *Lecture Notes in Economics and Mathematical Systems*, pages 155–171. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-00938-9. doi: 10.1007/978-3-642-00939-6_9. URL `http://dx.doi.org/10.1007/978-3-642-00939-6_9`. 81

J. Rieck and J. Zimmermann. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, 181:337–358, 2010. ISSN 0254-5330. doi: 10.1007/s10479-010-0748-4. URL http://dx.doi.org/10.1007/s10479-010-0748-4. 25, 30, 35, 39, 95

D. Riera, A. Juan, D. Guimarans, and E. Pagans. A constraint programming-based library for the vehicle routing problem. In *21st European Modeling and Simulation Symposium*, pages 105–110, Santa Cruz de Tenerife: Universidad de La Laguna - Canary Islands, Spain, september 2009. ISBN 9788469254141. 209

A.E. Rizzoli, R. Montemanni, E. Lucibello, and L.M. Gambardella. Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, 1:135–151, 2007. ISSN 1935-3812. doi: 10.1007/s11721-007-0005-x. URL http://dx.doi.org/10.1007/s11721-007-0005-x. 29, 35, 39

A. Rodríguez and R. Ruiz. A study on the effect of the asymmetry on real capacitated vehicle routing problems. *Computers & Operations Research*, 39(9):2142–2151, 2012. 9, 96, 101, 102

S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006a. ISSN 0377-2217. doi: 10.1016/j.ejor.2004.09.004. URL http://www.sciencedirect.com/science/article/pii/S0377221704005831. Feature Cluster: Heuristic and Stochastic Methods in Optimization Feature Cluster: New Opportunities for Operations Research. 28, 30, 33, 35, 39, 205, 207

S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006b. ISSN 00411655. URL http://0-search.ebscohost.com.cataleg.uoc.edu/login.aspx?direct=true&AuthType=ip,cookie,url,uid&db=bth&AN=23022889&lang=es&site=ehost-live. 28, 35, 39, 205, 207

L.M. Rousseau, M. Gendreau, and G. Pesant. Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, 8:43–58, 2002. ISSN 1381-1231. doi: 10.1023/A:1013661617536. URL http://dx.doi.org/10.1023/A%3A1013661617536. 209

L. Ruinelli. *Column Generation for a Rich VRP: VRP with simultaneous distribution, collection and pickup-and-delivery.* PhD thesis, University of Applied Sciences and Arts of Southern Switzerland, 2011. 32, 35, 39

R. Ruiz, C. Maroto, and J. Alcaraz. A decision support system for a real vehicle routing problem. *European Journal of Operational Research*, 153(3):593–606, 2004. ISSN 0377-2217. doi: 10.1016/S0377-2217(03)00265-0. URL http://www.sciencedirect.com/science/article/pii/S0377221703002650. EURO Young Scientists. 55

S. Salhi and R.J. Petch. A ga based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modelling and Algorithms*, 6:591–613, 2007. ISSN 1570–1166. doi: 10.1007/s10852-007-9069-2. URL http://dx.doi.org/10.1007/s10852-007-9069-2. 97

J. Sanders and E. Kandrot. *CUDA by example: an introduction to general-purpose GPU programming.* Addison-Wesley Professional, 2010. 143

C.G. Santillán, L.C. Reyes, M.L.M. Rodríguez, J.J.G. Barbosa, O.C. López, G.R. Zarate, and P. Hernández. Variants of vrp to optimize logistics management problems. In Ortiz Zezzatti, Carlos Alberto Ochoa, Camelia Chira, Arturo Hernandez, and Miguel Basurto, editors, *Logistics Management and Optimization Through Hybrid Artificial Intelligence Systems*, volume 2012, pages 207–237. IGI Global, 2012. doi: 10.4018/978-1-4666-0297-7.ch008. 33, 35, 39

M.W.P. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154, 1992. 122

REFERENCES AND BIBLIOGRAPHY

A. Sbihi and R.W. Eglese. Combinatorial optimization and green logistics. *4OR*, 5(2):99–116, 2007. ISSN 1619-4500. doi: 10.1007/s10288-007-0047-3. URL http://dx.doi.org/10.1007/s10288-007-0047-3. 221, 225

V. Schmid, K.F. Doerner, and G. Laporte. Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224(3):435–448, 2013. ISSN 0377-2217. doi: 10.1016/j.ejor.2012.08.014. URL http://www.sciencedirect.com/science/article/pii/S0377221712006376. 25

A. Şen and K. Bülbül. A survey on multi trip vehicle routing problem. In *Proceedings of VI. International Logistics & Supply Chain Congress 2008*, stanbul, Trkiye, 2008. İstanbul Bilgi Üniversitesi. 58

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In M. Maher and J.F. Puget, editors, *Principles and Practice of Constraint Programming - CP98*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer-Verlag, Berlin Heidelberg, 1998. ISBN 978-3-540-65224-3. doi: 10.1007/3-540-49481-2_30. URL http://dx.doi.org/10.1007/3-540-49481-2_30. In Proceedings of 4th International Conference on Principles and Practice of Constraint Programming (CP-98). 28, 208

M.M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987. 9, 27, 28, 31, 33, 121, 123, 124, 125, 126

K. Sörensen. Route stability in vehicle routing decisions: a bi-objective approach using metaheuristics. *Central European Journal of Operations Research*, 14:193–207, 2006. ISSN 1435-246X. doi: 10.1007/s10100-006-0168-3. URL http://dx.doi.org/10.1007/s10100-006-0168-3. 29, 35, 39

K. Sörensen, M. Sevaux, and P. Schittekat. “multiple neighbourhood” search in commercial vrp packages evolving towards self-adaptive methods. In C. Cotta, M. Sevaux, and K. Sörensen, editors, *Adaptive and Multilevel Metaheuristics*, volume 136 of *Studies in Computational Intelligence*, pages 239–253. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-79437-0. doi: 10.1007/978-3-540-79438-7_12. URL http://dx.doi.org/10.1007/978-3-540-79438-7_12. 23

S.K. Srivastava. Green supply-chain management: A state-of-the-art literature review. *International Journal of Management Reviews*, 9(1):53–80, 2007. ISSN 1468-2370. doi: 10.1111/j.1468-2370.2007.00202.x. URL http://dx.doi.org/10.1111/j.1468-2370.2007.00202.x. 225

L. Steg and R. Gifford. Sustainable transportation and quality of life. *Journal of Transport Geography*, 13 (1):59–69, 2005. ISSN 0966-6923. doi: http://dx.doi.org/10.1016/j.jtrangeo.2004.11.003. URL http://www.sciencedirect.com/science/article/pii/S0966692304000870. Sustainability and the Interaction Between External Effects of Transport (Part Special Issue, pp. 23-99). 1

T. Stützle and H. Hoos. Max-min ant system and local search for the traveling salesman problem. In *Evolutionary Computation, 1997., IEEE International Conference on*, pages 309–314, apr 1997. doi: 10.1109/ICEC.1997.592327. 29

A. Subramanian. *Heuristic, Exact and Hybrid Approaches for Vehicle Routing Problems*. PhD thesis, UNIVERSIDADE FEDERAL FLUMINENSE, 2012. 18, 32, 39, 206, 207

A. Subramanian, L. Drummonda, C. Bentesb, L. Ochia, and R. Farias. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911, 2010. ISSN 0305-0548. 142

A. Subramanian, P.H.V. Penna, E. Uchoa, and L.S. Ochi. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295, 2012. 18, 32, 35, 39, 57, 58, 207

I. Sungur, F. Ordóñez, and M. Dessouky. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5):509–523, 2008. doi: 10.1080/07408170701745378. URL `http://www.tandfonline.com/doi/abs/10.1080/07408170701745378`. 150

E.D. Taillard. A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO - Operations Research*, 33:1–14, 0 1999. ISSN 1290-3868. doi: 10.1051/ro:1999101. URL `http://dx.doi.org/10.1051/ro:1999101`. 9, 57, 58, 67, 219

E.D. Taillard, G. Laporte, and M. Gendreau. Vehicle routeing with multiple use of vehicles. *Journal of the Operational research society*, pages 1065–1070, 1996. 58

E.G. Talbi. *Metaheuristics: from design to implementation.* John Wiley & Sons, Inc., 2009. ISBN 9780470496916. doi: 10.1002/9780470496916.fmatter. URL `http://dx.doi.org/10.1002/9780470496916.fmatter`. 3, 17, 18, 19, 139

E.G. Talbi. *Parallel Combinatorial Optimization.* John Wiley & Sons, Inc., 2012. ISBN 978-0-471-72101-7. 6, 139, 143

K.C. Tan, L.H. Lee, Q.L. Zhu, and K. Ou. Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering*, 15(3):281–295, 2001. ISSN 0954-1810. doi: 10.1016/S0954-1810(01)00005-X. URL `http://www.sciencedirect.com/science/article/pii/S095418100100005X`. 123, 126

K.C. Tan, C.Y. Cheong, and C.K. Goh. Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *European Journal of Operational Research*, 177(2):813–839, 2007. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.12.029. URL `http://www.sciencedirect.com/science/article/pii/S0377221706000208`. 150

C.D Tarantilis, C.T Kiranoudis, and V.S Vassiliadis. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research*, 152(1):148–158, 2004. ISSN 0377-2217. doi: 10.1016/S0377-2217(02)00669-0. URL `http://www.sciencedirect.com/science/article/pii/S0377221702006690`. 57, 58

F. Teklu, A. Sumalee, and D. Watling. A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):31–43, 2007. ISSN 1467-8667. doi: 10.1111/j.1467-8667.2006.00468.x. URL `http://dx.doi.org/10.1111/j.1467-8667.2006.00468.x`. 130

P. Toth and D. Vigo. *An overview of vehicle routing problems*, pages 1–26. SIAM - Society for Industrial and Applied Mathematics, Philadelphia, 2001. 2, 8, 12, 14, 24, 25, 208, 210

P. Trudeau and M. Dror. Stochastic inventory routing: Route design with stockouts and route failures. *Transportation Science*, 26(3):171–184, 1992. 173

T.H. Truong and F. Azadivar. Simulation based optimization for supply chain configuration design. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 2, pages 1268–1275, 2003. doi: 10.1109/WSC.2003.1261560. 130

S. Ubeda, F.J. Arcelus, and J. Faulin. Green logistics at eroski: A case study. *International Journal of Production Economics*, 131(1):44–51, 2011. ISSN 0925–5273. doi: 10.1016/j.ijpe.2010.04.041. URL `http://www.sciencedirect.com/science/article/pii/S092552731000174X`. Innsbruck 2008. 225

M. Vallejo, P.A. Vargas, and D.W. Corne. A fast approximative approach for the vehicle routing problem. In *Computational Intelligence (UKCI), 2012 12th UK Workshop on*, pages 1 –8, sept. 2012. doi: 10.1109/UKCI.2012.6335752. 33, 35, 39, 95

P. Van-Hentenryck and R. Bent. *Online stochastic combinatorial optimization*. Boston: The MIT Press, 2009. 149

B. Verweij, S. Ahmed, A.J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24:289–333, 2003. ISSN 0926-6003. doi: 10.1023/A:1021814225969. URL `http://dx.doi.org/10.1023/A%3A1021814225969`. 133

T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. Technical report, Tech. rep., CIRRELT 2012-05, 2012a. 33

T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. A unified solution framework for multi-attribute vehicle routing problems. Technical report, Tech. rep., CIRRELT 2012-23, 2012b. 33, 34, 35, 39, 40, 207

T. Vidal, T.G. Crainic, M. Gendreau, and C. Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1):475–489, 2013. ISSN 0305-0548. doi: 10.1016/j.cor.2012.07.018. URL `http://www.sciencedirect.com/science/article/pii/S0305054812001645`. 34, 35, 39, 207

D. Vigo. A heuristic algorithm for the asymmetric capacitated vehicle routing problem. *European Journal of Operational Research*, 89(1):108–126, FEB 22 1996. ISSN 0377-2217. 96

N.A. Wassan and I.H. Osman. Tabu search variants for the mix fleet vehicle routing problem. *The Journal of the Operational Research Society*, 53(7):768–782, 2002. URL `http://search.proquest.com/docview/231350071?accountid=15299`. Copyright - Copyright (c) 2002 Palgrave Macmillan Ltd; Idioma del resumen - English; Pginas - 768-782; ProQuest ID - 231350071; ltima actualizacin - 2010-06-10; CODEN - OPRQAK; Lugar de publicacin - Houndmills; Autor de la institucin corporativa - Wassan, N A; Osman, I H; DOI - 132905641; 1870001; 12717; OPRQAK; OQT. 57, 58

M. Wen. *Rich Vehicle Routing Problems and Applications*. PhD thesis. Management Department, Technical University of Denmark, 2010. ISBN 978-87-90855-89-5. 30

M. Wen, J. Larsen, J. Clausen, J.F. Cordeau, and G. Laporte. Vehicle routing with cross-docking. *Journal of the Operational Research Society*, 60(12):1708–1718, 2008. ISSN 0160-5682. doi: 10.1057/jors.2008.108. 30, 35, 39

M. Wen, J.F. Cordeau, G. Laporte, and J. Larsen. The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37(9):1615–1623, 2010a. ISSN 0305-0548. doi: 10.1016/j.cor.2009.12.002. URL `http://www.sciencedirect.com/science/article/pii/S0305054809003165`. 30, 35, 39

M. Wen, E. Krapper, J. Larsen, and T.K. Stidsen. A multilevel variable neighborhood search heuristic for a practical vehicle routing and driver scheduling problem. Technical report, Technical Report, Management Department, Technical University of Denmark, 2010b. 30

Tony Whiteing and Damian Stantchev. Environmental aspects of sustainable mobility. Technical report, European Commission DG Energy and Transport - Transport Research Knowledge Centre, 2008. URL `http://www.transport-research.info`. 1

F. Xiang, L. Junzhou, W. Jieyi, and G. Guanqun. Qos routing based on genetic algorithm. *Computer Communications*, 22(15-16):1392–1399, 1999. ISSN 0140-3664. doi: 10.1016/S0140-3664(99)00113-9. URL `http://www.sciencedirect.com/science/article/pii/S0140366499001139`. 130

G. Yazgı-Tütüncü. An interactive gramps algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls. *European Journal of Operational Research*, 201(2):593–600, 2010. ISSN 0377-2217. doi: 10.1016/j.ejor.2009.03.044. URL `http://www.sciencedirect.com/science/article/pii/S0377221709001957`. 57, 58

B. Yu and Z. Zhen-Yang. An ant colony optimization model: The period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 47(2):166–181, 2011. ISSN 1366-5545. doi: 10.1016/j.tre.2010.09.010. URL http://www.sciencedirect.com/science/article/pii/S136655451000089X. 39, 144

Y. Yu, F. Chu, and H. Chen. A model and algorithm for large scale stochastic inventory routing problem. In *Service Systems and Service Management, 2006 International Conference on*, volume 1, pages 355–360, Troyes, France, oct. 2006. doi: 10.1109/ICSSSM.2006.320640. 175

H. Zhang and H. Li. Simulation-based optimization for dynamic resource allocation. *Automation in Construction*, 13(3):409–420, 2004. ISSN 0926-5805. doi: 10.1016/j.autcon.2003.12.005. URL http://www.sciencedirect.com/science/article/pii/S0926580504000044. 130

16

Appendix

16.1 Paper Award Selection

José Cáceres-Cruz

De:	ICORES Secretariat [icores.secretariat@insticc.org]
Enviat:	Sunday, January 13, 2013 8:08 PM
Per a:	jose.caceres.cruz@gmail.com
A/c:	roman.buil@uab.cat; rherrero.math@gmail.com; drierat@uoc.edu; ajuanp@uoc.edu
Tema:	Candidate to ICORES 2013 best student paper award

Dear Mr. Jose Caceres-Cruz,

We would like to inform you that your paper 29 entitled "Hybrid Approach for Solving an Asymmetric Heterogeneous Vehicle Routing Problem in a Real Urban Context" is in a short list of candidates to win the ICORES 2013 best student paper award, based on marks provided by the reviewers. Only those papers presented at the conference that also get good marks by the session chairs qualify for the final stage. The awards will be announced at the closing session, on 18 February 13, therefore perhaps you'd like to organize your program in order to attend the closing session, just in case your paper is actually selected as the best.
A final note to mention that when a student paper quality is exceptional, it is possible that the best paper award is won by a candidate to best student paper award.

Kind regards
Vera Coelho
ICORES 2013 Secretariat

[Número de página]

Figure 16.1: Response email of ICORES committee of paper (Cáceres-Cruz et al., 2013).

16.2 First Page of Publications derived from this thesis

Biased Randomization of Classical Heuristics

Angel A. Juan, IN3-Open University of Catalonia, Spain
José Cáceres-Cruz, IN3-Open University of Catalonia, Spain
Sergio González-Martín, IN3-Open University of Catalonia, Spain
Daniel Riera, IN3-Open University of Catalonia, Spain
Barry B. Barrios, IN3-Open University of Catalonia, Spain

INTRODUCTION

In the context of combinatorial optimization problems, this chapter discusses how to randomize classical heuristics in order to transform these deterministic procedures into more efficient probabilistic algorithms. This randomization process can be performed by using a uniform probability distribution or, even more interesting, by using a non-symmetric distribution.

Combinatorial Optimization Problems (COPs) have posed numerous challenges to the human mind throughout the past decades. From a theoretical perspective, they have a well-structured definition consisting of an objective function that needs to be minimized or maximized, and a series of constraints that must be satisfied. From a theoretical point of view, these problems have an interest on their own due to the mathematics involved in their modeling, analysis and solution. However, the main reason for which they have been so actively investigated is the tremendous amount of real-life applications that can be successfully modeled as a COP. Thus, for example, decision-making processes in fields such as logistics, transportation, and manufacturing contain plentiful hard challenges that can be expressed as COPs (Faulin et al., 2012; Montoya et al., 2011). Accordingly, researchers from different areas –e.g. Applied Mathematics, Operations Research, Computer Science, and Artificial Intelligence– have directed their efforts to conceive techniques to model, analyze, and solve COPs.

A considerable number of methods and algorithms for searching optimal or near-optimal solutions inside the solution space have been developed. In some small-sized problems, the solution space can be exhaustively explored. For those instances, efficient exact methods can usually provide the optimal solution in a reasonable time. Unfortunately, the solution space in most COPs is exponentially astronomical. Thus, in medium- or large-size problems, the solution space is too large and finding the optimum in a reasonable amount of time is not a feasible task. An exhaustive method that checks every single candidate in the solution space would be of very little help in these cases, since it would take exponential time. Therefore, a large amount of heuristics and metaheuristics have been developed in order to obtain near-optimal solutions, in reasonable computing times, for medium- and large-size problems, some of them even considering realistic constraints.

The main goal of this chapter is to present a hybrid scheme which combines classical heuristics with biased-randomization processes. As it will be discussed later, this hybrid scheme represents an efficient, relatively simple, and flexible way to deal with several COPs in different fields, even when considering realistic constraints.

BACKGROUND

In the context of this chapter, we will refer to any algorithm which makes use of pseudo-random numbers to perform 'random' choices during the exploration of the solution space by the term randomized search method, or simply randomized algorithm. This includes most current metaheurisics and stochastic local-search processes. Thus, since it does not follow a determinist path, even for the same input, a randomized

Figure 16.2: Front page of publication Juan et al. (2014a).

Original Article

Vehicle routing in a Spanish distribution company: Saving using a savings-based heuristic

Alex Grasas[a,*], Jose Caceres-Cruz[b], Helena R. Lourenço[a], Angel A. Juan[b] and Mercè Roca[c]

[a]Department of Economics and Business, Universitat Pompeu Fabra, Ramon Trias Fargas 25–27, 08005 Barcelona, Spain.
E-mails: alex.grasas@upf.edu; helena.ramalhinho@upf.edu
[b]Department of Computer Science, Multimedia and Telecommunication, IN3-Open University of Catalonia, Roc Boronat 117, 08018 Barcelona, Spain.
E-mails: jcaceresc@uoc.edu; ajuanp@uoc.edu
[c]ESCI International Business School, Pg. Pujades 1, 08003, Barcelona, Spain.
E-mail: merce.roca@esci.upf.edu

*Corresponding author.

Abstract In this article we present a Vehicle Routing Problem (VRP) faced by a large distribution company in the Northeast of Spain. The company distributes products from its central facilities to a chain of around 400 stores all over the country. One of the peculiarities of the VRP of this company – which is common among real-life VRPs – is the presence of a heterogeneous fleet where vehicles with different capacities can make multiple trips during a single day. This variant of the problem, which we refer as Heterogeneous Fleet and Multi-trip VRP, has been barely studied in the literature. To solve the problem, we use an algorithm based on the well-known savings heuristic with a biased-randomization effect and three local search operations. Our approach is simple to implement as it needs few parameters and no fine-tuning processes, which are usually cumbersome and require experts' involvement. We obtain savings of around 12 per cent in transportation costs, which represent around €30 000 saved per week.
OR Insight (2013) **26,** 191–202. doi:10.1057/ori.2013.2; published online 13 March 2013

Keywords: heterogeneous vehicle routing problem; multi-trip vehicle routing; savings heuristic; randomized algorithms; real-life applications

Received 11 October 2012; accepted 31 January 2013 after one revision

 OR Insight Vol. 26, 3, 191–202
www.palgrave-journals.com/ori/

Figure 16.3: Front page of publication Grasas et al. (2013).

Aplicación de un algoritmo randomizado a un problema real de enrutamiento de vehículos heterogéneos

José Cáceres[1], Alex Grasas[2], Helena R. Lourenço[3], Angel A. Juan[4], Mercè Roca[5], Rosa Colomé[6]
[1,4] Universitat Oberta de Catalunya, Barcelona, España. [2,3] Universitat Pompeu Fabra, Barcelona, España. [5,6] ESCi, Barcelona, España.

Resumen—**La problemática del enrutamiento de vehículos cobra cada vez más importancia en el plano empresarial y estatal. Esta área de estudio ha experimentado grandes avances teóricos, pero se ha mantenido a cierta distancia de la práctica. La mejora en las técnicas para obtener soluciones factibles y de calidad está permitiendo aplicar resultados teóricos en la resolución de escenarios reales. En este estudio, se presenta la resolución de un problema de enrutamiento de vehículos con una flota heterogénea utilizando un algoritmo que combina una heurística clásica con un factor aleatorio y una memoria temporal de las mejores rutas encontradas. El experimento se ha ejecutado con los datos de una empresa española de distribución con más de 370 tiendas en el noreste de España. Los resultados reflejan mejoras con respecto al plan de rutas concebido de forma manual por los expertos de la empresa.**

Palabras clave—**Problema de Enrutamiento de Vehículos Heterogéneos, Algoritmos Randomizados, Heurísticas.**

I. Introducción

En los últimos años, las empresas de logística y transporte se están enfrentando a situaciones cada vez más exigentes y con menos recursos disponibles, producto de la inestabilidad de los mercados y el competitivo contexto empresarial. El transporte por carretera representa el principal medio para el intercambio de bienes en Europa y otras partes del mundo. Desde el año 2000, el impacto económico y ambiental asociado al transporte terrestre ha ido incrementando. Los gobiernos y empresas de todo el mundo han posado su atención en la optimización de los procesos logísticos y de distribución terrestres. Dicha optimización se ha hecho necesaria en todo tipo de empresa (grande, mediana, o pequeña) para beneficiar la calidad del servicio, la satisfacción del cliente, y la reducción de costes.

Distintas áreas del conocimiento han enfocado sus esfuerzos para concebir técnicas útiles para este tipo de problemática. La optimización de procesos parece, a simple vista, un marco natural para las Matemáticas Aplicadas y la Investigación Operativa. A este grupo de disciplinas, se le suma la Ciencia de la Computación que, con sus continuos avances tecnológicos, colabora en el desarrollo de algoritmos de optimización eficientes y personalizables a cada problemática concreta. A esto hay que sumar, además, el progresivo aumento en la capacidad de cómputo que ofrecen los procesadores modernos, así como las técnicas de paralelización que se pueden emplear en entornos *multi-core*, *cluster*, o *grid*.

Este estudio presenta la aplicación de un algoritmo híbrido para la resolución de un caso real en una empresa de distribución de alimentos española. Las siguientes secciones describen el marco teórico y algunos trabajos relacionados con el problema de optimización de rutas, el contexto actual de planificación de rutas de la empresa considerada, la metodología de resolución aplicada, algunos resultados preliminares y, finalmente, las conclusiones.

II. Trabajos Previos

El Problema de Enrutamiento de Vehículos (VRP) se ha estudiado durante más de 50 años (**Laporte, 2009**). Su versión más simple es conocida como el Problema de Enrutamiento de Vehículos con Capacidades limitadas (CVRP), definido por **Datzing & Ramser (1959)**. Este problema consiste en definir un conjunto de rutas para servir a un conjunto de clientes con una flota de vehículos desde un almacén o nodo central. Cada vehículo tiene la misma capacidad (flota homogénea) y cada cliente tiene una cierta demanda conocida que debe ser satisfecha. Además, existe un coste asociado al traslado de un vehículo desde un nodo a otro, que bien podría representar las distancias, el tiempo de viaje o algún otro coste en particular. El objetivo es definir las rutas que minimicen el coste total, la distancia recorrida, o el tiempo empleado, de manera que la demanda de cada nodo cliente sea satisfecha y que la capacidad máxima de cada camión sea respetada.

En las últimas décadas, diferentes enfoques para el CVRP han sido explorados (**Toth y Vigo 2002, Golden et al. 2008, Juan et al. 2011a, Faulin y Juan 2008**). Estos enfoques tienen un amplio espectro que se inicia con el uso de métodos de optimización pura, como la programación lineal, para resolver problemas de tamaño pequeño con restricciones relativamente simples, hasta el uso de heurísticas y metaheurísticas que ofrecen soluciones casi óptimas para los problemas de mediano y gran

[1] E-mail: jcaceresc@uoc.edu
[2] E-mail: alex.grasas@upf.edu
[3] E-mail: helena.ramalhinho@upf.edu
[4] E-mail: ajuanp@uoc.edu
[5] E-mail: merce.roca@esci.upf.edu
[6] E-mail: rosa.colome@esci.upf.edu

Figure 16.4: Front page of publication Cáceres-Cruz et al. (2012a).

European J. of Industrial Engineering, Vol. X, No. Y, xxxx 1

A successive approximations method for the heterogeneous vehicle routing problem: analysing different fleet configurations

Angel A. Juan

Department of Computer Science,
Multimedia and Telecommunication,
Open University of Catalonia,
08018 Barcelona, Spain
E-mail: ajuanp@uoc.edu

Javier Faulin*

Department of Statistics and Operations Research,
Campus Arrosadia, Public University of Navarra,
Los Magnolios Building, First Floor,
31006 Pamplona, Navarre, Spain
E-mail: javier.faulin@unavarra.es
*Corresponding author

Jose Caceres-Cruz, Barry B. Barrios and Enoc Martinez

Department of Computer Science,
Multimedia and Telecommunication,
Open University of Catalonia,
08018 Barcelona, Spain
E-mail: jose.caceres.cruz@gmail.com
E-mail: barry.brian.barrios@gmail.com
E-mail: e@enocmartinez.com

Abstract: In this paper, we propose a relatively simple-to-implement procedure for solving the heterogeneous-fleet vehicle routing problem (HeVRP), in which different types of vehicle loading capacities are considered. Our approach is based on the so called successive approximations method (SAM), which 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 (HoVRP) is solved. After several rounds, a global solution for the HeVRP is obtained by merging routes from different HoVRP solutions. In the first part of the paper, we analyse how distance-based costs vary when slight deviations from the homogeneous fleet assumption are considered. In the second part of the article, the SAM approach is adapted so it can simultaneously deal with both fixed and variable costs in HeVRPs. An experimental comparison is then made with other HeVRP algorithms.
[Received: November 12, 2012; Revised: March 25, 2013; Revised: June 29, 2013; Accepted: July 5, 2013]

Figure 16.5: Front page of publication Juan et al. (2014b).

Solving Vehicle Routing Problems with Asymmetric Costs and Heterogeneous Fleets

Rosa Herrero

Telecommunications and Systems Engineering Dept., Autonomous University of Barcelona, Spain

Alejandro Rodríguez

Departamento de Organización de Empresas, Universitat Politècnica de València, València, Spain

José Cáceres-Cruz, Angel A. Juan

Computer Science Dept., IN3-Open University of Catalonia, Barcelona, Spain

Abstract: The Vehicle Routing Problem (VRP) is a flourishing research area with clear applications to real-life distribution companies. However, most VRP-related academic articles assume the existence of a homogeneous fleet of vehicles and/or a symmetric cost matrix. These assumptions are not always reasonable in real-life scenarios. To contribute closing this gap between theory and practice, we propose a hybrid methodology for solving the Asymmetric and Heterogeneous Vehicle Routing Problem (AHVRP). In our approach we consider: (i) different types of vehicle loading capacities (heterogeneous fleets), and (ii) asymmetric distance-based costs. The proposed approach combines a randomized version of a well-known savings heuristic with several local searches specifically adapted to deal with the asymmetric nature of costs. A computational experiment allows us to discuss the efficiency of our approach and also to analyze how routing costs vary when slight departures from the homogeneous fleet assumption are considered.

Keywords: Real-Life Vehicle Routing Problem; Heterogeneous Fleets; Asymmetric Costs; Randomized Algorithms.

1 Introduction

Vehicle Routing Problems (VRPs) deal with the physical distribution of goods from a central depot to customers, see for instance **?** and **?**. The best-known VRP variant is the so-called Capacitated Vehicle Routing Problem ($CVRP$). In the CVRP it is assumed the existence of a homogeneous fleet of vehicles with limited capacity. Another frequent assumption is that distance-based costs associated with traveling from one node i (customer or depot) to another node j, c_{ij}, are symmetric, i.e., $c_{ij} = c_{ji}$ for all pair of nodes. A wide number of VRP variants have been developed during the last years, each of them considering different

Figure 16.6: Front page of publication Herrero et al. (2014).

Multi-start Approach for Solving an Asymmetric Heterogeneous Vehicle Routing Problem in a Real Urban Context

José Cáceres-Cruz[1], Daniel Riera[1], Roman Buil[2], Angel A. Juan[1] and Rosa Herrero[2]
[1]*IN3-Computer Science Department, Open University of Catalonia, Barcelona, Spain*
[2]*Department of Telecommunications and Systems Engineering, Universitat Autònoma de Barcelona, Bellaterra, Spain*
{jcaceresc, drierat, ajuanp}@uoc.edu, roman.buil@uab.cat, rherrero.math@gmail.com

Keywords: Heterogeneous Vehicle Routing Problem, Asymmetric Cost Matrix, Clarke and Wright, Randomized Algorithms, Heuristics.

Abstract: Urban transportation is a strategic domain that has become an important issue for client satisfaction in distribution companies. In academic literature, this problem is categorized as a Vehicle Routing Problem, a popular research stream that has undergone significant theoretical advances but has remained far from practice implementations. Most Vehicle Routing Problems usually assume homogenous fleets, that is, all vehicles are considered of the same type and size. In reality, this is usually not the case as most companies use different types of trucks to distribute their products. Also, researchers consider symmetric distances between customers. However, in intra-urban distribution it is more appropriate to consider asymmetric costs. In this study, we address the Heterogeneous Fixed Fleet Vehicle Routing Problem with some additional constraints: (a) Asymmetric Cost matrix, (b) Service Times and (c) Routes Length restrictions. Our objective function is to reduce the total routing costs. We present an approach using a multi-start algorithm that combines a randomized Clarke & Wright's Savings heuristic and a local search procedure. We execute our algorithm with data from a company that distributes food to more than 50 customers in Barcelona. The results reveal promising improvements when compared to an approximation of the company's route planning.

1 INTRODUCTION

In the last years, logistics and transportation companies are facing growingly demanding situations with fewer available resources. Market instability and the competitive business environment have caused an increasing optimization of logistic processes. Several fields of research have directed their efforts to conceive techniques to fulfil this purpose, like applied mathematics, operations management and computer sciences. The main challenge for these theoretical domains is the consideration of real contexts including real constraints into their approaches.

Vehicle routing is a complex logistics management problem and represents a key phase for the logistic optimization. There are many variations for the routing problem. Particularly, we have considered a special variant where several restrictions are considered at the same time. The set of defined constraints are taken from a real case provided by a food distribution company located in Barcelona, Spain. The distribution inside cities has special conditions like little time for delivery, congestion, traffic lights, and different types of vehicles related to the size and velocity issues. Also, there are many possible configurations (routes) to visit a customer because the street direction creates a special network of available arcs. The purpose of this study is to develop and apply a randomized multi-start algorithm based on a Clarke & Wright savings heuristic for the Asymmetric Heterogeneous Fleet Vehicle Routing Problem (AHVRP) with service times and routes length restrictions. The main advantage of the proposed approach is to design a simple algorithm that does not need any special fine-tuning.

The paper is organized as follows: *Section 2* describes the theoretical background and previous works. In *Section 3* we develop the details of the proposed algorithm. *Section 4* presents the data instances from the distribution company. *Section 5* shows the results of applying the proposed methodology to a real context case. To conclude, *Section 6* summarizes with some final remarks and

168

Figure 16.7: Front page of publication Cáceres-Cruz et al. (2013).

Lecture Notes in Management Science (2013) Vol. **5**: 84–92
5th International Conference on Applied Operational Research, Proceedings

ISSN 2008-0050 (Print), ISSN 1927-0097 (Online)

Applying a savings algorithm for solving a rich vehicle routing problem in a real urban context

José Cáceres-Cruz [1], Daniel Riera [1], Roman Buil [2] and Angel A. Juan [1]

[1] *IN3-Computer Science Department, Open University of Catalonia, Barcelona, Spain*
jcaceresc@uoc.edu; drierat@uoc.edu; ajuanp@uoc.edu

[2] *Department of Telecommunications and Systems Engineering,*
Universitat Autònoma de Barcelona, Bellaterra, Spain
roman.buil@uab.cat

Abstract. Nowadays urban transportation is a strategic domain for distribution companies. In academic literature, this problem is categorized as a Vehicle Routing Problem, a popular research stream that has undergone significant theoretical advances but has remained far from practice implementations. In fact, a general combinatorial routing problem has emerged as Rich Vehicle Routing Problem for considering problems inspired in real situations. Intra-urban distribution required a special combination of routing characteristics. In this study, we consider a routing problem with asymmetric cost matrix, heterogeneous fleet of vehicles, service times, limited routes length, open routes, and balanced loads in routes' restrictions. Our objective function is to reduce the total traveling time. We present an algorithm based on a randomized Clarke & Wright's Savings heuristic. We execute our algorithm with data from a company that distributes prepared food to more than 50 customers in Barcelona. The results reveal promising improvements in different scenarios.

Keywords: rich vehicle routing problem; clarke and wright; heuristics

Introduction

Vehicle routing is a complex logistics management problem and represents a key phase for the logistic optimization. We have considered a variant where several restrictions are considered at the same time. The set of defined constraints are taken

Figure 16.8: Front page of publication Cáceres-Cruz et al. (2013b).

Hybrid Approach combining Insertion Heuristic and Biased Random Sampling for the Vehicle Routing Problem with Time Windows

José Cáceres-Cruz[1], Daniel Riera[1], Angel A. Juan[1], Silvia Padrón[2]

[1] Open University of Catalonia - IN3, Barcelona, Spain.
{drierat,ajuanp,jcaceresc}@uoc.edu

[2] Universitat Autònoma de Barcelona, Bellaterra, Spain.
silvia.padron@uab.cat

Abstract. The Vehicle Routing Problem with Time Windows is a NP-hard routing problem where the demand is located in nodes and should be completely served fulfilling certain scheduling constraints for delivering. This paper presents a randomized algorithm which combines a classical heuristic with biased random sampling in order to solve this type of problem. This new algorithm is compared with one version of the Insertion heuristic of Solomon, reaching results which outperform it. As discussed in the paper, the methodology presented is flexible, can be easily parallelized and it does not require any complex fine-tuning process. Some preliminary tests show the potential of the proposed approach as well as its limitations.

Keywords. Vehicle Routing Problem with Time Windows, combinatorial optimization, randomized algorithms, metaheuristics, simulation.

1 Introduction

In last decades, optimization routing problems have been the target of many studies **(Golden et al., 2008)**. The Vehicle Routing Problem with Time Windows (**VRPTW**) is probably one of the most developed research lines inside of the classical Vehicle Routing Problem **(Potvin & Bengio, 1996**; **Potvin et al., 1996)**. On this, a set of vehicles must deliver the goods to a set of customers. Unlike the original problem, VRPTW must to respect some delivery time windows on each customer and considering arrival, waiting and service times among others. The objective of this study is to adapt one of the VRPTW heuristics proposed in **Solomon (1987)**. This heuristic is known as *Insertion* and basically consists on the iterative construction routes with the insertion of proper customers. Therefore our main idea is to apply the randomization concepts presented in **Juan et al. (IN PRESS)** in order to generate a new promising metaheuristic algorithm. A concrete example of this randomized process can be found in **Juan et al. (2010)**.

Figure 16.9: Front page of publication Cáceres-Cruz et al. (2013).

Ann Oper Res
DOI 10.1007/s10479-011-0918-z

Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands

Angel A. Juan · Javier Faulin · Josep Jorba · Jose Caceres · Joan Manuel Marquès

Abstract This paper focuses on the Vehicle Routing Problem with Stochastic Demands (VRPSD) and discusses how Parallel and Distributed Computing Systems can be employed to efficiently solve the VRPSD. Our approach deals with uncertainty in the customer demands by considering safety stocks, i.e. when designing the routes, part of the vehicle capacity is reserved to deal with potential emergency situations caused by unexpected demands. Thus, for a given VRPSD instance, our algorithm considers different levels of safety stocks. For each of these levels, a different scenario is defined. Then, the algorithm solves each scenario by integrating Monte Carlo simulation inside a heuristic-randomization process. This way, expected variable costs due to route failures can be naturally estimated even when customers' demands follow a non-normal probability distribution. Use of parallelization strategies is then considered to run multiple instances of the algorithm in a concurrent way. The resulting concurrent solutions are then compared and the one with the minimum total costs is selected. Two numerical experiments allow analyzing the algorithm's performance under different parallelization schemas.

Keywords Vehicle routing problem with stochastic demands · Parallel and distributed computing · Monte Carlo simulation · Probabilistic algorithms · Heuristics

A.A. Juan · J. Jorba · J. Caceres · J.M. Marquès
Department of Computer Science, Multimedia, and Telecommunication, Open University of Catalonia, 08018 Barcelona, Spain

A.A. Juan
e-mail: ajuanp@uoc.edu

J. Jorba
e-mail: jjorbae@uoc.edu

J. Caceres
e-mail: jcaceresc@uoc.edu

J.M. Marquès
e-mail: jmarquesp@uoc.edu

J. Faulin (✉)
Department of Statistics and OR, Public University of Navarre, 31006 Pamplona, Spain
e-mail: javier.faulin@unavarra.es

Published online: 23 June 2011

Springer

Figure 16.10: Front page of publication Juan et al. (2013a).

Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds

COMBINING MONTE CARLO SIMULATION WITH HEURISTICS FOR SOLVING THE INVENTORY ROUTING PROBLEM WITH STOCHASTIC DEMANDS

José Cáceres-Cruz
Angel A. Juan

IN3-Computer Science Dept.
Open University of Catalonia
Barcelona, 08018, Spain

Scott E. Grasman

Industrial and Systems Engineering Dept.
Rochester Institute of Technology
Rochester, NY 14623-5603, USA

Tolga Bektas

School of Management
University of Southampton
Southampton, SO17 1BJ, UK

Javier Faulin

Dept. Statistics and Operations Research
Public University of Navarre
Pamplona, 31006, Spain

ABSTRACT

In this paper, we introduce a simulation-based algorithm for solving the single-period Inventory Routing Problem (IRP) with stochastic demands. Our approach, which combines simulation with heuristics, considers different potential inventory policies for each customer, computes their associated inventory costs according to the expected demand in the period, and then estimates the marginal routing savings associated with each customer-policy entity. That way, for each customer it is possible to rank each inventory policy by estimating its total costs, i.e., both inventory and routing costs. Finally, a multi-start process is used to iteratively construct a set of promising solutions for the IRP. At each iteration of this multi-start process, a new set of policies is selected by performing an asymmetric randomization on the list of policy ranks. Some numerical experiments illustrate the potential of our approach.

1 INTRODUCTION

Today, one of the most important concepts in supply chain management is that of replacing sequential decision making with global decision making, where all parties in the supply chain determine the best policy for the entire system. Inventory and transportation systems are good examples of sequential decision making. However, driven by business practices such as vendor managed inventory (VMI), integrated inventory and transportation systems have received much recent attention (Kleywegt et al. 2004). VMI is a supply chain centralized control initiative where the supplier is authorized to manage inventories of the retailers and to make decisions such as when and how much inventory to ship to the retailer. VMI is seen as an effective means of managing inventory through the strategic use of technologies which enable the flow of information throughout the entire supply chain. Despite the potential benefits, and probably due to its complexity, only a relatively small number of articles have analytically approached the issue of integrating decisions. This issue is known in the literature as the Inventory Routing Problem or IRP (Campbell et al. 2002). Therefore, model formulations with exact or approximate solution procedures are still needed to assist with the widespread adoption of VMI and use of synchronized inventory and transportation systems.

In this paper, a hybrid approach is proposed. Our approach combines Monte Carlo simulation (MCS) with a multi-start asymmetric randomization of a classical routing heuristic. We consider a single-period

978-1-4673-4781-5/12/$31.00 ©2012 IEEE

Figure 16.11: Front page of publication Cáceres-Cruz et al. (2012b).

Proceedings of the 2013 Winter Simulation Conference
R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, eds

A SIMULATION-BASED ALGORITHM FOR THE INTEGRATED LOCATION AND ROUTING PROBLEM IN URBAN LOGISTICS

Andrés Muñoz-Villamizar
Jairo R. Montoya-Torres

Escuela de Ciencias Económicas y Administrativas
Universidad de La Sabana
Chía (Cundinamarca), COLOMBIA

Angel A. Juan
José Cáceres-Cruz

IN3 - Computer Science Department
Open University of Catalonia
Barcelona, 08018, SPAIN

ABSTRACT

In most medium and large sized cities around the world, freight transportation operations might have a noticeable impact on urban traffic mobility as well as on city commercial activities. In order to reduce both traffic congestion and pollution levels, several initiatives have been traditionally implemented. One of the most common strategies concerns the allocation of urban distribution warehouses near the city center in order to consolidate freight delivery services. This paper considers the integrated problem of locating distribution centers in urban areas and the corresponding freight distribution (vehicle routing). The combined problem is solved by using a hybrid algorithm which employs Monte Carlo simulation to induce biased randomness into several stages of the optimization procedure. The approach is then validated using real-life data and comparing our results with results from other works already available in the existing literature.

1 INTRODUCTION

The idea of implementing freight consolidation platforms within urban areas is known in the academic literature as Urban Distribution Centers (UDC) (Taniguchi et al. 1999). The general goal of this research area is to solve –or at least to reduce– traffic problems within urban areas, considering some extra variables like environmental pollution and excessive energy consumption. According to Muñuzuri et al. (2012), this is a critical issue in most large sized European cities. In effect, due to their inherited radial structure these cities tend to show a high concentration of shopping areas, restaurants, and other social attraction poles in the city center, which not only influence mobility and commercial activities but also impose a series of restrictions in flows of freight deliveries. Thus, most urban centers in these cities contain narrow streets with no parking lots or back alleys, which are not well designed to support asymmetric flows of people going to work, shop, eat, or visit tourist attractions (Ligocki and Zonn 1984). In addition, according to several authors (Topp and Pharoah 1994; Muñuzuri et al. 2005; Geroliminis and Daganzo 2006; Delaître 2008), infrastructure investments in these cities have often been implemented in order to promote environmental sustainability, such as bike lanes, underground and tram systems, more efficient bus systems and the enlargement of pedestrian areas (Daganzo 2010). Despite the clear advantages of these policies, they also led to larger and stricter restrictions regarding freight deliveries.

Among the advantages described by Taniguchi et al. (1999), creating UDCs allows the implementation of a much more efficient urban logistics system, with the same capacity of service than conventional systems but with lower environmental impact. Thus, several cities have decided to put into practice these UDCs in order to take advantage of some of the benefits they offer, including:

- The use of electric vehicles, whose limited autonomy prevents them from travelling long distances.

Figure 16.12: Front page of publication Muñoz-Villamizar et al. (2013).