

*Nafarroako  
Unibertsitate  
Publikoa*



Universidad  
Pública de  
Navarra

Department of Statistics, Computer Science and Mathematics  
Institute of Smart Cities

PhD Thesis  
**Simheuristic Algorithms for the  
Sustainable Freight Transport Problem**

Author:  
**Lorena Silvana Reyes-Rubiano**

Supervisors:  
**Dr. Javier Faulin Fajardo**  
**Dr. Angel A. Juan Perez**

Pamplona - Spain, April 2019



## Acknowledgments

I would like to thank all my family, especially my parents and my friends for their confidence.

I like to express my sincere gratitude to my supervisors, Professor Javier Faulin and Professor Angel A. Juan for their continuous encouragement, help and guidance since my arrival at the Public University of Navarre until the final stage of this dissertation. This experience gave me the opportunity to know different countries and cultures, to meet a lot of friendly people, to do new friends, and to improve my academic and research skills.

Similarly, I express my gratitude to the Ibero-American program for Science and Technology for Development (CYTED2014- 515RT0489), the Erasmus+ programm (2016-1-ES01-KA108-023465), the Public University of Navarre, DECYL Research Group, for their financial support for developing my research in the period 2016-19. Particularly, I am very grateful to the Department of Statistics, Computer Science and Mathematics and the Institute of Smart Cities for their help during these years.

I wish to thank Dr. Javier Panadero, Professor Angel A. Juan and his research team for their stimulus and hospitality during my research stay at the DPCS-ICSO group at Universitat Oberta de Catalunya, Barcelona, Spain. Likewise, I am very grateful to Professor Patrick Hirsch and his research team for their guidance, kindness and friendliness during my research stay at the Institute of Production and Logistics at University of Natural Resources and Life Sciences, Vienna, Austria. Also, I wish to thank Professor Djamila Ouelhadj and her research team for their affability during my research stay at the Mathematics Department at the University of Portsmouth, Portsmouth, United Kingdom.

Finally, my warmest thank you all the people who helped me during this *journey*.



## Nomenclature

Nomenclature	Description
AFV	Alternative fuel vehicles.
API	Air pollution index.
ALNS	Adaptive large neighborhood search metaheuristics.
BKS	Best known solution.
BR	Biased-randomization.
BRCWS	Biased-Randomized CWS.
BR-VNS	Biased-randomized variable neighborhood.
COP	Combinatorial optimization problem.
CO <sub>2</sub>	Carbon dioxide emission.
CVRP	Capacitated vehicle routing problem
CWS	Clarke and Wright's savings heuristics.
EV	Electric vehicle.
EVRPST	Electric vehicle routing problem with stochastic travel time.
Green VRP	Green vehicle routing problem.
GDP	Gross domestic product.
IT	Information technologies.
ICE	Internal combustion engine vehicles.
MCS	Monte Carlo Simulation.
MDVRP	Multi-depot vehicle routing problem.
MILP	Mixed integer linear programming.
NP-hard problem	A COP problem that cannot be solved in a polynomial time.
OP	Orienteering problem.

Nomenclature	Description
PRP	Pollution routing problem.
Rich VRP	Rich vehicle routing problem.
SAA	Sample average approximation.
SCEVRP-LC	Sustainable capacitated electric vehicle routing problem with length constraints.
SCEVRP-SDT-LC	Sustainable capacitated electric vehicle routing problem with stochastic demands, travel times and length constraints.
SIDRA	Signalized intersection design and research aid
SIM-BR-MS	Biased-randomized version of multi-start simheuristic algorithm.
SIM-BR-ILS	Biased-randomized version of iterated local search simheuristic algorithm.
SIM-BR-VNS	Biased-randomized version of variable neighborhood search simheuristic algorithm.
TOP	Team orienteering problem.
UAV	Unmanned aerial vehicle.
VNS	Variable neighborhood search.
VRP	Vehicle routing problem.

# Contents

<b>List of Algorithms</b>	<b>v</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Motivation . . . . .	3
1.2 Smart city logistics . . . . .	5
1.3 Sustainability indicators . . . . .	8
1.3.1 Economic dimension . . . . .	10
1.3.2 Social dimension . . . . .	11
1.3.3 Environmental dimension . . . . .	12
1.3.4 The trade-off costs among sustainability dimensions . . . . .	14
1.4 VRP variants for the sustainable freight transport . . . . .	15
1.5 Overview of the thesis . . . . .	17
<b>2 Methodology</b>	<b>21</b>
2.1 Constraints and attributes for COPs . . . . .	21
2.2 Solving approaches for COPs . . . . .	22
2.2.1 Exact methods . . . . .	23
2.2.2 Approximated methods: heuristics . . . . .	23
2.2.3 Metaheuristics . . . . .	24
2.3 Biased randomization of heuristics . . . . .	25

2.4	Solving approaches for stochastic COPs: optimization and simulation techniques . . . . .	29
2.5	Simheuristic algorithms . . . . .	31
2.6	Stochastic programming . . . . .	32
<b>3</b>	<b>The sustainable multi-depot vehicle routing problem</b>	<b>37</b>
3.1	A BR-VNS simheuristic algorithm for solving the sustainable multi-depot vehicle routing problem . . . . .	38
3.2	Literature review . . . . .	39
3.3	Problem description . . . . .	41
3.4	The BR-VNS solving approach . . . . .	44
3.5	Computational experiments . . . . .	46
3.5.1	Benchmark instances . . . . .	47
3.5.2	Validation . . . . .	48
3.6	Computational results . . . . .	49
3.7	Contributions . . . . .	55
<b>4</b>	<b>The stochastic electric vehicle routing problem using energy safety stocks</b>	<b>61</b>
4.1	A BR-MS simheuristic algorithm for solving the stochastic electric vehicle routing problem . . . . .	62
4.2	Literature review . . . . .	63
4.2.1	The deterministic green VRP . . . . .	64
4.2.2	The stochastic green VRP . . . . .	65
4.3	Additional details on the EVRPST . . . . .	66
4.4	The SIM-BR-MS algorithm solving approach . . . . .	69
4.5	Computational experiments . . . . .	71
4.5.1	Benchmark instances . . . . .	71
4.5.2	Parameters setting . . . . .	71
4.5.3	Computational results . . . . .	73
4.6	Analysis of results . . . . .	75
4.7	Contributions . . . . .	76
<b>5</b>	<b>The stochastic electric vehicle routing problem with sustainability indicators</b>	<b>81</b>
5.1	A BR-ILS simheuristic algorithm for solving the stochastic electric vehicle routing problem with sustainability indicators . . . . .	82
5.2	Literature review . . . . .	82
5.3	Problem description . . . . .	83
5.3.1	Deterministic version . . . . .	84
5.3.2	Stochastic version . . . . .	86



---

5.4	The SIM-BR-ILS solving approach . . . . .	88
5.4.1	General framework . . . . .	88
5.4.2	Proposed policies . . . . .	89
5.5	Computational experiments . . . . .	91
5.5.1	Benchmark instances . . . . .	91
5.5.2	Parameters setting . . . . .	91
5.5.3	Computational results . . . . .	93
5.6	Analysis of results . . . . .	96
5.7	Contributions . . . . .	101
<b>6</b>	<b>The stochastic team orienteering problem with driving ranges</b>	<b>105</b>
6.1	A BR-VNS simheuristic algorithm for solving the stochastic team orienteering problem . . . . .	106
6.2	Literature review . . . . .	107
6.3	Problem description . . . . .	108
6.4	Two approaches for solving the stochastic TOP . . . . .	110
6.4.1	The SAA solving approach . . . . .	111
6.4.2	The SIM-BR-VNS algorithm solving approach . . . . .	112
6.5	Computational experiments and results . . . . .	113
6.5.1	Analysis of the deterministic TOP . . . . .	113
6.5.2	Analysis of the stochastic TOP . . . . .	114
6.6	Contributions . . . . .	122
<b>7</b>	<b>Conclusions and further work</b>	<b>125</b>
<b>8</b>	<b>Appendix A</b>	<b>129</b>
<b>9</b>	<b>Appendix B</b>	<b>137</b>
<b>10</b>	<b>Appendix C</b>	<b>141</b>
<b>11</b>	<b>Appendix D</b>	<b>147</b>
<b>12</b>	<b>Appendix E</b>	<b>151</b>
<b>13</b>	<b>Appendix F</b>	<b>155</b>
	<b>References</b>	<b>159</b>



## List of Algorithms

1	Procedure to calculate the priority list of edges. . . . .	26
2	Procedure to the biased randomization of heuristics . . . . .	27
3	Procedure to generate a feasible solutions. . . . .	28
4	Procedure to assess the solution performance . . . . .	30
5	The BR-VNS algorithm for the sustainable MDVRP . . . . .	57
6	Procedure to calculate the priority list . . . . .	58
7	Procedure to generate a new feasible solution . . . . .	58
8	Procedure to destroy-and-rebuild a solution . . . . .	59
9	Procedure of local search . . . . .	59
10	The SIM-BR-MS algorithm for the EVRPST. . . . .	78
11	The SIM-BR-ILS algorithm for the SCEVRP-SDT-LC . . . . .	104
12	The SIM-BR-VNS algorithm for the TOP . . . . .	123



## List of Figures

1.1	Sustainability dimension and sustainable solution representation . . .	4
1.2	The background of the smart city concept . . . . .	7
1.3	Structure and contributions of this thesis . . . . .	18
2.1	Estimation of savings for sustainable problems. . . . .	25
2.2	Estimation of sustainability cost according to the route structure. . .	26
2.3	Sorted saving list and geometric distribution with parameter $\lambda = 0.3$	28
2.6	The SAA approach . . . . .	33
2.4	The biased randomization of heuristics . . . . .	34
2.5	The simheuristic algorithm . . . . .	35
3.1	Average proportion of each cost component in the sustainable solutions.	50
3.2	Gaps (%) between the sustainability solution and the solutions minimizing one measure per level of congestion. . . . .	51
3.3	Minimizing distances . . . . .	53
3.4	Minimizing time . . . . .	53
3.5	Use of radar plots for decision-making. Criteria: travel distance and time. . . . .	53
3.6	Minimizing $CO_2$ emissions . . . . .	54
3.7	Minimizing social cost . . . . .	54
3.8	Use of radar plots for decision-making. Criteria: $CO_2$ emissions and social cost. . . . .	54
4.1	A simple representation of the EVRPST with driving-range constraints.	63
4.2	Frequent attributes and constraints in the green VRP . . . . .	64
4.3	Corrective action for type I failure . . . . .	68

4.4	Preventive action for type II failure . . . . .	68
4.5	Different actions to deal with route failures while using electric vehicles . . . . .	68
4.6	Visual comparison among BDS-Det, BDS-Stoch, and BSS-Stoch. . . . .	75
4.7	Reliability values for different safety stock levels. . . . .	76
4.8	Expected travel times for different safety stock levels . . . . .	77
5.1	Representation of a solution for the SCEVRP-SDT-LC. . . . .	87
5.2	Multiple boxplots comparing solutions found with different weights. TC: total cost; Ec: economic cost; Co2: environmental cost; Sc: social cost; sol: solution. . . . .	94
5.3	Example of different solutions for the instance A-n33-k6. . . . .	96
5.4	Effect of policies on the expected total cost and the reliability. . . . .	100
5.5	Effect of the computing time (seconds) and the number of seeds on the expected total cost for the instance A-n69-k9. . . . .	100
6.1	The SAA framework. . . . .	111
6.2	A typical simheuristic framework . . . . .	112
6.3	Comparison of gaps w.r.t. the BKS of the deterministic TOP. . . . .	114
6.4	The gap comparison . . . . .	121
6.5	The computing time comparison . . . . .	121
6.6	The comparison between the SAA and the BR-VNS simheuristic. . . . .	121

## List of Tables

1	Summary of the thesis objectives addressed in each chapter . . . . .	2
1.1	Core of sustainability indicators. Based on Kiba-Janiak (2016) . . . . .	9
1.2	Externalities cost. Based on Ranaiefar and Amelia (2011) . . . . .	10
2.1	Attributes and constraints addressed in this thesis . . . . .	22
2.2	The relationship between the BKS and the best stochastic solution (BSS) . . . . .	31
3.1	Comparison of the BR-VNS algorithm for the classical MDVRP. . . . .	49
3.2	Comparison of the BR-VNS algorithm against CPLEX. . . . .	50
3.3	Gaps (%) between the sustainable solution and the solutions minimizing one measure considering a high level of congestion. . . . .	55
4.1	An illustrative set of works covering the most popular green VRP variants. . . . .	66
4.2	Characteristics of the benchmark instances. . . . .	72
4.3	Performance of best deterministic and stochastic solutions. . . . .	74
4.4	Solution performance considering different safety stock levels. . . . .	79
5.1	Parameters of the algorithm. . . . .	91
5.2	Parameters to quantify and monetize impacts. . . . .	92
5.5	Comparison between the best deterministic and the best stochastic solutions with a low level of stochasticity. . . . .	98
5.6	Comparison between the best deterministic and the best stochastic solutions with a high level of stochasticity. . . . .	99

5.3	Comparison of our approach against BKS for deterministic instances when minimizing distance. . . . .	102
5.4	Total cost and gaps (%) for solutions found with different weights. . .	103
6.1	Deterministic TOP - Results by CPLEX and our BR-VNS algorithm.	115
6.2	Stochastic TOP with low variance ( $c = 0.05$ ) - Results by SAA and SIM-BR-VNS. . . . .	117
6.3	Stochastic TOP with medium variance ( $c = 0.25$ ) - Results by SAA and SIM-BR-VNS. . . . .	118
6.4	Stochastic TOP with high variance ( $c = 0.75$ ) - Results by SAA and SIM-BR-VNS. . . . .	119



## Abstract

The sustainable freight transport entails the design of the distribution plans with the least negative impacts. On one hand, this distribution problem relies on determining the routes to visit a set of customers, which can be geographically scattered. On the other hand, the operational constraints and the attributes involved in urban transport need to be considered for designing the distribution plan. Distribution plans encompass not only the classical routing constraints but also a set of economic, social and environmental criteria implicated with transport sector. These attributes link social and industrial needs taking into account the triple bottom of objectives sustainability. Those attributes may be difficult to address because they can be progressing in different directions. This thesis contributes to integrate these challenges by means of analysis of transport problems, and structured method developments for supporting the decision making process.

To attain these challenges the following objectives have been proposed:

- Identification of attributes and constraints for problems related to freight transport in smart cities, with especial focus on environmental, economic and social impacts.
- Modeling of sustainability indicators in the vehicle routing problems with the purposes of producing greener transport in smart cities.
- Design and implementation of hybrid algorithms combining metaheuristics with simulation to provide sustainable solutions.
- Validation of the algorithms using realistic data and well-known solutions.

The first objective is to provide a characterization in problems related to freight transport, considering a special focus on sustainability dimensions. Some measures

to estimate the negative impacts caused by transport activities have been also included. In [Chapter 1](#), the classical issues related to urban transport and the sustainability dimensions are presented.

Afterwards, the [Chapter 2](#) provides a general description of solving approaches for combinatorial optimization problems considering also an overview of the most common attributes and constraints related to the current sustainability initiatives. Then, the framework of biased randomized simheuristic algorithm is described together with the most classical methods to solve rich vehicle routing problems. The proposed algorithms are well described across the chapters of this thesis. For the second objective of this dissertation, a formal description for routing problems with single depot and multi depot configuration. In [Chapter 3](#) a sustainable multi-depot problem is defined and solved by a mixed integer programming and a variable neighborhood search framework. From [Chapter 4](#) to [Chapter 6](#), vehicles routing problem with electric is described assuming a single depot and stochastic variables.

The third objective is a global one which will be addressed over the course of the whole dissertation. Easy to implement and competitive simheuristic algorithms are proposed to cope with stochastic problems. Particular attention is paid on the inclusion of sustainable criteria and consideration of current operational constraints from freight transport.

The fourth objective is to implement and test the algorithms using benchmarks for deterministic and stochastic problems. The results show the sustainability influence of the optimization criteria and the effect of stochastic data on the performance of the solution approaches and solutions quality.

Finally, this dissertation ends with some conclusions and comments on further research lines. For a brief summary, [Table 1](#) shows the objectives develop through the chapters of this dissertation.

Table 1: Summary of the thesis objectives addressed in each chapter

		Objectives			
		O1	O2	O3	O4
Freight transport and sustainability dimensions	Chapter 1	✓			
	Methodology		✓		
Applications	Chapter 3		✓	✓	✓
	Chapter 4		✓	✓	✓
	Chapter 5			✓	✓
	Chapter 6		✓	✓	✓

## 1.1 Motivation

City logistics and freight transport logistics have an important role in the economic growth of a city. Hence, transport activities involve many stakeholders and, in turn, different targets which could be contradictory with each other. This heterogeneity could disturb the decision-making process putting in evidence the problems of freight urban mobility. According to [Eurostat \(2011\)](#), transport sector employs around 10 million people and accounts for about 5% of the GDP in the EU. The efficiency of most companies heavily depends on this industry, since logistics activities account for 10-15% of the cost of a finished product for European companies. The total EU-28 road transport accounted for just over three-quarters of the total inland freight transport in 2015 based on tonne-kilometers performed ([Eurostat, 2017](#)). The road freight transport performance was around 1.8 millions of tonne-kilometers in 2016, showing an increase of about 4.2% compared with 2010. From a social perspective, it can be highlighted that slightly over 26 mil persons lost their lives in road accidents within the EU-28 in 2015. There has been a steady decrease in the number of persons killed on European roads over the last decade. Last but not least, we consider the environmental impacts. Eurostat states that transport and storage activities accounted for almost 14% of the greenhouse gas emissions and 24% of the emissions of ozone precursors considering all the economic activities in 2014.

Consequently, the sustainability concept into logistics systems deal with transport problems in smart cities to ensure efficient urban mobility, not just for people, but for goods as well. Sustainability integrates the three dimensions (economic, social and environmental) which are mutually influence each other. [Figure 1.1](#) illustrates the complexity of achieving a balance between the economic, social and environmental impacts.

From a logistic system perspective, sustainable transport should provide a suit-

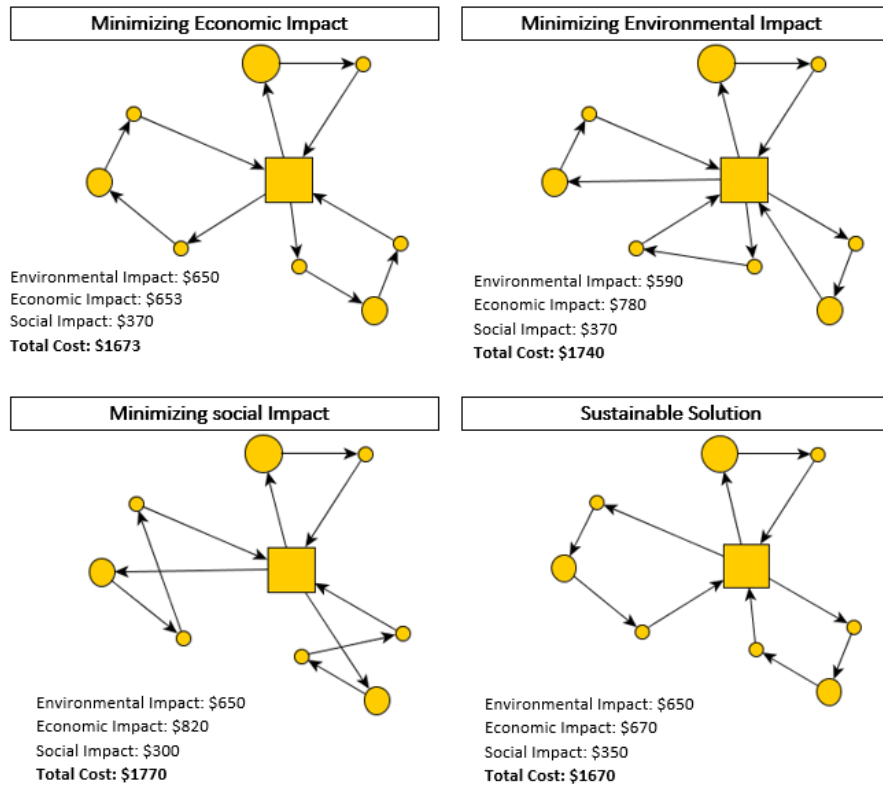


Figure 1.1: Sustainability dimension and sustainable solution representation

able balance among economic, social and environmental aspects. The sustainability into a logistic system has gained attention in transport management. This perspective allows identifying the key factors which should provide a synergy among sustainability dimensions (Ritzinger et al., 2016). Furthermore, there are other multiple external factors such as the weather, traffic accidents, traffic signs and rush hours which have a strong influence on the routes performance (Gendreau et al., 2015). These factors are closely related to random events, which raise the risk of route failures, increasing the costs and negative impacts (International Electro-Technical Commissions, 2014). Consequently, the recent sustainable strategies rely on including new technology and structured tools to manage a massive amount of random data for supporting decision making on the fly.

Generally speaking, the environmental and social concerns have been increasing along with the industrial needs. Companies are required to become more sustainable, i.e., getting a suitable balance between economic, environmental and social dimensions in their processes. Therefore, the concept of efficiency has to be expanded to include multiple criteria in the decision-making process. Green initiatives have led

to the emergence of smart cities, which combine economic growth, improvements in living standards and reduction of the negative impacts caused by commercial activities. All logistic systems in smart cities aim to implement optimization techniques, technological advances and information and communication systems to make the freight transport a flexible system adept at meeting the social and industrial needs (Smart Freight Transport Center, 2017).

Ahvenniemi et al. (2017) present a discussion about the concept of a sustainable city and a smart city. necessarily, the concept of the smart city is strongly related to technology but not on sustainability issues. In contrast, a sustainable city is a much stronger focus on the triple bottom line (economic, environmental and social dimensions) aiming at balancing the trade-off among dimensions. In this context, we address the concept of the smart city as a headway of a sustainable city. Then, a smart city is all about providing products and services in a smart way. It is also about connectivity where technology allows to share and update real-time data through an online information system. As a result, a constant flow of information, money and goods provides accurate information to make smart decisions.

Accordingly, we argue that a smart city is a sustainable city which is supported by technology Faulin et al. (2018). The main contributions of this chapter are: *i*) a comprehensive differentiation of the concept of smart city and sustainable city; *ii*) the characterization of the three sustainability lines into the freight transport. Moreover, this chapter provides an overview of the effect of sustainability on the freight transport system. Here, operations research and management science techniques play an important role to formalize, model and solve such complex problems allowing their integration with information and communication technologies. Some variants of the classical product distribution problem are discussed.

## 1.2 Smart city logistics

Taniguchi et al. (1999) introduce the concept of city logistics as “the process for totally optimizing the logistics and transport activities by private companies in urban areas while considering the traffic environment, the traffic congestion and the energy consumption within the framework of a market economy”. Then, city logistics lies in profitable logistic systems where the efforts aim at minimizing transport cost. Later, the need for efficient and environmentally acceptable urban transport system is conjoined by the idea of a green city. Thus, environmental issues are critical concerns all over the world. As a result, the logistic system is aiming at environmentally responsible and friendly operations. Due to global warming and overuse of natural resources caused by transport activities, the natural environment has become an important variable in the decision-making process (Bektaş et al., 2018). Therefore, the interest in developing green logistics from companies, government and the pub-

lic is increasing in order to change the environmental performance of suppliers and customers.

Following, the social and industrial needs are not only concerned with the economic impact of decisions, but also with the effects on society, such as the effects of pollution on the environment. Consequently, the sustainable city concept has gained importance due to the increasing pressure for the balance among economic, social and environmental dimensions in the logistic system. The sustainability initiatives are introduced as a way to conciliate the economic development and natural resource consumption in city logistics. Thus, sustainability indicators arise from the philosophy to get a development that meets the current industrial and social needs and preserve the resources for the future generations (McKinnon et al., 2015). Then, the growing public concern about living conditions and environmental preservation, especially in the context of modern cities.

On one hand, the sustainability concept in city logistics promotes to determine educational programs about sustainable actions which lead freight transport on a greener system. On the other hand, the transition from a sustainable city to smart city refers to sustainability-oriented processes and technology integration. This initiative leads to the emergence and integration of the new technology, which aims at an optimal synchronization of transport operations (Bibri and Krogstie, 2017). In this sense, an optimal synchronization links the operation research for supporting the transport decisions. Thus, the decisions making process meets the multidimensional needs and fits rapid market changes by decisions on the fly. Therefore, information, communication and technology are key factors to turn a city into a smart place. Thus, a smart city is a place well connected, sustainable and resilient against the urban dynamic (Faulin et al., 2018).

Currently, there is an extensive discussions around the smart city concept (Letaifa, 2015). The smart city definition could come up from a combination of definitions above: a smart city is a metropolitan place that is embedded with harmonic systems that provides a balance between economic development, environmental preservation and promote a high quality of life (Montoya-Torres et al., 2016). In this context, there is a consensus around the multidimensional factors or triple bottom related to sustainability dimensions (Vega-Mejía et al., 2017). Therefore, the smart cities concept arise to face the constant changes of economy, environment and society, besides environmental preservation concerns and city development.

Consequently, the sustainability concept has been gaining increasing attention. Following an extensive literature review, Vega-Mejía et al. (2017) and McKinnon et al. (2015) evidence that the sustainability concept is usually limited to the environmental dimension but it also involves economic and social issues. In this sense, there are criteria for sustainability which could against each other. As a consequence, the smart city concept appears as an engaging approach to get an agreement

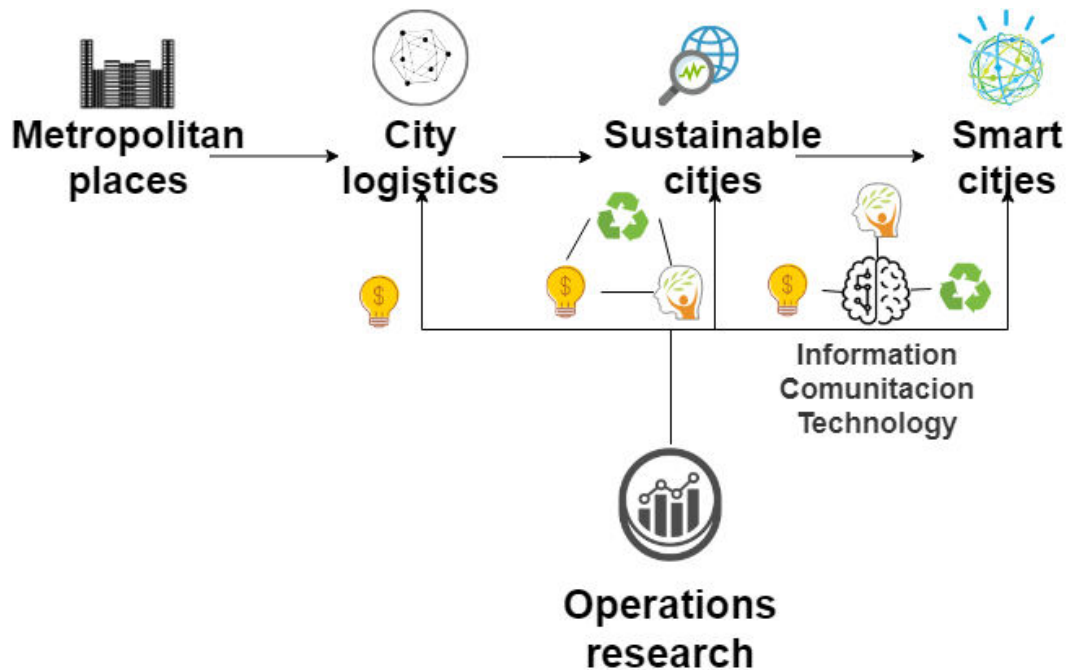


Figure 1.2: The background of the smart city concept

on how to trade-off the different sustainability dimensions. Thus, a sustainable development followed by the technology integration might turn a metropolitan place into a smart city. The Figure 1.2 summarizes the city concept transition to smart city. Then, it is necessary to integrate sustainability criteria to get ‘smart’ decisions to reach an optimal balance between economic, social and environmental benefits. Then, sustainability in the smart city concept appears as a solution to support the decision making involved in transport logistics (Faulin et al., 2018). Many actors inter-operate for the freight mobility in cities. As a result, new information and communication technology have integrated to support the city operations sustainably (Alizadeh, 2017). Similarly, new technology such as electric vehicles have integrated as innovative ways to handle freight transport aiming at the minimal negative impact on the environment, quality life and economic growth (Eurostat, 2017). In this sense, the definition of sustainable city evolves to the concept of the smart city through the integration of new technology (Silva et al., 2018).

Likewise, there are few tools to support the measurement of economic, social and environmental impacts caused by the freight transport. These impacts could be subjective because of the heterogeneity of the involved stakeholders. As a result, some objectives cannot be filled for all stakeholders, even after considering the three sustainability dimensions (Montoya-Torres et al., 2016). Thus, after the objectives and stakeholders interests are identified, the next step is building a consensus which

influences on the city performance, in sustainability terms (Kiba-Janiak, 2016).

### 1.3 Sustainability indicators

From the freight transport perspective, the main challenge of smart city are the economic growth and quality of life by implementing sustainable strategies. Thus, multidimensional indicators have been integrated as benchmarks to determine the well-performing of cities. These indicators involves the economic, social and environmental dimensions which are also nested to stakeholders objectives (Kiba-Janiak, 2016). In this section, we encompass and summarize the contributions about the sustainability dimensions addressing smart logistics systems.

In transport sector, one of the main stakeholders is the government, who plans, controls and imposes regulations for transport activities, for example the noise boundaries permitted (The European Parliament and the Council of the European Union, 2014). Furthermore, there are many initiatives which are aimed to enhance the quality life. For example, the World Health Organization program “Living well, within the limits of our planet” aimed at decrease the noise and the pollution by the city design and integration of new regulations (The European Parliament and the Council of the European Union, 2013). As a way of responding to the aforementioned initiatives, a number of relevant initiatives have been released by private and public organizations, e.g.: *i*) Lean and Green Europe ([www.lean-green.eu](http://www.lean-green.eu)); *ii*) US / Canada Smartway Transport Partnership ([www.nrcan.gc.ca](http://www.nrcan.gc.ca)); or *iii*) UNCTAD Sustainable Freight Transport and Finance ([www.unctad.org](http://www.unctad.org)). In summary, the traditional paradigm of freight distribution in urban zones is changing with the introduction of the sustainability concept and the integration of new technology.

These initiatives are linked to monetary factors, such as taxes and willingness. For example, the monetary indicators which encourage companies to adopt a sustainable behavior. Generally speaking, monetary incentives and indicators quickly yield an effect on the behaviour of any system (Schall and Mohnen, 2015). Thus, the negative impacts caused by the transport activity are refereed as negative externalities. These costs have received increased attention in the decision making (Ranaiefar and Amelia, 2011). In the literature there are different classification costs to capture the negative impacts and understand the stakeholders objectives (McKinnon et al., 2015; Ranaiefar and Amelia, 2011). From the viewpoint of sustainability dimensions and the external costs operational indicators could be developed. These indicators extent decision criteria for fulfilling the stakeholders objectives. Table 1.1 visualizes the core of indicators related to sustainability dimensions and stakeholders objectives.

The mentioned indicators summarize objective or challenges the cities to become in a smart city. These attribute cover the objectives for metropolitan places to



Table 1.1: Core of sustainability indicators. Based on Kiba-Janiak (2016)

Sustainability dimensions	Initiatives from stakeholders			
	Objectives	Government	Companies	Customers
Economic	Congestion reduction	Introduction of time windows for freight transport	Reduction in travel time	Road traffic and accessibility to locations
	Profitability objective	Introduction of subsidies for less polluting transport system Foster consolidation strategies	Shipping and delivery times Logistic cost reduction Undertaking vertical and horizontal cooperation	Service levels Lower product price Designated unloading and loading zones
Social	Health objectives	Public health costs	Workload balance regulations	Human health
	Safety objectives	Heavy vehicles within the city	Insurance companies costs	Road safety (Accident and deaths)
Environmental	Nuisance reduction	Legislation about allowed boundaries	Introduction of new technologies (vehicles with alternative fuel consumption)	Social welfare
	Fuel consumption reduction Emission reduction Infrastructure protection	Road protection	Promotes multimodal transport and energy efficiency in buildings.	Infrastructure use

become in a smart city. In fact, this indicators adhere attributes to the logistic systems. In Table 1.1 is evident that the three dimensions are interlaced each other (Silva et al., 2018). The cities development is push by the natural resource protection and management enhancing the environmental and economic indicators which have an effect over the social dimension.

Consequently, the transport problems are enriched by attributes and constraints that aim to take into account the attributes sustainability. The new approach of urban freight transport problems are supported by the literature, including a large variety of models to measure the negative impacts. This section provides a summary of the methods aimed at assessing the cities performance from a sustainability perspective. This summary leads to the identification of main factors in each dimension for designing effective decision criteria.

### 1.3.1 Economic dimension

The economic impact covers the synergies between the social and the environmental effects. In brief, it concerns the management of natural resources use, promotion of innovation, the costs because of negative externalities (e.g., public health costs), the city monetary situation and regulations effects on the transport system. As a result, governments have designed instruments and monetary indicators to estimate the cost of impacts (Santos et al., 2010). Ranaiefar and Amelia (2011) present an estimations about externalities and costs involved. Thus, an estimation of the transport cost can be made involving the externalities impact. Table 1.2 displays a summary of the externalities cost.

Table 1.2: Externalities cost. Based on Ranaiefar and Amelia (2011)

Externality	<i>USD/km</i>	<i>USD/ton-km</i>
	Nash (2003)	Delucchi and McCubbin (2010)
Congestion	[2.28-14.82]	0.3375
Accident	0.096	[0.68-1.25]
Air pollution	[2.38-19.98]	[0.0625-11.6875]
Climate change	[2.60-3.74]	[0.0125-3.68]
Noise pollution	[0.0-89.21]	[0-3.31]
Water pollution	-	[0.0019 - 0.031]
Energy security	-	[0.14-0.52]
Infrastructure	[4.13-5.90]	-

Furthermore, the new technology integration demands large investment on infrastructure. For instance, the integration of electric vehicles implies infrastructure

changes such as charging stations and places to batteries storage. The use of electric vehicles in transport activities is related to several urban changes in terms of infrastructure and distribution strategies. On one hand, some of these challenges relate to infrastructure and fleet configurations (Juan et al., 2014c; Shao et al., 2017). On the other hand, electric vehicles have started to replace conventional vehicles in city logistics, redefining transport operations (Hof et al., 2017).

### 1.3.2 Social dimension

The performance in social terms is a combination of the environmental and the economic impacts. Consequently, the social impacts are subjective and a no clear distinction can be often made between economic, social and environmental impacts. On one hand, the environmental impacts are focused on the resources management and receptors such as the nature. On the other hand, economic impact refers to capital issues such as the job creation, business activity and earnings; whereas, social impacts concentrate on the human beings. Thus, the social impact is an effect of the economic and environmental one. For instance, health condition, safety condition and city livability are attributes from the air pollution, noise and climate change among other factors. These externalities involve the social and environmental issues (Silva et al., 2018). Meanwhile the economic and environmental are quantitative measures, the social dimension involves intangible factors (Navarro et al., 2016). As a result, social impact is a measure hard to estimate because of the stakeholders perceptions (McKinnon et al., 2015). Furthermore, this indicator may be measured from a customer or employee viewpoint (Delucchi and McCubbin, 2010).

Likewise, social impacts of transport are caused by a multiplicity of factors, which might also reinforce each other. In order to mitigate this snowball effect, in this section, we conceive road safety as a social indicator. Road safety constitutes one of the most critical indicators and is related to infrastructure, driver fatigue (workloads) distractions and high speed. According to Wang et al. (2016), speed variations are directly related to the accident risk of both pedestrians and vehicles.

In addition, having multiple traffic signs may encourage drivers to carry out dangerous maneuvers which affect the road safety (Xie et al., 2013). Recently, a set of social rules have been established through regulations in Europe concerning driving and working hours and rest times in order to tackle the driver fatigue and improve the working conditions (European Transport Safety Council, 2011, 2017).

Workload and accident road index might be mutually dependent indicators. Thus, governments have imposed regulations concerning the introduction of time windows for freight transport, driving and working hours of drivers. These rules concern stricter limits on the number of driving hours, working hours and breaks that must take a driver. Therefore, these regulations and the service times have to be taken into account while designing schedule and distribution routes. Matl et al.

(2017) review equity functions (mainly referring to allocating workloads and balancing the utilization of resources) for bi-objective vehicle routing problem models. Bashiri et al. (2016) present two mixed integer programming models to tackle the economic and social aspects related to workload balance and its influence on the accident risk.

### 1.3.3 Environmental dimension

In the environmental dimension, the travel time, travel distance and vehicle weight play a crucial role in the fuel/energy consumption and carbon emissions. Thereby, Ubeda et al. (2011) aimed at reducing transport costs and emissions by considering the distance and some variations in the vehicle maximum capacity. It is concluded that enhancing load factors (which may be achieved by using heterogeneous fleets) is an efficient way to get significant savings and the environmental benefit.

Demir et al. (2011) present a comparison among four methods that evaluate the fuel consumption and emissions level which are based on the model of Arvis et al. (2018); the model proposed by Arvis et al. (2018) measures the fuel consumption per second (mL/S). Additionally, the model measures in adding the energy spent when the vehicle is moving, accelerating and slowing down, aerodynamic drag, rolling resistance, weight, velocity and road gradient are taken into account in required energy. This model has an error of 5% considering few variations in road gradient. Despite such a small gap, Arvis et al. (2018) affirm that the model is not able of measuring the some stops.

The model proposed by Demir et al. (2011) conserves the same parameters but split the function in four parts. The first and the second part define the fuel consumption according to acceleration and the speed from initial until the end point of each route. Also, the models consider the changes of kinetic energy per traveled kilometer in the acceleration process. The third part defines the fuel consumption when the vehicle is moving, it is stopping (idle time); also the model considers an average travel speed, average travel distance and the changes kinetic energy. The fourth part integrated the consumed fuel in acceleration and deceleration. According to Demir et al. (2011), the model presents an error of 1% although a method with four equations is complex to implement.

The third model is another variant of the onw proposed by Arvis et al. (2018) which considers two status in an equation: when the vehicles are traveling and idling. This model takes into account the average speed, average traveled distance, kinetic energy or are idle, in order to reflect different situations of traffic. The last model is composed of three parts. The first part relates to the engine force, which represents the power of traction. In this part, the model takes into account the requirements of the motor such as weight, air density, rolling resistance, aerodynamic drag and then calculates the demanded energy by the engine. The second part of the equation

considers the vehicle speed ration in top gear. Finally, in the last part, the model calculates the fuel consumption considering the demanded energy by the engine force, engine speed and other parameters related to efficiency and engine friction.

Moreover the fuel consumption from an economic point of view, Kuo (2010) considered the fuel consumption for a Vehicle Routing Problem with time dependent in order to take into account the travel speed and the travel time. For calculating the fuel consumption, the route is split per sections. Equation 1.1 allows calculating the fuel consumption from  $i$  to  $j$  ( $F_{ij}$ ), in where  $d_{ij}$  is the distance from  $i$  to  $j$ .  $GPH_{i,j}^k$  represents the gallons per hour by the vehicle  $k$  and  $v_{ij}^k$  represents the travel speed.

$$F_{ij} := GPH_{i,j}^k \cdot \frac{d_{ij}}{v_{ij}^k} \quad (1.1)$$

Zhang et al. (2015) based on Kuo (2010) solved a classical model for the vehicle routing problem taking into account carbon emissions and fuel consumption cost which are defined by the load capacity. Usually, fuel consumption cost is estimated from oil cost and the released emissions are measured from a pollution benchmark which allows calculating a marginal cost. Equation 1.2 shows the calculate considered by Zhang et al. (2015),  $F_{ij}$  represents the fuel consumption from  $i$  to  $j$ , in where  $d_{ij}$  is the distance from  $i$  to  $j$ .  $LPH_{ij}^k$  represents the fuel consumption per unit time,  $p$  is the penalization factor of the additional load ( $M$ ) and  $L_{ij}$  is the weight of the transfers goods.

$$FC_{ij} := LPH_{ij}^k \cdot \frac{d_{ij}}{v_{ij}^k} \cdot \left\{ 1 + p \cdot \frac{L_{ij}}{M} \right\} \quad (1.2)$$

Currently, some tools have been developed to measure the pollution level in urban zones. For example, *MEET* is a model that assesses the released emissions by the heavy trucks, which takes into account the speed, weight, gradient road and distance (European Commission, 1999). Another model is *COPEPT* which considers almost same parameters than the *MEET* model except the gradient road (Demir et al., 2015). The *COPEPT* focuses on lineal regression that it considers the type of vehicle and weight. Signalized Intersection Design and Research Aid (*SIDRA*) system is similar to the *COPEPT* model, but *SIDRA* considers some constraint related to driver Demir et al. (2011); Ntziachristos et al. (2011).

Dhingra et al. (2003) present an analysis of the environmental impact in the city of Mumbai, India. Then, Equation 1.3 involves the economic, social and ecological factors to assess the air pollution index (*API*).

$$API := \frac{\sum_{(i)} Pd_i \cdot L_i \cdot wAPI_i}{\sum_{(i)} Pd_i \cdot L_i \cdot w} \quad (1.3)$$

Dhingra et al. (2003) split the city in grids, where  $i$  represents each grid of the zone;  $Pd_i$  is the population density in the grid  $i$ ;  $L_i$  is the length of each grid  $j$  and  $w$  is estimated value of concentration per grid. The value  $API_i$  describes the impact relevance according to population density of zone  $i$ .

Moreover, the sustainability concept promotes the use of vehicles running on alternative fuel technologies. In particular, electric vehicles (EVs) represent a promising option to mitigate the negative impacts caused by transport activities in city logistics. The specific benefits depend on the sources employed to generate energy (Muñoz-Villamizar et al., 2017). The main technical disadvantages of electric vehicles are: short driving range, reduced payload and long time for charging. Holland et al. (2015) demonstrate that the energy production usually has a lower the environmental impact than the gasoline production. Electric vehicles demand high levels of energy production and, for that reason, recent studies integrate this estimation to assess the performance of using electric vehicles instead of traditional ones (Lee et al., 2013). A survey on the use of electric vehicles in logistics and transport, discussing opportunities and challenges, is proposed by Juan et al. (2016). Wang et al. (2011) study the influence of the environmental criteria on the total cost and demonstrate that additional criteria imply additional costs and require an accurate operation synchronization in the supply chain. Sawik et al. (2017a) deal with a Green Vehicle Routing Problem (Green-VRP) with distance and capacity restrictions. There, the main objective is to minimize the effect of carbon emissions and noise in urban zones while consider the driving distance, driving altitude and driving times.

In conclusion, a balance among sustainability dimensions tends to be hard because of the interrelationships between socioeconomic actions and the environmental ones. In contrast, the sustainability perspective is a way to get a synergy between the stakeholders interests aimed at sustainable development. The three dimensions are interdependent and mutually influence each other. Therefore, the challenge relies on minimizing the trade-offs between the dimensions.

### 1.3.4 The trade-off costs among sustainability dimensions

The sustainability objectives are a triple bottom line of strategies for balancing sustainability dimensions. As mentioned before, the integration of the three dimensions could be the basis to get synergy between the involved stakeholders in freight transport. Therefore, the decision making process should conduct a thorough analysis of the logistic system to identify the key factors allowing corresponding synergy. The synergy concern on reaching an optimal or suitable balance between interest and perspective of stakeholders. According to the literature, the government, consumers, shopkeepers and transport companies are the main stakeholders for freight transport. In fact, the interests synergy could lead transport systems on a balance

sustainability-oriented. Likewise, building a interests consensus from stakeholders is a relevant part of integrating sustainability dimensions into the decision criteria (Kiba-Janiak, 2016).

Consequently, sustainability pillars are starting to be considered as decision criteria in distribution processes. While economic impacts can be measured through increases in operational costs, both social and environmental assessments tend to be subjective. As before mentioned, the externalities are perceived as economic indicators in order to measure the performance and impacts of transport activities. Besides, the economic perspective allows to take the negative impacts into account in the decision-making (Ranaiefar and Amelia, 2011). Economic, environmental, and social impacts are strongly interrelated (McKinnon et al., 2015). Prevention and mitigation costs for negative impacts need to be considered in financial reports. Prevention costs are due to the economic regulations associated with natural resources consumption or pollutant emissions. These costs are typically imposed by governments to avoid, or minimize, social and environmental consequences of transport operations. Regarding mitigation costs, they are related to penalties associated with the generation of more emissions than allowed (Santos et al., 2010). In addition, companies design the preventive and mitigating actions implementing focus on sustainable strategies; for instance, the use of alternative fuel vehicles (Scheuer, 2005).

Given the importance of these facts, the European Union sustainable development strategy defines sustainable transport as one of its seven key challenges. In this context, the increasing social concern is compelling companies to change purely commercial objectives in order to consider sustainability. This new vision seeks to compensate the negative impacts of transport activities without neglecting economic profits. Despite the fact that the literature on transport is extensive, there is a lack of works on urban transport taking into account social and environmental issues simultaneously (Geurs et al., 2009).

## 1.4 VRP variants for the sustainable freight transport

This section provides a summary of sustainability problems encountered in urban freight transport. The increasing social concern for the environment and sustainable growth, in general, requires the transformation of cities. In this context, urban freight transport problems have been analyzed cross-referencing impacts on sustainability. As mentioned previously, operations research methods contribute to sustainable management and address usually the logistic issues from a city. In this context, designing a sustainable distribution system by means of vehicle routing

models is the most relevant task for this thesis (Lin et al., 2014).

Another important aspect is the use of electric vehicles which are integrated as an alternative to reduce the environmental impacts generated by freight transport. The inclusion of this transport means dramatically change the urban freight distribution because of the poor infrastructures for alternative fuel vehicles. Accordingly, the main challenge of electric vehicles routing is related to the battery life. In this context, this thesis focuses on solving the existing transport problems by the inclusion of the sustainability indicators and electric vehicles.

In practice, the transport field has faced the VRP ever since the vehicles were introduced to meet the needs mobility-related. Consequently, the classical VRP focuses on specific constraints imposed by the involved resources in the distribution process. Nowadays, VRPs not only include the classical constraints but new specifications given by the stakeholders or by the new technology advances. Thus, new constraints and attributes set the rich VRPs aimed at sustainable city logistics.

The Multi-depot VRP is an important variant of VRP that represents a distribution network with several depots from which it can serve its customers. From the tactical and strategic viewpoint, a variant of the MDVRP is the first problem to be solved. Since the classical objective of the problem is to minimize the number of vehicles and travel distance, here sustainable MDVRP is solved. In order to address and solve the rich VRPs in a practical way, sustainable MDVRP is fragmented into various CVRPs. As a result, the following sustainable VRPs problems have been studied in this thesis:

- **The sustainable multi-depot vehicle routing problem** is about how the companies decide to make a sustainable freight distribution in urban or rural zones. Usually, a transport company has multiple depots from which their vehicles depart and arrive, and has multiple customers being served from the different depots. Generally speaking, urban freight transport is becoming increasingly complex due to an increase in the number of journeys, and the associated volume and frequency. Transport activities have a significant negative impact on the environment and population welfare, which motivates decision-makers to study the transport efficiency from a sustainability perspective. Consequently, the challenge is to make a route for each vehicle individually so that the vehicles drive in a sustainable way.
- **The electric vehicle routing problem with stochastic travel time** is focused on the integration of automated vehicles for the freight distribution. The automated driving system on the vehicle performs itself all driving task and monitor the driver conditions. According to The National Highway Traffic Safety Administration the automated vehicles could reduce the frequency of crashes by eliminating some human error on the roads, improving the safety on



the road (US. Department of transport, 2016). This type of vehicles represents a potential benefit beyond safety, including the environmental benefits and increased mobility for those otherwise unable to drive.

- **The sustainable electric vehicles routing problem with stochastic demand and travel time**, this problem relates to the freight transport system using the green technologies, such as the plug-in hybrid EVs. In particular the EVs in freight transport raise some additional challenges from the strategic, tactical, and operational perspectives. For instance, this distribution system requires the recharge stations for electric-based vehicles, meaning that strategic decisions need to be made about the changing of the cities infrastructure. Similarly, the limited driving-range capabilities of EVs, which are restricted by the amount of electricity stored in their batteries, impose non-trivial additional constraints when designing efficient distribution routes.
- **The team orienteering problem** this problem is inspired on logistic systems supported by a fleet of unmanned aerial vehicles (UAVs, or drones). Logistic activities using UAVs enables customers located further from the depot to receive the deliveries. Each UAV may be launched from a vehicle to deliver a single customer package, and then return to the vehicle to be loaded for the next delivery. The scope of delivery UAVs could also be beyond packages delivery in the last mile of client logistics. UAV could be used for monitoring in warehousing providing real-time time information by scanning inventory. In addition, UAVs offer an alternative way of gathering data (e.g., by taking pictures) and delivering products. On the one hand, in congested urban areas UAVs might represent a faster way of performing some operations than employing road vehicles.

## 1.5 Overview of the thesis

This thesis extends the literature on development of solving methods for the VRP, particularly rich VRPs. This last problem, rich VRPs are routing problems which integrate extra attributes and constraints that make more complex the classical VRPs and aim for real-life applications. Under this consideration, the [Figure 1.3](#) summarizes the problems studied through this dissertation. The purpose of this thesis is to contribute to sustainable VRPs by proposing structured algorithms and by studying new rich VRPs sustainability-oriented. Consequently, this thesis includes the development of a metaheuristics and simheuristic algorithms targeting to the objective of the tackled problem. In consequence, each chapter is self-contained in terms of the notation and literature review.

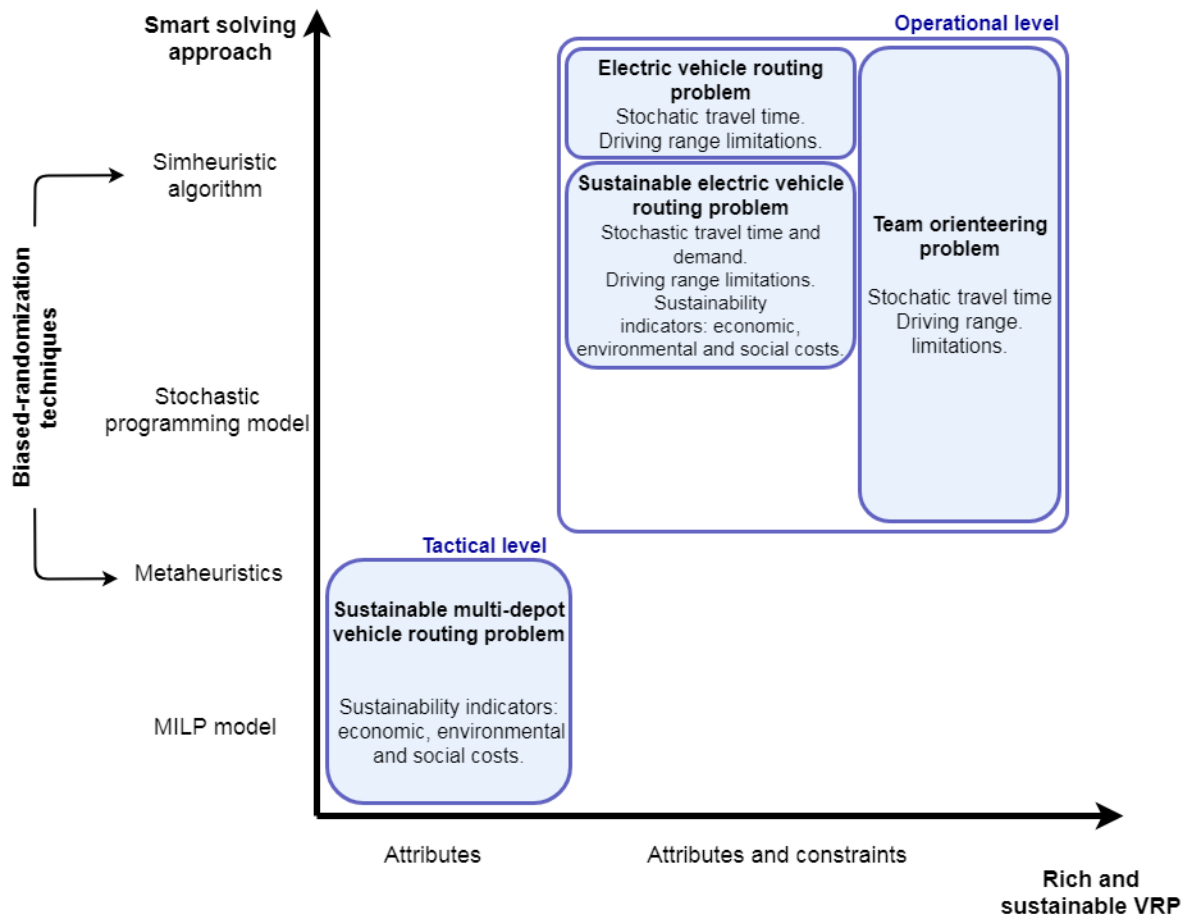


Figure 1.3: Structure and contributions of this thesis

Chapter 2 describes the theoretical background of solution methods in combinatorial optimization problems. The solutions approaches are aiming at adjusting the problem and algorithms for constraints and attributes related to sustainable freight transport. This thesis is devoted to solving approaches for deterministic and stochastic routing problems. The details of implemented metaheuristics are presented through the chapters included in this dissertation.

Chapter 3 studies an enriched multi-depot vehicle routing problem in which economic, environmental, and social dimensions are considered. Sustainability dimensions are integrated as operational indicators. These indicators are: travel time and distance (economic dimension), carbon emissions (environmental dimension), and risk of accidents (social dimension). In this chapter, we focused on the suitable integration of the three dimensions to facilitate sustainable distribution routes for urban zones. Note that these indicators may be in conflict. For instance, minimizing

travel time or distance may not lead to the same solution in a scenario with congestion. Thus, this chapter presents the trade-off among sustainability dimensions. The solving approach relies on a variable neighborhood search framework extended by biased-randomization strategies to better guide the solution-construction process. In addition, the mathematical formulation of the problem is presented and solved, which enables a direct comparison between exact methods and the proposed heuristic-based approach.

In order to tackle a real problem related with EVs, [Chapter 4](#) introduces a capacitated routing problem with stochastic travel time. From sustainability perspective, EVs represent a promising option to mitigate the negative impacts caused by transport activities in city logistics. Most of studies in the literature assume that the EVs can be fully charged by planned detours to charging stations. However, the inefficient operations, poor infrastructures, or lack of sustainable policies are barriers to integrate the EVs. In order to design reliable routing plans, a simheuristic algorithm is proposed to manage the operational risk of run out of energy. The objective is to evaluate the impact of a preventive police on the quality and feasibility of solutions for the electric VRP with stochastic travel time (EVRPST).

Similarly, the [Chapter 5](#) tackles the stochastic routing problem using EV and considering the working schedule and a maximum driving range. Hence, length constraints are introduced to ensure an appropriate balance in routes duration, which in practical terms concerns of workload balance. There, a preventive and corrective are implementing to mitigate the risk of routes failure. Regarding sustainability dimensions, an analysis of the trade-off each other and solution performance for stochastic conditions is presented.

Finally, [Chapter 6](#) introduces a the team orienteering problem as a routing problem in which the goal is to determine the set and sequence of nodes to visit. This routing problem aims at maximizing a profit from the customer visit, considering a maximum travel time. This problem can be easily linked with the use of UAVs. These type of vehicles offer an alternative way of gathering data and delivering products. In this chapter a simheuristic algorithm is proposed and compared against a stochastic programming model in order to evaluate the contribution of the proposed solving approach. This protocol integrates simulation inside the heuristic framework. A series of computational experiments contribute to illustrate the potential benefits of a simheuristic biased-randomized algorithm. Finally, the last chapter presents a general conclusion and some future research directions.

The contribution of this thesis can be summarized in solving realistic freight transport problems in urban areas in a smart city considering sustainability dimensions in a deterministic and stochastic environment. Accordingly, this thesis proposes and develops some simheuristic algorithms involving biased-randomization techniques for solving rich VRPs.



*This chapter presents an overview about solving approaches aimed at optimizing an objective function defined for a deterministic and stochastic combinatorial optimization problem (COP). Firstly, the main classification of solving methods is presented. Secondly, the integration of the biased randomization strategies into a metaheuristic algorithm is described. Finally, the simheuristic approach which combines the simulation with biased-randomization techniques into a heuristic algorithm is characterized. This is a transversal content which is addressed throughout the whole dissertation.*

## 2.1 Constraints and attributes for COPs

Vidal et al. (2013) present a literature review of heuristics and meta-heuristics for solving rich VRPs. For these problems, they present a classification of main attributes and outstanding the performance of the solving approaches. Based on the attribute classification presented in Vidal et al. (2013), the attributes can be classified by resources, routes structure, and evaluation criterion. The attributes resources-related involve the vehicle type used for performing the route. While the routes structure refers to any constraint that affect the route design, for example the driving ranges, the working hours, load capacity, depot capacity, information nature among other resource limitations. Finally, the evaluation criterion relies on the main factors that allow to determine the solution quality and efficiency of the solving approach for setting the sustainable VRP. Table 2.1 summarizes the attributes and constraints addressed in this thesis. This rich variants encompass the solving approach to routing problems sustainability-related. Moreover, following a brief description about the main solving approaches are depicted.

Table 2.1: Attributes and constraints addressed in this thesis

		Rich VRP variants sustainability-oriented.	
	Classification	Attribute	Constraint
Resources	EV	x	
	UAV	x	
	Multi-depot	x	
Route Structure	Driving ranges		x
	Working hours		x
	Balance workload		x
	Recovery strategies	x	
	Stochastic information	x	
	Evaluation criterion	Co2 emissions	x
Fuel consumption cost		x	
Accident cost		x	
Travel distance cost		x	
Maximizing profit		x	
	Driver cost	x	

## 2.2 Solving approaches for COPs

It is well-known that the VRP is a *NP*-hard problem (Garrido and Riff, 2010). Since the sustainable VRP is a variant of the classical VRP which complements the rich VRPs for conducting at real-life applications, it is also an NP-hard problem. From our knowledge, there is no polynomial time bounded exact algorithm for solving the VRPs for large instances. This makes solving the vehicle routing problem is computationally difficult. The additional constraints and attributes make the exact methods and other classical methods are hard to implement. As a result, the development of new solution methods is increasingly challenged by the growing variety of attributes and constraints. In this chapter the most appropriate solving approaches are presented. The evaluation of solution methods is done according to the objective of the problem. Thus each chapter of this thesis evaluates the solving approach by

considering the computational time and solution quality.

### 2.2.1 Exact methods

A large number of exact algorithms have been proposed for solving the VRP. These are based on integer linear programming, Stochastic programming, Branch-and-Bound and Branch-and-Cut (Toth and Vigo, 2014).. This last one solving approach is widely used for solving deterministic VRP however instances with 50 customers could not be solved to optimality (Lysgaard et al., 2004). Consequently, the previous solving approaches have been extended with relaxation techniques. For instance, the Branch-Cut-and-Price which refer to the combination of Branch-and-Cut with column generation. This hybrid method is able to solve VRP considering up to 199 nodes in a large computational time (Pecin et al., 2017). For solving the MDVRP, Contardo and Martinelli (2014) propose an exact algorithm based on set partitioning which solve instances with maximum 151 nodes.

### 2.2.2 Approximated methods: heuristics

In the literature, there is a vast number of approximated methods for solving COPs. These solving approaches relies on heuristic algorithms which work in a greedy and iterative way. Particularly, for solving VRP The most often used constructive heuristics is the Clarke and Wright's savings heuristics (CWS) (Clarke and Wright, 1964). This method starts from an initial solution  $s_0$  in which each customer is served by a different route, the heuristics looks for merging two routes extremities  $i$  and  $j$ , maximizing the cost saved  $s_i$ , where  $s_i = c_{i0} + c_{0j} - c_{ij}$ , under the condition that the merged route is feasible. As a result, a list of edges by their savings can be constructed, it is called saving list. Then, the first edge in the list represents the most promising merge for the corresponding routes. This algorithm is popular due to its conceptual simplicity and its fairly good results. Other class of constructive methods are the sequential and parallel insertion methods.

Another heuristic algorithm is the sweep method which explores the nodes set circularly, in increasing polar angle around the depot (Gillett and Miller, 1974). Each node is successively inserted in this order at the end of the current route. If this insertion is infeasible because of the route constraints, then a new route is initiated. Similarly, there are heuristics which relies on sequential procedures. For example, The *route-first cluster-second* and *cluster-first route-second* (Toth and Vigo, 2014). On one hand, the first approach constructs a solution with just one route which visits all nodes. Then, this initial solution is cut into several routes from the depot. Thus, the problem is decomposed by smaller VRPs which can be solved exactly as the shortest path problem. As well, the *cluster-first-route-second* approach, builds node clusters and then optimizes the order of visits for all clusters.

The creation of the clusters is performed by assignment rules based on priorities (Fisher and Jaikumar, 1981). At the end of this construction phase, the solutions need to be improved.

### 2.2.3 Metaheuristics

The term *metaheuristic* relies on a broad class of heuristic methods that enhance the search once the first encountered local optimum (Glover, 1986). The metaheuristics could be neighborhood-centered methods or hybrids methods. The first one relies on an iterative searching into the neighborhoods of an initial solution. For instance the population-based metaheuristics which consider a set of solutions by generating new ones obtained after the combinations of existing ones. The hybrids methods that combine procedures of different metaheuristics. This method relies on intensive iterative explorations (Lourenço et al., 2010). Metaheuristics methods involve intensification and diversification procedures. In intensification, promising regions are explored more thoroughly in the hope of finding better solutions. In diversification, unexplored 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 reduce number of regions (Caceres-Cruz et al., 2015).

Considering the attributes and the winning methods reported by Vidal et al. (2013) for solving rich VPRs, this thesis extends a variable neighborhood search (VNS), multi-start (MS) and iterated local search (ILS) framework. These methods are neighborhood-and population-based search. In the literature, they tend to be compared to determine the best type of metaheuristics.

The MS framework relies on (i) generation of solution; and (ii) local search. Each iteration produces a solution, usually a local optimum, and the best one is returned. The constructive heuristics is a biased randomized BR-CWS approach which later is explained. On the other hand, the VNS framework concerns on a successive exploration of multiple neighborhoods. It allows to find a local minimum by intensifying the search, and to escape from the associated valley by diversifying Hansen et al. (2010). In essence, it relies on three facts: (i) a local minimum with respect to one neighborhood structure is not necessarily so for another; (ii) a global minimum is a local minimum with respect to all possible neighborhood structures; and (iii) for many problems, local minimum with respect to one or several neighborhoods are relatively close to each other. Finally the ILS framework is a flexible metaheuristics that facilitate its quick implementation in practical applications (Lourenço et al., 2010). First, an initial solution is generated and then it is perturbed. Afterwards, a local search is applied to the initial solution. An important advantage of this metaheuristics is its modular structure.



## 2.3 Biased randomization of heuristics

Many uniformly-randomized algorithms rely on selecting randomly a promising element. The main goal of such approaches is to determine a priority list which includes and ranks all the candidate elements by its *goodness* (Grasas et al., 2017). In this thesis, a saving is estimated which represents the improvement on the objective function because of selecting an edge to be inserted in the solution (see the Algorithm 1). For the classical BRCWS, the saving is a measure distance-based. Here is proposed the inclusion of the sustainability dimensions for the savings estimation, assuming a total cost of sustainability for each edge ( $tc_{ij}$ ). The Algorithm 1 describes the inclusion of the sustainability dimensions for the savings estimation. Instead of being a symmetric list of savings, i.e., the savings for edge  $(h, i)$  is equal to the one for the edge  $(i, h)$ , the sustainable problems are tackled from an asymmetric savings list including all costs. As a result the total cost (Figure 2.1).

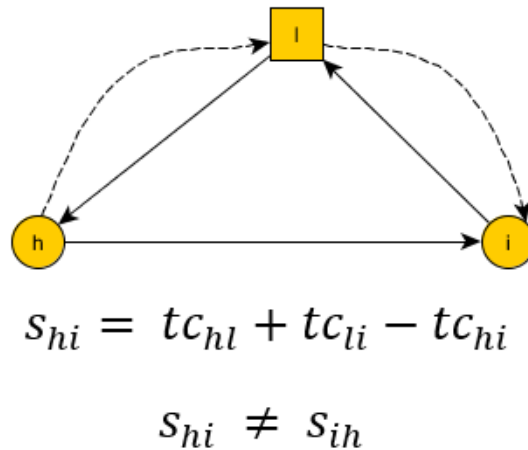


Figure 2.1: Estimation of savings for sustainable problems.

The total cost of sustainability as a distribution cost depends on multiple sustainability indicators. These indicators are: traveling time and distance (economic dimension), carbon emissions (environmental dimension), and risk of accidents (social dimension). Note that these indicators may be in conflict. For instance, minimizing traveling time or distance may not lead to the same solution in a scenario with congestion. As a result, here the savings list catch the sustainability dimensions in a measure for each edge. Figure 2.2 illustrates and quantifies their effect on the total cost of a given route with different nodes sequence. Accordingly, high-quality solutions visit first the customers with higher demands, thus minimizing the amount of freight transported over a long stretch of roads.

---

**Algorithm 1** Procedure to calculate the priority list of edges.

---

```

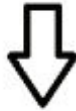
1: procedure CALCPRIORITYLIST(inputs, depot)
2:   priorityList  $\leftarrow$  empty
3:   for each (customer in inputs) do
4:     Savings(customer)  $\leftarrow$  OptCost(customer, altDepot(customer)) -
     OptCost(customer, depot)            $\triangleright$  attributes inclusion
5:     priorityList  $\leftarrow$  add(customer, priorityList)
6:   end for
7:   priorityList  $\leftarrow$  sort(priorityList)
8:   return priorityList
9: end procedure

```

---

Route	1	2	3	4	5	6	7	8	9	10	11	1
Demand		70	71	56	14	50	79	57	50	22	14	
Distance		1.2	0.5	0.7	0.9	1.3	0.8	0.9	0.3	0.7	0.6	
Social Cost		555.5	198.2	236.0	263.1	359.0	166.5	124.4	25.8	24.1	7.8	

**Total Cost**    **\$1,961**



Route	1	11	10	9	8	7	6	5	4	3	2	1
Demand		14	22	50	57	79	50	14	56	71	70	
Distance		0.6	0.7	0.3	0.9	0.8	1.3	0.9	0.7	0.5	1.2	
Social Cost		270.5	314.2	134.1	345.4	255.0	344.5	194.1	135.9	67.7	80.5	

**Total Cost**    **\$2,142**

Figure 2.2: Estimation of sustainability cost according to the route structure.

For the previous selected metaheuristics, the solution-construction process relies on the progressive insertion of edges which set the customers visited and customers sequence. Thus, the biased randomization refers to the introduction of randomization in the solution-construction phase. Under this condition, a saving is computed for each edge. Since the savings list, a priority list of edges is defined for guiding the optimization process towards good solutions. Notice, the selected metaheuristics are iterative algorithms which at each iteration, the next edge to insert is chosen from the priority list which has been previously sorted according to the savings.

In this sense, biased means that the procedure is guided by the selection prob-

---

**Algorithm 2** Procedure to the biased randomization of heuristics
 

---

```

1: procedure SELECTEDGE(seed,  $\alpha$ ,  $\beta$ , List)
2:    $\rho \leftarrow$  getRandomValue( $\alpha$ ,  $\beta$ ) ▷ uniform distribution
3:   index  $\leftarrow$  pgeometricDistribution( $\rho$ )
4:    $\xi \leftarrow$  getEdge(index, List)
5:   return  $\xi$ 
6: end procedure

```

---

ability of each edge. [Juan et al. \(2010\)](#) improve the classical CWS by a biased-randomization in the merging process of routes. This improvement relies on introduces non-uniform randomness in the selection of edges to be considered from the savings list. Since the classical version of CWS considers iteratively each edge in the list, the biased randomized CWS (BRCWS) algorithm sets a selection probability. In particular, the selection probability for each edge depends on its saving. As show in [Figure 2.3](#) the higher the edge is in the list, the higher selection probability. Then, the edge in the position of the random number is selected and deleted from the savings list, and the merging of the corresponding routes is performed. Afterwards, a new random number is generated and the same steps are applied until the savings list is empty. A probability distribution is required to follow the logic behind the CWS heuristics. The assignation of probabilities is done by a geometric distribution which uses a single parameter that needs to be tuning. However, this assignment can be done by using any other theoretical probability distributions ([Juan et al., 2011b](#)).

Hence, once the savings list is computed, a dummy solution is structure by connecting each node with the depot node. As a result, each node has an associated route. Then, the savings list allows leading the BRCWS on a good solution by the merging of routes (see the [Algorithm 3](#)). Thus, the BRCWS generates iteratively a different solution because of the biased randomization. By modifying the greedy behavior of the CWS heuristic, the BRCWS algorithm returns the best solution found.

From the initial solution a base solution is set to start the iterative process. This loop includes the following steps: *i*) Diversification process, *ii*) intensification process and *iii*) an acceptance criterion. Diversification process relies on partial-or-total destruction of the current solution, then a new solution is created. Partial destruction consists of moving a portion of nodes from one route to other ones. The total destruction is a greedy solution search. This behaviour is evident in the multi-start algorithm where a new solution is created by a BRCWS algorithm.

For diversification process, a local search operator is applied for improving the current solution by exchanging nodes inter each route. The last step is an acceptance criteria which is used to avoid entrapment at local optimum. The Flowchart [2.4](#)

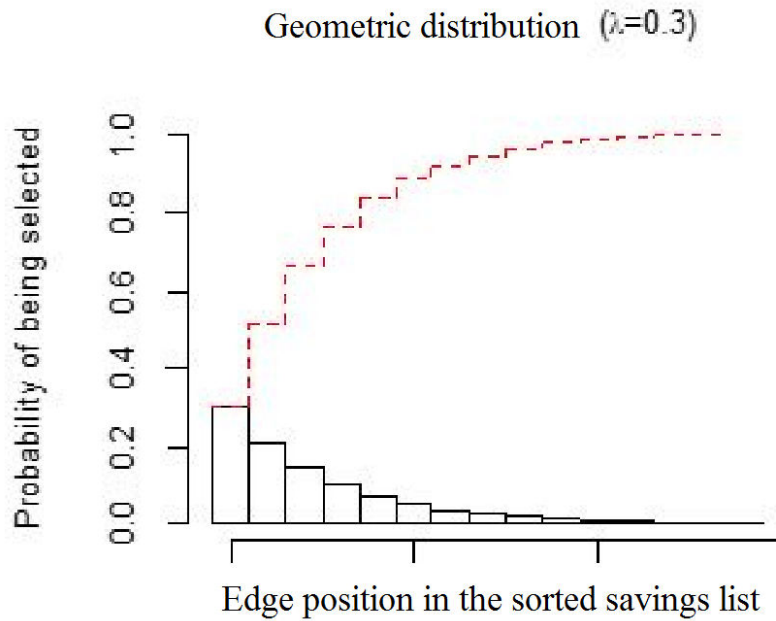


Figure 2.3: Sorted saving list and geometric distribution with parameter  $\lambda = 0.3$

---

**Algorithm 3** Procedure to generate a feasible solutions.

---

```

1: procedure FEASIBLESOLUTION(inputs, priorityList, paramAlgorithm)
2:   List  $\leftarrow$  copyList(priorityList) ▷ attributes inclusion
3:   sol  $\leftarrow$  constructInitialSol(inputs)
4:   while List is no empty do
5:      $\xi \leftarrow$  selectEdge(seed,  $\alpha$ ,  $\beta$ , List)
6:     iNode  $\leftarrow$  getOrigen( $\xi$ )
7:     jNode  $\leftarrow$  getEnd( $\xi$ )
8:     iR  $\leftarrow$  getRoute(iR, sol)
9:     jR  $\leftarrow$  getRoute(jR, sol)
10:    if feasiblemerging(sol, iR, jR, inputs) then ▷ constraints inclusion
11:      sol  $\leftarrow$  mergingRoutes(sol, iR, jR)
12:    end if
13:    deleteEdge(edge, List)
14:  end while
15:  returnsol
16: end procedure

```

---

illustrates a framework of the biased randomization of the selected heuristics.

The algorithm which uses biased randomization techniques is designed to opti-

mize a sustainable VRP. The inputs of this procedure are the algorithm parameters (*paramAlgorithm*) and the problem parameters (*inputs*). The data *inputs* includes related the attributes sustainability-related and problem data. For instance, the characteristics of the distribution network, type of nodes inside it (customers, depots), the nodes information (resources, requirements), the classical and particular constraints focused on sustainability.

Once the base solution is defined the main heuristic procedure starts from a good solution - the one obtained by the biased randomized of the heuristic-. Firstly, a diversification is performed to generate a new solution. Following, the new solution tend to be improved applying an intensification operator. As a result, if the performance of the new solution is better than the current base solution, the new one becomes the base solution for the next iteration. Moreover an acceptance criterion inspired on simulated annealing is implemented (Lourenço et al., 2010). It allows that new solution becomes the base one even when this one is a worst solution. Then entire procedure is repeated up to meet the stop criterion which could be a maximum computational time or a maximum number of iterations (*paramAlgorithm*).

## 2.4 Solving approaches for stochastic COPs: optimization and simulation techniques

Uncertain conditions not only persuade the operational costs but also it could affect negatively the society and the environment. Consequently, decision making under uncertain plays an important role to mitigate the effects of uncertainty on the performance of cities logistics. solving approaches for deterministic problems are powerful but have two main drawbacks: the underlying probability distributions must be known and the solutions can become infeasible for some conditions of random events. Thus, this thesis presents simheuristic framework to handle uncertainty while avoiding these drawbacks. The literature reports a wide number of studies related to deterministic rich VRP which disregard the impact of the of uncertainty. In practice, the uncertain conditions are an attribute related to the urban freight transport challenging the classical solving approaches for COPs.

In this sense, the term of *reliability* is integrated as an evaluation criterion related to the quality solution. The reliability and the objective function value of the problem define the quality of a solution for a stochastic problem. Thus, each solution for an stochastic problem has associated an expected objective function value of the problem and a reliability index. Both measures are evaluation criteria useful for decision making under uncertain conditions (Juan et al., 2015a). In addition, the solution reliability might reflect the robustness of the solving approach. In this context, the robustness can be defined as the performance of solving approach to

capture the problem attributes and achieve good quality solutions (see the Algorithm 4).

---

**Algorithm 4** Procedure to assess the solution performance

---

```

1: procedure MCS(sol, inputs, nSim)
2:   iter  $\leftarrow$  0
3:   reliabilities[failureType]  $\leftarrow$  0
4:   failures[failureType]  $\leftarrow$  0
5:   accumObjectiveValue  $\leftarrow$  0
6:   while (iter < nSim) do
7:     for (each route r in sol) do
8:       for (each edge (i, j) in r) do
9:          $\bar{\mu} \leftarrow$  generateStochasticVariable( $\mu$ )
10:        r.checkFeasibility(inputs)
11:        r.updateFailures()
12:        r.updateObjectiveValue(inputs)
13:      end for
14:      accumObjectiveValue  $\leftarrow$  accumObjectiveValue + r.getObjectiveValue()
15:    end for
16:    iter  $\leftarrow$  iter + 1
17:  end while
18:  reliabilities[failureType]  $\leftarrow$  1 - failures[failureType]/nSim
19:  expObjectiveValue  $\leftarrow$  accumObjectiveValue/nSim
20:  return statistics(sol)  $\triangleright$  expObjectiveValue and reliabilities
21: end procedure

```

---

For deterministic problems the solution quality is commonly estimated by a comparison between obtained solutions against the best known solutions (BKS) which are reported in the literature. There is not BKS reported in the literature for rich VRP, particularly for the sustainable VRP in stochastic conditions. Generally, for a deterministic problem, the BKS could be the optimal solution but it is not usually the optimal one. Despite, the solution quality is estimated from the gap between the BKS and the obtained solution for the deterministic problem. Similarly, some boundaries might be well-determined assuming the deterministic and stochastic conditions as the best (static and known information) and worst (dynamic and unknown information) solving conditions. Under this consideration, the relationship between the BKS and the best one obtained for stochastic problem (BSS) is depicted in 2.2. Thus,  $f(\text{solution}, \text{condition})$  is the expected objective function value reached by a *solution* in a specific *condition* of the problem. For a maximization problem,  $f(\text{BKS}, \text{best conditions})$  is the upper bound for the performance of the BSS assessed under stochastic conditions, i.e., the worst conditions. Then,  $f(\text{BKS}, \text{worst conditions})$  is

the lower bound. In contrast, for a minimization problem,  $f(\text{BKS}, \text{best conditions})$  is the lower bound meanwhile the  $f(\text{BKS}, \text{worst conditions})$  is the upper bound for a good solution in stochastic conditions.

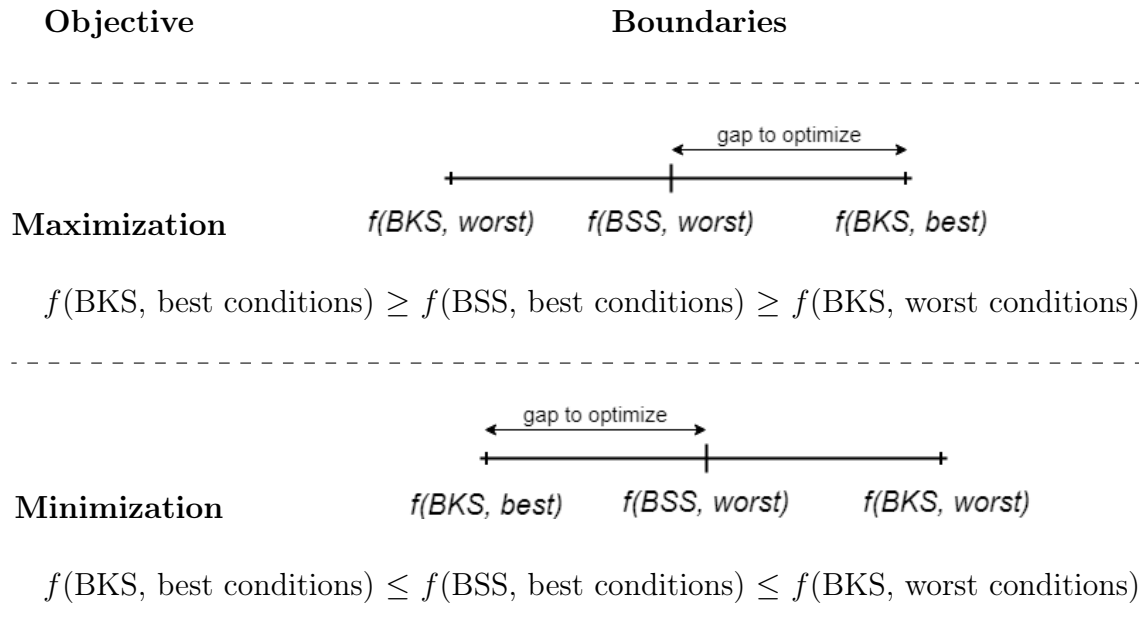


Table 2.2: The relationship between the BKS and the best stochastic solution (BSS)

## 2.5 Simheuristic algorithms

Juan et al. (2015a) describe the structure for solving a stochastic COPs by the current simheuristic approach. The Flowchart 2.5 provides oversight to simheuristic algorithm. First, a deterministic version of the stochastic instance is obtained assuming the stochastic variables  $\mu$  can be replaced by their expected value  $\bar{\mu}$ . Afterwards, the iterative process from the heuristic algorithm is extended by a Monte Carlo Simulation (MCS). The MCS addresses the metaheuristic algorithm to perform an efficient search inside the solution space associated with the deterministic version of the problem. For stochastic conditions, the quality and feasibility of the deterministic solution are estimated using simulation techniques.

According to the expected value of the objective function (*ObjectiveValue*), the solution performance is defined and employed to make a ranking of the best solutions for the stochastic problem. Therefore, the best deterministic solution refers to the one with the best objective function value. Moreover, an acceptance criterion is included. The acceptance criterion is employed to decide whether the

new solution is classified as promising or not. If this one is not promising, then it is discarded and another iteration starts. Otherwise, a MCS is applied to assess the expected objective function value, feasibility and reliability of the solution under stochastic conditions. Therefore, if the best solution found at that moment presents a worse expected performance, it is replaced by the new solution. In this stage, the simulation component only considers a short simulation to avoid jeopardizing the time of the optimization component. At the end of this stage, the solutions ranking contains the ones which report a good performance under stochastic conditions. Finally, in order to obtain more accurate estimates on solution performance, a large simulation is carried out for each of the solutions reported in the ranking.

In summary, the MCS relies on the following steps:

- Using random sampling from the assigned probability distributions, different executions of the routing plan is run in order to obtain random observations of the stochastic variable.
- For each sample the solution feasibility and its objective function are assessed and computed.
- From an intensive random observations, the solution reliability and expected the objective function value are computed for each routing plan. The reliability of each routing plan as the quotient between the number of route failures and the number of simulation runs.

The advantages of this approach are numerous: it benefits from the extensive literature research related to solve the deterministic version of the problem, and is capable of solving realistic problems, simple, easy-to-understand and to-implement, and efficient. After identifying a set of high-quality solutions (those that provide the highest average performance). A recent literature review on simheuristic methods for solving COP can be found in [Juan et al. \(2018\)](#).

## 2.6 Stochastic programming

Stochastic programming models assumes that the stochastic data can be estimated by a probability distribution. More generally, these models are formulated and solved analytically in order to provide information to decision maker. The most widely and studied programming model are the sample average approximation (SAA) that solving stochastic problems by using MCS. [Shapiro et al. \(2009\)](#) present a literature review of the most recent studies related with this solving approach.



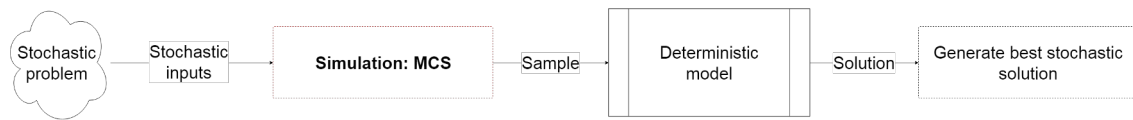


Figure 2.6: The SAA approach

In Figure 2.6 is described the SAA model which relies on two stages: firstly the sampling process and secondly the problem optimization. A MCS is performed to generate the sample of conditions which follow a probability distribution. From the sample a set of feasible solutions are defined assuming unknown the set of scenarios. In SAA, for each scenario in the sample an occurrence probability is assigned. Assuming  $x$  s the set of feasible solutions and  $\omega$  the set of scenarios which refer to the most likely conditions for problem. Thus, the value of  $f(x_i, \omega_i)$  represents the performance of the solution  $x_i$  under the condition  $\omega_i$ . Therefore, the complexity relies on the estimation of the expected objective function value. The complexity of this solving approach relies on the estimation of the expected objective function value. For a finite number  $n$  of scenarios in  $\omega$  with an occurrence probability ( $p(\omega_i)$ ) the expected value is computed as the  $f(x, \omega) = \sum_{i=1}^n f(\mathbf{x}_i, \omega_i) * p(\omega_i)$ . Finally, the value of  $f(x, \omega)$  indicates whether number of scenarios are suitable for solving the problem. Evidently, the number of scenarios grows exponentially the problem data, however the quality of  $f(x, \omega)$  depends on the sample, particularly on the variability between scenarios. The candidate solutions are given by a comparison between the obtained solution and the optimal one in the scenario  $\omega_i$ . Thus, the best solution is given by the one with the best expected value for  $f(x, \omega)$ .

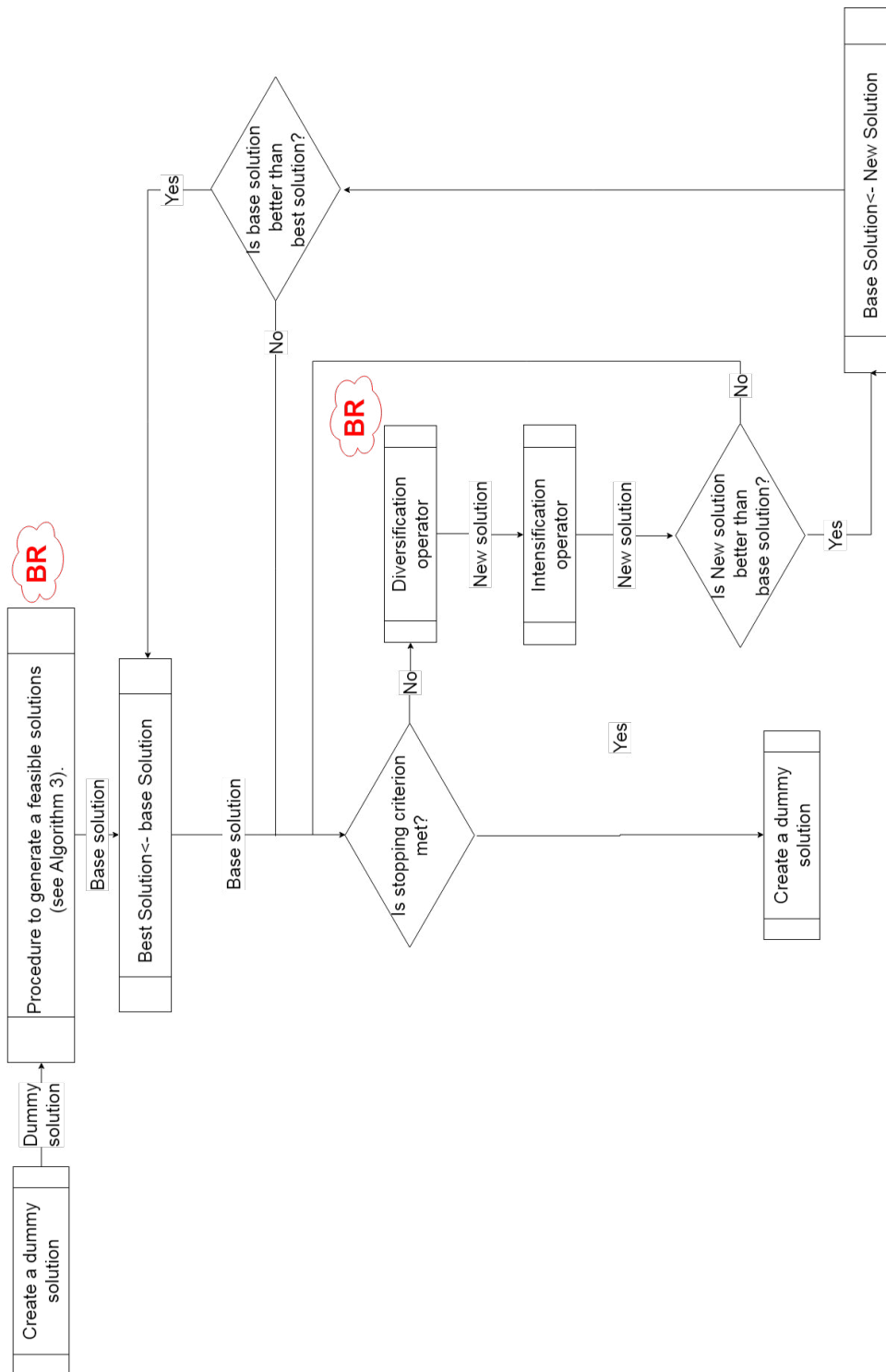


Figure 2.4: The biased randomization of heuristics

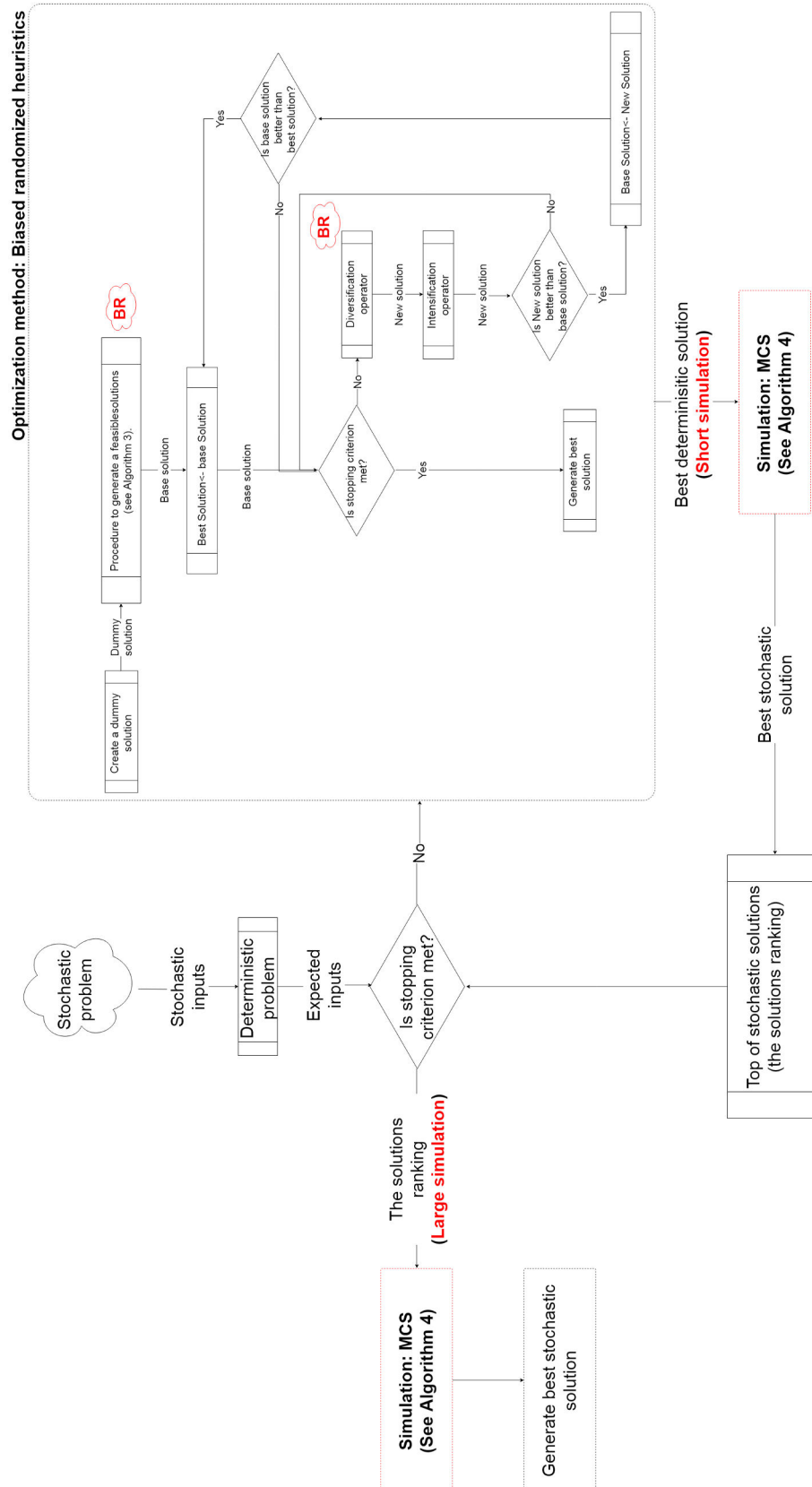


Figure 2.5: The simheuristic algorithm



## The sustainable multi-depot vehicle routing problem

*This chapter focuses on the distribution process in urban zones modeled as a multi-depot vehicle routing problem with several cost dimensions: economic, social, and environmental. A metaheuristic-based approach is proposed for tackling an enriched multi-depot vehicle routing problem in which economic, environmental, and social dimensions are considered. A series of computational experiments illustrates how the aforementioned dimensions can be integrated in realistic transport operations.*

The work presented in this chapter has been published in the Journal of Heuristics:

- Reyes-Rubiano, L., Calvet, L., Juan, A. A., Faulin, J., Bove, L. (2018). A biased-randomized variable neighborhood search for sustainable multi-depot vehicle routing problems. *Journal of Heuristics*, 1-22.

Part of the contents of this chapter has been presented at the following conferences:

- Reyes-Rubiano, L., Calvet, L., Juan, A. A., Faulin, J., (2017). Sustainable Urban Freight Transport Considering Multiple Capacitated Depots. In MIC 2017, Barcelona, Spain.
- Reyes-Rubiano, L., Calvet, L., Juan, A. A., Faulin, J. (2017). Solving the Multi-Depot Vehicle Routing Problem with Sustainability Indicators. In VeRoLog 2017, Amsterdam, Netherlands.

### 3.1 A BR-VNS simheuristic algorithm for solving the sustainable multi-depot vehicle routing problem

This chapter focuses on the distribution process in urban zones modeled as a multi-depot vehicle routing problem (MDVRP) with several cost dimensions: economic, social, and environmental. The MDVRP is a challenging combinatorial optimization problem, which has been widely studied (Pisinger and Ropke 2007; Cordeau and Maischberger 2012; Vidal et al. 2012; Subramanian et al. 2013; Vidal et al. 2014; Escobar et al. 2014; Juan et al. 2015b). A metaheuristic-based approach is proposed to tackle this problem when distribution costs depend on multiple sustainability indicators. As a consequence, for large instances of the problem –as the ones we might find in urban transport–, using a metaheuristic approach becomes a rational alternative (Talbi, 2009; Salhi, 2017). The proposed solving approach relies on the integration of biased-randomized (BR) techniques (Grasas et al., 2017) into a variable neighborhood search (VNS) framework (Hansen et al., 2010). In addition, a mixed-integer mathematical formulation is presented to define the problem and get optimal solutions for some small-size instances. The computational experiments also allow to compare the BR-VNS algorithm with an exact solver. More computational experiments are performed adapting benchmark MDVRP instances to gain insights into the problem and the relationships among the sustainable indicators.

To the best of our knowledge, this is the first work addressing a rich MDVRP including sustainability indicators (Caceres-Cruz et al., 2015). Accordingly, the main contributions of this chapter are: *(i)* the proposal of a rich MDVRP extension considering different sustainability dimensions; *(ii)* a BR-VNS algorithm which introduces biased-randomization techniques into a metaheuristic framework to better guide the solution-construction process; *(iii)* a mathematical formulation of the problem, which enables a direct comparison between exact methods and a heuristic-based approach; and *(iv)* a comprehensive analysis (including suitable visualization techniques) of the trade-off among the different sustainability indicators.

The rest of the chapter is structured as follows: Section 3.2 provides a literature review. Section 3.3 offers a detailed description of the problem analyzed, including a mathematical formulation. Section 3.4 proposes a BR-VNS algorithm. The computational experiments are explained in Section 3.5, while Section 3.6 discusses the results. Finally, Section 3.7 gathers the contribution of the chapter.

## 3.2 Literature review

The increasing social concern for the environment and a sustainable growth in general requires the transformation of cities. In this context, the classical VRP may be enriched to include characteristics that allow the reduction of environmental and social impacts in urban zones concerning transport activities. During the last decade, this problem has been complimented by a large number of variants including: the green VRP (green VRP) and the pollution routing problem (PRP). While the former is focused on the environmental impact caused by the fuel or energy consumption of transport, the latter takes into account the pollution and different emissions generated. Thus, both problems analyze the emissions and fuel/energy consumption levels, which depend on traffic congestion, speed, acceleration, type of road, type of vehicle, and load, among other internal and external factors of the operation (Bektaş and Laporte, 2011; Koc et al., 2014). Rich VRPs encompass special characteristics from city logistics and smart cities, *e.g.*, the integration of information technologies (IT) in transport operations or the inclusion of dynamism, stochasticity and other attributes related to the urban transport (Caceres-Cruz et al., 2015). The reader interested in a comprehensive review on the MDVRP is referred to Calvet et al. (2016).

Regarding environmental impacts, the distance and vehicle weight play a crucial role in the fuel/energy consumption and carbon emissions. Thereby, Ubeda et al. (2011) aimed at reducing transport costs and emissions by considering the distance and some variations in the vehicle maximum capacity. It is concluded that enhancing load factors (which may be achieved by using heterogeneous fleets) is an efficient way to get significant savings and environmental benefits. The authors also discuss negative externalities of transport, such as noise, air pollution, congestion, accident rate, energy consumption, and land use. There are studies tackling the negative impacts from three different perspectives: negative externalities, emissions released, and fuel consumption. Faulin et al. (2011), Liu et al. (2014), and Zhang et al. (2015) considered environmental indicators for the capacitated vehicle routing problem (CVRP). They affirm that the load variation defines fuel consumption and emissions caused by transport. Besides, the load variation influences the distribution processes profitability. In this line, Kuo (2010), Demir et al. (2015), and Xiao and Konak (2015) developed methodologies for the green heterogeneous VRP (green HVRP). These authors considered traffic congestion, road gradients, speed variations, and distance traveled as variables that influence fuel consumption in urban transport (Jabbarpour et al., 2015). More recently, Niknamfar and Niaki (2016) studied the MDVRP with time windows to optimize the customers-depots allocation and the vehicles selection, with the purpose of minimizing the environmental impact of the transport activity. They proved that a suitable allocation and coordination among stakeholders does not only reduce the negative impacts but also enhances the

total profit. [Juan et al. \(2014b\)](#) considered a supply chain with multiple suppliers for minimizing the empty trips and the travel distance in each route. They concluded that it is possible to reduce the  $CO_2$  emissions by 23% when the distribution process is carried out in collaboration with multiple suppliers. [Wang et al. \(2014\)](#) proposed the use of environmental criteria in order to reduce total operation costs. These authors developed an algorithm to integrate the economic and environmental goals based on the MDVRP with backhauls. [Demir et al. \(2015\)](#) considered the MDVRP with freight pick-up and delivery to ensure that any customer demand can be met from any depot, thus reducing the operation cost too.

Some studies have focused on the analysis of the environmental impact caused by transport activities in urban zones. However, there is no estimation of the real impact of these activities. About 60% of transport activities take place in urban regions, which concentrate around 80% of the population in some countries ([European Commission, 2015](#)). Social impact refers to health problems and other factors such as quietness, air quality, urban aesthetic, accessibility, and urban safety. The associated social costs are estimated through penalties, taxes, or willingness to pay. According to some studies, about 0.4%, 0.2%, 1.5%, and 2% of the GDP is related to air pollution problems, noise, accidents, and traffic congestion, respectively ([Caceres-Cruz et al., 2015](#)). Therefore, the sustainability concept has started to take part in the decision-making process. However, there is a lack of structured tools that allow the integration of the three dimensions and provide support to decision-makers ([Chen et al., 2013](#)).

There are only a few works analyzing sustainability criteria in freight transport. [Chibeles-Martins et al. \(2016\)](#) pose ecological criteria to determine an optimal structure of distribution networks. They solved a bi-objective problem focused on determining the suitable locations, capacities, and attributes in factories, warehouses, and a distribution center. The solution method is based on the simulating annealing metaheuristic framework, and Pareto optimality is considered to get a balance between economic and ecological concerns. In the same sense, [Zhang et al. \(2016\)](#) implement evolutionary algorithms to determine the optimal design of supply chains considering two possible scenarios: in the first one the transport is outsourced, while in the second one the transport is leased. It is a multi-objective problem aimed at minimizing  $CO_2$  emissions, fine dust, and costs. The authors implement the non-dominated sorting genetic algorithm-II (NSGA-II) as well as the strength Pareto evolutionary algorithm 2 (SEAP2). Both methods take into account Pareto optimality through a scalarization method computed by a weighted sum. Similarly, [Kadziński et al. \(2017\)](#) define a sustainable objective to design an optimal distribution structure considering a supply chain with multi-distribution channels. The considered objectives are two: maximizing customer coverage, and minimizing cost and environmental impacts.



### 3.3 Problem description

This chapter studies a supply chain problem with multiple suppliers and customers, which is formulated as a rich MDVRP. It is assumed that the distribution process is carried out by a homogeneous fleet of vehicles. The problem consists of optimizing the distribution routing plan considering different sustainability dimensions. The three components of sustainability (economic, environmental, and social impacts) are represented by travel distances and times, carbon emissions, and risk of accidents. Several studies have addressed the economic impacts as a variable mainly influenced by travel distances. Therefore, most existing models seek to minimize travel distances. However, achieving this goal does not guarantee a minimum economic impact, since many time-related factors are not being considered –e.g., congestion, speed limits, traffic signs, vehicle crashes, etc. (Wang et al., 2016). In fact, the shortest paths in urban zones are sometimes the slowest ones too. Accordingly, the tackled problem also considers travel times to represent these urban attributes.

Formally, the MDVRP can be defined on a complete undirected graph  $G=(N,A)$ , where  $N = \{N_d, N_c\}$  is a set of nodes,  $N_d$  and  $N_c$  represent the subsets of depots and customers respectively, and  $A = \{(i, j) : i, j \in N, i \neq j\}$  is the set of edges connecting all nodes in  $N$ . Each depot  $i$  ( $\forall i \in N_d$ ) has a capacity  $s_i$  ( $s_i > 0$ ), and each customer  $j$  ( $\forall j \in N_c$ ) has a demand  $r_j$  ( $r_j > 0$ ). The vehicle fleet  $K$  is composed of  $o$  identical vehicles ( $K = \{1, 2, \dots, o\}$ ). Finally,  $Q$ ,  $D$ , and  $Q_d$  denote, respectively, the capacity and the maximum-distance-allowed associated with each vehicle, and the capacity of each depot. Each edge  $(i, j) \in A$  has an associated travel time ( $t_{ij}$ ), and travel distance ( $d_{ij}$ ). Typically, the aim is to design a set of routes minimizing the total cost, which depends on travel distances or times. Besides satisfying the constraints related to the capacities and the maximum-distance for each vehicle, a feasible solution has to ensure that each customer is visited only once. In addition, each route must start and end at the same depot. The binary variable  $x_{ijk}$  is employed to represent the solution (i.e., the set of routes):  $x_{ijk} = 1$  if the edge  $(i, j)$  is traversed by vehicle  $k$ , and  $x_{ijk} = 0$  otherwise.

The rich MDVRP variant presented in this chapter aims to find a sustainable solution by assessing and minimizing the negative impacts associated. The following sustainability dimensions are considered:

- *Economic dimension.* This dimension is composed by total travel time and fuel consumption ( $f_{ij}$ ), which are monetized based on the driver wage ( $DW$ ), the vehicle fixed cost ( $FC$ ), and the oil price ( $C_f$ ). The costs of the route associated with vehicle  $k$  are computed as follows:

$$\sum_{(i,j) \in A} (DW + FC) \cdot t_{ij} \cdot x_{ijk} \quad (3.1)$$

$$\sum_{(i,j) \in A} C_f \cdot f_{ij} \cdot x_{ijk} \quad (3.2)$$

- *Environmental dimension.*  $CO_2$  emissions estimates assume that the internal combustion process of vehicles burns the carbon of the fuel and it is released as carbon dioxide. Thus, emissions are assumed to depend on fuel consumption. Expression (3.3) computes the cost of environmental impacts for the route associated with vehicle  $k$ , considering a factor for carbon emissions ( $C_e$ ).

$$\sum_{(i,j) \in A} C_e \cdot f_{ij} \cdot x_{ijk} \quad (3.3)$$

- *Social dimension.* Accidents are an externality caused by speed variations on roads, among other factors. These variations represent the state and stability of the roads, and are associated with an accident risk for pedestrians and vehicles (Wang et al., 2016). Expression (3.4) represents the social cost, and depends on a given coefficient ( $a_{ij}$ ), vehicle loading ( $y_{ijk}$ ), and travel distance. In particular, flow variables  $y_{ijk}$  represent the load in the route associated with vehicle  $k$  servicing customer  $j$  after visiting customer  $i$ .

$$\sum_{(i,j) \in A} a_{ij} \cdot d_{ij} \cdot y_{ijk} \quad (3.4)$$

The fuel consumption is estimated as suggested in Kuo (2010) and Zhang et al. (2016). Thus, in Equation (3.5),  $lph_{ij}$  represents the fuel consumption per unit of time, and  $p$  is a penalty for each additional load ( $M$ ). This value is determined by the average miles per fuel liter ( $kpl_{ij}$ ) and velocity ( $v_{ij}$ ) (Equation (3.6)):

$$f_{ijk} = lph_{ij} \cdot \frac{d_{ij}}{v_{ij}} \cdot \left( 1 + p \cdot \frac{y_{ijk}}{M} \right) \quad \forall (i, j) \in A, k \in K \quad (3.5)$$

$$lph_{ij} = \frac{v_{ij}}{kpl_{ij}} \quad \forall (i, j) \in A \quad (3.6)$$

Without loss of generality, it will be assumed that  $p$  is equal to 0. Thus,  $f_{ijk}$  can be represented by  $f_{ij}$ . All in all, the objective is defined on a multi-criteria function to minimize, which considers the total travel time, the total travel distance, the environmental cost, and the social cost:

$$\sum_{k \in K} \sum_{(i,j) \in A} (DW \cdot t_{ij} + FC + C_f \cdot f_{ij} + C_e \cdot f_{ij}) \cdot x_{ijk} + a_{ij} \cdot d_{ij} \cdot y_{ijk} \quad (3.7)$$

Based on the model presented by Mirabi et al. (2010), the constraints are as

follows:

$$\sum_{i \in N} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in N_c \quad (3.8)$$

$$\sum_{(i,j) \in A | j \in N_c} r_j \cdot x_{ijk} \leq Q \quad \forall k \in K \quad (3.9)$$

$$\sum_{(i,j) \in A} d_{ij} \cdot x_{ijk} \leq D \quad \forall k \in K \quad (3.10)$$

$$U_{lk} - U_{jk} + |N| \cdot x_{ljk} \leq |N| - 1 \quad \forall l, j \in N_c, k \in K \quad (3.11)$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} \quad \forall i \in N_c, k \in K \quad (3.12)$$

$$\sum_{j \in N_c} \sum_{k \in K} x_{ijk} \leq p \quad \forall i \in N_d \quad (3.13)$$

$$\sum_{i \in N_d} \sum_{j \in N_c} x_{ijk} \leq 1 \quad \forall k \in K \quad (3.14)$$

$$\sum_{j \in N_c} x_{ijk} = \sum_{j \in N_c} x_{jik} \quad \forall i \in N_d, k \in K \quad (3.15)$$

$$\sum_{k \in K} \sum_{j \in N_c} y_{ijk} \leq Q_d \quad \forall i \in N_d \quad (3.16)$$

$$\sum_{i \in N} y_{ijk} - \sum_{i \in N} y_{jik} = r_j \cdot \sum_{i \in N} x_{ijk} \quad \forall j \in N_c, k \in K \quad (3.17)$$

$$r_j \cdot x_{ijk} \leq y_{ijk} \leq (Q - r_i) \cdot x_{ijk} \quad \forall (i, j) \in A, k \in K \quad (3.18)$$

$$x_{ijk} = 0 \quad \forall i, j \in N_d, k \in K \quad (3.19)$$

$$y_{ijk} \geq 0 \quad \forall (i, j) \in A, k \in K \quad (3.20)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (3.21)$$

$$U_{lk} \geq 0 \quad \forall l \in N_c, k \in K \quad (3.22)$$

Equation (5.6) assigns each customer to exactly one route. Equation (5.7) limits the total demand that may be served by a vehicle. Equation (5.8) defines the maximum distance allowed per vehicle. Equation (5.9) eliminates sub-tours. The flow conservation is introduced by Equation (3.12). Equation (3.13) limits the number of routes to the same number of vehicles available in each depot ( $p$ ). Equation (3.14) imposes that each vehicle relates to a single route or zero. Equation (5.10) ensures that each route starts and ends at the same depot. Equation (5.11) makes sure that the total demand of the customers allocated to a depot is not greater than its capacity. Equation (5.12) states that the load in the vehicle arriving at customer  $j$

minus the demand of that customer equals the load in the vehicle leaving it after the service. Equation (5.13) sets lower and upper bounds for the loads. Equation (5.14) avoids the creation of routes among depots. Finally, Equations (5.15) and (5.17) define variable domains.

### 3.4 The BR-VNS solving approach

The methodology proposed to solve the MDVRP with different sustainability dimensions relies on the integration of Biased Randomization (BR) techniques within a variable neighborhood search (VNS) metaheuristic framework (Hansen et al., 2010, 2019). As discussed in Juan et al. (2013a) and Gragas et al. (2017), BR techniques make use of a skewed (non-uniform) probability distribution to efficiently transform a constructive heuristic into a randomized algorithm without losing the logic behind the heuristic. This strategy allows to guide the solution-construction process not only when the initial solution is generated, but also whenever a destruction-reconstruction process is employed inside the metaheuristic framework. Several works illustrate the effectiveness of this strategy when combined with a simple multi-start framework (Dominguez et al., 2014, 2016b).

In this chapter is extended these principles by employing a VNS framework. This metaheuristic relies on three facts: (i) a local minimum with respect to one neighborhood structure is not necessarily so for another; (ii) a global minimum is a local minimum with respect to all possible neighborhood structures; and (iii) for many problems, local minimum with respect to one or several neighborhoods are relatively close to each other. The basic structure of the solving approach is summarized in Algorithm 10. Algorithms 6, 7, 8 and 9 describe the following specific functions: creation of a sorted list of customers per depot, generation of feasible solutions, shaking of solutions to diversify the search, and local search. They are explained in detail below.

The inputs of the main procedure (Algorithm 10) are: the instance (describing information about the customers, the depots, and the parameters needed to compute the costs of the impacts), the dimension or optimization criterion, two parameters, so called *minPercent* and *maxPercent*, which are used during the shaking process, and two parameters for the Geometric distributions used in the BR process of the assignment and the routing processes (*betaMap* and *betaRoute*), respectively. The main procedure may be split in three stages. Initially, a priority list of eligible customers for each depot is created. The second stage generates top promising customers-depots maps. It starts by building a feasible solution, which is stored as the base solution (*baseSol*), and the best solution found so far (*bestSol*). In addition, this initial solution is included in a list of best solutions (*bestSols*). This list will store a few good solutions, which may be similar in terms of the impact of the dimension

chosen, but very different in terms of the other impacts. Thus, it allows a posteriori comparison of solutions considering the trade-off between dimensions not included in the objective function. Afterwards, a parameter  $p$  is defined, which will determine the degree of destruction in the shaking process, and is set to  $minPercent$ . A loop with a stopping criterion based on the elapsed time starts. First, it builds a new solution ( $newSol$ ) by applying a shaking procedure to  $baseSol$ , which is followed by a fast local search. The relative percentage difference ( $rpd$ ) (lines 14 and 15) between the costs of  $newSol$  and  $baseSol$  is computed. If  $newSol$  improves  $baseSol$  (i.e.,  $rpd < 0$ ), then the latter is replaced by the former, and  $p$  is set to  $minPercent$  minus 1. In addition,  $bestSols$  is updated and sorted if at least one of the following conditions are satisfied: (i) the number of solutions in the list is less than the maximum; and (ii) the worst solution is worse than  $newSol$ . While in the first case  $newSol$  is added to the list, in the second case the worst solution is replaced by  $newSol$ . Here an acceptance criterion proposed by [Hatami et al. \(2015\)](#) is introduced to avoid entrapment at local optimum. It determines that when  $newSol$  does not improve  $baseSol$  (i.e.,  $rpd$  is zero or negative), The former has to be replaced by the latter with a probability of  $exp(-rpd)$ . If the replacement takes place, then  $p$  is set to  $minPercent$  minus 1. The last step in a loop iteration is to increase  $p$  by one unit (provided that  $p < maxPercent$ ) in order to diversify the search. Notice that the increment sets this parameter to the minimum in those iterations where  $baseSol$  is updated. By doing this, the BR-VNS algorithm intensifies the search by considering a small neighborhood of  $newSol$ . Finally, an intensive routing process is applied to the map of each solution in  $bestSols$ . It consists of running the same BR routing algorithm with more iterations. After sorting the list, it is returned. Regarding the procedure to create priority lists (Algorithm 6), it is applied for each depot and starts with an empty list. Then, for each customer a measure called marginal savings is computed. It is a function of two measures: (i) the cost of a route going from the given depot to the customer and going back to the depot; and (ii) the same measure but considering the depot that is different to the given one and provides the lowest possible cost. The marginal saving per pair depot-customer is computed as the second measure minus the first. If positive and large, the corresponding depot-customer assignation is promising and should be prioritized. If negative and large, it will probably be better to assign that customer to a different depot. Having computed a list of all the marginal savings related to the depot, it is sorted in a decreasing order and returned. That list can not include those customers that are so far from the depot that the dummy route depot-customer-depot already exceeds the maximum distance allowed per route (if it exists).

The generation of a feasible solution (Algorithm 7) relies on a loop that creates solutions until a feasible one is returned. Each iteration of the loop starts by building a map of customer-depot assignations. These assignations are done by following

these steps: (i) BR is applied to the priority lists; (ii) the depot with the highest level of remaining capacity is selected; (iii) the first customer in the priority list of the depot chosen is assigned to that depot and erased from all the priority lists; and (iv) steps (ii) and (iii) are repeated until all customers are assigned. Provided that the resulting map satisfies the constraint of depots' capacity, a fast routing procedure is applied to each sub-map of depot and assigned customers. This procedure relies on the BR version (Juan et al., 2011b) of the classical Clarke and Wright savings (CWS) heuristic (Clarke and Wright, 1964). Notice that the two aforementioned papers compute savings from symmetric distances, while in this case it is employed the costs based on the user-chosen dimension. All the impacts are symmetric except the social one, which is a function of the load. Attempting to minimize the computing effort, if the user aims to minimize the social cost, then the routes are created assuming symmetric costs, which are estimated by the mean of the cost of the arcs. Finally, it is checked whether the restriction of number of vehicles is satisfied. In that case the solution found is labeled as feasible and returned.

The shaking procedure (Algorithm 8) is applied to generate a new solution by destroying and rebuilding *baseSol*. It selects a  $p$  percentage of the routes, rounding up the result. The customers of these routes are extracted, and a new partial solution is created considering only that subset of customers. Finally, the original and the new partial solutions are merged, creating a new solution that is returned.

The local search (Algorithm 9) uses a fast access storage of customers-set (hash) and associated route (object) to improve each new route in a solution following a sequential order. First, a repair procedure is applied per route, which aims to undo any possible dimension-based knot without exceeding the length of the route in the solution with the maximum length. Then, it is checked whether that subset of customers has been previously visited. In that case, we have the best route found stored. If the route being considered is worse than that stored, the latter is copied into the former. Otherwise, the new route replaces the stored one. If the subset of customers has not been considered previously, both the subset and the route are stored. After analyzing all the routes, the resulting solution is returned.

### 3.5 Computational experiments

The algorithm proposed in the previous section was implemented in Java and run on a personal computer with 8 GB of RAM and an Intel Core i7 of 1.8 GHz. In order to test it and illustrate its use to assess the trade-off among sustainability indicators, a set of computational experiments have been performed, which are described in this section.

### 3.5.1 Benchmark instances

Since there are no benchmark instances for the Sustainable MDVRP, here we adapt the benchmark instances for the classical MDVRP proposed by [Chao et al. \(1993\)](#) and recently used in [Vidal et al. \(2014\)](#). There are 23 instances with 50 to 360 customers and 2 to 9 depots. The vehicles' efficiency parameters are based on a light-duty vehicle employed for freight distribution in urban zones.

The cost coefficient proposed in [Zhang et al. \(2015\)](#) for  $CO_2$  emissions ( $C_e = 0.1$  USD/L) is used. Regarding the travel time cost, it is defined in [Koc et al. \(2014\)](#) as the sum of a vehicle fixed cost (FC) and a driver wage (DW), which are set to  $FC = 1.4$  USD/h and  $DW = 6.3$  USD/h, respectively. The travel distance cost is based on the price of fuel ( $C_f = 1.1$  USD/L) and the average miles per fuel liter ( $kpl_{ij} = 5.56$  km/L,  $\forall (i, j) \in A$ ). [Delucchi and McCubbin \(2010\)](#) propose the interval  $[1 \cdot 10^{-4}, 1.3 \cdot 10^{-3}]$  USD per kg-km for the coefficient  $a_{ij}$  needed to estimate the social cost. Without loss of generality, times are generated from distances using the formula  $t_{ij} = \alpha \cdot d_{ij} + \varepsilon_{ij}$ , where  $\alpha$  is a constant based on an estimated speed ( $\alpha^{-1} = 35$  km/h) and  $\varepsilon_{ij}$  represents external factors that define the correlation between travel time and distance. It is set to follow a truncated Normal distribution with a lower bound and mean equal to 0 and a standard deviation equal to 3.5, 2, and 0.5, respectively. These deviations are set in order to get a correlation around 0.5, 0.7, and 0.9, which may represent different road states: a high, medium and low congested zone, respectively. Thus, three scenarios are generated per instance. For example, given an edge with a travel distance of 10 km and a probability of 95%, its travel time could range inside one of the following intervals [0.59, 1.69]h, [0.64, 5.06]h, and [0.69, 8.42]h, depending on road states.

Each instance has been solved 10 times (employing a different seed for the random number generator) and the best solutions are reported. A maximum computing time of 6 minutes per run is considered. The parameter fine-tuning is performed empirically by testing 'reasonable' ranges. The parameters for the Geometric distributions related to the allocation and the routing process are randomly chosen in the intervals (0.5, 0.8) and (0.1, 0.2), respectively. A maximum of 5 solutions are stored in the list. The minimum and maximum levels of shaking, which define the neighborhoods, are set to 10%, and 50%, respectively.

The experimentation process consists of analyzing how the solution space changes according to the optimization criterion, and how it influences the remaining indicators. Thus, the following five options are considered: the optimization criterion is based on minimizing each of the four components of the objective function or the sum of all of them. The specific combination of goals in a real-life application will depend on the particular utility function of each particular decision-maker.

### 3.5.2 Validation

Aiming at validating and testing the efficiency of the proposed solving approach for the classical MDVRP, we run experiments considering the MDVRP instances. Table 3.1 compares the obtained results against the best known solutions (BKS), and the results of Vidal et al (2014b) (HGSADC+). The average gaps in terms of distances (shown in the last row) are 0.280%, and 0.276%, respectively. Regarding computational resources, HGSADC+ has been implemented in C++, compiled with g++ -O3, and run on an Opteron 250, 2.4 GHz CPU. The average computational time of HGSADC+ over the 10 associated runs to each instance is 10.10 minutes. In contrast, the average computational time of the proposed algorithm over the 10 associated runs to each instance is 4.76 minutes. Notice that BR-VNS algorithm has been implemented in Java (see Appendix B), while HGSADC+ has been implemented in C++.

The next step is to compare the BR-VNS algorithm against an exact solver. For doing that, the mathematical model described in Section 3.3 has been implemented using the GAMS language (Version 23.5.2), and the CPLEX solver Version 12.2.0.0 (see Appendix C).

First, 4 customers-depot clusters called sub-maps have been created from instance p07 (4 depots). Then, 3 types of sub-maps combinations have been defined, which leads to 10 tests. The combinations have been defined by the integration of different groups of sub-maps. In the first combination, all sub-maps have been considered as a data set. In the second combination, 3 sub-maps define a data set and the remaining sub-map (customers-depot) is another data set for the same test. Similarly, the third combination refers to a combination of pairs of sub-maps (considering each of them exactly once). Therefore, 14 different data sets have been solved in tests 1 to 9. For test 10, all sub-maps have been solved independently. The CPLEX solver stops when a solution with a duality gap lower than 10% is found, or when a maximum runtime, which depends on the number of nodes, has been exceeded. For data sets from 3 sub-maps, which have a total number of nodes between 77 and 104, the time is limited to 100,000 seconds. If we consider data sets from 2 sub-maps, which have at least 41 nodes, that time is limited to 5,000 seconds. Finally, the time is limited to 2,000 seconds when each sub-map is independently solved. Table 3.2 summarizes the results. The second column shows the distance cost while the third one reports the computational time employed by the solver.

As reported in the Table 3.2, CPLEX was unable to obtain solutions for tests 1 to 5 (*i.e.*, for sub-maps with 77 to 104 nodes). In the case of the remaining tests, CPLEX provides a feasible solution in a runtime of 4,981 seconds. However, for test 10 we reported a runtime of 1,567.88 seconds assuming that all sub-maps were solved through a parallel procedure. Regarding the BR-VNS algorithm, The best solution in distance terms reaches a cost around 9% lower than CPLEX solutions,



Table 3.1: Comparison of the BR-VNS algorithm for the classical MDVRP.

Inst.	$(p,  N_c ,  N_d )$	BKS (0)	HGSADC+ (1)	BR-VNS (2)	Gap (%) (2)-(0)	Gap (%) (2)-(1)
p01	(4,50,4)	576.87	576.87	576.87	0.00	0.00
p02	(2,50,4)	473.53	473.53	473.87	0.07	0.07
p03	(3,75,2)	640.65	640.65	641.19	0.08	0.08
p04	(8,100,2)	999.21	999.21	1003.49	0.43	0.43
p05	(5,100,2)	750.03	750.03	751.94	0.25	0.25
p06	(6,100,3)	876.50	876.50	876.50	0.00	0.00
p07	(4,100,4)	881.97	881.97	885.74	0.43	0.43
p08	(14,249,2)	4372.78	4375.49	4410.87	0.87	0.81
p09	(12,249,3)	3858.66	3859.17	3882.12	0.61	0.59
p10	(8,249,4)	3631.11	3631.11	3646.67	0.43	0.43
p11	(6,249,5)	3546.06	3546.06	3547.08	0.03	0.03
p12	(5,80,2)	1318.95	1318.95	1318.95	0.00	0.00
p13	(5,80,2)	1318.95	1318.95	1318.95	0.00	0.00
p14	(5,80,2)	1360.12	1360.12	1360.12	0.00	0.00
p15	(5,160,4)	2505.42	2505.42	2511.43	0.24	0.24
p16	(5,160,4)	2572.23	2572.23	2573.78	0.06	0.06
p17	(5,160,4)	2709.09	2709.09	2709.09	0.00	0.00
p18	(5,240,6)	3702.85	3702.85	3702.85	0.00	0.00
p19	(5,240,6)	3827.06	3827.06	3839.79	0.33	0.33
p20	(5,240,6)	4058.07	4058.07	4063.64	0.14	0.14
p21	(5,360,9)	5474.84	5474.84	5576.25	1.85	1.85
p22	(5,360,9)	5702.16	5702.16	5731.64	0.52	0.52
p23	(5,360,9)	6078.75	6078.75	6084.32	0.09	0.09
Average					0.280	0.276

on average, in a runtime of 227.26 seconds.

### 3.6 Computational results

This section assesses the performance of the BR-VNS algorithm under different road states and sustainability dimensions. Firstly, we analyze the impact of road states (congestion level) on the total cost and the sustainability dimensions. Secondly, a study of the trade-offs among the different dimensions is presented.

Figure 3.1 provides information regarding the proportions of costs per congestion level. In particular, it represents the average proportion of each cost component and

Table 3.2: Comparison of the BR-VNS algorithm against CPLEX.

Test	CPLEX		BR-VNS		
	Distance Cost (USD)	Time (s)	Distance Cost (USD)	Time (s)	Gap (%)
1	Not found	100000			-
2	Not found	100000			-
3	Not found	100000			-
4	Not found	100000			-
5	Not found	100000			-
6	457.52	4981.49	310.73	227.26	-32.08
7	310.71	4941.64			0.01
8	329.46	4911.94			-5.69
9	329.46	4911.94			-5.69
10	315.78	1567.88			-1.60
	Average				-9.01

its range per congestion level considering the solution found, for each instance, by minimizing the total cost (*i.e.*, the sustainable solutions). It can be observed that travel time represents the main cost, and its magnitude is the most sensitive to the scenario.

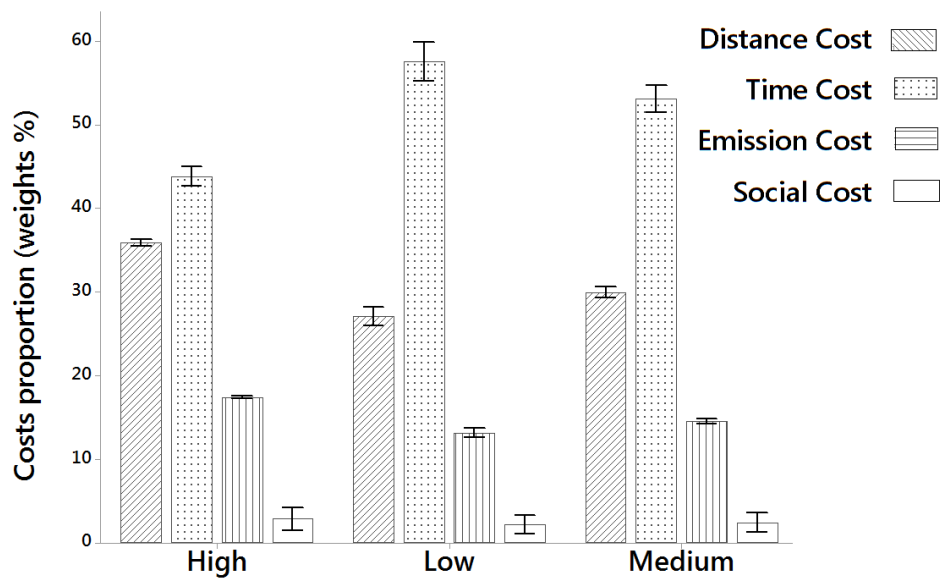


Figure 3.1: Average proportion of each cost component in the sustainable solutions.

Radar plots allow the study of the trade-offs between the sustainability dimensions. Figures 3.5 and 3.8 illustrate their use for instance p07 considering a medium

level of congestion. Each radar plot represents the solution minimizing the total cost (light polygon) and a solution minimizing one of the four measures (dark polygon): distances, time,  $CO_2$  emissions and social cost, respectively. The five axes of each plot represent the total cost and each component of that cost. The smallest and biggest pentagons link the minimum and the maximum values found (considering these 5 solutions). According to the radar plot 3.3, minimizing the distance cost leads to an increase in the time cost and, as a consequence, the total cost. The importance of the time dimension can be observed with radar plot 3.4, where the time cost is slightly reduced, but the distance,  $CO_2$  and social costs take the maximum values.

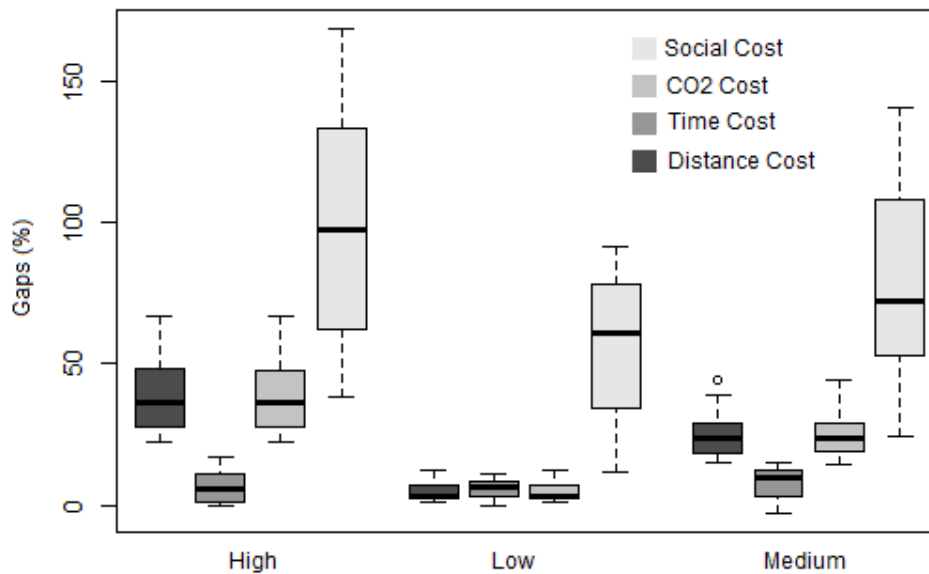


Figure 3.2: Gaps (%) between the sustainability solution and the solutions minimizing one measure per level of congestion.

Table 3.3 summarizes the trade-offs among solutions minimizing a different cost component for all instances with a high level of congestion. The second column shows the gaps of distance costs considering the solution minimizing the total cost and that minimizing the distance cost. Thus, applying the sustainable solution instead of the one minimizing the distance leads to a traveled distance 38.62% higher, on average. The third, fourth and fifth columns represent the trade-offs considering the other components: time cost,  $CO_2$ , and social cost, respectively. The highest mean gap (and variance) is related to the social cost, and the lowest to the time cost. Figure 3.2 displays the same gaps but studying the different levels of congestion. It illustrates the importance of considering a sustainable-aware approach for freight transport for scenarios with a non-negligible level of congestion: while the gaps are

small when the level is low (except for the social dimension, which represents the smallest proportion), they tend to increase and become more variable when that level increases.

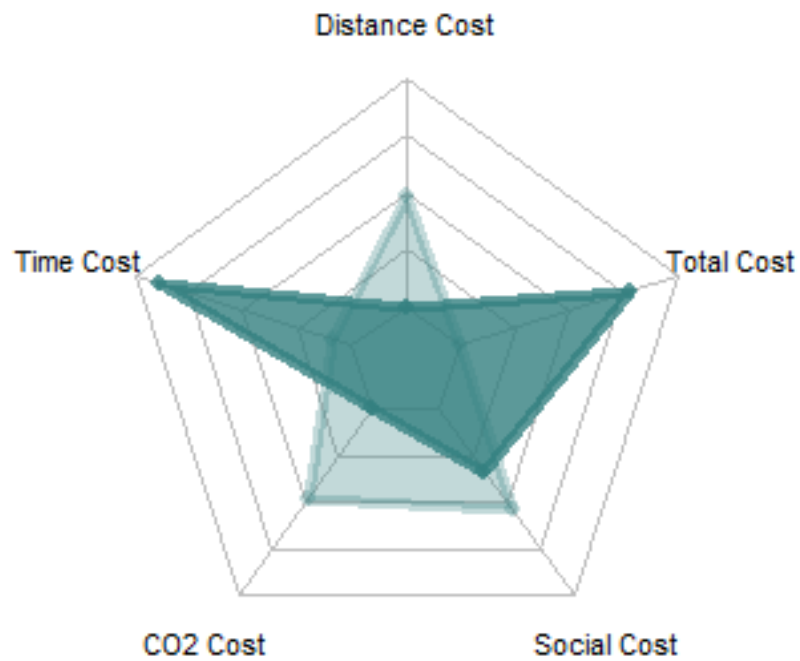


Figure 3.3: Minimizing distances



Figure 3.4: Minimizing time

Figure 3.5: Use of radar plots for decision-making. Criteria: travel distance and time.

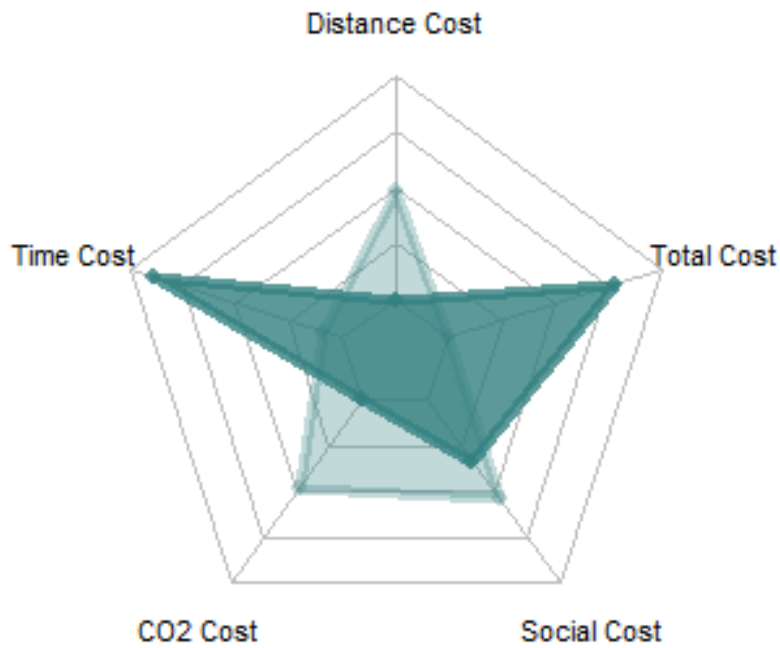
Figure 3.6: Minimizing  $CO_2$  emissions

Figure 3.7: Minimizing social cost

Figure 3.8: Use of radar plots for decision-making. Criteria:  $CO_2$  emissions and social cost.

Table 3.3: Gaps (%) between the sustainable solution and the solutions minimizing one measure considering a high level of congestion.

Instance	Distance Cost	Time Cost	$CO_2$ Cost	Social Cost
p01	39.83	2.64	39.83	76.26
p02	51.74	6.02	51.74	97.81
p03	60.69	3.18	60.69	161.80
p04	48.91	6.19	48.47	70.92
p05	67.07	16.35	67.07	140.18
p06	55.45	9.43	55.45	145.48
p07	48.98	7.23	48.98	94.36
p08	25.03	0.00	24.75	37.98
p09	34.25	5.98	34.34	46.82
p10	35.60	10.11	35.61	55.08
p11	32.81	11.26	32.63	63.71
p12	36.62	16.13	36.62	141.79
p13	36.44	0.00	36.44	124.12
p14	23.76	0.00	23.76	61.18
p15	45.66	16.63	45.66	168.62
p16	29.19	3.28	29.39	126.31
p17	25.50	0.00	25.50	74.02
p18	40.46	16.14	40.46	125.56
p19	25.62	3.47	24.88	109.66
p20	23.66	0.55	23.50	46.87
p21	47.22	16.83	47.22	152.93
p22	31.36	1.62	31.67	97.54
p23	22.32	0.73	22.43	51.54
Mean	38.62	6.69	38.57	98.72
S.D.	12.69	6.19	12.71	41.32

### 3.7 Contributions

This chapter aims at solving a richer version of the multi-depot vehicle routing problem including sustainability indicators. Estimates from the literature are considered in order to quantify and monetize the economic, environmental, and social impacts. While the first one is based on travel distances and times, the second and the third ones rely on carbon emissions and risk of accidents, respectively. In order to solve the corresponding optimization problem, we developed a solving approach based on the integration of biased-randomized strategies within a variable neigh-

borhood search framework. Thus, biased randomization is used at different stages of the metaheuristic in order to better guide the searching process. This includes both the generation of customer-to-depot assignment maps as well as the routing process itself. A set of computational experiments are carried out in order to test the BR-VNS algorithm and illustrate its use. The proposed algorithm is able to report high-quality solutions in short computing times, and enables decision-makers to assess solutions under particular interests regarding the impacts considered. Also, visualization techniques are used to compare solutions in different dimensions, which might be useful for decision-makers in order to understand the trade-offs between them.



**Algorithm 5** The BR-VNS algorithm for the sustainable MDVRP

---

```

1: procedure BR-VNS-SMDVRP(instance, dimension, minPercent, maxPercent,
   betaMap, betaRoute)
    $\triangleright$  instance: customers, depots, impactsParameters
    $\triangleright$  dimension: optimization criterion
 $\triangleright$  minPercent, maxPercent: min and max % of routes destroyed during the shaking
   process
    $\triangleright$  betaMap: parameter of the Geometric dist. used in the BR assignment process
    $\triangleright$  betaRoute: parameter of the Geometric dist. used in the BR routing process
   STAGE 0: Generate a priority list of customers for each depot
2:   priorityLists  $\leftarrow$  emptyList
3:   for each (depot in instance) do
4:     priorityList(depot)  $\leftarrow$  calcPriorityList(instance, dimension, depot)
5:     priorityLists  $\leftarrow$  add(priorityList(depot))
6:   end for
   STAGE 1: Fast generation of top promising customers-depots maps
7:   baseSol  $\leftarrow$  buildFeasibleSol(instance, priorityLists, dimension, betaMap,
   betaRoute)
8:   bestSol  $\leftarrow$  copy(baseSol)
9:   bestSols  $\leftarrow$  add(bestSol)  $\triangleright$  sorted array of top sols
10:  p  $\leftarrow$  minPercent
11:  while (stopping criterion is not met) do
12:    newSol  $\leftarrow$  shake(p, baseSol, priorityLists, dimension, betaMap, betaRoute)
13:    newSol  $\leftarrow$  fastLocalSearch(newSol, instance, dimension)  $\triangleright$  fast local search
14:     $\delta$   $\leftarrow$  cost(newSol, dimension, instance) - cost(baseSol, dimension, instance)
15:    rpdc  $\leftarrow$   $\delta$ /cost(baseSol, dimension, instance)  $\cdot$  100
16:    if (rpdc < 0) then  $\triangleright$  newSol improves baseSol
17:      baseSol  $\leftarrow$  newSol
18:      p  $\leftarrow$  minPercent - 1  $\triangleright$  reset neighborhood size
19:      if (number of sols in bestSols < arraySize(bestSols) or cost(newSol,
   dimension, instance) < cost(worst sol in bestSols, dimension, instance)) then
20:        bestSols  $\leftarrow$  update(bestSols, newSol)
21:        bestSols  $\leftarrow$  sort(bestSols, dimension)
22:      end if
23:    else
24:      u  $\leftarrow$  randomUniform(0,1)
25:      if (u < exp(-rpdc)) then  $\triangleright$  acceptance criterion
26:        baseSol  $\leftarrow$  newSol
27:        p  $\leftarrow$  minPercent - 1  $\triangleright$  reset neighborhood size
28:      end if
29:    end if
30:    p  $\leftarrow$  min{p + 1, maxPercent}  $\triangleright$  neighborhood increment
31:  end while
   STAGE 2: Intensive routing process of each map in top solutions
32:  for each (sol in bestSols) do
33:    sol  $\leftarrow$  intensiveRouting(sol, dimension, betaRoute)
    $\triangleright$  uses an intensive BR routing algorithm
34:  end for
35:  bestSols  $\leftarrow$  sort(bestSols, dimension)
36:  return best sol in bestSols
37: end procedure

```

---

---

**Algorithm 6** Procedure to calculate the priority list

---

```

1: procedure CALCPRIORITYLIST(instance, dimension, depot)
2:   priorityList  $\leftarrow$  empty
3:   for each (customer in instance) do
4:     marginalSavings(customer)  $\leftarrow$  cost(customer, altDepot(customer),
       dimension, instance) - cost(customer, depot, dimension, instance)
5:     priorityList  $\leftarrow$  add(customer, priorityList)
6:   end for
7:   priorityList  $\leftarrow$  sort(priorityList)
8:   return priorityList
9: end procedure

```

---



---

**Algorithm 7** Procedure to generate a new feasible solution

---

```

1: procedure BUILDFEASIBLESOL(instance, priorityLists, dimension, betaMap,
   betaRoute)
2:   feasible  $\leftarrow$  false
3:   while (feasible is false) do
4:     map  $\leftarrow$  buildBiasedRandMap(instance, priorityLists, betaMap)
        $\triangleright$  uses marginal savings and a BR round-robin process
5:     if (maxDemandInDepot(map)  $\leq$  depotCapacity) then
        $\triangleright$  checks if all depots can serve their assigned demand
6:       sol  $\leftarrow$  fastRouting(map, dimension, betaRoute)
        $\triangleright$  uses one execution of a BR routing heuristic
7:       if (maxRoutesInDepot(sol)  $\leq$  vehPerDepot) then
        $\triangleright$  checks if all depots have enough vehicles to cover their assigned routes
8:         feasible  $\leftarrow$  true
9:       end if
10:    end if
11:  end while
12:  return sol
13: end procedure

```

---

**Algorithm 8** Procedure to destroy-and-rebuild a solution

---

```

1: procedure SHAKE( $p$ ,  $baseSol$ ,  $priorityLists$ ,  $dimension$ ,  $betaMap$ ,  $betaRoute$ )
2:    $mutableSubSol \leftarrow \text{selectRoutesAtRandom}(p, baseSol)$   $\triangleright$  uniform random
   selection
3:    $immutableSubSol \leftarrow \text{extract}(baseSol, mutableSubSol)$ 
4:    $subInstance \leftarrow \text{extractInstance}(mutableSubSol)$ 
    $\triangleright$  extracts customers related to the selected routes
5:    $newSubSol \leftarrow \text{buildFeasibleSol}(subInstance, priorityLists, dimension, betaMap,$ 
    $betaRoute)$ 
6:    $newSol \leftarrow \text{aggregate}(immutableSubSol, newSubSol)$ 
7:   return  $newSol$ 
8: end procedure

```

---

**Algorithm 9** Procedure of local search

---

```

1: procedure FASTLOCALSEARCH( $newSol$ ,  $instance$ ,  $dimension$ )
2:    $maxLength \leftarrow \text{getMaxRouteLength}(instance)$ 
3:   for each ( $route$  in  $newSol$ ) do
4:      $route \leftarrow \text{repairRoute}(route, dimension, maxLength)$ 
    $\triangleright$  fast repair of any possible dimension-based knot in the route
5:     if ( $\text{customersSet}(route)$  is in storage) then
6:        $storedRoute \leftarrow \text{getStoredRoute}(\text{customersSet}(route))$ 
7:       if ( $\text{cost}(route, dimension, instance) < \text{cost}(storedRoute, dimension,$ 
    $instance)$ ) then
8:          $storedRoute \leftarrow route$ 
9:       else
10:         $route \leftarrow storedRoute$ 
11:      end if
12:    else
13:       $storage \leftarrow \text{add}(\text{customerSet}(route), route)$ 
14:    end if
15:  end for
16:  return  $newSol$ 
17: end procedure

```

---



## The stochastic electric vehicle routing problem using energy safety stocks

*The electric vehicle routing problem with stochastic travel time is a concern in the transport sector. For example, the freight distribution using automated vehicles. In this context, the automated driving system on the vehicle performs itself all driving tasks and monitor the battery energy consumption rate. Thus, a preventive policy is implemented for handling the operational risk of getting run out of battery over the course of the route. The proposed simheuristic algorithm combines Monte Carlo simulation with a multi-start metaheuristics, which also employs biased-randomization techniques. Thus, the main goal is to minimize the expected time-based cost required to complete the freight distribution plan.*

The work presented in this chapter has been published in the Statistics and Operations Research Transactions:

- Reyes-Rubiano, L.; Ferone, D., Juan, A. A.; Faulin. J. (2019). A Simheuristics for Routing Electric Vehicles with Limited Driving Ranges and Stochastic Travel Times. Progress in Statistics and Operations Research Transactions. Accepted for publication.
- Reyes-Rubiano, L. S., Faulin, J., Calvet, L., Juan, A. A. (2017, December). A simheuristic approach for freight transport in smart cities. In Proceedings of the 2017 Winter Simulation Conference (p. 274). IEEE Press.

## 4.1 A BR-MS simheuristic algorithm for solving the stochastic electric vehicle routing problem

The use of electric vehicles in freight transport is a promising option to mitigate the negative impacts caused by transport activities in city logistics. For instance, EVs require extra operational efforts due to the limited life of their batteries, the amount of time required to refill them, and the lack of recharging stations in modern cities. These technical limitations introduce driving-range constraints that do not exist in the case of traditional internal combustion vehicles (Juan et al., 2016). Thus, the EVs integration leads to new optimization problems.

The battery life is an issue that can be conceived as a hard constraint for new routing problems. The energy consumption rate depends on a wide range of random or difficult to predict factors, such as traffic congestion, road characteristics affecting the energy consumption, weather conditions, driving style, among other factors that involve uncertainty. Accordingly, this chapter analyzes the electric vehicle routing problem with stochastic travel times (EVRPST), which also considers time-based driving-range constraints (Figure 4.1). Being a rich extension of the classical vehicle routing problem (VRP), the EVRPST is also an *NP-hard* optimization problem, which justifies the use of heuristic-based solving approaches. The main goal is to design an ‘efficient’ routing plan that satisfies a set of customers’ demands using a homogeneous fleet of electric vehicles, each of them characterized by a limited loading capacity and driving range. Furthermore, this chapter considers a more realistic VRP in which transport times are not deterministic but random variables instead. The main goal is to the minimization of the total expected time necessary to complete the delivery. Notice that random travel times could cause the exhaustion of the vehicle battery before completing its assigned route. Such a route failure will require a costly corrective action, which will be also measured in time units (Eshtehadi et al., 2017).

To solve the EVRPST, a novel simheuristic approach integrating Monte Carlo simulation within a multi-start framework is proposed (SIM-BR-MS). A review on basic concepts of simheuristic algorithms can be found in Juan et al. (2015b). Also, the generation of solutions inside the multi-start framework is based on the use of biased-randomized techniques, which allow to extend deterministic heuristics into enhanced probabilistic algorithms. Grasas et al. (2017) provide an updated review of biased-randomized algorithms. The SIM-BR-MS algorithm considers the use of energy safety stocks, i.e.: during the design of the routing plan, a certain percentage of the battery is reserved for covering emergency situations with higher-than-expected travel times. Notice that using higher levels of safety stock leads to shorter routes and a higher number of required vehicles. In contrast, using lower levels of safety stock will increase the probability of suffering a route failure. Whenever this occurs,

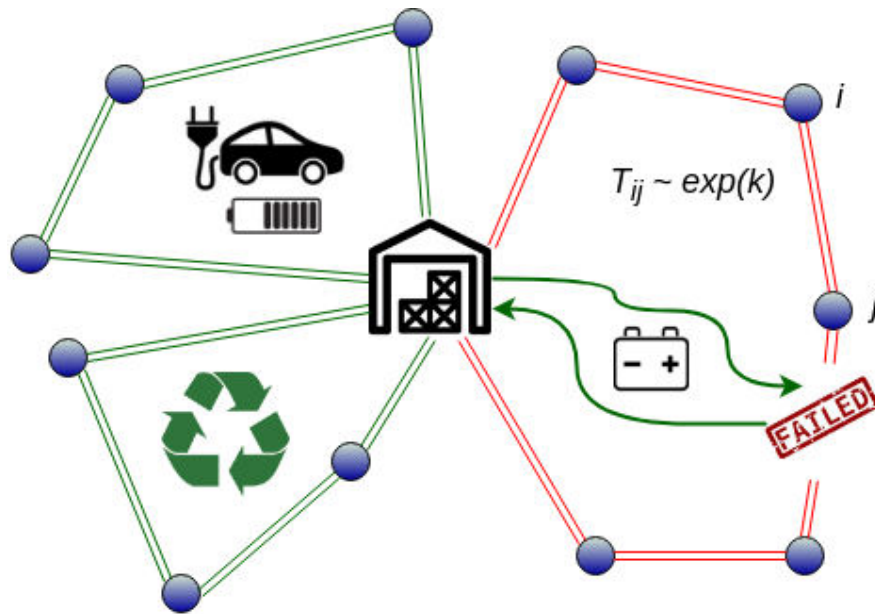


Figure 4.1: A simple representation of the EVRPST with driving-range constraints.

it is assumed that the failing battery has to be replaced by a new one. In the computational experiments, this corrective action has a time-based penalty cost equivalent to a round-trip from the depot to the current position of the battery that needs to be replaced. All in all, the main contributions of this chapter are: *(i)* to mitigate the lack of works on vehicle routing problems considering both driving-range limitations and uncertainty conditions; *(ii)* to develop and test the SIM-BR-MS algorithm approach for the EVRPST; and *(iii)* to analyze the effect of random travel times and the use of energy safety stocks on the routing plans.

The remaining of the chapter is structured as follows: Section 4.2 reviews related work in the transport literature; Section 4.3 provides some additional details on the problem under study; the SIM-BR-MS algorithm is explained in Section 4.4; Section 4.5 describes a series of computational experiments, while the associated results are discussed in Section 4.6; finally, Section 4.7 highlights the findings and identifies potential lines for VRP using EVs.

## 4.2 Literature review

The use of EVs in transport activities is related to several urban changes in terms of infrastructure and distribution strategies. On one hand, some of these challenges relate to infrastructure and fleet configurations (Juan et al., 2014c; Shao et al., 2017). On the other hand, EVs have started to replace conventional vehicles in city logis-

tics, redefining transport operations (Hof et al., 2017). Many logistics and transport problems in smart cities can be modeled as rich VRP variants (Caceres-Cruz et al., 2015). The rich VRP has been a very active research line in combinatorial optimization problems. This is partly due to the difficulty of managing multiple attributes and constraints, such as the different sustainability dimensions: economic, social, and environmental McKinnon et al. (2015). In particular, the green VRP is a rich VRP which considers routing problems using alternative fuel vehicles Erdogan and Miller-Hooks (2012). Figure 4.2 provides a scheme that summarizes different attributes and constraints frequently associated with the green VRP Lin et al. (2014).

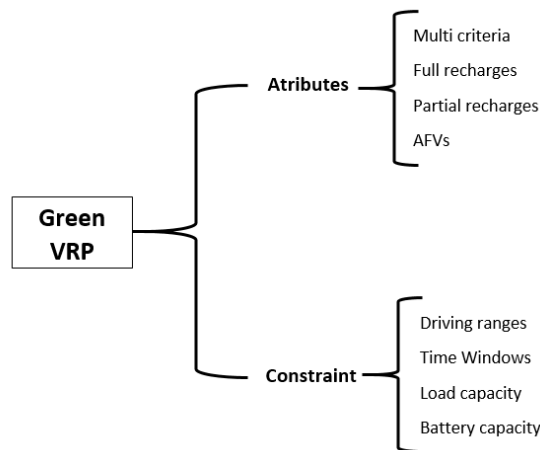


Figure 4.2: Frequent attributes and constraints in the green VRP

### 4.2.1 The deterministic green VRP

A key restriction in VRPs with EVs is the limited capacity of their batteries, which might require multiple recharging stops. Hence, Erdogan and Miller-Hooks (2012) solve a green VRP allowing intermediate stops by implementing procedures based on the well-known savings heuristics (Clarke and Wright, 1964) and the popular density-based clustering algorithm. Demir et al. (2012) solves a PRP with time windows, where customer sequences are first defined and, afterwards, the travel speeds are optimized by means of an adaptive large neighborhood search (ALNS) metaheuristics. Juan et al. (2014c) address the green VRP with multiple driving ranges. The goal of this work is to define alternative fleet configurations based on EVs and hybrid-electric vehicles. The authors describe an integer programming formulation and a multi-round heuristic algorithm that iteratively constructs a solution. Schneider et al. (2014) propose an ALNS metaheuristics with some local searches with the aim of minimizing the total distribution cost –which includes the



cost of using a fleet of vehicles plus the actual routing cost. Additionally, these authors considered intermediate stops in recharging stations. Similarly, the ALNS metaheuristics is hybridized with the adaptive variable neighborhood search framework by [Schneider et al. \(2015\)](#), who deal with a routing problem with EVs-related constraints and also consider intermediate stops. [Koc and Karaoglan \(2016\)](#) design a simulated annealing metaheuristics, based on an exact method, to solve the green VRP for the small-scale instances proposed by [Erdogan and Miller-Hooks \(2012\)](#). [Hiermann et al. \(2016\)](#) study the VRP with EVs, time windows, and recharging stations. [Hof et al. \(2017\)](#) consider EVs to solve a location-routing problem where the objective is to determine whether the battery swap stations should be defined from candidate locations or closer to the set of customers. Finally, the green VRP with multiple objectives –including both monetary and environmental costs– is discussed by [Sawik et al. \(2017b\)](#).

### 4.2.2 The stochastic green VRP

Stochastic combinatorial optimization has received increasing interest during the last decades ([Bianchi et al., 2009](#); [Ritzinger et al., 2016](#)). Solving a stochastic VRP requires a methodology able to deal with the random components of the problem, which is not straightforward, as discussed in [Juan et al. \(2011a, 2013b\)](#). The most frequent random variables are: customers' demands, service and travel times, and frequency of order placing ([Bozorgi et al., 2017](#)). The previous articles highlight the importance of dealing with uncertainty, and study realistic characteristics such as urban transport dynamics. In most existing works, travel times are assumed to be constant, but this is not a realistic assumption. Hence, [Ritzinger et al. \(2016\)](#) propose to deal with uncertain travel times by modeling them as stochastic and time-dependent variables.

Uncertainty conditions are sometimes addressed by means of stochastic programming. This approach provides high quality solutions for small instances ([Bozorgi-Amiri et al., 2013](#)). [Erdogan and Miller-Hooks \(2012\)](#) present an exact model to solve the VRP with stochastic travel times. These authors assess the influence of route duration on environmental indicators, such as energy consumption. Another relevant problem is the time-dependent VRP, where the travel times are different depending on the specific period. [Gendreau et al. \(2015\)](#) provides a literature review on these topics. Travel times may vary by exogenous variables, such as traffic congestion, weather conditions, moving targets, or mobile obstacles. They might also be influenced by endogenous variables, e.g.: by varying the vehicles' speeds or by choosing highways over standard roads.

Recently, [Eshtehadi et al. \(2017\)](#) address a VRP with stochastic demands and travel times. These authors develop a solving approach based on an exact method that is able to solve instances with up to 20 nodes considering multiple scenarios.

The authors tackle the stochasticity describing two scenarios that represent the best and the worst conditions for demand and travel times. To conclude this literature review, Table 4.1 summarizes some illustrative works providing evidence about the most studied green VRP variants.

Table 4.1: An illustrative set of works covering the most popular green VRP variants.

Studies	Attributes	Constraints	Solution Approach
(Shao et al., 2017)		Driving range	GA
(Eshtehadi et al., 2017)	Stochastic demands Stochastic travel times	Driving range	SP
(Sawik et al., 2017b)	Multi criteria	Driving range	EM
(Koc and Karaoglan, 2016)		Driving range	SAM and EM
(Hiermann et al., 2016)	Full recharges	Time windows	EM
(Desaulniers et al., 2016)	Full recharges Partial recharges	Driving range Time windows	EM
(Schneider et al., 2015)	Full recharges	Driving range	ALNS
(Felipe et al., 2014)	Full recharges	Driving range	GSA
(Juan et al., 2014d)	heterogeneous fleet	Driving range	RMS
(Schneider et al., 2014)	Full recharges	Driving range	ALNS
(Erdogan and Miller-Hooks, 2012)	Stochastic travel time Full recharges	Driving range	SP

ALNS: Adaptive large neighborhood search. EM: Exact method. SP: Stochastic programming.  
RMS: A randomized multi-start algorithm. GA: Genetic algorithm. GSA: Greedy algorithm. SH: Savings heuristics.  
SAM: Simulated annealing metaheuristics. DC: Density-based clustering algorithm

### 4.3 Additional details on the EVRPST

The EVRPST is defined on an undirected graph  $G = (N, A)$ . Here,  $N$  contains the depot (node 0) and a set of customers  $N^* = \{1, 2, \dots, n\}$ . Also,  $A = \{(i, j) \mid i, j \in N, i \neq j\}$  is the set of edges connecting any two nodes in  $N$ . Each customer  $i \in N^*$  has a demand  $d_i > 0$ . There is a set  $V$  of homogeneous vehicles, each of them with a loading capacity of  $q > \max\{d_i\}$ . As it is usual in most VRPs (Toth and Vigo, 2014), the following assumptions hold: (i) all customers' demands must be satisfied; (ii) each vehicle route starts and ends at the depot; (iii) each customer is visited exactly once; and (iv) the demand to be served in each route does not exceed the vehicle loading capacity. Moreover, the time-based cost of traversing each edge  $(i, j)$  is given by an independent random variable  $T_{ij} = T_{ji} > 0$ , which follows a known probability distribution with mean  $E(T_{ij}) = t_{ij}$ . Thus, the additional constraint is considered as well: the expected travel time employed by a vehicle to complete its route is limited by the battery duration,  $t_{max} > \{\sum E[T_{ij}]\}$ .

However, considering stochastic travel times implies introducing uncertainty about how much energy will be required to complete a route. Energy consumption and travel times depend on multiple factors, such as: current load of the vehicle, road type, vehicle speed, driving skills, etc. This uncertainty makes it hard to guarantee feasible solutions when hard time-related constraints on batteries duration are considered. In particular, electric vehicles have a risk of batteries exhaustion during the trip, which is considered as a *route failure*. Decision makers may define corrective actions to properly address these failures when they happen. They might also define preventive actions to be applied before the vehicle runs out of battery. Figure 4.5 illustrates some examples of these types of actions.

On the one hand, a corrective action to resume the routing plan is required when a vehicle  $A$  runs out of energy after visiting a customer  $j$  (failure type I). In the computational experiments, it is assumed that the cost of this corrective action is the time needed for a new vehicle  $B$  to complete a round-trip from the depot to the current location of  $A$  to supply a new battery. On the other hand, a preventive action could also be applied: if there is a high risk of running out of battery after serving a customer  $j$ , vehicle  $A$  might decide to return from  $j$  to the depot for re-charging or swapping batteries (failure type II); after that, it might resume its planned route from the next customer,  $k$ . The time-based cost of such a preventive action could be estimated as the time requested to visit the depot for recharging batteries plus the time employed in moving from the depot to the next customer in the original route,  $k$ .

Although the simheuristic methodology is quite flexible and could be easily extended to consider preventive actions, in the computational experiments it is only considered corrective actions (i.e., type I failures). Accordingly, the objective function minimizes the expected time-based cost required to complete the delivery process. Notice that this time-based cost is a non-smooth function, since it includes the ‘penalty’ cost associated with applying these corrective actions whenever route failures occur. Hence, if  $T_v$  represents the total time employed by vehicle  $v$  in completing its route, the objective function can be expressed as:

$$\min E \left( \sum_{v \in V} T_v \right) \quad (4.1)$$

with:

$$T_v = \begin{cases} \sum_{\substack{i,j \in N \\ i \neq j}} T_{ij} \cdot z_{ijv} & \text{if } \sum_{\substack{i,j \in N \\ i \neq j}} T_{ij} \cdot z_{ijv} \leq t_{max} \\ \sum_{\substack{i,j \in N \\ i \neq j}} T_{ij} \cdot z_{ijv} + 2 \cdot T_{j0} & \text{otherwise} \end{cases} \quad (4.2)$$

where the decision variable  $z_{ijv}$  takes the value 1 if vehicle  $v$  covers the edge  $(i, j)$ ,

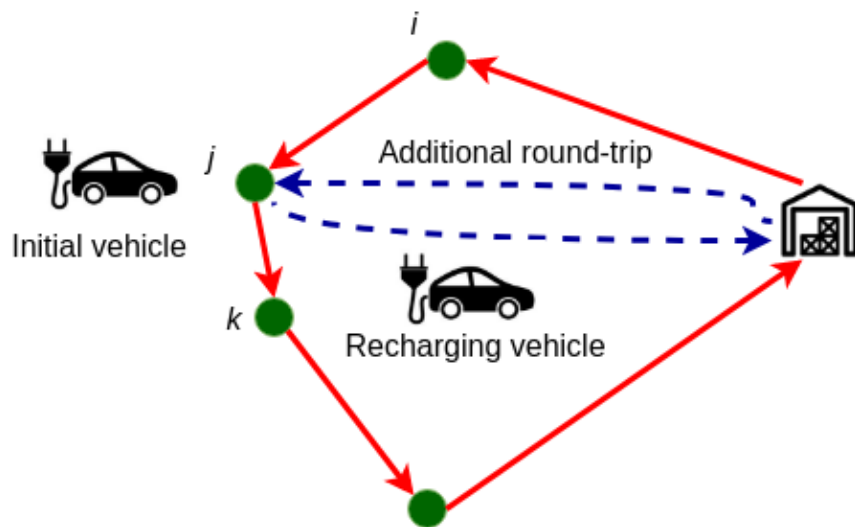


Figure 4.3: Corrective action for type I failure

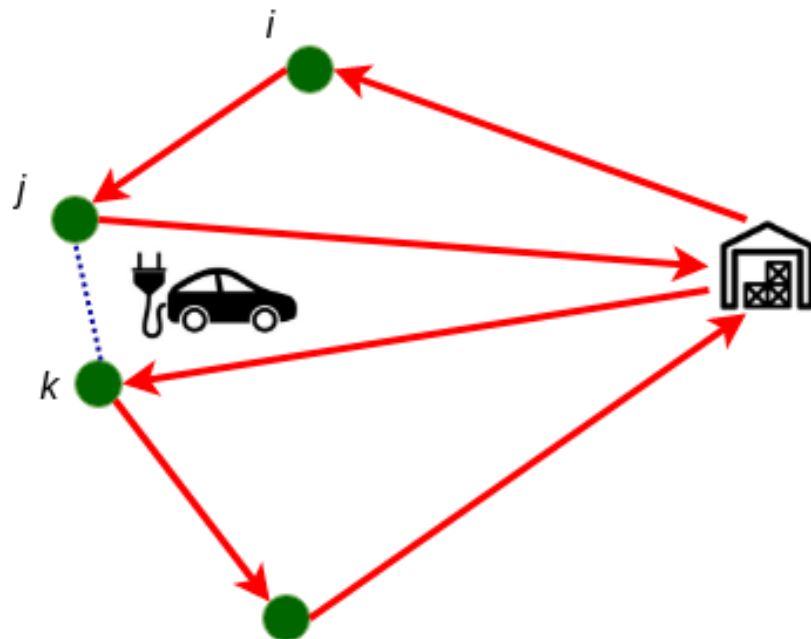


Figure 4.4: Preventive action for type II failure

Figure 4.5: Different actions to deal with route failures while using electric vehicles

while it takes the value 0 otherwise.

## 4.4 The SIM-BR-MS algorithm solving approach

The solving methodology relies on a simheuristics approach, which proposes the integration of simulation techniques within a heuristic framework to address stochastic optimization problems in a natural way (Juan et al., 2018). Biased-randomized versions of a constructive heuristics allows for fast generation of high-quality solutions (Grasas et al., 2017). When complemented with some local search and encapsulated inside a multi-start (or similar) framework, they constitute a strong basis that can be easily extended into a simheuristic framework (Grasas et al., 2016). The Sim-BR-MS algorithm builds upon the enhanced version of the Clarke and Wright Savings (CWS) heuristics (Clarke and Wright, 1964), which is called Biased-Randomized CWS (BRCWS) (Juan et al., 2011b). The complete algorithm is summarized in Algorithm 10 and described next in more detail.

First, the stochastic instance is transformed into a deterministic one by using expected travel times as initial estimates for the real stochastic values. Then, following the Clarke and Wright (1964) heuristics, a dummy solution is created and the savings associated with traversing each edge are computed. This initial solution (*initSol*) is improved by the classical *2-Opt* local search operator, and its expected travel time (stochastic cost) is estimated by using a short MCS with just *sSim* runs –typically in the order of a few hundreds. Notice that, as any other solution the SIM-BR-MS algorithm will generate, *initSol* will have two time-based costs: the one associated with the deterministic version of the problem (*detCost*) and the one associated with the stochastic one (*stochCost*). At this stage, *initSol* is stored as the temporary reference or ‘base’ solution (*baseSol*) and included in a list of ‘elite’ stochastic solutions (*bestStochSolList*). Afterwards, a multi-start process is repeated until a termination criterion (*maxTime*) is met. In each iteration, a new deterministic solution (*newSol*) is generated by using the BRCWS procedure. Once a fast local search is applied, this solution is labeled as ‘promising’ if its deterministic time-based cost is lower than that of *baseSol*. If it is not promising, *newSol* is discarded and a new iteration starts. If it is promising, a new short MCS is applied to estimate the stochastic cost (expected time) associated with *newSol*. Whenever appropriate, *baseSol* is replaced by *newSol* and the *bestStochSolList* is updated. Once the ending criterion is met, the expected time associated with each elite solution in *bestStochSolList* is assessed again, this time using a more intensive MCS with *lSim* runs –typically in the order of a few thousands. Notice that while the assessments in the main loop are required to be fast, because the number of solutions to assess may be extraordinarily high, those applied to a reduced list of elite solutions can employ more computing time.

The computational time of the algorithm is bounded by *maxTime*. Regarding its computational complexity, each iteration has three stages: the construction with BRCWS, the local search, and the simulation phase. The computational complexity

of BRCWS is bounded by the number of the edges  $m$ , since the merging can be done in constant time but it is necessary to examine all savings. Since each client is served exactly once, the local search swapping moves are bounded to  $O(n^2) = O(m)$ . Finally, the complexity of the simulation stage is  $O(m \cdot sSim)$ . Therefore, the complexity of each iteration is dominated by the simulation phase, and it is  $O(m \cdot sSim)$ .

As usually done in the related literature (Grasas et al., 2017), the biased-randomized procedure is based on the use of a geometric probability distribution, which makes use of a parameter  $\beta$  ( $0 < \beta < 1$ ). The BRCWS heuristics is adapted from the one proposed by Juan et al. (2011b) to ensure the feasibility of the generated solutions. In particular, it is guaranteed that the expected travel time of each vehicle will not exceed the duration of the batteries. However, as discussed before, under stochastic conditions it is not possible to guarantee that a route is failure-free. Accordingly, the reliability of each solution (i.e., the probability that a solution does not suffer any route failure) is also estimated from the data obtained in the previous simulation runs. As a way to increase these reliability levels, different levels of safety stock are considered for each vehicle. In other words, during the route-design stage, a given percentage of the vehicle driving-range capacity ( $s\%$ ) is reserved as a safety stock to be used in case of higher-than-expected travel times. The specific value of  $s$  is a decision variable to be determined during the simulation-optimization process, since it will depend on the specific instance being analyzed as well as on the probability distribution used to model travel times.

Notice that a relatively high value of  $s$  leads to short and reliable routes, i.e., routes employing short travel times and with a low probability of experiencing a failure due to the existence of a noticeable safety stock. Unfortunately, this also requires the use of more vehicles to cover all customers. On the contrary, a relatively value of  $s$  produces longer routes with a higher probability of suffering a failure (low reliability), but it requires a lower number of routes to cover all customers.

Regarding the MCS module, the steps followed to assess the stochastic performance (expected travel time) of a given solution are: (i) using random sampling from the assigned probability distributions, different executions of the routing plan is run in order to obtain random observations of the total travel time associated with it; (ii) from these random observations, different statistics can be computed for each routing plan, e.g.: average time, variability of these times, etc.; (iii) using the same simulation outcomes, it is estimated the reliability of each routing plan as the quotient between the number of route failures and the number of simulation runs. These experiments are repeated for different percentages of the safety stock level,  $s$ .

## 4.5 Computational experiments

This section presents a set of extensive computational experiments carried out to test the SIM-BR-MS algorithm for the EVRPST. Firstly, the section introduces the instances that will be used to test our approach. Secondly, the algorithm parameters are discussed. Finally, the computational results are provided –they will be fully analyzed in the next section. The algorithm has been implemented as a Java application (see Appendix D). A standard personal computer with an Intel Core i5 CPU at 3.2 GHz and 4 GB RAM has been employed to perform all the experiments.

### 4.5.1 Benchmark instances

As a benchmark for the test, a set of 27 instances originally proposed by [Uchoa et al. \(2017\)](#) are selected. The original instances already included a maximum distance per route. They have been adapted so they use time-based costs instead of distance-based ones –i.e., Euclidean distances are considered to be travel times and the maximum distance per route is transformed into a maximum time per route. These instances are derived from the ones proposed by [Christofides et al. \(1981\)](#), [Golden et al. \(1998\)](#), and [Li et al. \(2005\)](#). Table 4.2 shows the main characteristics of these instances.

In order to perform numerical experiments under uncertainty conditions, the aforementioned deterministic instances have been extended to consider stochastic travel times as follows: if the original instance shows a deterministic travel time  $t_{ij} = t_{ji} > 0$  when moving from node  $i$  to node  $j$  (with  $i \neq j$ ), then the stochastic travel time  $T_{ij}$  is considered as a random variable following an exponential probability distribution with  $E[T_{ij}] = t_{ij}$  and  $Var[T_{ij}] = t_{ij}^2$ . In a real-life scenario, the specific probability distributions associated with each stochastic travel time would need to be fitted from historical observations, but the SIM-BR-MS algorithm would still be valid. Furthermore, different levels of safety stock –as a percentage of the battery capacity (i.e., vehicle driving range)– have been considered in the experiments:  $s \in \{0\%, 5\%, 10\%, \dots, 35\%\}$ .

### 4.5.2 Parameters setting

One of the advantages of the SIM-BR-MS algorithm is that it does not require a complex fine-tuning process. In fact, after some quick trial-and-error experiments, the following values were set for each parameter:

- The biased-randomized selection during the construction process was generated by using a geometric probability distribution with parameter  $\beta \in (0.23, 0.30)$  –i.e., at each iteration a random value inside the previous interval was assigned to  $\beta$ .

Table 4.2: Characteristics of the benchmark instances.

Instance	$n$	$ V $	$q$	$t_{max}$
Golden_1	240	9	550	650
Golden_2	320	10	700	900
Golden_3	400	10	900	1200
Golden_4	480	10	1000	1600
Golden_5	200	5	900	1800
Golden_6	280	7	900	1500
Golden_7	260	9	900	1300
Golden_8	440	10	900	1200
CMT6	50	6	160	200
CMT7	75	11	140	160
CMT8	100	9	200	230
CMT9	150	14	200	200
CMT10	199	18	200	200
CMT13	120	11	200	720
CMT14	100	11	200	1040
Li_21	560	10	1200	1800
Li_22	600	15	900	1000
Li_23	640	10	1400	2200
Li_24	720	10	1500	2400
Li_25	760	19	900	900
Li_26	800	10	1700	2500
Li_27	840	20	900	900
Li_28	880	10	1800	2800
Li_29	960	10	2000	3000
Li_30	1040	10	2100	3200
Li_31	1120	10	2300	3500
Li_32	1200	11	2500	3600

$n$  = number of customers;  $|V|$  = number of vehicles  
 $q$  = capacity of each vehicle  
 $t_{max}$  = maximum time allowed per route

- The number of simulation runs was set to  $sSim = 400$  for short simulations (on each promising solution) and to  $lSim = 10,000$  for large simulations (on each elite solution).



- For each instance, the algorithm was run 20 times, each time employing a different seed for the pseudo-random number generator.
- For each instance and seed, the algorithm was executed for  $maxTime = 90$  seconds. Notice that this time does not include the time employed in computing the intensive simulations –however, since the number of elite solutions is reduced, this final step takes just a few additional seconds.

### 4.5.3 Computational results

Table 4.3 summarizes the results obtained both using the BRCWS procedure –a deterministic component inside the simheuristics– and the complete BR-MS simheuristic algorithm. Both approaches were run using the same parameters setting as described in Section 4.5.2. Also, in this comparison, no safety stock is considered, i.e.,  $s = 0\%$ .

Hence, column *BDS-Det* shows the cost (in total travel time) associated with the best-found solution obtained for the deterministic version of the problem when it is applied in a deterministic scenario (without uncertainty); column *BDS-Stoch* provides the expected cost of the same solution when it is employed in a stochastic scenario; the reliability column gives an estimate of the probability that the best deterministic solution can be used in a stochastic scenario without suffering any route failure –notice that reliabilities can be low in some cases since no safety stock is considered. Similarly, column *BSS-Stoch* shows the expected cost of the best-found solution for the stochastic version of the problem when applied in a stochastic scenario. Finally, the reliability column provides an estimate of the probability that this solution can be completed as designed –without route failures. As depicted in Figure 4.6, *BDS-Det* and *BDS-Stoch* act as a lower bound and an upper bound, respectively, for *BSS-Stoch*. Thus, in general, it is not a good idea to apply the best-found solution for the deterministic version of the problem to a scenario under uncertainty, since it might often result in a sub-optimal plan. Instead, it is better to use a simulation-optimization approach to generate solutions with a better performance under stochastic conditions (usually by offering a higher reliability level and thus avoiding expensive corrective actions).

For each instance and safety stock level  $s$ , Table 4.4 shows the expected cost (in total travel time) provided by the SIM-BR-MS algorithm in a stochastic scenario. The table also shows the reliability associated with each solution –which tends to increase with the safety stock level–, as well as the gap with respect to the solution obtained without using any safety stock.

One should notice that, in most cases, using a safety stock during the design stage might be a good strategy to reduce the impact of route failures whenever travel times are higher than expected. This concept is further discussed in the next

Table 4.3: Performance of best deterministic and stochastic solutions.

Instance	BRCWS (deterministic component)			MS-BR Simheuristics	
	BDS-Det	BDS-Stoch (a)	Reliability	BSS-Stoch	Reliability
CMT6	546.59	586.75	0.97	586.75	0.97
CMT7	856.26	1060.14	0.86	1040.29	0.88
CMT8	870.60	911.39	0.97	911.14	0.97
CMT9	1118.03	1189.43	0.95	1183.26	0.96
CMT10	1375.31	1439.11	0.95	1431.04	0.96
CMT13	1537.88	1544.24	0.99	1539.03	0.99
CMT14	823.11	823.24	0.99	823.24	0.99
Golden_1	5786.96	9939.65	0.02	9298.79	0.05
Golden_2	8646.93	13376.35	0.01	12754.47	0.03
Golden_3	12828.23	17757.94	0.01	16416.42	0.06
Golden_4	17963.58	23019.70	0.02	21764.50	0.06
Golden_5	7334.24	7679.08	0.78	7602.17	0.83
Golden_6	9829.11	12119.12	0.14	11371.87	0.30
Golden_7	12270.11	15998.37	0.04	15274.38	0.08
Golden_8	13753.22	18831.50	0.01	17869.64	0.03
Li_21	20465.47	24826.35	0.03	23939.78	0.08
Li_22	16612.02	23985.19	0.00	23330.96	0.00
Li_23	23192.07	27986.58	0.02	27176.38	0.07
Li_24	26160.76	30327.41	0.04	30086.13	0.06
Li_25	17618.46	27426.64	0.00	26942.85	0.00
Li_26	28728.31	34534.97	0.01	32076.98	0.09
Li_27	18460.02	28341.25	0.00	28160.91	0.00
Li_28	32654.00	35986.88	0.08	35547.75	0.20
Li_29	35230.52	38188.93	0.10	36485.80	0.87
Li_30	40363.61	44088.03	0.07	42891.96	0.48
Li_31	44248.09	47195.81	0.13	46263.44	0.58
Li_32	45959.99	50720.75	0.04	49407.09	0.15
Average	16490.84	19996.24	0.31	19340.31	0.40

BDS-Det: Best deterministic solution in a deterministic scenario.

BDS-Stoch: Best deterministic solution in a stochastic scenario.

BSS-Stoch: Best stochastic solution in a stochastic scenario.

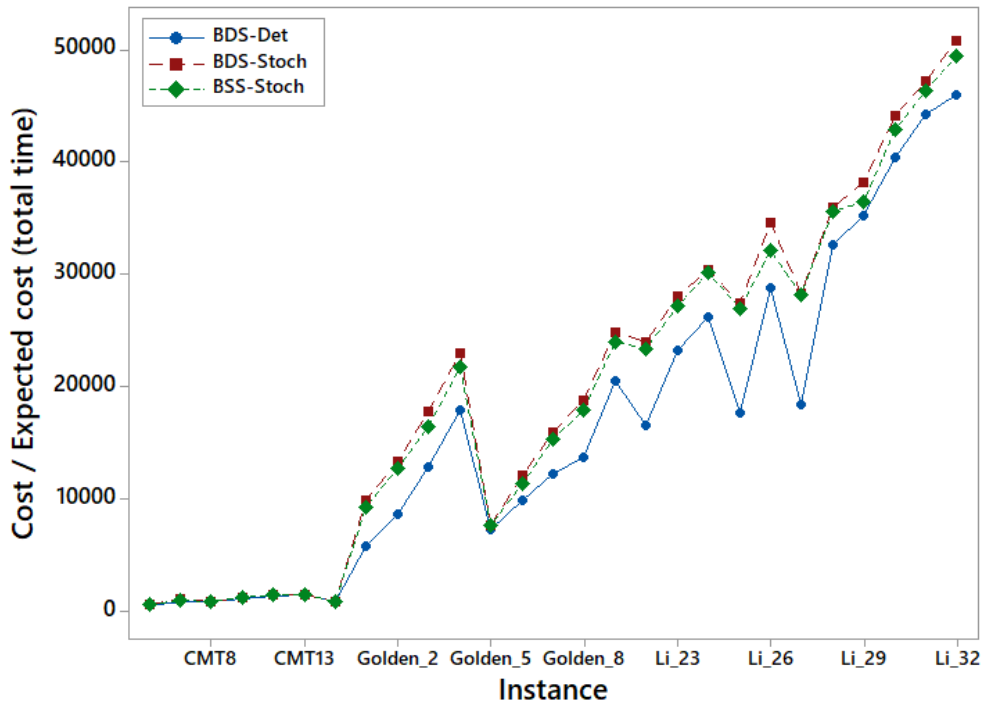


Figure 4.6: Visual comparison among BDS-Det, BDS-Stoch, and BSS-Stoch.

section. Also, note that for a safety stock level of 35% (or higher), there are some instances that cannot be solved during the design stage, i.e., assuming such a high safety stock level, some customers in instances *Li\_25* and *Li\_27* cannot be reached from the depot in the reduced ‘standard’ time of the batteries (that is, without considering the extra time that can be provided by the energy safety stock). That justifies that this chapter focus on safety stock levels between 0% and 35% of the original battery capacity.

## 4.6 Analysis of results

For each considered safety stock level,  $s \in \{0\%, 5\%, 10\%, \dots, 35\%\}$ , Figure 4.7 uses boxplots to illustrate the distribution of the reliability indices associated with the best-found stochastic solutions for each instance.

Notice that the higher the safety stock level, the higher the average reliability index is. Moreover, increasing the safety stock level also contributes to reduce the variability in these reliability indices, i.e., increasing the safety stock has the expected effect of reducing the number of route failures, which in turn reduces the extra costs generated by corrective actions. Of course, increasing the safety stock

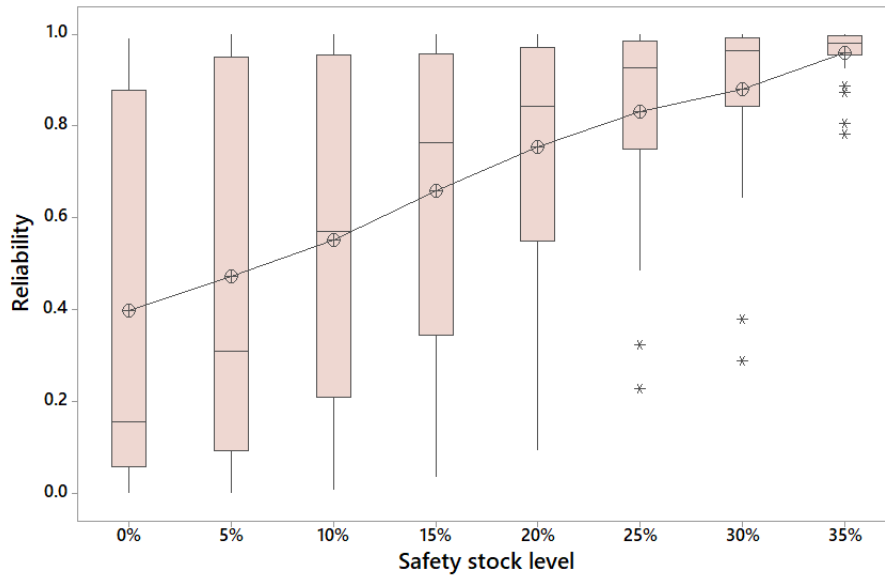


Figure 4.7: Reliability values for different safety stock levels.

level makes the solution more ‘robust’ against uncertainty (thus reducing the cost due to corrective actions), but it also requires the use of additional routes in the solution, which raises the cost (total time employed) of the final distribution plan. Therefore, this trade-off must be taken into account when finding the right level of safety stock for each individual instance.

Finally, Figure 4.8 shows the expected travel times, across all instances, for each safety stock level. The most relevant observation here, is that the expected cost (total travel time) can be reduced, on average, by using safety stock levels between 20% and 25% of the original capacity. Of course, the specific safety stock level to use will depend upon the actual instance as well as on the probability distribution employed to model the travel times. Still, the point here is that the use of safety stocks can contribute to reduce the total expected cost of the distribution plan by making this plan less sensitive to the risk of route failures.

## 4.7 Contributions

This chapter analyzes the EVRPST problem considering also driving-range limitations, which might cause route failures when the vehicle runs out of battery. The SIM-BR-MS algorithm combines Monte Carlo simulation with a multi-start framework, which also integrates a biased-randomized constructive heuristics. Also, the SIM-BR-MS makes use of safety stocks during the routing design stage, thus de-

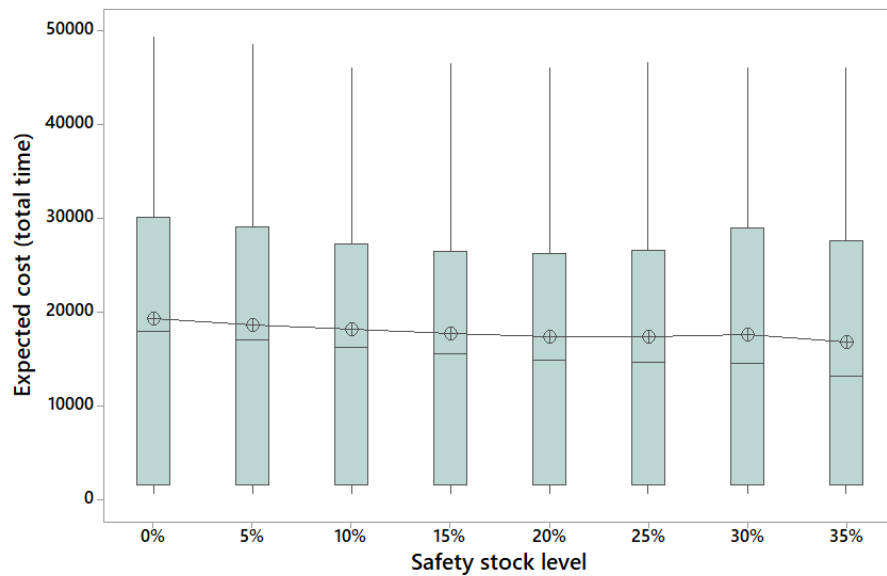


Figure 4.8: Expected travel times for different safety stock levels

creasing the risk of suffering route failures. In other words, this chapter focus on constructing reliable solutions with a low risk of requesting corrective actions. The results prove that using deterministic solutions in stochastic scenarios might lead to sub-optimal distribution plans that can be easily improved by using a simulation-optimization technique such as the one proposed here. They also illustrate how the use of the suitable energy safety stock levels during the routing design stage can increase the reliability of the distribution plans, thus reducing the total expected costs.

---

**Algorithm 10** The SIM-BR-MS algorithm for the EVRPST.

---

```

1: procedure SIMHEURISTIC solve(test, nodes, edges)
    ▷ test: maxTime,  $\beta$ , sSim, lSim, s
    ▷ nodes: coordinates, demand
    ▷ edges: travel time

2:   savings  $\leftarrow$  computeSavings (nodes, edges)
3:   initSol  $\leftarrow$  savingsHeuristic (nodes, savings)      ▷ Clarke and Wright (1964)
4:   initSol  $\leftarrow$  localSearch (initSol)                  ▷ 2-Opt
5:   stochCost(initSol)  $\leftarrow$  simulation (initSol, sSim)
6:   baseSol  $\leftarrow$  initSol
7:   bestStochSolList  $\leftarrow$  add (initSol)                  ▷ elite solutions
8:   while (elapsedTime < maxTime) do
9:     newSol  $\leftarrow$  BRCWS (nodes, savings,  $\beta$ , s)    ▷ Juan et al. (2011b)
10:    newSol  $\leftarrow$  localSearch (newSol)                 ▷ 2-Opt
11:    if (detCost(newSol) < detCost(baseSol)) then
12:      stochCost(newSol)  $\leftarrow$  simulation (newSol, sSim)
13:      if (stochCost(newSol) < detCost(baseSol)) then
14:        baseSol  $\leftarrow$  newSol
15:      end if
16:      update (bestStochSolList)
17:    end if
18:  end while
19:  for (each sol in bestStochSolList) do
20:    stochCost(sol)  $\leftarrow$  simulation (sol, lSim)
21:  end for
22:  return bestSol in bestStochSolList
23: end procedure

```

---

Table 4.4: Solution performance considering different safety stock levels.

Instance	BDS-Stoch			BSS-Stoch with $s = 5\%$			BSS-Stoch with $s = 15\%$			BSS-Stoch with $s = 25\%$			BSS-Stoch with $s = 35\%$		
	E[Cost] (a)	E[Cost] (c)	Reliab.	E[Cost] (a)	Gap (c)-(a)	Reliab.	E[Cost] (c)	Reliab.	Gap (e)-(a)	E[Cost] (g)	Reliab.	Gap (g)-(a)	E[Cost] (i)	Reliab.	Gap (i)-(a)
CMT6	586.75	584.80	0.97	-0.33%	0.97	0.03%	586.94	0.97	0.03%	580.22	0.97	-1.11%	577.34	0.98	-1.60%
CMT7	1060.14	1044.42	0.87	-1.48%	0.87	-1.88%	1040.23	0.87	-1.88%	1041.44	0.88	-1.76%	1021.55	0.89	-3.64%
CMT8	911.39	910.92	0.97	-0.05%	0.97	-0.23%	909.31	0.97	-0.23%	895.55	0.99	-1.74%	890.08	0.98	-2.34%
CMT9	1189.43	1185.18	0.96	-0.36%	0.96	-0.78%	1180.13	0.96	-0.78%	1165.24	0.96	-2.03%	1150.64	0.97	-3.26%
CMT10	1439.11	1433.89	0.96	-0.36%	0.96	-0.69%	1429.24	0.96	-0.69%	1440.93	0.95	0.13%	1423.00	0.98	-1.12%
CMT13	1544.24	1554.70	1.00	0.68%	1.00	0.54%	1552.56	1.00	0.54%	1554.78	1.00	0.68%	1555.34	1.00	0.72%
CMT14	823.24	820.79	1.00	-0.30%	1.00	-0.40%	819.92	1.00	-0.40%	822.17	1.00	-0.13%	820.66	1.00	-0.31%
Golden_1	9939.65	8814.54	0.09	-11.32%	0.27	-20.41%	7911.15	0.27	-20.41%	7698.07	0.56	-22.55%	8094.93	0.81	-18.56%
Golden_2	13376.35	12159.70	0.07	-9.10%	0.32	-20.24%	10669.63	0.32	-20.24%	10373.62	0.59	-22.45%	10883.65	0.87	-18.64%
Golden_3	17757.94	16135.89	0.09	-9.13%	0.34	-17.95%	14569.99	0.34	-17.95%	13520.73	0.75	-23.86%	13127.24	0.95	-26.08%
Golden_4	23019.70	20693.91	0.15	-10.10%	0.48	-16.64%	19188.46	0.48	-16.64%	18460.61	0.82	-19.81%	18498.44	0.97	-19.64%
Golden_5	7679.08	7584.29	0.95	-1.23%	0.95	-1.22%	7585.41	0.95	-1.22%	7335.11	0.97	-4.48%	7231.07	0.99	-5.83%
Golden_6	12119.12	10963.40	0.45	-9.54%	0.76	-15.78%	10206.53	0.76	-15.78%	9812.63	0.91	-19.03%	9657.49	0.97	-20.31%
Golden_7	15998.37	14250.60	0.22	-10.92%	0.44	-16.66%	13333.32	0.44	-16.66%	12568.18	0.75	-21.44%	12115.77	0.96	-24.27%
Golden_8	18831.50	16971.04	0.08	-9.88%	0.32	-17.69%	15499.76	0.32	-17.69%	14579.05	0.69	-22.58%	14314.21	0.93	-23.99%
Li_21	24826.35	22865.97	0.19	-7.90%	0.47	-12.44%	21736.92	0.47	-12.44%	20683.79	0.85	-16.69%	20724.91	0.97	-16.52%
Li_22	23985.19	21542.09	0.02	-10.19%	0.11	-13.39%	20772.52	0.11	-13.39%	19038.31	0.49	-20.62%	20424.35	0.78	-14.85%
Li_23	27986.58	25514.42	0.24	-8.83%	0.59	-13.86%	24106.71	0.59	-13.86%	23381.52	0.90	-16.45%	23126.42	0.99	-17.37%
Li_24	30327.41	29102.20	0.11	-4.04%	0.87	-12.82%	26439.78	0.87	-12.82%	26200.40	0.93	-13.61%	26121.15	1.00	-13.87%
Li_25	27426.64	26020.12	0.00	-5.13%	0.06	-12.19%	24084.62	0.06	-12.19%	24501.08	0.32	-10.67%	N/A	N/A	N/A
Li_26	34534.97	29439.93	0.75	-14.75%	0.80	-15.11%	29316.25	0.80	-15.11%	28893.00	0.96	-16.34%	29010.87	0.99	-16.00%
Li_27	28341.25	27203.26	0.00	-4.02%	0.04	-9.12%	25756.63	0.04	-9.12%	26592.44	0.23	-6.17%	N/A	N/A	N/A
Li_28	33986.88	33494.04	0.76	-6.93%	0.76	-8.35%	32982.25	0.76	-8.35%	31505.10	0.98	-12.45%	31559.27	1.00	-12.30%
Li_29	38188.93	36379.32	0.92	-4.74%	0.71	-6.81%	35586.57	0.71	-6.81%	33713.97	0.99	-11.72%	34091.61	1.00	-10.73%
Li_30	44088.03	42149.09	0.31	-4.40%	0.90	-7.66%	40709.81	0.90	-7.66%	40839.56	1.00	-7.37%	40826.65	1.00	-7.40%
Li_31	47195.81	45952.96	0.45	-2.63%	0.88	-6.32%	44212.40	0.88	-6.32%	44151.37	0.99	-6.45%	44985.68	1.00	-4.68%
Li_32	50720.75	48539.65	0.16	-4.30%	1.00	-8.17%	46575.65	1.00	-8.17%	46595.38	0.99	-8.13%	46018.94	1.00	-9.27%
Averages			0.47	-5.60%	0.66	-9.49%		0.66	-9.49%		0.83	-11.44%		0.96	-11.67%





## The stochastic electric vehicle routing problem with sustainability indicators

*This chapter focuses on the urban distribution process considering the driver conditions and limitations the EVs. This section analyzes the sustainable capacitated electric routing problem with stochastic demands, travel times and length constraints for estimating the effect of uncertain demand and travel time on the solution performance. Here, a preventive and corrective policy are implemented for mitigating the impact of route failures. A simheuristic algorithm is proposed which integrates Monte Carlo simulation into an iterated local search metaheuristics. The SIM-BR-ILS algorithm includes preventive and corrective policies to deal with the stochasticity of the problem.*

The work presented in this chapter has been submitted in the journal of International Transactions in Operational Research:

- Calvet, L.; Reyes-Rubiano, L.; Abdullahi, H.; Ouelhadj, D. A.; Faulin, J. (2019). A hybrid approach for sustainable freight transport considering stochastic travel times and demand. International Transactions in Operational Research. Under review.

Part of the contents of this chapter has been presented at the following conferences:

- Reyes-Rubiano, L., Calvet, L., Juan, A., Faulin. J. (2018). Optimizing Routes with Sustainability Criteria in Transport Management. In EURO 2018, Valencia, Spain.
- Faulin. J., Reyes-Rubiano, L., Calvet, L., Juan, A. (2018). The Sustainability

Dimension as Optimization Criteria: A Capacitated Vehicle Routing Problem. In EURO/ ALIO 2018, Bologna, Italy.

## 5.1 A BR-ILS simheuristic algorithm for solving the stochastic electric vehicle routing problem with sustainability indicators

This chapter focuses on the sustainable capacitated electric vehicle routing problem with stochastic demands, travel times and length constraints (SCEVRP-SDT-LC). Fundamentally, length constraints are introduced to ensure an appropriate balance in routes duration, which in practical terms concerns driving and working hours. A simheuristic approach that integrates MCS into an Iterated Local Search (ILS) metaheuristic framework is proposed (SIM-BR-ILS). To the best of our knowledge, the SCEVRP-SDT-LC has not yet been addressed in the literature. The main contributions of this chapter are: *(i)* a set of realistic instances created from CVRP benchmarks; *(ii)* a formal description of the sustainable capacitated electric vehicle routing problem with length constraints (SCEVRP-LC) through an optimization model; *(iii)* a simheuristic algorithm to solve the SCEVRP-SDT-LC, which includes corrective and preventive policies defined to handle the stochasticity; and *(iv)* a comprehensive set of computational experiments to analyze the trade-off between the dimensions of sustainability and the effects of the stochasticity on the solutions performance. These contributions give insights into current logistics challenges and powerful optimization solving approaches.

The rest of the chapter is structured as follows. Section 5.2 reviews research works on sustainable transport, considering both deterministic and stochastic approaches. Section 5.3 presents a mathematical model for the deterministic version of the problem addressed. Afterwards, the proposed simheuristics is explained in section 5.4. While the computational experiments are described in section 5.5. Finally, section 5.7 draws the main conclusions and identifies potential lines of future work.

## 5.2 Literature review

The CVRP consists of a set of delivery vehicles with limited capacity, which are available to serve customers with known demands. Typically, the aim is to minimize the total travel distance. The travel distances between nodes (customers and depot) are known. Some additional decision criteria and constraints may be introduced, which leads to new variants called Rich VRPs. These variants include attributes

such as constraints related to route length, route duration, and workload, which describe realistic settings. This section reviews works on two Rich VRPs related to the problem addressed in this chapter: the sustainable routing problems and the stochastic routing problems.

In general, the environmental and economic impacts of the routing activities have been much more studied than the social ones. For instance, [Erdogan and Miller-Hooks \(2012\)](#) describe the green VRP, which is characterized by vehicles using alternative fuels, and propose tools to make decisions regarding routes duration and refueling distance. Similarly, the PRP ([Kramer et al., 2015](#)) focuses on measuring the  $CO_2$  emissions. The reader interested in works on the green VRP and the PRP is referred to [Bektaş and Laporte \(2011\)](#); [Dabia et al. \(2014\)](#); [Tajik et al. \(2014\)](#); [Demir et al. \(2015\)](#); [McKinnon et al. \(2015\)](#); [Soysal et al. \(2015\)](#); [Zhang et al. \(2015\)](#).

The indicators of the social dimension relate to intangible effects, which usually are hard to measure ([Navarro et al., 2016](#); [Demir et al., 2015](#); [McKinnon et al., 2015](#)). These indicators may be measured from a customer or employee perspective ([Delucchi and McCubbin, 2010](#)). Road safety constitutes one of the most critical indicators and is related to infrastructure, driver fatigue (workloads) distractions, and high speed. According to [Wang et al. \(2016\)](#), speed variations are directly related to the accident risk of both pedestrians and vehicles. In addition, having multiple traffic signs may encourage drivers to carry out dangerous maneuvers to avoid them, which deteriorates the road safety ([Xie et al., 2013](#)). Recently, a set of social rules have been established through regulations in Europe concerning driving and working hours and rest times in order to tackle the driver fatigue and improve the working conditions ([European Transport Safety Council, 2011, 2017](#)).

[Matl et al. \(2017\)](#) review equity functions (mainly referring to allocating workloads and balancing the utilization of resources) for bi-objective VRP models. [Bashiri et al. \(2016\)](#) present two mixed integer programming models to tackle the economic and social aspects related to workload balance and its influence on the accident risk. [Wang et al. \(2011\)](#) study the influence of environmental criteria on the total cost and demonstrate that additional criteria imply additional costs and require an accurate operation synchronization in the supply chain.

## 5.3 Problem description

This section presents a formal description of the deterministic and stochastic version of the problem tackled.

### 5.3.1 Deterministic version

The CVRP considering EVs is a Rich VRP characterized by restrictions related to electric vehicles and may be described as follows. Let  $G = (N, A)$  be a complete undirected graph where  $N = 0 \cup N_c$  is a set of nodes; 0 corresponds to the depot and  $N_c = \{1, 2, \dots, n\}$  to the subset of customers.  $A = \{(i, j) : i, j \in N, i \neq j\}$  is the set of arcs that connect all nodes in  $N$ . Each customer  $i$  has a known positive demand  $q_i$ , while the demand of the depot is zero. There is a fleet  $K = \{1, 2, \dots, \kappa\}$  of identical vehicles with a capacity of  $Q$ . Each arc  $(i, j)$  is characterized by a travel distance  $(d_{ij})$  and time  $(t_{ij})$ . Each route must start and end at the depot, and all customers' demands must be satisfied. Often, the goal is to minimize the travel distance and/or travel time. Recently, some authors have started to propose objective functions including environmental indicators, such as:  $CO_2$  emissions, noise pollution, use of electric vehicles, and ecological deterioration. The solution of the problem (*i.e.*, the set of routes) is represented by the binary variable  $x_{ijk}$ , which is equal to one if the arc  $(i, j)$  is traversed by the vehicle  $k$ , and zero otherwise. An upper bound is set to the travel time per route, which represents the battery life ( $B$ ).

We describe below the mathematical model, which includes sustainable indicators in the objective function and constraints, after listing the notation required.

#### Sets

$N$	Set of nodes
$A$	Set of arcs connecting all nodes
$N_c$	Set of customers nodes
$K$	Set of electric vehicles
$i$	Index of origin nodes
$j$	Index of destination nodes
$k$	Index of vehicles
$s$	Index of sustainability dimensions

Equation 5.1 represents the *economic dimension*. It depends on the driver wages ( $DW$ ), the fixed costs of the vehicles ( $FC$ ), the energy cost per kilowatt hour ( $C_f$ ), the fuel consumption rate ( $kpl$ ), and the travel distance.

$$z_1 := \sum_{(i,j) \in A} \sum_{k \in K} DW \cdot t_{ij} \cdot x_{ijk} + \sum_{j \in N_c} \sum_{k \in K} FC \cdot x_{0jk} + \sum_{(i,j) \in A} \sum_{k \in K} C_f \cdot kpl \cdot d_{ij} \cdot x_{ijk} \quad (5.1)$$

Equation 5.2 refers to the *environmental dimension*. The cost associated considers the energy price per ton of  $CO_2$  released ( $C_e$ ), the energy consumption rate ( $\gamma$ ), and the travel distance. The function is a simplified version of one proposed by Kuo (2010) and Zhang et al. (2015).

**Parameters**


---

$q_i$	Demand of node $i$
$Q$	Vehicle capacity
$B$	Battery life (travel time)
$DW$	Driver wages
$FC$	Vehicle fixed costs
$\gamma$	Energy consumption rate
$C_f$	Price of energy per kWh consumed
$C_e$	Price of energy per ton of $CO_2$ released
$kpl$	Fuel consumption rate
$a$	Cost of risk for a heavy vehicle
$\alpha_s$	Weight for indicator $s$

---

**Variables**


---

$x_{ijk}$	Binary variable. 1 if arc $(i, j)$ is traversed by vehicle $k$ , 0 otherwise
$f_{jk}$	Continuous variable. Remaining tank fuel of vehicle $k$ when arrives at node $j$
$z_s$	Continuous variable. Value of indicator $s$
$U_{jk}$	Auxiliary variable to eliminate sub-tours

---

$$z_2 := \sum_{(i,j) \in A} \sum_{k \in K} C_e \cdot \gamma \cdot d_{ij} \cdot x_{ijk} \quad (5.2)$$

Finally, the *social dimension* is described by Equation 5.3, which employs the accident risk cost ( $a$ ), and monetizes the risk associated with a heavy vehicle traversing the arc  $(i, j)$ . This value varies according to the distance of the arc and the load of the vehicle  $k$  after visiting customer  $i$  ( $y_{ijk}$ ).

$$z_3 := \sum_{(i,j) \in A} \sum_{k \in K} a \cdot d_{ij} \cdot y_{ijk} \quad (5.3)$$

The proposed weighted objective function (Equation 5.4) is defined as a holistic approach combining the costs related to the different dimensions.  $\alpha_s$  (where  $0 \leq \alpha_s \leq 1$  and  $\sum_{s=1}^3 \alpha_s = 1$ ) constitutes the weight or relative importance for indicator  $s$ .

$$\text{Min. } \alpha_1 \cdot z_1 + \alpha_2 \cdot z_2 + \alpha_3 \cdot z_3 \quad (5.4)$$

The constraints, which are based on [Erdogan and Miller-Hooks \(2012\)](#), are de-

scribed below. Equations 5.5 and 5.6 ensure that each customer is visited exactly once. The flow conservation is introduced by Equation 5.7. Equation 5.8 guarantees that the total demand served by a vehicle does not exceed its capacity. Equation 5.9 imposes that each vehicle with an assigned route can leave the depot at most once.

Equation 5.10 states that the load in the vehicle arriving at customer  $j$  minus his demand equals the load after the visit. Equations 5.11 and 5.12 set an upper and lower bound for the load of an arc when it is traversed. Equation 5.13 sets an upper bound ( $B$ ) for the total travel time of a route. Equation 5.14 avoids sub-tours, where  $U_{jk}$  is an auxiliary variable and  $|N_c|$  is the number of customers. Finally, Equations 5.15 to 5.18 define variable domains.

$$\sum_{j \in N} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in N_c \quad (5.5)$$

$$\sum_{i \in N} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in N_c \quad (5.6)$$

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{jik} \quad \forall i \in N_c, k \in K \quad (5.7)$$

$$y_{ijk} \leq Q \quad \forall (i, j) \in A, k \in K \quad (5.8)$$

$$\sum_{j \in N_c} x_{0jk} \leq 1 \quad \forall k \in K \quad (5.9)$$

$$\sum_{i \in N} y_{jik} = \sum_{i \in N} y_{ijk} - \sum_{i \in N} q_j \cdot x_{jik} \quad \forall j \in N_c, k \in K \quad (5.10)$$

$$y_{ijk} \leq (Q - q_i) \cdot x_{ijk} \quad \forall (i, j) \in A, k \in K \quad (5.11)$$

$$y_{ijk} \geq q_j \cdot x_{ijk} \quad \forall (i, j) \in A, k \in K \quad (5.12)$$

$$\sum_{(i,j) \in A} t_{ij} \cdot x_{ijk} \leq B \quad \forall k \in K \quad (5.13)$$

$$U_{ik} - U_{jk} + |N_c| \cdot x_{ijk} \leq |N_c| - 1 \quad \forall i, j \in N_c, k \in K \quad (5.14)$$

$$x_{ijk} \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (5.15)$$

$$y_{ijk} \geq 0 \quad \forall (i, j) \in A, k \in K \quad (5.16)$$

$$f_{ik} \geq 0 \quad \forall i \in N, k \in K \quad (5.17)$$

$$U_{ik} \geq 0 \quad \forall i \in N_c, k \in K \quad (5.18)$$

### 5.3.2 Stochastic version

The stochastic version of the problem introduces stochastic demands and travel times, which are modeled as random variables following specific probability distri-

butions (either theoretical or empirical ones). Figure 5.1 illustrates a solution with two routes, where each customer has an expected demand,  $E[q_i]$ , and each arc has an expected travel time,  $E[t_{ij}]$ . Each time a driver visits a new customer both the load of the vehicle and the remaining time are reduced. The size of the circles is proportional to the demands. Due to the definitions of the environmental and social costs, it is expected that the first customers to be visited will be those with heavier demands.

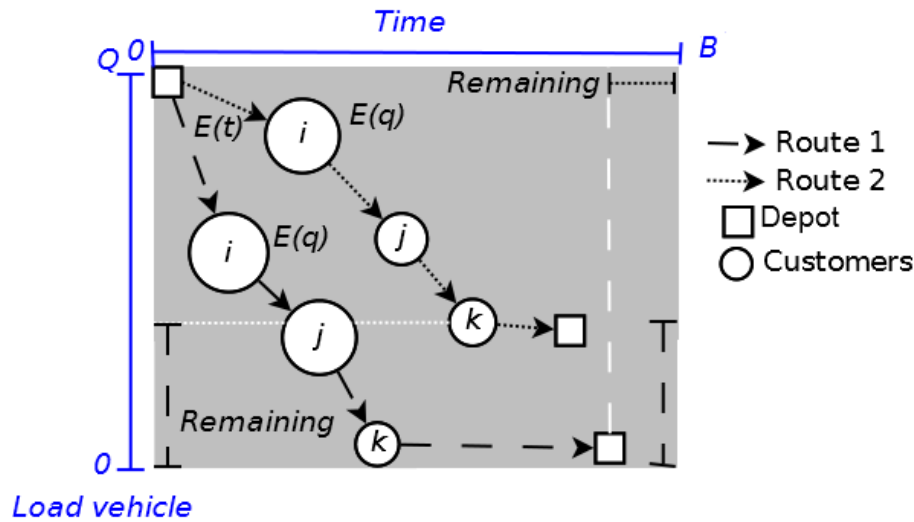


Figure 5.1: Representation of a solution for the SCEVRP-SDT-LC.

While deterministic conditions vehicles capacity can performances routes satisfying the constraints related to the capacity of the vehicles and the maximum time of the batteries, it is not possible to guarantee that they are not violated in a stochastic environment. However, we may associate a probability to each constraint ( $p_c$  and  $p_b$ , respectively). In other words, the decision-maker may require a solution with probabilities of satisfying each restriction non-lower than  $p_c$  and  $p_b$ , respectively (*i.e.*, these values constitute lower bounds). The current probabilities for each solution are named reliabilities and are denoted by  $r_c$  and  $r_b$ , respectively.

When a restriction is violated, it is called a route failure, which has associated a penalty. There are two types of route failures: (*i*) the remaining load of a vehicle is not sufficient to satisfy the demand of the customer being visited; and (*ii*) a vehicle runs out of energy. Two types of policies regarding route failures may be applied: preventive and corrective. While the preventive policies aim to reduce the risk of route failure, the corrective policies are applied after a route fails. In most cases, the corrective policies are more expensive than the preventive ones in terms of travel distance and time and, as a consequence, in terms of environmental and social costs. Both types of policies are interdependent. The specific preventive and

corrective policies applied for the different types of failures are described in the next section.

## 5.4 The SIM-BR-ILS solving approach

The proposed approach to tackle the SCEVRP-SDT-LC consists of a simheuristic algorithm based on the ILS metaheuristics (Lourenço et al., 2010). As mentioned previously, a simheuristics algorithm assumes that there is a relevant correlation between the best stochastic solutions (BSSs) and the best deterministic solutions (BDSs), which are those minimizing the lowest expected cost in the stochastic environment and the cost in the deterministic environment, respectively. In other words, most of the top BSSs and the top BDSs may be the same, but the ranks are likely to differ. The ILS metaheuristics is chosen because it constitutes a simple yet powerful single solution metaheuristics, which basically combines a perturbation procedure and a local search. Moreover, it is parameter-free, which makes it easier to replicate our experiments.

### 5.4.1 General framework

The main procedure (Algorithm 11) starts building a solution (*baseSol*) based on the BRCWS algorithm. Then, it is further improved by means of a local search. The next step is to call the MCS procedure to assess the solution. This procedure determines if the solution is feasible, *i.e.*, if the reliabilities ( $r_c, r_b$ ) are equal to or greater than the probabilities set by the decision-maker ( $p_c, p_b$ ). Provided the solution is feasible, the procedure continues. Otherwise, new solutions are created, improved and assessed until a feasible one is found (lines 2-8).

Afterwards, the initial solution is stored as the base solution (*baseSol*) and the best solution found so far (*bestSol*) and is added to a list of BSSs (*bestStochSolList*) (lines 9-10). Then, a loop with a stopping criterion based on the elapsed time is started (lines 11-29). Inside, the current base solution is perturbed and a local search is applied. The next step is to compare the resulting new solution (*newSol*) against *baseSol*. If *newSol* has an equal to or greater cost (*i.e.*, considering the deterministic instance), then that solution is discarded and a new iteration of the loop starts. Otherwise, the MCS procedure is applied to *newSol*. Provided this procedure determines a feasible solution in the stochastic environment, the relative percentage difference (*rpdc*, line 17) between the expected costs of *newSol* and *baseSol* is computed. If it is negative (*i.e.*, *newSol* is better), *newSol* replaces *baseSol* and is introduced in *bestStochSolList*. Then, the worst solution in the list is deleted if its size exceeds a given maximum value ( $L$ ). Oppositely, if *rpdc* is positive, *newSol* may replace *baseSol* with a probability of  $e^{-rpdc}$  (lines 22-25). This acceptance criterion



was first proposed by [Hatami et al. \(2015\)](#) and is intended to avoid getting trapped in a local optimal.

Finally, the MCS procedure is applied again to each solution in *bestStochSolList* but with a higher number of scenarios. By doing this, we obtain estimates of performance measures with a higher level of accuracy (lines 30-32).

### 5.4.2 Proposed policies

Three combinations of policies are discussed and compared:

- No policies. This represents a case where stochasticity is ignored and no actions are planned to try to avoid or reduce the risk of route failures.
- Only corrective policies. These policies make it possible to resume the route after a route failure. There is a policy for each kind of route failure:
  - The remaining load of a vehicle is not enough to satisfy the demand of the customer being visited. In this case, the vehicle goes back to the depot for reloading. The driver delivers the remaining goods before going back to the depot and completes the deliver after reloading. This policy has impacts in terms of travel time and distance.
  - A vehicle runs out of energy. Another vehicle is sent to supply a battery to the first one, which then resumes the route. This policy has impacts in terms of travel time. This time is computed as twice the mean of the expected travel time for going from the depot to the origin of the edge and the expected travel time for going from the end of the edge to the depot. If the end of the edge is the depot, no penalization is added (in other words, the additional time is considered negligible).
- Corrective and preventive policies. Preventive policies are introduced to avoid or reduce the number of route failures and the corresponding costs of the corrective policies, which tend to be relatively high. There are two preventive policies.
  - When a vehicle goes back to the depot to reload because the demand of a customer exceeded its remaining capacity, we consider the battery level and the expected travel times. If the vehicle is expected to be able to serve the remaining customers, then the vehicle is only reloaded (first corrective policy). Otherwise, in addition to reload the vehicle, the battery is replaced (preventive policy).
  - The remaining load and the state of the battery are checked after each visit to assess if it is feasible to reach the next customer and satisfy

its demand (considering the expected demand and the expected travel time). If needed, the driver returns to the depot to reload and replace the battery, or another vehicle supply a new battery if no reloading is arranged.

Notice that the corrective and preventive policies may imply a higher travel distance and time and, as a consequence, a higher cost. We do not consider other penalties. It is also important to highlight that the use of preventive policies do not ensure that a route failure never takes place.

In the next section, the chapter presents a comparison between the reliabilities and the monetized impacts of the solution returned for each combination of policies. The three reliabilities computed for each solution may be interpreted as the probabilities of suffering at least one route failure due to the limited capacity of the vehicles, the duration of the battery or any of these factors, respectively. The impacts associated to the first option (no policies) cannot be compared with those of the other options. The reason is that when a route failure happens in a given scenario and there are no policies to be applied, the route can not be finished and some customer demands may remain unsatisfied. Thus, the impacts may be lower if the solution becomes unfeasible.

In addition to these policies, we also introduce three different mechanisms to obtain more reliable solutions. The first two are considered while searching for a feasible initial solution (Algorithm 11, lines 3-8). In particular, if the solution built by the BRCWS heuristics is not feasible because the reliability related to the capacity is below the required probability ( $r_c < p_c$ ), then we reduce the capacity of the vehicles for the design of routes in 2%. By doing this, the heuristics will provide a solution with smaller routes and a higher reliability. Similarly, if the solution built by the heuristics is not feasible because the reliability related to the battery is below the required probability ( $r_b < p_b$ ), we reduce the limit of the battery of the vehicles for the design of routes in 2%. These decreases are applied until a feasible solution is found. The third mechanism refers to the loading of the vehicles when they leave the depot. The higher the load, the higher the social cost. For that reason, in some cases it may be cheaper not using all the capacity of the vehicle. However, if we only load the expected demand, the probability of having a route failure may be high. During the execution of the main loop in the simulation stage we load the double of the expected demand (or the capacity, if this is lower), but when assessing the top best solutions we keep reducing this amount by 2% while the expected total cost reduces and the solution remains feasible.

## 5.5 Computational experiments

The proposed solving approach has been implemented as a Java application (see Appendix E). A standard personal computer, Intel QuadCore i5 CPU at 3.2 GHz and 4 GB RAM with Windows 7, has been used to execute all tests.

### 5.5.1 Benchmark instances

In order to validate and assess the performance of our algorithm for a deterministic environment, the 40 instances have been solved minimizing only the distance. The solutions provided by our approach (SIM-BR-ILS) are compared against the BKS from literature. The maximum computing time has been set to 120 seconds, which is a reasonable time. The results are gathered in Table 5.3. In 26 out of 40 instances our approach achieves a gap lower than 1% and the average gap is 0.94%. This means that our solutions are slightly worst than the BKS. Our approach has an average computing time of 36.47 seconds.

### 5.5.2 Parameters setting

Table 5.1 presents the algorithm parameters. The number of seeds has been set to 10 and only the best results are stored, which is usual in the literature. The number of scenarios simulated and the size of the list of best solutions have been set taking into account the computing time (defined below). Obviously, if we allocate more time, then these values may be increased to obtain better solutions and more accurate estimates. The distributions of probability of  $\beta$  and  $p$  are set after a few quick experiments. By defining a distribution for these parameters instead of a constant value we aim to explore a wider search area. The remaining of this section describes the instances, the validation process, and some numerical results. A comprehensive analysis of the results is provided in the next section.

Table 5.1: Parameters of the algorithm.

Parameter	Value
Number of seeds	10
$nSim_s, nSim_l$	500, 5000
$L$	5
$\beta$	U(0.7,0.8)
$p$	U(0,100)

Additionally, there are no benchmark instances for the SCEVRP-SDT-LC. For this reason, we have generated new instances based on CVRP instances from the

CVRPLIB library (<http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>). In particular, 40 instances proposed by Augerat et al. (1998) were selected. The number of nodes ranges from 31 to 80. First, we set deterministic travel times by simulating speeds. As suggested by Zhang et al. (2015), three speed levels are considered: high, moderate and low. These speeds represent the (high) transport speed on a freeway, the (low) urban transport speed, and an intermediate speed, respectively. A uniform probability distribution is used to describe each level. The following parameters are set: (90,110), (50,70), and (25,45) km/h, respectively. In order to set the speed of an arc, we assume the following proportions of vehicles driving at a high, moderate, and low speed: 20%, 20%, and 60%, respectively. Table 5.2 gathers the remaining parameters for the deterministic model, including units and references.

Table 5.2: Parameters to quantify and monetize impacts.

Input Value	Converted to	Reference
$DW$ 0.0022 £/s	19.36 €/h	
$FC$ 59.90 £/day	67.62 €/day	Koc et al. (2014)
$C_f$ 0.271 USD/kWh	0.23 €/kWh	Global Energy Prices (2018)
$kpl$ 12.9 kWh/100 km	0.129 kWh/km	Muñoz-Villamizar et al. (2017)
$C_e$ 22 USD/ton of $CO_2$	0.02 €/kg of $CO_2$	World Bank (2015)
$\gamma$ 0.024 kg of $CO_2$ /km		Nealer et al. (2015)
$a$ [0.1-2] USD/ton-mile	0.0005 €/kg- km	Delucchi and McCubbin (2010)
$B$ 6 h		

Without loss of generality, we assume that stochastic demands and travel times follow log-normal probability distributions. Note that any other distribution could be applied, but this distribution is a suitable option to model non-negative random variables. In particular, the demand of customer  $i$  follows a log-normal probability with an expected value of  $q_i$  and a variance of  $c \cdot q_i$ . The location  $\mu$  and scale parameters  $\sigma$  may be computed as shown in Equations 5.19 and 5.20. The same description holds for travel times. Two values of  $c$ , 0.1 and 1, are tested representing a low and high level of stochasticity, respectively. For example, if  $q_i = 20$ , the smallest interval containing 90% of the generated values would be approximately (17.75, 22.41) for  $c = 0.1$  and (13.57, 28.07) for  $c = 1$ . The probabilities  $p_c$  and  $p_b$  are set to 0.6.

$$\mu = \ln(E[q_i]) - \frac{1}{2} \cdot \ln \left( 1 + \frac{\text{Var}[q_i]}{E[q_i]^2} \right) \quad (5.19)$$

$$\sigma = \left| \sqrt{\ln \left( 1 + \frac{\text{Var}[q_i]}{E[q_i]^2} \right)} \right| \quad (5.20)$$

### 5.5.3 Computational results

Before introducing stochasticity, it is interesting to study the effect of the weights on the solution's performance. In particular, we would like to analyze the correlation among dimensions. Being able to set different weights allows the decision-maker to compare different 'high-quality' solutions. 4 combinations of weights are tested:

- $\alpha_s = 1/3$  (*i.e.*, a balanced solution  $-\alpha_1 = 1/3, \alpha_2 = 1/3, \alpha_3 = 1/3$ )  $\alpha_1 = 1$  (*i.e.*, a solution minimizing the economic impact  $-\alpha_1 = 1, \alpha_2 = 0, \alpha_3 = 0$ )
- $\alpha_2 = 1$  (*i.e.*, a solution minimizing the environmental impact  $-\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0$ )
- $\alpha_3 = 1$  (*i.e.*, a solution minimizing the social impact  $-\alpha_1 = 0, \alpha_2 = 0, \alpha_3 = 1$ )

Results are summarized in Table 5.4. The first column identifies the instance. The second column shows the total cost (TC) of the balanced solution. The next two columns represent two gaps (%) comparing the solution minimizing the economic impact (economic sol) and the balanced one. While the first gap compares the total cost, the second one compares the economic impact. Most values of these gaps are zero and the others (3 out of 40 instances) are positive for the first gap and negative for the second one. All are small in terms of absolute values. Thus, as expected, solutions minimizing the economic impact may have a lower economic impact and a higher total cost than a balanced solution. This suggests that both solutions have similar performance measures or are the same. The fifth and the sixth columns represent two gaps comparing the solution minimizing the environmental impact (co2 sol) and the balanced one. In particular, these gaps show the relative differences in terms of total cost and and environmental impact. The solution minimizing the environmental impact has a significantly higher total cost and a lower environmental impact. These gaps are higher in absolute values than those in the third and fourth columns. Finally, the last two columns compare the solution minimizing the social impact (social sol) and the balanced one. The comparison in terms of total cost and social impacts. These gaps are the highest, which suggest that the corresponding solutions are strongly different. The average gaps are shown in the last row of the table. According to our results, it may be concluded that designing routes aiming to

minimize only the environmental or the social impact increases the total cost 7.21% or 31.01%, respectively, on average in comparison with a balanced objective function. However, this allows to reduce the environmental or the social impacts in 6.29% or 21.15%, respectively. The information in Table 5.4 is graphically represented in Figure 5.2, which shows the behaviour of each gap by means of a boxplot. It helps to visualize the differences of expected values and variability. All boxplots seem relatively symmetric and have few outliers.

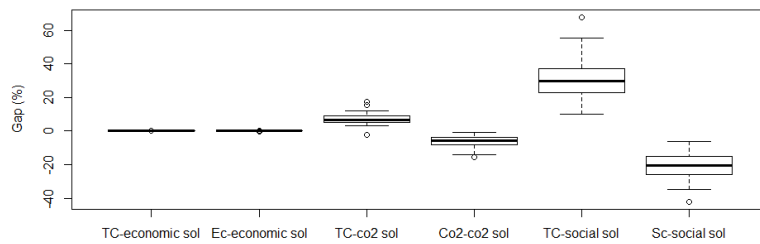


Figure 5.2: Multiple boxplots comparing solutions found with different weights. TC: total cost; Ec: economic cost; Co2: environmental cost; Sc: social cost; sol: solution.

Likewise, this section explores the suitability of a simheuristic approach in environments with a relatively low or high level of stochasticity. In particular, we compare two solutions: the best deterministic solution (BDS), which minimizes the total cost ignoring stochasticity (*i.e.*, considering expected values for demands and travel times), and the best stochastic solution (BSS), which minimizes the expected total cost. An example that helps us to illustrate how one solution may be the best in a deterministic environment and a different one may be the best in a stochastic one is shown in Figure 5.3. It shows the BSS for the instance A-n33-k6 (right) and the BDS (left) considering a single seed and a low level of stochasticity. The axes represent the geographic coordinates, each circle is a customer, and the black square identifies the depot. The value inside each route reveals the expected demand of that route. The maximum capacity of the vehicles is 100. The total costs, the expected total costs and the reliabilities of the solutions are shown. Notice that these solutions do not differ significantly from the type of solutions we would expect if only travel distances were minimized. While the BSS (right) has a higher reliability and a lower expected total cost (*i.e.*, is a better solution in the stochastic environment described), the BDS has a lower total cost (*i.e.*, is a better solution in the deterministic environment). One reason may be that the routes of the BSS are more balanced in terms of expected demand assigned, thus reducing the probability of route failures for running out of capacity.

When designing simheuristics, researchers tend to assume that there is a relatively strong correlation between the best deterministic and stochastic solutions or, in other words, the solutions with a relatively high (low) total cost tend to have also a relatively high (low) expected total cost. However, this correlation is not equal to one and significant differences may be observed.

Tables 5.5 and 5.6 summarize the results found in scenarios with a low and a high level of stochasticity, respectively. The maximum computing time for these experiments is 30 seconds. The first column identifies the instance. The next three columns refer to the BDS and describe the total cost (TC), the expected total cost (ETC), and the reliability. The following column shows the gap between the ETC and the TC; *i.e.*, measures the relative increment in terms of cost observed, on average, if the stochasticity is ignored. Afterwards, columns sixth and seventh represent the ETC and the reliability of the BSS. Finally, the last column reveals the gap between the BSS and the BDS in terms of ETC. Mean reliabilities and gaps are shown in the last row. The mean reliability for the BDS in the environment with a low stochasticity is 0.84. Thus, in 84% of the scenarios simulated there are no route failures. The mean gap between the ETC and TC for the BDS is 3.34%. It reveals that if we ignore the stochasticity and apply the solution found, it will be, on average, 3.34% more expensive than what we expected. The mean reliability of the BSS is slightly higher: 0.86. The mean gap between the BDS and the BSS in terms

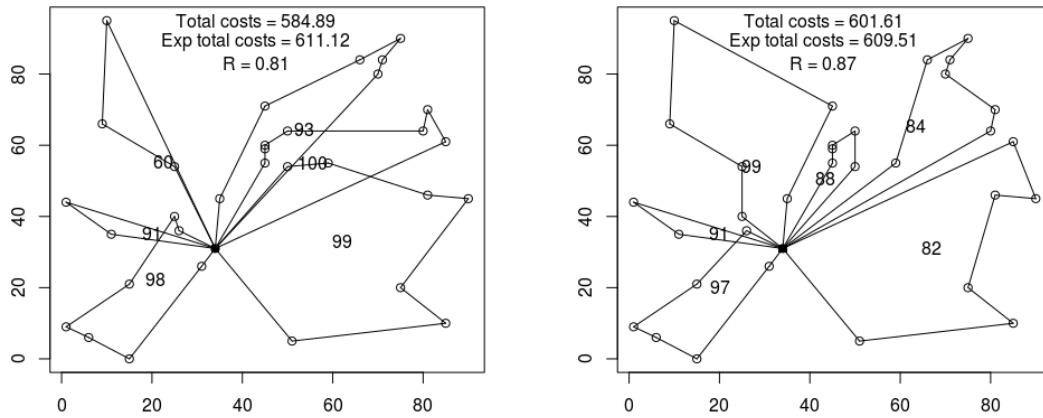


Figure 5.3: Example of different solutions for the instance A-n33-k6.

of ETC is  $-0.42\%$ , which means that considering stochasticity helps us to reduce the cost significantly. However, this gap is null for some instances. Focusing now on the results obtained for a high level of stochasticity (Table 5.6), it can be stated that the reliabilities are significantly lower but the difference between the mean reliability of the BDS and the mean reliability of the BSS is the same: 0.02. As expected, the mean gap between the ETC and the TC for the BDS increases, from  $3.34\%$  to  $6.05\%$ . Moreover, the mean gap between the BDS and the BSS in terms of ETC also increases in absolute values, from 0.42 to 0.47. These results suggest that ignoring the stochasticity becomes more expensive as the level of stochasticity increases.

## 5.6 Analysis of results

The sixth instances with the highest number of nodes have been selected to study the effect of applying different policies. The instances are: A-n63-k10, A-n65-k9, A-n69-k9, B-n66-k9, B-n67-k10, and B-n68-k9. Regarding policies applied, three options are studied: no policies, only corrective policies, and both corrective and preventive policies. The maximum computing time is set to 30 seconds. Figure 5.4 summarizes the results. The plot on the left shows the ETC while the plot on the right reveals the reliabilities. The  $x$ -axis identifies the policies applied and each line is associated to a different instance. In many cases (A-n63-k10, A-n69-k9, and B-n66-k9), introducing corrective policies leads to a reduction of the ETC



and an increase in the reliability. It is possible to obtain a higher ETC and/or a lower reliability if we take into account that implementing no policies means that a solution becomes unfeasible (*i.e.*, not all demands are satisfied) when a route failure happens. By introducing corrective policies, we ensure that all the routes finish. When we further introduce preventive policies, all the ETCs decrease (by a different amount) and all the reliabilities except two increase. It proves the suitability of considering different kinds of policies

Table 5.5: Comparison between the best deterministic and the best stochastic solutions with a low level of stochasticity.

Instance	BDS-TC	BDS-ETC	BDS-rel	ETC-TC (Gap %)	BSS-ETC	BSS-rel	BSS.ETC-BDS.ETC (Gap %)
A-n32-k5	502.62	519.54	0.92	3.37	517.87	0.92	-0.32
A-n33-k5	494.55	502.16	0.93	1.54	502.16	0.93	0.00
A-n33-k6	576.40	597.87	0.80	3.72	585.84	0.93	-2.01
A-n34-k5	511.96	523.38	0.92	2.23	523.38	0.92	0.00
A-n36-k5	528.17	545.60	0.91	3.30	543.02	0.95	-0.47
A-n37-k5	489.23	494.99	0.99	1.18	494.99	0.99	0.00
A-n37-k6	626.44	651.49	0.78	4.00	645.04	0.84	-0.99
A-n38-k5	522.58	547.17	0.78	4.71	544.29	0.85	-0.53
A-n39-k5	524.00	544.80	0.85	3.97	543.09	0.82	-0.31
A-n39-k6	603.73	611.44	0.94	1.28	611.44	0.94	0.00
A-n44-k6	635.03	650.02	0.84	2.36	648.21	0.83	-0.28
A-n45-k6	682.94	700.21	0.91	2.53	698.71	0.99	-0.21
A-n45-k7	721.90	758.21	0.80	5.03	739.26	0.95	-2.50
A-n46-k7	672.38	685.98	0.87	2.02	685.98	0.87	0.00
A-n48-k7	720.63	734.98	0.91	1.99	734.28	0.96	-0.10
A-n53-k7	713.82	728.26	0.81	2.02	728.26	0.81	0.00
A-n54-k7	727.29	753.98	0.75	3.67	753.98	0.75	0.00
A-n55-k9	876.15	903.55	0.76	3.13	903.55	0.76	0.00
A-n60-k9	907.31	941.04	0.79	3.72	941.04	0.79	0.00
A-n61-k9	907.24	925.80	0.87	2.05	925.80	0.87	0.00
A-n62-k8	818.45	839.91	0.86	2.62	839.35	0.85	-0.07
A-n63-k10	952.17	975.79	0.78	2.48	975.52	0.80	-0.03
A-n63-k9	1007.89	1046.31	0.77	3.81	1046.15	0.84	-0.02
A-n65-k9	919.30	965.39	0.63	5.01	962.79	0.63	-0.27
A-n69-k9	883.34	910.29	0.77	3.05	903.34	0.76	-0.76
B-n31-k5	474.83	487.21	0.91	2.61	486.54	0.94	-0.14
B-n34-k5	502.73	524.47	0.89	4.32	520.82	0.92	-0.70
B-n38-k6	569.12	589.37	0.92	3.56	587.40	0.98	-0.33
B-n41-k6	606.17	650.59	0.77	7.33	628.89	0.84	-3.34
B-n43-k6	550.37	570.37	0.86	3.63	565.87	0.93	-0.79
B-n44-k7	658.60	684.02	0.79	3.86	682.06	0.89	-0.29
B-n45-k5	519.00	549.03	0.79	5.79	549.03	0.79	0.00
B-n50-k7	628.44	643.05	0.87	2.32	641.30	0.90	-0.27
B-n50-k8	810.97	849.00	0.78	4.69	832.37	0.82	-1.96
B-n57-k9	947.26	985.88	0.87	4.08	985.88	0.87	0.00
B-n63-k10	1017.18	1046.74	0.79	2.91	1045.94	0.79	-0.08
B-n64-k9	863.98	876.30	0.91	1.43	876.30	0.91	0.00
B-n66-k9	911.21	959.44	0.76	5.29	958.89	0.76	-0.06
B-n67-k10	916.70	947.62	0.76	3.37	946.27	0.76	-0.14
B-n68-k9	895.53	929.08	0.81	3.75	929.08	0.81	0.00
Average			0.84	3.34		0.86	-0.42

TC: total cost; ETC: expected total cost; rel: reliability; BDS: best deterministic solution; BSS: best stochastic solution.

Table 5.6: Comparison between the best deterministic and the best stochastic solutions with a high level of stochasticity.

Instance	BDS-TC	BDS-ETC	BDS-rel	ETC-TC (Gap %)	BSS-ETC	BSS-rel	BSS.ETC-BDS.ETC (Gap %)
A-n32-k5	513.62	538.83	0.81	4.91	537.98	0.81	-0.16
A-n33-k5	489.53	505.82	0.74	3.33	504.63	0.74	-0.24
A-n33-k6	581.00	612.81	0.62	5.48	608.76	0.58	-0.66
A-n34-k5	527.34	556.68	0.66	5.56	556.68	0.66	0.00
A-n36-k5	522.06	543.34	0.76	4.08	542.67	0.76	-0.12
A-n37-k5	500.18	517.43	0.83	3.45	516.41	0.85	-0.20
A-n37-k6	637.53	676.92	0.53	6.18	674.44	0.58	-0.37
A-n38-k5	523.71	555.85	0.60	6.14	555.31	0.60	-0.10
A-n39-k5	524.00	565.76	0.62	7.97	563.35	0.62	-0.43
A-n39-k6	603.23	627.33	0.68	4.00	624.14	0.65	-0.51
A-n44-k6	632.07	666.74	0.62	5.49	666.74	0.62	0.00
A-n45-k6	678.68	706.43	0.69	4.09	705.24	0.69	-0.17
A-n45-k7	721.02	764.08	0.61	5.97	764.08	0.61	0.00
A-n46-k7	681.42	721.73	0.62	5.92	719.12	0.69	-0.36
A-n48-k7	716.27	773.22	0.56	7.95	756.38	0.63	-2.18
A-n53-k7	757.66	807.04	0.48	6.52	807.04	0.48	0.00
A-n54-k7	778.58	831.29	0.59	6.77	831.29	0.59	0.00
A-n55-k9	913.37	971.82	0.42	6.40	951.20	0.58	-2.12
A-n60-k9	976.21	1031.89	0.59	5.70	1031.89	0.59	0.00
A-n61-k9	924.18	960.67	0.54	3.95	960.67	0.54	0.00
A-n62-k8	846.70	912.75	0.50	7.80	912.40	0.52	-0.04
A-n63-k10	1018.39	1052.37	0.64	3.34	1052.37	0.64	0.00
A-n63-k9	1021.23	1079.26	0.56	5.68	1079.26	0.56	0.00
A-n65-k9	954.68	1022.97	0.40	7.15	1005.06	0.58	-1.75
A-n69-k9	930.42	988.84	0.45	6.28	978.02	0.58	-1.09
B-n31-k5	470.36	507.85	0.71	7.97	506.49	0.72	-0.27
B-n34-k5	502.73	544.43	0.69	8.29	540.98	0.64	-0.63
B-n38-k6	566.94	602.77	0.70	6.32	599.80	0.72	-0.49
B-n41-k6	606.80	668.69	0.60	10.20	648.38	0.60	-3.04
B-n43-k6	575.33	612.30	0.64	6.43	610.90	0.68	-0.23
B-n44-k7	642.94	687.43	0.62	6.92	686.45	0.62	-0.14
B-n45-k5	518.34	565.10	0.59	9.02	562.17	0.62	-0.52
B-n50-k7	631.25	658.95	0.63	4.39	651.29	0.73	-1.16
B-n50-k8	874.04	911.91	0.68	4.33	911.20	0.67	-0.08
B-n57-k9	969.56	1060.70	0.49	9.4	1056.53	0.53	-0.39
B-n63-k10	1076.53	1151.43	0.53	6.96	1151.43	0.53	0.00
B-n64-k9	865.97	904.06	0.51	4.4	904.06	0.51	0.00
B-n66-k9	983.04	1043.78	0.54	6.18	1036.51	0.58	-0.70
B-n67-k10	973.05	1027.55	0.47	5.6	1027.55	0.47	0.00
B-n68-k9	960.57	1014.51	0.62	5.62	1004.31	0.65	-1.01
Average			0.61	6.05		0.63	-0.47

TC: total cost; ETC: expected total cost; rel: reliability; BDS: best deterministic solution; BSS: best stochastic solution.

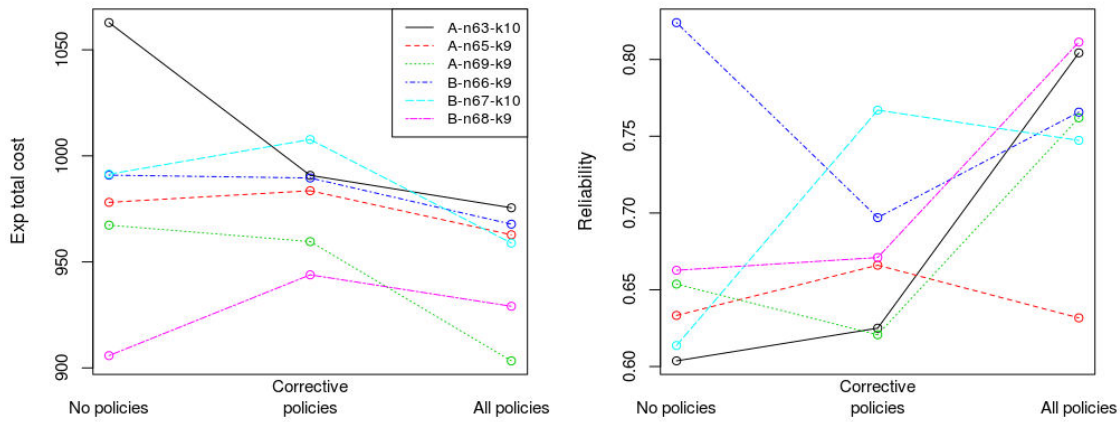


Figure 5.4: Effect of policies on the expected total cost and the reliability.

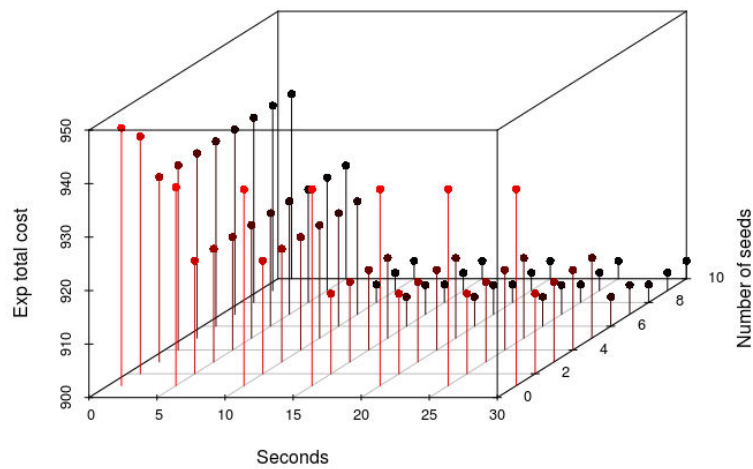


Figure 5.5: Effect of the computing time (seconds) and the number of seeds on the expected total cost for the instance A-n69-k9.

Additionally, in this section is studied the effect of different number of seeds and maximum computing time. The instance A-n69-k9 has been solved considering different limits of time, (1, 5, 10, 15, 20, 25, and 30) seconds, and different numbers

of seeds, (1, 2, 3, 4, 5, 6, 7, 8, 9, and 10). Figure 5.5 shows the ETC as a function of the computing time and the number of seeds. According to the results, both parameters have an important effect. Setting the number of seconds to 15 and the number of seeds to 6, it is difficult to obtain better solutions by increasing any of the parameters. This same pattern has been observed for other instances of a similar size (or number of nodes). However, the reliabilities do not show such a regular landscape, which indicates that the correlation among expected total cost and reliability is not perfect.

## 5.7 Contributions

This chapter addresses the sustainable capacitated vehicle routing problem under stochastic travel times and demands. For solving this problem, a SIM-BR-ILS algorithm is proposed that integrates Monte Carlo simulation into an iterated local search metaheuristics. The SIM-BR-ILS algorithm includes preventive and corrective policies to deal with the stochasticity of the problem. According to our results, the SIM-BR-ILS algorithm is competitive in deterministic environments. Moreover, even allowing a low level of stochasticity, the solution found ignoring randomness present a poor performance. Similarly, the policies effects, the seeds and the maximum computing time are studied in order to achieve a better knowledge of the problem.

Table 5.3: Comparison of our approach against BKS for deterministic instances when minimizing distance.

Instance	BKS (Km)	SIM-BR-ILS (Km)	Run time (s.)	Gap (%)
A-n32-k5	784.00	787.08	6.02	0.39
A-n33-k5	661.00	662.11	0.11	0.17
A-n33-k6	742.00	742.69	65.69	0.09
A-n34-k5	778.00	780.94	0.08	0.38
A-n36-k5	799.00	809.71	56.39	1.34
A-n37-k5	669.00	672.47	0.22	0.52
A-n37-k6	949.00	950.85	0.51	0.20
A-n38-k5	730.00	733.95	0.96	0.54
A-n39-k5	822.00	829.44	8.16	0.90
A-n39-k6	831.00	833.20	1.91	0.27
A-n44-k6	937.00	938.84	0.40	0.20
A-n45-k6	944.00	957.88	6.55	1.47
A-n45-k7	1146.00	1146.91	15.30	0.08
A-n46-k7	914.00	917.72	35.73	0.41
A-n48-k7	1073.00	1074.34	1.26	0.12
A-n53-k7	1010.00	1020.76	76.45	1.07
A-n54-k7	1167.00	1171.68	15.80	0.40
A-n55-k9	1073.00	1081.71	1.68	0.81
A-n60-k9	1354.00	1360.59	31.67	0.49
A-n61-k9	1034.00	1070.49	4.74	3.53
A-n62-k8	1288.00	1319.59	90.29	2.45
A-n63-k9	1616.00	1622.14	199.99	0.38
A-n63-k10	1314.00	1343.77	76.60	2.27
A-n65-k9	1174.00	1193.45	3.74	1.66
A-n69-k9	1159.00	1179.76	86.61	1.79
B-n31-k5	672.00	676.09	2.36	0.61
B-n34-k5	788.00	789.84	5.06	0.23
B-n38-k6	805.00	807.88	38.35	0.36
B-n41-k6	829.00	833.66	26.54	0.56
B-n43-k6	742.00	746.98	0.10	0.67
B-n44-k7	909.00	922.19	21.71	1.45
B-n45-k5	751.00	754.85	0.43	0.51
B-n50-k7	741.00	744.23	0.25	0.44
B-n50-k8	1312.00	1325.47	99.59	1.03
B-n57-k9	1598.00	1609.26	98.65	0.70
B-n63-k10	1496.00	1507.59	95.06	0.77
B-n64-k9	861.00	888.41	6.04	3.18
B-n66-k9	1316.00	1330.10	98.79	1.07
B-n67-k10	1032.00	1053.20	85.40	2.05
B-n68-k9	1272.00	1298.70	93.52	2.10
Average			36.47	0.94

Table 5.4: Total cost and gaps (%) for solutions found with different weights.

Instance	TC-balanced sol	TC-economic sol (Gap %)	Ec-economic sol (Gap %)	TC-co2 sol (Gap %)	Co2-co2 sol (Gap %)	TC-social sol (Gap %)	Sc-social sol (Gap %)
A-n32-k5	518.01	0.00	0.00	7.49	-11.58	44.96	-14.67
A-n33-k5	496.58	0.00	0.00	5.17	-7.86	26.32	-32.80
A-n33-k6	576.86	0.00	0.00	5.66	-2.22	44.86	-26.71
A-n34-k5	522.19	0.00	0.00	6.82	-4.56	22.75	-27.24
A-n36-k5	520.99	0.25	-0.01	7.17	-9.54	32.69	-21.20
A-n37-k5	494.86	0.00	0.00	6.24	-7.35	9.82	-32.26
A-n37-k6	643.18	0.32	-0.50	3.27	-7.95	46.07	-24.57
A-n38-k5	508.03	0.00	0.00	6.66	-0.90	22.16	-25.25
A-n39-k5	530.31	0.00	0.00	6.64	-4.88	30.26	-25.24
A-n39-k6	595.35	0.00	0.00	7.59	-3.43	39.17	-27.57
A-n44-k6	640.35	0.00	0.00	4.26	-5.30	18.57	-29.98
A-n45-k6	683.62	0.00	0.00	9.06	-11.70	16.71	-42.04
A-n45-k7	853.28	0.00	0.00	11.15	-8.34	34.62	-19.36
A-n46-k7	682.21	0.00	0.00	6.45	-8.13	23.06	-16.83
A-n48-k7	769.16	0.00	0.00	0.00	0.00	24.29	-29.28
A-n53-k7	715.18	0.00	0.00	5.07	-5.74	32.47	-27.35
A-n54-k7	749.95	0.00	0.00	17.39	-6.44	46.32	-23.54
A-n55-k9	873.50	0.00	0.00	4.17	-2.84	31.74	-23.45
A-n60-k9	922.74	0.00	0.00	7.45	-12.74	31.22	-20.04
A-n61-k9	911.99	0.00	0.00	4.97	-6.95	43.39	-22.05
A-n62-k8	1109.94	0.00	0.00	6.59	-12.53	21.39	-24.72
A-n63-k10	974.65	0.00	0.00	6.61	-4.11	33.91	-24.32
A-n63-k9	1551.93	0.00	0.00	4.78	-15.36	41.95	-18.25
A-n65-k9	894.25	0.00	0.00	4.38	-4.28	27.77	-26.46
A-n69-k9	888.75	0.00	0.00	5.31	-3.39	25.24	-34.73
B-n31-k5	473.94	0.00	0.00	10.01	-2.11	55.22	-8.29
B-n34-k5	502.73	0.00	0.00	10.21	-1.28	42.38	-8.90
B-n38-k6	569.12	0.05	-0.29	10.55	-5.02	33.87	-12.17
B-n41-k6	606.17	0.00	0.00	4.83	-5.52	29.16	-6.31
B-n43-k6	563.21	0.00	0.00	8.76	-5.26	27.15	-15.63
B-n44-k7	646.96	0.00	0.00	12.18	-6.93	67.63	-11.46
B-n45-k5	519.00	0.00	0.00	5.39	-1.79	26.67	-25.20
B-n50-k7	634.85	0.00	0.00	8.26	-3.67	22.94	-11.66
B-n50-k8	905.47	0.00	0.00	15.69	-13.57	28.75	-16.98
B-n57-k9	2211.33	0.00	0.00	6.78	-3.93	14.98	-6.96
B-n63-k10	1538.83	0.00	0.00	9.22	-14.28	31.91	-12.49
B-n64-k9	868.01	0.00	0.00	6.08	-3.40	24.25	-14.38
B-n66-k9	914.51	0.00	0.00	6.05	-3.61	16.15	-17.94
B-n67-k10	923.95	0.00	0.00	4.83	-6.64	16.12	-18.24
B-n68-k9	885.72	0.00	0.00	9.11	-6.49	31.66	-19.52
Average		0.02	-0.02	7.21	-6.29	31.01	-21.15

TC: total cost; Ec: economic cost; Co2: environmental cost; Sc: social cost; sol: solution.

---

**Algorithm 11** The SIM-BR-ILS algorithm for the SCEVRP-SDT-LC

---

```

1: procedure SIM-BR-ILS(inputs, parameters)
    ▷ inputs : nodes, Q, B, imp, weights, pc, pb
    ▷ imp: parameters to compute impacts
    ▷ pc, pb: minimum probabilities required
    ▷ parameters: maxTime, L, β, p, nSims, nSiml
    ▷ maxTime: maximum computing time allowed
    ▷ L: maximum number of solutions stored
    ▷ β: parameter for biased randomization
    ▷ p: parameter for destruction stage
    ▷ nSims, nSiml: number of scenarios during the loop and after
2: feasibleSol ← FALSE
3: while (feasibleSol is FALSE) do
4:   baseSol ← BRCWS (inputs, β)           ▷ Based on ‘rich’ savings
5:   baseSol ← localSearch(baseSol, inputs)
6:   statistics(baseSol) ← MCS(baseSol, inputs, nSims)
7:   feasibleSol ← checkFeasibility(baseSol, inputs)
8: end while
9: bestSol ← baseSol
10: bestStochSolList ← add(bestSol)           ▷ Store best stochastic solutions
11: while (stopping criterion is not met) do
12:   newSol ← perturb(baseSol, inputs, p)
13:   newSol ← localSearch(newSol, inputs)
14:   if (cost(newSol) ≤ cost(baseSol)) then           ▷ Search for promising solutions
15:     statistics(newSol) ← MCS(newSol, inputs, nSims)
16:     if (checkFeasibility(newSol, inputs) is TRUE) then
17:       rpdcost ← (expCost(newSol) – expCost(baseSol))/expCost(baseSol) · 100
18:       if (rpdcost ≤ 0) then
19:         baseSol ← newSol
20:         update(bestStochSolList, L)
21:       else           ▷ Avoid local optima
22:         u ← generateU()
23:         if (u < exp(–rpdcost)) then
24:           baseSol ← newSol
25:         end if
26:       end if
27:     end if
28:   end if
29: end while
30: for (each sol in bestStochSolList) do
31:   statistics(sol) ← MCS(sol, inputs, nSiml)
32: end for
33: return bestStochSolList
34: end procedure

```

---



## The stochastic team orienteering problem with driving ranges

*These optimization approaches provide good solutions, but they use to require from intensive computations. This chapter describes the strengths and weaknesses of a stochastic programming approach when compared to a simheuristic one for solving a stochastic combinatorial optimization problem. As a benchmark test, the team orienteering problem with random travel times and driving-range limitations is considered. Applications of the TOP to the field of transport can be found, for example, in the use of unmanned aerial vehicles. In the context of smart cities, this type of vehicles offers an alternative way of gathering data (e.g., by taking pictures) and delivering products. A series of computational experiments allow for comparing both approaches in terms of the solutions they can generate.*

The work presented in this chapter has been submitted to the journal of Simulation Modelling Practice and Theory journal:

- Reyes-Rubiano, L., Panadero, J., Juan, A., Faulin, J. (2019). Comparing Stochastic Programming with Simheuristics for Solving the Stochastic Team Orienteering Problem. Progress in Simulation Modelling Practice and Theory. Under review.

Part of the content of this chapter has been published in the proceeding of Winter Simulation Conference 2018:

- Reyes-Rubiano, L. S., Ospina-Trujillo, C. F., Faulin, J., Mozos, J. M., Panadero, J., Juan, A. A. (2018). The Team Orienteering Problem with Stochastic Service Times and Driving-Range Limitations: A Simheuristic Approach. In 2018 Winter Simulation Conference (WSC) (pp. 3025-3035). IEEE.

## 6.1 A BR-VNS simheuristic algorithm for solving the stochastic team orienteering problem

As mentioned in [Chapter 1](#), urban freight transport has to deal with uncertainty. This condition can be an attribute of urban transport problems. Then, the negative impacts caused by transport activities may increase if the uncertain data is disregarded for designing of distribution plans. To overcome the effect of uncertainty in operational level, stochastic versions of routing problems have been introducing.

Over the last decades, deterministic and stochastic optimization problems have been studied and solved using both exact and approximated methods. These optimization approaches provide good solutions, but they use to require from intensive computations. This chapter is focused on analyzing the strengths and weaknesses of a stochastic programming approach when compared to a simheuristic one for solving a stochastic combinatorial optimization problem. Both solving approaches require an accurate process of definition, implementation, and validation. First, a sample average approximation method is proposed to cope with the problem. Then, a simheuristic algorithm is designed by integrating Monte Carlo simulation within a metaheuristic framework.

As a variant of routing problems, in this chapter a stochastic team orienteering problem (TOP) is studied and solved using two different approaches. In the deterministic version of the TOP, the goal is to find the vehicle routes that maximize the rewards obtained by visiting a subset of requesting customers. Usually, a time- or distance-based threshold is imposed on each route. In the stochastic version of the problem, it is usual to consider either random rewards or random travel / servicing times. Several variants of the TOP have been employed in a number of applications related to different areas, such as city logistics, humanitarian logistics, and military logistics ([Gunawan et al., 2016](#)). Nevertheless, the literature related to stochastic versions of the TOP is still scarce, and most of the solving approaches rely on exact methods ([Gunawan et al., 2018](#)).

A formal description of the TOP with deterministic and stochastic travel times is provided. On one hand, the deterministic version of the problem is tackled by a mathematical programming model, which is solved by the CPLEX commercial solver. On the other hand, a sample average approximation (SAA) method ([Shapiro et al., 2009](#)) is implemented for dealing with the stochastic version. This approach provides a solution based on a sampling of travel times. Considering this random sampling, the expected value of the probabilistic rewards is approximated by a deterministic objective function. Finally, a simheuristic algorithm (SIM-BR-VNS) is also proposed to solve the stochastic version of the TOP, and its performance is compared to that of the SAA method. As described in [Juan et al. \(2018\)](#), simheuristics are composed of two different components: an optimization one based on a meta-

heuristic –which searches for promising solutions–, and a simulation one –which assesses the quality of these promising solutions in a stochastic environment and guides the searching process. For the metaheuristic component, a variable neighborhood search (VNS) is employed. Generally speaking, both the SAA and the simheuristic approaches can be classified as simulation-optimization methods. Extensive computational experiments on large instances with up to 102 customers are analyzed using both approaches.

The remaining of the chapter is structured as follows: Section 6.2 briefly reviews the related work; Section 6.3 provides a more detailed description of the TOP version we consider; Section 6.4 provides a formal description of the stochastic TOP. Section 6.4 describes the proposed simheuristic algorithm and the SAA method; Section 6.5 reports the results of the computational experiments. Finally, the main findings and future research lines are given in Section 6.6.

## 6.2 Literature review

The team orienteering problem was introduced by [Chao et al. \(1996\)](#), as a multi-vehicle extension of the orienteering problem (OP) ([Golden et al., 1987](#)). In the classical TOP, it is assumed a perfect knowledge of each customer’s scoring reward and the time incurred in traversing the edges connecting any pair of nodes (customers or depots). In addition, each node can be visited only once, except for the starting and the finishing depots. Some practical applications of the TOP are the home fuel deliver problem and the tourist trip-design problem. More advanced variants of the TOP might consider time windows constraints or even time dependencies ([Gunawan et al., 2016](#)).

Despite of the raising interest in the TOP –which is partly due to the increasing use of unmanned aerial vehicles (UAVs)–, the literature regarding stochastic versions of the problem is still quite limited. Whereas deterministic variants assume the existence of perfect information, more realistic problems use to consider scenarios under uncertainty. In many cases, this uncertainty can be modeled by means of random variables following certain probability distributions, which have been fitted from historical data. These solving approaches are mostly based on the combination of optimization methods with simulation techniques ([Gosavi et al., 2015](#)). Accordingly, the traditional goal when solving a stochastic TOP is the maximization of the expected reward, while other relevant statistics of the proposed solution (e.g., variance, quartiles, etc.) are rarely considered.

Several authors have implemented stochastic programming models to solve vehicle routing problems. Thus, for example, [Lei et al. \(2014\)](#) studied and formulated the mobile facility routing-and-scheduling problem as a two-stage stochastic programming model. Similarly, [Evers et al. \(2014\)](#) formulated an OP with stochastic weights

as a two-stage stochastic programming model. These authors proposed a linearization in order to apply the sample average approximation (SAA) solving method. The state of the art on probabilistic aspects of OPs contains different works that extend deterministic problems and propose original methods to find solutions. For instance, [Varakantham et al. \(2018\)](#) claimed that the implementation of the OP is limited in practice, proposing its extension in order to consider a broader class of problems which contains random factors such as time-dependent travel times. These authors also proposed to solve a stochastic programming problem with chance constraints using SAA. Likewise, [Angelelli et al. \(2017\)](#) proposed a novel stochastic version of the OP and also a metaheuristics to solve problems with a large number of customers.

Moreover, [Ilhan et al. \(2008\)](#) are among the first authors to introduce uncertainties in the collected rewards, discussing the OP with stochastic rewards. [Royset and Reber \(2009\)](#) discussed a TOP application using unmanned aerial vehicles. Likewise, [Erdogan and Laporte \(2013\)](#) tackled the TOP with stochastic rewards using an exact method. Here, service times were based on a finite number of different scenarios. Similarly, [Afsar and Nacima \(2013\)](#) also analyzed the TOP with stochastic rewards using column generation. More recently, [Panadero et al. \(2017\)](#) proposed a simheuristic algorithm to solve the TOP with stochastic travel times, where the expected reward is maximized and the reliability of the generated solutions is analyzed. Also, [Gunawan et al. \(2018\)](#) proposed an iterated local search for solving a similar problem. Furthermore, [Dolinskaya et al. \(2018\)](#) addressed the problem of searching and rescuing operations in a post-disaster situation. Finally, some TOP applications to UAVs are discussed in [Marcosig et al. \(2017\)](#).

### 6.3 Problem description

The TOP with stochastic travel times and maximum travel time per route is an extension of the deterministic TOP, which is a *NP-hard* problem ([Chao et al., 1996](#)). We provide next a model for the stochastic version considered in this chapter, which extends the formulation introduced by [Poggi et al. \(2010\)](#) for the deterministic version. Let us consider a directed graph  $G = (N, A)$ , where: (i)  $N = \{0, 1, \dots, n + 1\}$  is a set of  $n + 2$  nodes including  $n$  customers as well as an origin depot (node 0) and a destination depot (node  $n + 1$ ); and (ii)  $A = \{(i, j)/i, j \in N, i \neq j\}$  is the set of arcs connecting the nodes. A fleet of  $m$  homogeneous vehicles travels through the graph  $G$  visiting some of its nodes, starting from the origin depot and finishing in the destination one. The first time a customer  $i$  is visited, a reward  $u_i \geq 0$  is obtained ( $\forall i = 1, 2, \dots, n$ ). Thus, visiting a customer more than once will not pay off, since no additional reward is gathered. The origin and destination depots have no associated rewards, i.e.:  $u_0 = u_{n+1} = 0$ . In our stochastic version, each arc  $(i, j) \in A$  is associated with a random travel time,  $T_{ij} = T_{ji}$ , which is assumed to

follow a best-fit probability distribution. The total time employed in completing any route is limited by a threshold value,  $t_{max}$  (e.g., the maximum duration of the batteries in case of electric vehicles or the maximum number of hours a driver can work per day). Hence, the main goal is to find the  $m$  visiting routes that maximize the expected aggregated reward. Since travel times are random, whenever a vehicle cannot complete the designed route on or before the deadline, the reward collected so far in that route is considered to be lost, i.e., we are assuming here that partial rewards obtained during a route are only consolidated if the vehicle reaches the destination depot. For each arc  $(i, j) \in A$  and each vehicle  $d \in \{1, 2, \dots, m\}$ , consider the binary variable  $x_{ij}^d$ , which takes the value 1 if vehicle  $d$  covers arc  $(i, j)$  and takes the value 0 otherwise. Likewise, consider the binary variable  $y_j$ , which takes the value 1 if customer  $j$  is visited (i.e.,  $\sum_{d=1}^m \sum_{i \neq j} x_{ij}^d \geq 1$ ), and 0 otherwise. Notice that the actual reward collected from each node  $j$  is also a random variable,  $U_j = U_j(x_{ik}^d, T_{ik})$ , which depends on whether  $j$  is visited by a vehicle  $d_j$  and, if so, on whether the route covered by vehicle  $d_j$  is completed before  $t_{max}$ , i.e.:  $U_j = u_j$  if  $y_j = 1$  and  $\sum_{(i,k) \in A} x_{ik}^{d_j} \cdot T_{ik} \leq t_{max}$ . Otherwise,  $U_j = 0$ . Equation (6.1) denotes the objective function to be maximized, where  $E[U_j]$  represents the expected value of the  $U_j$  random variable:

$$\max \sum_{j=1}^n E[U_j(x_{ik}^d, T_{ik})] \quad (6.1)$$

Constraints 6.2 state that each customer is visited at most once by a single vehicle:

$$\sum_{d=1}^m \sum_{i \neq j} x_{ij}^d \leq 1 \quad \forall j \in \{1, 2, \dots, n\} \quad (6.2)$$

Constraints 6.3 guarantee connectivity of the routes, where  $S$  refers to a subset of nodes and  $\delta^-(S)$  refers to the set of arcs arriving to nodes in  $S$  (notice that for each  $j \in S$ , if  $j$  is visited then there has to be an arc arriving at  $j$ ):

$$\sum_{d=1}^m \sum_{(i,k) \in \delta^-(S)} x_{ik}^d \geq y_j \quad \forall S \subset V, \forall j \in S \quad (6.3)$$

Constraints 6.4 impose that the expected time in traversing any route does not exceed the threshold (in the deterministic variant, this is a hard constraint, while in the stochastic one it becomes a soft one that can be violated at the cost of losing

the rewards in the associated route):

$$\sum_{(i,j) \in A} x_{ij}^d \cdot E[T_{ij}] \leq t_{max} \quad \forall d = 1, 2, \dots, m \quad (6.4)$$

Constraints 6.5 avoids that the same arc can be employed by more than one route:

$$\sum_{d=1}^m x_{ij}^d \leq 1 \quad \forall (i, j) \in A \quad (6.5)$$

Constraint 6.6 states that the total number of arcs departing from the initial depot (node 0) should be equal to the number of vehicles ( $m$ ):

$$\sum_{j=1}^n \sum_{d=1}^m x_{0jd} = m \quad (6.6)$$

Constraint 6.7 states that the total number of arcs arriving to the destination depot (node  $n + 1$ ) should be equal to the number of vehicles ( $m$ ):

$$\sum_{i=1}^n \sum_{d=1}^m x_{i(n+1)d} = m \quad (6.7)$$

Constraints 6.8 refer to the binary character of the  $y_j$  variables:

$$y_j \in \{0, 1\} \quad \forall j \in \{1, 2, \dots, n\} \quad (6.8)$$

Finally, constraints 6.9 refer to the binary character of the  $x_{ij}^d$  variables:

$$x_{ij}^d \in \{0, 1\} \quad \forall i, j \in \{1, 2, \dots, n\}, \forall d \in \{1, 2, \dots, m\} \quad (6.9)$$

## 6.4 Two approaches for solving the stochastic TOP

In the scientific literature, both simulation and optimization approaches have been used to deal with complex systems under stochastic conditions. Usually, simulation is employed to generate sampling observations for the random variables that represent the uncertain behavior of the system, as well as to compute statistics on the system performance. In this section, two approaches that hybridize simulation with optimization techniques are discussed: the sample average approximation method and the simulation-optimization approach known as simheuristics, which has been applied in solving stochastic optimization problems in the areas of vehicle

routing (Juan et al., 2014d; Guimarans et al., 2018), scheduling (Juan et al., 2014a; Gonzalez-Neira et al., 2017; Hatami et al., 2018), computer networks (Cabrera et al., 2014), and facility location (De Armas et al., 2017).

### 6.4.1 The SAA solving approach

The SAA method is a stochastic programming approach that relies on two stages (Figure 6.1). In the first one, a stochastic objective function is estimated (and then transformed into a deterministic one) from a number  $l$  of randomly-sampled scenarios. In the second one, a candidate solution is obtained by solving the mathematical program involving the deterministic objective function generated in the first stage. As a result, the stochastic programming performed by a SAA is a clear application of mathematical-programming optimization via simulation.

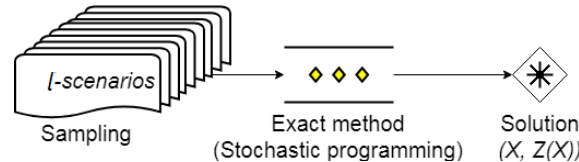


Figure 6.1: The SAA framework.

This solving method is commonly used to deal with stochastic optimization problems. The approach relies on using probability distributions to represent the uncertain parameters of the stochastic problem. Thus, the SAA estimates the expected value of the objective function as a function of decision variables. In order to do so, it uses a random sample of scenarios. As a result, this approach aims at decomposing the stochastic problem into a finite number of scenarios with an associated probability of occurrence (Shapiro et al., 2009). Let us represent by  $t_{ij}^r$  the sampled value obtained for  $T_{ij}$  in scenario  $r$  ( $r = 1, 2, \dots, l$ ).

Then, Equation (6.10) estimates the objective function for our stochastic problem:

$$\sum_{j=1}^n E[U_j(x_{ik}^d, T_{ik})] \approx \frac{1}{l} \cdot \sum_{j=1}^n \sum_{r=1}^l U_j(x_{ik}^d, t_{ik}^r) \quad (6.10)$$

Now, the original stochastic optimization problem can be solved as a deterministic one by employing the approximated objective function. Notice that the larger the number  $l$  of scenarios, the better the approximation. However, increasing this number also raises the computational effort required to solve the problem.

### 6.4.2 The SIM-BR-VNS algorithm solving approach

A simheuristic algorithm extends a metaheuristic one by integrating simulation into it. In [Grasas et al. \(2016\)](#), the authors discuss how to extend an iterated local search framework into a simheuristic. Likewise, in [Ferone et al. \(2018\)](#) the authors explain how to extend a GRASP framework into a simheuristic. A simheuristic algorithm is typically composed of two different components: an optimization one –which searches for promising solutions– and a simulation one –which assesses the promising solutions in a stochastic environment and provides feedback to the optimization one (Figure 6.2). Regarding the optimization component, we use a variable neighborhood search (VNS) framework, in which the constructive phase uses biased-randomization (BR) techniques [Grasas et al. \(2017\)](#). BR techniques have been successfully applied in the past to improve the performance of classical heuristics, both in scheduling applications [Martin et al. \(2016\)](#) as well as in vehicle routing ones [Juan et al. \(2015b\)](#); [Dominguez et al. \(2016a,b\)](#).

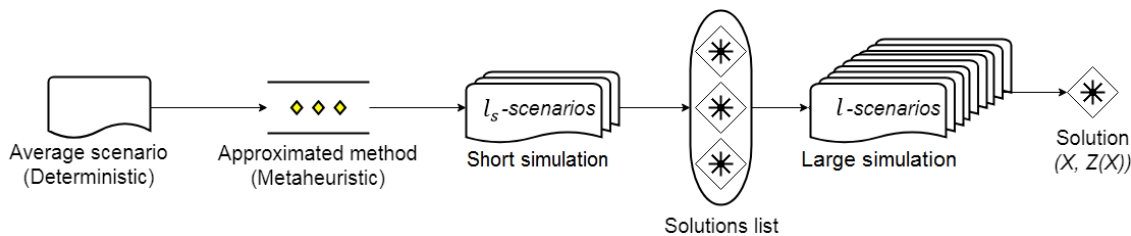


Figure 6.2: A typical simheuristic framework

The main steps of our BR-VNS simheuristic algorithm are described next:

- Firstly, an initial ‘dummy’ solution is built by constructing a route connecting each customer with the origin and destination nodes. In order to merge some of these routes -so that a single vehicle can visit more than one customer- a concept of ‘preference’ level is used: the time-based savings generated by merging any two routes is given by the savings in time associated with completing the merged route instead of the two original ones; this concept is extended to the concept of preference level, which is a linear combination of time-based savings and accumulated reward. This concept of preference level is used to generate a sorted list of potential merges, and these are completed following the corresponding order, from higher to lower preference level. Also, a merge can be completed only if the total expected time after the operation does not exceed the maximum time allowed for any route.
- Secondly, we employ BR techniques ([Grasas et al., 2017](#)) to transform the previously described heuristic into a probabilistic algorithm. Accordingly, an



initial base solution (*baseSol*) is constructed by the BR algorithm. The expected reward provided by the *baseSol* is estimated using a Monte Carlo simulation (MCS). At this stage, this solution is also the best stochastic solution, *bestSol*. Then, a BR-VNS algorithm is implemented to extend the BR algorithm. An overview of the BR-VNS is given in pseudo-code Algorithm 12. A destruction-reconstruction shaking procedure, involving a percentage  $p_r$  of routes in *baseSol*, is used to explore new solutions, *newSol*, inside the search space. Subsequently, the solution is enhanced employing three local search operators. The first one includes a *2-opt* local search. The second one is based on the deletion and reinsertion of percentage  $p_n$  of nodes inside each route. The last local search considers the insertion of non-visited nodes, allowing violations of the maximum-time constraint. In the next step, each *newSol* is compared against the current *baseSol*. If *newSol* has a greater expected reward, it replaces *baseSol*. Otherwise, *newSol* is discarded and a new iteration is started. Similarly, *newSol* is also compared with *bestSol*. The search is interrupted after meeting the stopping criteria. This loop generates a set of ‘elite’ solutions for the stochastic version of the TOP (*poolBestSols*).

- Finally, a more intensive simulation (one with a larger number of runs) is carried out over the elite solutions in order to obtain more accurate estimates on their expected reward.

## 6.5 Computational experiments and results

This section summarizes the results of a numerical study designed to analyze the accuracy and the effectiveness of the two simulation-optimization methods described above in the solving of the stochastic TOP. Both approaches were implemented in Java, the SAA method was compiled using the CPLEX 12.6 library (see Appendix F). Both approaches were run on a computer with 64 GB of RAM and an Intel Xeon at 3.7 GHz. A total of 1,000 scenarios were considered for the SAA method. In the case of our Sim BR-VNS algorithm, 200 runs were employed for each of the fast simulations, while this value was increased to 1,000 runs for the more intensive ones.

### 6.5.1 Analysis of the deterministic TOP

The deterministic version of the TOP with a maximum duration per route was solved both using the CPLEX and the BR-VNS algorithm. The goal here was to validate the BR-VNS performance for solving the deterministic TOP and compare it with the best results from the literature as well as with the results provided by CPLEX

in a limited amount of computing time. The classical TOP instances presented by Chao et al. (1996) were employed for this initial test. Each instance reports a driving range, a number of available vehicles, the coordinates of each node, and their associated rewards. For CPLEX, a maximum computing time of 100,000 seconds was allowed. Table 6.1 presents: (i) the best-known solutions (BKS) from the literature, which correspond to the ones obtained with the PSO-inspired Algorithm (PSOiA) by Dang et al. (2013); (ii) the results generated by CPLEX; and (iii) the ones provided by our BR-VNS algorithm. Notice that our BR-VNS algorithm is able to provide, in short computing times, extremely competitive solutions (average gap of 0% in an average time of 105 seconds). Also, notice that the results provided by our BR-VNS algorithm outperform those provided by CPLEX in the specified time period.

Figure 6.3 shows how the gap between the CPLEX solution (in the maximum time allowed) and the BKS grows as the size of the problem increases. In particular, this gap starts to be noticeable for instances with 66 customers and over. On the contrary, our BR-VNS algorithm is able to provide competitive gaps for instances with up to 102 nodes and employing short computing times.

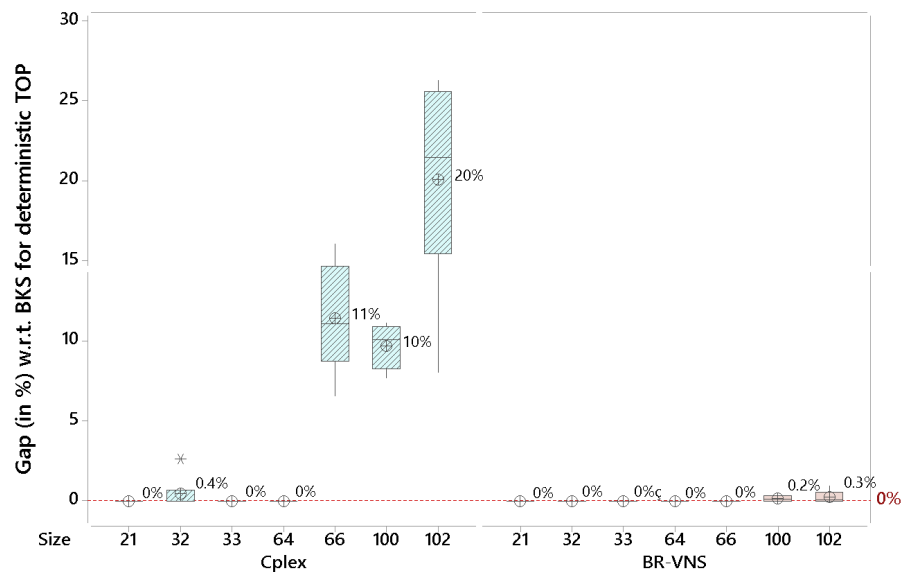


Figure 6.3: Comparison of gaps w.r.t. the BKS of the deterministic TOP.

### 6.5.2 Analysis of the stochastic TOP

In this section both the SAA method and our BR-VNS simheuristic are used to solve the stochastic version of the TOP. With that purpose, we modified and ex-

Table 6.1: Deterministic TOP - Results by CPLEX and our BR-VNS algorithm.

Instance	n	PSOiA (a) Dang et al. (2013)	CPLEX			BR-VNS		
			Reward (b)	Time (s)	Gap (a) - (b)	Reward (c)	Time (s)	Gap (a) - (c)
p1.4.m	32	130	130	74	0%	130	0	0%
p1.4.n		155	155	391	0%	155	0	0%
p1.4.o		165	165	114	0%	165	0	0%
p1.4.p		175	175	63	0%	175	6	0%
p1.4.q		190	185	5070	3%	190	6	0%
p1.4.r		210	210	6109	0%	210	0	0%
p2.4.f	21	105	105	30	0%	105	0	0%
p2.4.g		105	105	34	0%	105	0	0%
p2.4.h		120	120	36	0%	120	0	0%
p2.4.i		120	120	110	0%	120	0	0%
p2.4.j		120	120	599	0%	120	0	0%
p2.4.k		180	180	575	0%	180	0	0%
p3.4.o	33	500	500	10000	0%	500	22	0%
p3.4.p		560	560	6228	0%	560	1	0%
p3.4.q		560	560	8308	0%	560	18	0%
p3.4.r		600	600	10000	0%	600	18	0%
p3.4.s		670	670	10000	0%	670	0	0%
p3.4.t		670	670	10000	0%	670	0	0%
p4.4.o	100	1061	946	100000	11%	1061	149	0%
p4.4.p		1124	1004	100000	11%	1124	543	0%
p4.4.q		1161	1063	100000	8%	1161	160	0%
p4.4.r		1216	1081	100000	11%	1213	464	0%
p4.4.s		1260	1163	100000	8%	1256	456	0%
p4.4.t		1285	1163	100000	9%	1281	65	0%
p5.4.u	66	1300	1115	100000	14%	1300	0	0%
p5.4.v		1320	1195	100000	9%	1320	8	0%
p5.4.w		1390	1245	100000	10%	1390	19	0%
p5.4.x		1450	1280	100000	12%	1450	40	0%
p5.4.y		1520	1420	100000	7%	1520	5	0%
p5.4.z		1620	1360	100000	16%	1620	25	0%
p6.3.n	64	1170	1170	100000	0%	1170	193	0%
p6.4.j		366	366	100000	0%	366	0	0%
p6.4.k		528	528	100000	0%	528	5	0%
p6.4.l		696	696	100000	0%	696	37	0%
p6.4.m		912	912	100000	0%	912	0	0%
p6.4.n		1068	1068	100000	0%	1068	184	0%
p7.4.o	102	781	641	100000	18%	780	208	0%
p7.4.p		846	778	100000	8%	846	54	0%
p7.4.q		909	735	100000	19%	909	437	0%
p7.4.r		970	715	100000	26%	966	512	0%
p7.4.s		1022	763	100000	25%	1022	207	0%
p7.4.t		1077	821	100000	24%	1067	582	1%
<b>Average</b>				<b>58756</b>	<b>6%</b>		<b>105</b>	<b>0%</b>

tended the deterministic data set proposed by [Chao et al. \(1996\)](#) by considering deterministic travel times as the expected values of random travel times following a Log-Normal probability distribution, which constitutes a ‘natural’ choice for modeling non-negative random variables. Hence,  $\forall(i, j) \in A$  we assume that  $T_{ij} \sim \text{LogNormal}(\mu_{ij}, \sigma_{ij})$  with  $E[T_{ij}] = t_{ij}$  and  $\text{Var}[T_{ij}] = c \cdot t_{ij}$ , being  $c > 0$  a design parameter that allows us to consider different levels of uncertainty. It is expected that as  $c$  converges to zero, the results from the stochastic version converge to those obtained in the deterministic scenario. Equations (6.11) define the Log-Normal behavior for a random variable  $T_{ij}$ .

$$\begin{aligned}\mu_{ij} &= \ln(E[T_{ij}]) - \frac{1}{2} \ln\left(1 + \frac{\text{Var}[T_{ij}]}{E[T_{ij}]^2}\right) \\ \sigma_{ij} &= \left| \sqrt{\ln\left(1 + \frac{\text{Var}[T_{ij}]}{E[T_{ij}]^2}\right)} \right|\end{aligned}\tag{6.11}$$

Tables 6.2 to 6.4 show the complete results obtained for the tested instances considering three different variability levels: low ( $c = 0.05$ ), medium ( $c = 0.25$ ), and high ( $c = 0.75$ ). The third column in each table provides the BKS for the deterministic version of the problem ( $c = 0$ ), which can be considered as an upper bound for optimal value in each stochastic scenario. In effect, the achievable expected rewards will tend to be lower as the variability in travel times raises: since travel times have a physical lower bound, but virtually no upper bound, the higher the uncertainty in these travel times the lower the average reward that can be achieved by a given routing plan. The next columns give the solutions, computing times, and gaps with respect to the deterministic solution associated with the SAA method and our BR-VNS simheuristic algorithm. Notice that the SAA method using CPLEX cannot find, after the specified time, solutions for instances sets  $p.4$ ,  $p.5$ , and  $p.7$ . On the contrary, our approach is always able to find ‘good’ solutions in reasonably low computing times, regardless of the instance size. Moreover, in most of the instances our BR-VNS simheuristic algorithm provides better solutions than the SAA approach.

Table 6.2: Stochastic TOP with low variance ( $c = 0.05$ ) - Results by SAA and SIM-BR-VNS.

Instance	n	Deterministic (PSOiA) (a) Dang et al. (2013)	SAA			Sim-BR-VNS			
			Reward (b)	Time (s)	Gap (a) - (b)	Reward (c)	Time (s)	Gap (a) - (c)	
p1.4.m	32	130	119.9	7870	8%	125.9	38	3%	
p1.4.n		155	125.0	7844	19%	135.3	9	13%	
p1.4.o		165	118.7	8008	28%	149.0	28	10%	
p1.4.p		175	139.7	7840	20%	161.7	32	8%	
p1.4.q		190	162.5	7855	15%	173.6	11	9%	
p1.4.r		210	172.5	7749	18%	192.5	17	8%	
p2.4.f	21	105	98.3	7624	6%	78.0	13	26%	
p2.4.g		105	104.4	4023	1%	103.0	35	2%	
p2.4.h		120	117.2	7838	2%	112.0	8	7%	
p2.4.i		120	120.0	4235	0%	112.0	9	7%	
p2.4.j		120	120.0	4694	0%	112.0	8	7%	
p2.4.k		180	159.1	7920	12%	160.0	41	11%	
p3.4.o	33	500	415.9	7800	17%	472.7	23	5%	
p3.4.p		560	477.9	8041	15%	513.8	16	8%	
p3.4.q		560	476.0	7548	15%	555.3	14	1%	
p3.4.r		600	335.1	7346	44%	563.0	99	6%	
p3.4.s		670	534.0	7847	20%	586.3	22	12%	
p3.4.t		670	552.0	7376	18%	664.9	38	1%	
<b>Average</b>				<b>7192</b>		<b>14%</b>		<b>26</b>	<b>8%</b>
p4.4.o	100	1061	N.F	100000	-	918.9	277	13	
p4.4.p		1124	N.F	100000	-	973.8	413	13%	
p4.4.q		1161	N.F	100000	-	1048.0	257	10%	
p4.4.r		1216	N.F	100000	-	1058.2	414	13%	
p4.4.s		1260	N.F	100000	-	1093.0	392	13%	
p4.4.t		1285	N.F	100000	-	1178.0	503	8%	
p5.4.u	66	1300	N.F	100000	-	1176.3	502	10%	
p5.4.v		1320	N.F	100000	-	1274.0	96	3%	
p5.4.w		1390	N.F	100000	-	1299.4	35	7%	
p5.4.x		1450	N.F	100000	-	1323.0	421	9%	
p5.4.y		1520	N.F	100000	-	1402.9	202	8%	
p5.4.z		1620	N.F	100000	-	1495.0	142	8%	
p6.3.n	64	1170	N.F	100000	-	1028.5	230	12%	
p6.4.j		366	N.F	100000	-	298.4	83	18%	
p6.4.k		528	N.F	100000	-	439.4	295	17%	
p6.4.l		696	N.F	100000	-	605.4	259	13%	
p6.4.m		912	N.F	100000	-	836.3	45	8%	
p6.4.n		1068	N.F	100000	-	816.2	272	24%	
p7.4.o	102	781	N.F	100000	-	753.9	193	3%	
p7.4.p		846	N.F	100000	-	792.5	210	6%	
p7.4.q		909	N.F	100000	-	887.0	359	2%	
p7.4.r		970	N.F	100000	-	912.0	382	6%	
p7.4.s		1022	N.F	100000	-	985.1	283	4%	
p7.4.t		1077	N.F	100000	-	980.5	391	9%	

Table 6.3: Stochastic TOP with medium variance ( $c = 0.25$ ) - Results by SAA and SIM-BR-VNS.

Instance	n	Deterministic (PSOiA) (a) Dang et al. (2013)	SAA			Sim-BR-VNS		
			Reward (b)	Time (s)	Gap (a) - (b)	Reward (c)	Time (s)	Gap (a) - (c)
p1.4.m	32	130	97.7	7983	25%	119.4	11	8%
p1.4.n		155	119.8	7976	23%	126.4	64	18%
p1.4.o		165	114.2	7951	31%	136.2	9	17%
p1.4.p		175	124.7	8036	29%	137.3	6	22%
p1.4.q		190	137.2	8034	28%	153.4	56	19%
p1.4.r		210	121.9	8030	42%	175.3	37	17%
p2.4.f	21	105	88.4	7352	16%	73.9	20	30%
p2.4.g		105	99.5	7776	5%	98.5	251	6%
p2.4.h		120	109.7	7576	9%	90.0	89	25%
p2.4.i		120	116.1	7917	3%	94.45	19	21%
p2.4.j		120	119.4	8004	0%	112.3	82	6%
p2.4.k		180	135.7	7646	25%	149.0	57	17%
p3.4.o	33	500	378.7	8034	24%	428.4	199	14%
p3.4.p		560	327.3	7890	42%	465.8	25	17%
p3.4.q		560	338.7	7917	40%	510.3	59	9%
p3.4.r		600	430.9	8039	28%	544.8	106	9%
p3.4.s		670	464.3	7929	31%	546.2	265	18%
p3.4.t		670	455.4	8002	32%	604.2	83	10%
<b>Average</b>			<b>7894</b>	<b>24%</b>		<b>80</b>	<b>16%</b>	
p4.4.o	100	1061	N.F	100000	-	789.8	256	26%
p4.4.p		1124	N.F	100000	-	834.4	400	26%
p4.4.q		1161	N.F	100000	-	899.9	247	22%
p4.4.r		1216	N.F	100000	-	914.3	120	25%
p4.4.s		1260	N.F	100000	-	922.4	394	27%
p4.4.t		1285	N.F	100000	-	999.7	351	22%
p5.4.u	66	1300	N.F	100000	-	847.4	312	35%
p5.4.v		1320	N.F	100000	-	1048.1	290	21%
p5.4.w		1390	N.F	100000	-	1224.0	392	12%
p5.4.x		1450	N.F	100000	-	1297.0	192	11%
p5.4.y		1520	N.F	100000	-	1298.0	384	15%
p5.4.z		1620	N.F	100000	-	1299.0	261	20%
p6.3.n	64	1170	N.F	100000	-	797.9	185	32%
p6.4.j		366.0	N.F	100000	-	269	555	26%
p6.4.k		528	N.F	100000	-	310.8	541	41%
p6.4.l		696	N.F	100000	-	414.2	433	40%
p6.4.m		912	N.F	100000	-	602.6	223	34%
p6.4.n		1068	N.F	100000	-	648.4	179	39%
p7.4.o	102	781	N.F	100000	-	700.6	367	10%
p7.4.p		846	N.F	100000	-	739.1	126	13%
p7.4.q		909	N.F	100000	-	816.3	352	10%
p7.4.r		970	N.F	100000	-	870.8	208	10%
p7.4.s		1022	N.F	100000	-	929.9	281	9%
p7.4.t		1077	N.F	100000	-	959.5	501	11 %

Table 6.4: Stochastic TOP with high variance ( $c = 0.75$ ) - Results by SAA and SIM-BR-VNS.

Instance	N	Deterministic (PSOiA) (a) Dang et al. (2013)	SAA			Sim-BR-VNS		
			Reward (b)	Time (s)	Gap (a) - (b)	Reward (c)	Time (s)	Gap (a) - (c)
p1.4.m	32	130	64.2	7602	51%	101.1	118	22%
p1.4.n		155	83.8	7827	46%	115.8	11	25%
p1.4.o		165	104.6	7803	37%	116.6	42	29%
p1.4.p		175	111.02	7680	37%	123.2	38	30%
p1.4.q		190	103.9	8002	45%	137.9	146	27%
p1.4.r		210	106.9	7895	49%	156.6	144	25%
p2.4.f	21	105	85.7	7119	18%	72.9	21	31%
p2.4.g		105	92.1	7409	12%	92.4	26	12%
p2.4.h		120	104.1	7512	13%	84.8	102	29%
p2.4.i		120	22.9	4827	81%	86.9	35	28%
p2.4.j		120	114.0	7837	5%	99.3	59	17%
p2.4.k		180	131.6	7706	27%	139.1	54	23%
p3.4.o	33	500	317.1	7759	37%	393.5	8	21%
p3.4.p		560	278.8	7630	50%	413.3	155	26%
p3.4.q		560	361.4	7518	35%	475.5	399	15%
p3.4.r		600	364.4	7836	39%	495.0	323	18%
p3.4.s		670	407.7	7924	39%	517.7	108	23%
p3.4.t		670	489.9	7808	27%	540.4	231	19%
<b>Average</b>				<b>7539</b>	<b>36%</b>		<b>112</b>	<b>23%</b>
p4.4.o	100	1061	N.F	100000	-	682.1	200	36
p4.4.p		1124	N.F	100000	-	743.9	206	34
p4.4.q		1161	N.F	100000	-	777.4	121	33
p4.4.r		1216	N.F	100000	-	811.2	248	33
p4.4.s		1260	N.F	100000	-	853.3	276	32
p4.4.t		1285	N.F	100000	-	879.7	305	32
p5.4.u	66	1300	N.F	100000	-	767.0	193	41%
p5.4.v		1320	N.F	100000	-	931.1	68	29%
p5.4.w		1390	N.F	100000	-	1080.9	480	22%
p5.4.x		1450	N.F	100000	-	1175.8	367	19%
p5.4.y		1520	N.F	100000	-	1241.8	383	18%
p5.4.z		1620	N.F	100000	-	1272.7	593	21%
p6.3.n	64	1170	N.F	100000	-	722.1	325	38%
p6.4.j		366	N.F	100000	-	207.5	525	43%
p6.4.k		528	N.F	100000	-	299.4	231	43%
p6.4.l		696	N.F	100000	-	401.7	12	42%
p6.4.m		912	N.F	100000	-	564.2	126	38%
p6.4.n		1068	N.F	100000	-	611.8	199	43%
p7.4.o	102	781	N.F	100000	-	640.4	570	18%
p7.4.p		846	N.F	100000	-	698.8	456	17%
p7.4.q		909	N.F	100000	-	743.1	435	18%
p7.4.r		970	N.F	100000	-	821.4	218	15%
p7.4.s		1022	N.F	100000	-	854.1	433	16%
p7.4.t		1077	N.F	100000	-	892.7	358	17%

For those instances that could be solved by the SAA approach, Figure 6.4 shows percentage gaps with respect to the BKS associated with the deterministic version of the TOP. Here the suffix ‘D’ refers to the deterministic version of the TOP, while the suffixes ‘L’, ‘M’, and ‘H’ refer to the low-, medium-, and high-variance

stochastic versions, respectively. As discussed before, these BKS can be seen as upper bounds for the optimal solution of the stochastic TOP. Notice that, for each considered scenario, the average gap provided by our BR-VNS simheuristic is always lower (i.e., better) than the one provided by the SAA approach. In the low-variance scenario, our BR-VNS simheuristic offers an average gap of about 8%, while the average gap raises to 14% for the SAA. As the level of variability increases, these gaps also go higher. Thus, for the high-variance scenario, the average gap provided by the BR-VNS is about 23%, while the one associated with the SAA goes up to 36%. Similarly, Figure 6.5 shows the associated computing times. Again, for considered scenarios, our BR-VNS algorithm clearly outperforms the SAA approach in this dimension.



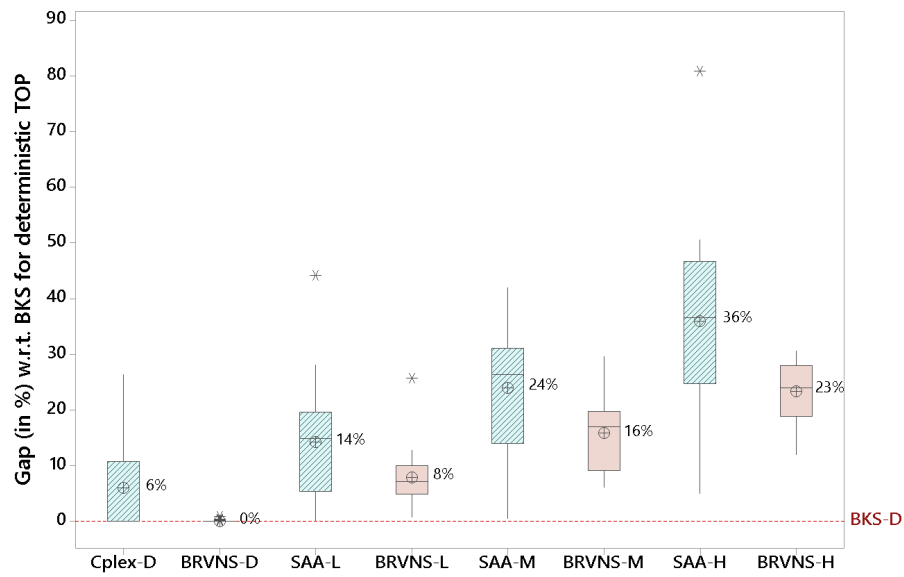


Figure 6.4: The gap comparison

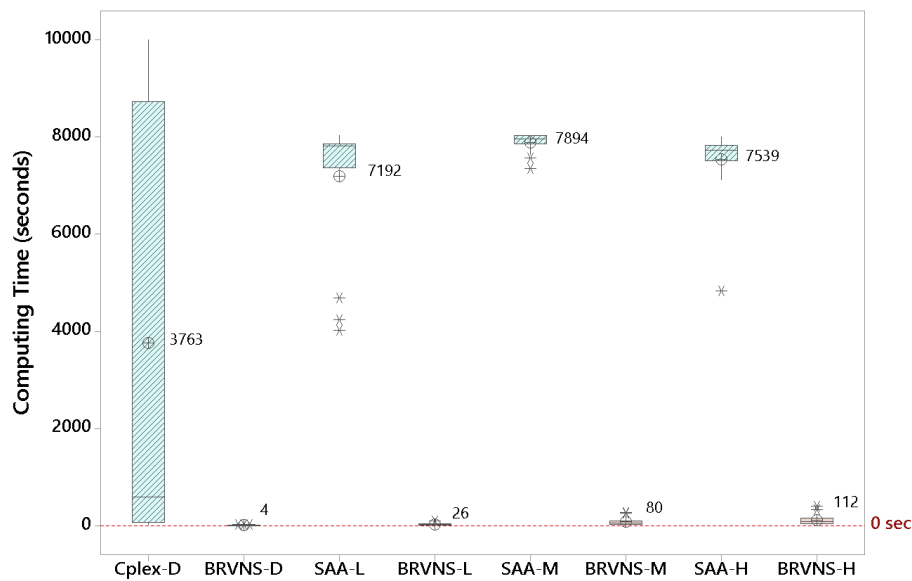


Figure 6.5: The computing time comparison

Figure 6.6: The comparison between the SAA and the BR-VNS simheuristic.

## 6.6 Contributions

The classical deterministic version of the TOP has been solved both using the CPLEX commercial optimizer as well Sim-BR-VNS algorithm. The contributions of this chapter relies on the numerical results which suggest that stochastic programming methods like the SAA can efficiently solve small scale instances. However, they find severe difficulties as the size of the instance grows. For this particular problem, the SAA approach has serious difficulties to solve, at least in reasonable computing times, instances with more than 65 customers. On the contrary, the Sim-BR-VNS algorithm –which do not rely on exact methods– can efficiently solve stochastic instances with up to 102 customers in low computing times.

**Algorithm 12** The SIM-BR-VNS algorithm for the TOP

---

```

1: procedure SIM-BR-VNS(inputs, parameters)
2:   baseSol  $\leftarrow$  genInitSol ▷ solve biased randomized heuristic
3:   shortSimulation(baseSol) ▷ MCS
4:   bestSol  $\leftarrow$  baseSol
5:   k  $\leftarrow$  1
6:   while stopping criteria not reached do
7:     newSol  $\leftarrow$  shaking(baseSol, pr) ▷ percentage pr of routes
8:     newSol  $\leftarrow$  localSearch(newSol, pn) ▷ percentage pn of nodes
9:     if detReward(newSol) > detReward(baseSol) then
10:      shortSimulation(newSol) ▷ MCS
11:      if stochReward(newSol) > stochReward(baseSol) then
12:        baseSol  $\leftarrow$  newSol
13:        if stochReward(newSol) > stochReward(bestSol) then
14:          bestSol  $\leftarrow$  newSol
15:          insert(poolBestSols, bestSol)
16:        end if
17:        k  $\leftarrow$  1
18:      end if
19:    end if k  $\leftarrow$  min{k + 1, 100}
20:  end while
21: end procedure

```

---



## Conclusions and further work

### Conclusions

This thesis is devoted to study sustainable vehicle routing problems. These problems extend rich VRPs by the inclusion of the main attributes and constraints of the sustainable freight transport system. Despite some studies include the sustainability dimensions, the gap with classical models and real-world applications is still significant. Therefore, the purpose of this thesis is to contribute to closing that gap. First, a review on works considering the vehicle routing problem and sustainability dimensions as the optimization criteria has been presented. While there is a high number of works about the vehicle routing problems, there is still the need to explore new solution approaches and to study new sustainable transport problems that are closer to reality. The sustainable vehicle routing problem can be classified by attributes and constraints of the problem. The attributes are additional characteristics that enrich the classical vehicle routing problem and aim to properly reach a sustainable freight transport. While constraints are problem features that have an effect on the routes structure, such as driving range, balance workload and working hours. Several solving approaches have been designed for dealing with these rich vehicle routing problems, but the development of a general framework to cover these particularities under realistic conditions remains a considerable challenge. Recent solvers are designed to address deterministic problems generalizing realistic attributes of urban freight transport. Obviously, the transport problems are stricken by stochastic conditions which usually the classical solver manage the problem as a deterministic one. Commonly, simplifying assumptions allows providing a solution which might be a good approximation for the stochastic problem. Under this consideration, feasible solutions are hard to ensure for the stochastic problem. Furthermore, such a development is critical for estimating the influence of stochasticity on the solution quality and the performance of the solving approach. Consequently, this

thesis extends the simheuristic framework by the inclusion of sustainability criteria in the solutions-construction stage. The simheuristics are a hybrid methodology which integrates optimization and simulation techniques. It has been designed to address combinatorial optimization problems with stochastic inputs, which depend on the problem nature. A number of potential applications in the sustainable freight transport problems have been identified, and solved. Hence, in this thesis the most successful heuristics were complemented by biased-randomization of techniques and extended by a Monte Carlo simulation.

Applications to urban transport constitutes the main topic in applications. The multi-depot vehicle routing problem has been introduced, and sustainability dimensions have been addressed in [Chapter 3](#). From the literature, this chapter proposes a set of sustainability indicators aiming at estimating the performance of solutions. These indicators are monetary measures quantify the economic, environmental, and social impacts of freight transport. The main objective is to find a sustainable solution minimizing the negative impacts associated with each dimension. For the solution of this sustainable multi-depot VRP a biased-randomized variable neighborhood search algorithm is proposed (BR-VNS). Thus, biased randomization is used at different stages of the metaheuristic in order to better guide the searching process. This includes both the generation of customer-to-depot assignment maps as well as the routing process itself. A set of computational experiments are carried out in order to test our approach and illustrate its use. The BR-VNS algorithm is able to report high-quality solutions in short computing times, and enables decision-makers to assess solutions under particular interests regarding the impacts considered. Furthermore, visualization techniques are used to represent the trade-offs among sustainability dimensions.

[Chapter 4](#) extends the capacitated electric VRP to a stochastic combinatorial optimization problem. This chapter analyzes the aforementioned problem considering stochastic travel time. Also, the driving-range limitations might cause route failures when the vehicle runs out of battery. The stochastic travel time is addressed by a Monte Carlo simulation. For solving this problem, a biased-randomized version of multi-start simheuristic algorithm is proposed. This solving approach is validated under deterministic and stochastic scenarios. This test allows concluding that a deterministic solution in stochastic scenarios might lead to a sub-optimal distribution plan that can be easily improved by using a simulation-optimization technique such as the one proposed here. Additionally, [Chapter 4](#) presents a preventive policy relies on the use of the suitable energy safety stock levels. This policy is considered for designing, enhancing the reliability of the distribution plans and reduce the total expected costs.

Similarly, [Chapter 5](#) addresses the sustainable capacitated vehicle routing problem under stochastic travel times and demands. For solving this problem, we have

proposed a simheuristic algorithm that integrates Monte Carlo simulation into an iterated local search metaheuristic (SIM-BR-ILS). This SIM-BR-ILS algorithm includes preventive and corrective policies to deal with the stochasticity of the problem. According to the computational results, the SIM-BR-ILS algorithm is competitive in deterministic environments. As expected, for scenarios with a high or low uncertainty level the deterministic solution presents a poor performance. As a conclusion, introducing corrective and preventive policies while designing the distribution plan might enhance the solution performance in terms of economic, environmental and social impacts. Moreover it improve the solution reliability.

Finally, [Chapter 6](#) introduces the team orienteering problem aimed at alternative fuel vehicles which performance is limited by driving range. In this chapter, the problem is solved by both a stochastic programming model and a simheuristic algorithm. First, a sample average approximation model (SAA) is developed and implemented as well as a biased-randomized variable neighborhood search simheuristic algorithm (SIM-BR-VNS). The computational results demonstrate that stochastic programming methods like the SAA can efficiently solve small-scale instances. However, they find severe difficulties as the size of the instance grows. For this particular problem, the SAA approach has serious difficulties to solve, at least in reasonable computing times, instances with more than 65 customers. On the contrary, the SIM-BR-VNS algorithm –which do not rely on exact methods– can efficiently solve stochastic instances with up to 102 customers in low computing times.

Generally speaking, the simheuristic framework applications are underlined by the computational results reported through the chapters of this thesis. The obtained results suggest the competitiveness of the simheuristics in comparison to other solving approaches reported in the literature. In addition, the computational results provide protocols sustainability-related for managing the decision-making process under stochastic problems.

## Further work

Despite this thesis contributes to bringing the rich VRPs closer to reality balancing the social and industrial needs, there are pending research lines about sustainability issues into VRPs. In this thesis, some research topics are suggested:

- The introduction of sustainability indicators in richer VRPs considering heterogeneous electric vehicles in terms of capacity, driving ranges, battery type, and fuel consumption rate.
- The effect of preventive and corrective strategies for stochastic problems, taking into account possible correlations among travel times associated with different edges.

- Several rich vehicles routing problems have been addressed. Many realistic characteristics may be added, which could increase the complexity of the problems. It could be interesting to extend the simheuristic algorithms incorporating the analysis of stochastic programming models.
- The methodology of simheuristic algorithm and the biased randomization of heuristics can be extended to address multi-objective optimization problems.
- An online simheuristic algorithm need to be developed, in which the real time information can be used to improve the simulation quality.
- The design and testing new approaches relying on simheuristic algorithm and the biased randomization of heuristics for solving stochastic problems, not necessarily giving more relevance to the optimization process than the one for the simulation process.



The discussions and results presented in this work are based on some research outputs in the form of publications in ISI-SCI journals and the international conferences. Developed research dissemination includes:

#### Conferences:

- Juan, A.; Faulin, J.; Raba, D.; Freixes, A.; Reyes, L. (2018). Simheuristic Algorithms for Transportation and Logistics Problems under Uncertainty. Actas del XIII Congreso de Ingeniería del Transporte. Gijon, Spain. June 6-8.
- Fajardo, J.; Juan, A.; Serrano, A.; Reyes, L. (2018). Biased Randomization Strategies in Metaheuristics to Solve Problems in Transportation. Actas del XXXVII Congreso Nacional de Estadística e Investigación Operativa. Oviedo, Spain.
- Juan, A.; Faulin, J.; Reyes, L.; Raba, D.; Freixes, A. (2018). Simheuristics: Extending Metaheuristics to solve Optimization Problems under Uncertainty Scenarios. Actas del XXXVII Congreso Nacional de Estadística e Investigación Operativa. Oviedo, Spain.
- Faulin, J.; Juan, A.; Reyes, L.; Panadero, J.; Ospina, C.; Mozos, J. (2018). A Simheuristic Algorithm for the Team Orienteering Problem. XVIII Spanish Conference on Artificial Intelligence. Granada, Spain.
- Reyes, L.; Faulin, J.; Juan, A. (2018). Urban Freight Distribution and Sustainability. 2018 CYTED Workshop. Rio de Janeiro, Brazil.

- Faulin, J.; Reyes, L.; Calvet, L.; Juan, A. (2018). Sustainability Dimensions as Optimization Criteria: A Capacitated Vehicle Routing Problem. Joint EURO/ALIO International Conference 2018 on Applied Combinatorial Optimization. Bologna, Italy.
- Reyes, L.; Calvet, L.; Juan, A.; Faulin, J. (2017). Sustainable Urban Freight Transport Considering Multiple Capacitated Depots. Metaheuristics International Conference, 1-3. Barcelona, Spain.
- Reyes-Rubiano, L., Calvet, L., Juan, A. A., Faulin, J. (2017). Solving the Multi-Depot Vehicle Routing Problem with Sustainability Indicators. In VeRoLog, Amsterdam, Netherlands.
- Faulin, J.; Juan, A.; Serrano, A.; Reyes, L.; Latorre, I. (2017). Biased Randomization Procedures in Metaheuristics to Solve Stochastic Combinatorial Problems in Transportation and Logistics. International Conference on Operations Research. Berlin, Germany.

#### Articles in JCR-Indexed journals ( Thomson Reuters):

- Reyes, L.; Ferone, D.; Juan, A.; Faulin, J. (2019): A Simheuristic for Routing Electric Vehicles with Limited Driving Ranges and Stochastic Travel Times. *Statistics and Operations Research Transactions*, 43(1), doi 10.2436/20-8080-02-77 (indexed in ISI SCI, 2017 IF = 1.344, Q2; 2017 SJR = 0.551, Q2). ISSN: 1696-2281.
- Reyes, L.; Calvet, L.; Juan, A.; Faulin, J.; Bove, L. (2018): Sustainable Urban Freight Transport: a multi-depot vehicle routing problem considering different cost dimensions. *Journal of Heuristics*, doi 10.1007/s10732-018-9366-0 (indexed in ISI SCI, 2017 IF = 1.129, Q3; 2017 SJR = 1.008, Q1). ISSN: 1381-1231.
- Reyes, L.; Faulin, J.; Calvet, L.; Juan, A. A. (2018): A simheuristic approach for freight transportation in smart cities. *Proceedings of Winter Simulation Conference*, 3346-3357 (Indexed in ISI WOS and Scopus, 2016 SJR = 0.228). ISSN: 0891-7736.
- Reyes, L.; Ospina-Trujillo, Carlos F.; Mozos Jose M.; Panadero Javier; Juan, A. Angel (2019): The team orienteering problem with stochastic service times and driving-range limitations: a simheuristic approach. *Proceedings of Winter Simulation Conference*, 3025–3035 (Indexed in ISI WOS and Scopus, 2017 SJR = 0.37). ISSN: 0891-7736.

**Journal articles submitted (under review):**

- Reyes, L.; Panadero, J.; Juan, A.; Faulin, J. (2019): Comparing Stochastic Programming with Simheuristics for Solving the Stochastic Team Orienteering Problem. *Simulation Modelling Practice and Theory* (indexed in ISI SCI, 2017 IF = 2.092, Q1; 2017 SJR = 0.684, Q1). ISSN: 1569-190X.
- Calvet, L.; Reyes-Rubiano, L., Abdullahi, H.; Ouelhadj, D. A.; Faulin, J. (2019): A hybrid approach for sustainable freight transport considering stochastic travel times and demand. *Int. Transactions in Operational Research*, DOI: 10.1111/itor.12625 (indexed in ISI SCI, 2017 IF = 2.400, Q1; 2017 SJR = 1.071, Q1). ISSN: 0969-6016.

# A biased-randomized variable neighborhood search for sustainable multi-depot vehicle routing problems

Lorena Reyes-Rubiano<sup>1</sup> · Laura Calvet<sup>2,3</sup> ·  
Angel A. Juan<sup>2,3</sup> · Javier Faulin<sup>1</sup> · Lluç Bové<sup>2,3</sup>

Received: 28 March 2017 / Revised: 20 January 2018 / Accepted: 13 February 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** Urban freight transport is becoming increasingly complex due to a boost in the volume of products distributed and the associated number of delivery services. In addition, stakeholders' preferences and city logistics dynamics affect the freight flow and the efficiency of the delivery process in downtown areas. In general, transport activities have a significant and negative impact on the environment and citizens' welfare, which motivates the need for sustainable transport planning. This work proposes a metaheuristic-based approach for tackling an enriched multi-depot vehicle routing problem in which economic, environmental, and social dimensions are considered. Our approach integrates biased-randomization strategies within a variable neighborhood search framework in order to better guide the searching process. A series of computational experiments illustrates how the aforementioned dimensions can be integrated

---

Lorena Reyes-Rubiano  
lorena.reyes@unavarra.es

Laura Calvet  
lcalvetl@uoc.edu

Angel A. Juan  
ajuanp@uoc.edu

Javier Faulin  
javier.faulin@unavarra.es

Lluç Bové  
lbovec@uoc.edu

<sup>1</sup> Institute of Smart Cities, Public University of Navarra, Pamplona, Spain

<sup>2</sup> IN3 - Computer Science, Multimedia and Telecommunication Department, Open University of Catalonia, Barcelona, Spain

<sup>3</sup> Euncet Business School, Terrassa, Spain

## **A SIMHEURISTIC APPROACH FOR FREIGHT TRANSPORTATION IN SMART CITIES**

Lorena S. Reyes-Rubiano  
Javier Faulin

Institute of Smart cities  
Public University of Navarra  
Campus Arrosadia  
31006 Pamplona, SPAIN

Laura Calvet  
Angel A. Juan

IN3 - Computer Science Dept.  
Open University of Catalonia  
Carl Friedrich Gauss Av.  
08860 Castelldefels, SPAIN

### **ABSTRACT**

In modern society, sustainable transportation practices in smart cities are becoming increasingly important for both companies and citizens. These practices constitute a global trend, which affects multiple sectors resulting in relevant socio-economic and environmental challenges. Moreover, uncertainty plays a crucial role in transport activities; for instance, travel time may be affected by road work, the weather, or accidents, among others. This paper addresses a rich extension of the capacitated vehicle routing problem, which considers sustainability indicators (i.e., economic, environmental and social impacts) and stochastic traveling times. A simheuristic approach integrating Monte Carlo simulation into a multi-start metaheuristic is proposed to solve it. A computational experiment is carried out to validate our approach, and analyze the trade-off between sustainability dimensions and the effect of stochasticity on the solutions.

### **1 INTRODUCTION**

Nowadays, the growing public concern for the environment preservation and social welfare is leading to more sustainable cities. In this sense, smart cities are places that have implemented information and communication technologies for getting an optimal transport system, considering economic, environmental, and social aspects in urban zones. For instance, most companies are starting to design and apply smart strategies to control environmental impacts. While a number of studies tackle the sustainability issues from an environmental perspective, the sustainability also involves social and economic factors (McKinnon et al. 2015). In addition, governments create monetary instruments such as road pricing and fines related to emissions excess and traffic noise, among others. Thus, a route cost varies according to the considered country or region, and the type of vehicle and road, to mention some examples.

In this context, we define the capacitated vehicle routing problem with stochastic traveling times (CVRP-ST). Traditionally, the goal of the CVRP is to design routing plans to serve a set of customers from one depot minimizing the traveling distance. A high correlation among traveling times and distances is typically assumed, which is unrealistic in urban routing (Figure 1). Recently, this problem has been enriched with goals related to fuel consumption. However, there are other negative factors that may be reduced by an efficient distribution planning. In fact, there is a lack of works focused on sustainability indicators (Eshtehadi, Fathian, and Demir 2017). Here, we consider the three cost dimensions: i.e., economic, environmental, and social dimensions. Moreover, authors tend to work on deterministic problems, but this assumption is too demanding for routing problems, where there is a wide range of elements with unpredictable but potentially significant effects on traveling times such as road works, the weather, accidents, etc. The high number of agents interacting also adds stochasticity to the traffic flow. Here, we model traveling times as random variables following specific probability distributions, either theoretical or empirical ones.

# A simheuristic for routing electric vehicles with limited driving ranges and stochastic travel times

Lorena Reyes-Rubiano<sup>1,\*</sup>, Daniele Ferone<sup>2</sup>, Angel A. Juan<sup>3,4</sup>  
and Javier Faulin<sup>1</sup>

---

## Abstract

Green transportation is becoming relevant in the context of smart cities, where the use of electric vehicles represents a promising strategy to support sustainability policies. However the use of electric vehicles shows some drawbacks as well, such as their limited driving-range capacity. This paper analyses a realistic vehicle routing problem in which both driving-range constraints and stochastic travel times are considered. Thus, the main goal is to minimize the expected time-based cost required to complete the freight distribution plan. In order to design reliable routing plans, a simheuristic algorithm is proposed. It combines Monte Carlo simulation with a multi-start metaheuristic, which also employs biased-randomization techniques. By including simulation, simheuristics extend the capabilities of metaheuristics to deal with stochastic problems. A series of computational experiments are performed to test our solving approach as well as to analyse the effect of uncertainty on the routing plans.

---

*MSC:* 90B08.

*Keywords:* Vehicle routing problem, electric vehicles, green transport and logistics, smart cities, simheuristics, biased-randomized heuristics.

## 1. Introduction

The growing public concern about living conditions and environmental preservation, specially in the context of modern cities, leads to the emergence and consolidation of the sustainable city concept, which integrates social, environmental, and economic dimensions (McKinnon et al., 2015). Smart sustainable cities call for an intelligent management of resources considering the social welfare in order to achieve a sustainable growth (Bibri and Krogstie, 2017). On the one hand, companies need to satisfy an in-

---

\* *Corresponding author:* lorena.reyes@unavarra.es

<sup>1</sup> Institute of Smart Cities, Public University of Navarra, Pamplona, Spain

<sup>2</sup> Dept. of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Milano, Italy

<sup>3</sup> IN3 – Computer Science Dept., Open University of Catalonia, Barcelona, Spain

<sup>4</sup> Euncet Business School, Terrassa, Spain

Received: March 2018

Accepted: November 2018

**THE TEAM ORIENTEERING PROBLEM WITH STOCHASTIC SERVICE TIMES AND  
DRIVING-RANGE LIMITATIONS: A SIMHEURISTIC APPROACH**

Lorena S. Reyes-Rubiano  
Carlos F. Ospina-Trujillo  
Javier Faulin

Jose M. Mozos  
Javier Panadero  
Angel A. Juan

Institute of Smart Cities  
Public University of Navarra  
Campus Arrosadia  
Pamplona, 31006, SPAIN

Universitat Oberta de Catalunya – IN3  
Euncet Business School  
Carl Friedrich Gauss Av.  
Castelldefels, 08860, SPAIN

**ABSTRACT**

In the context of smart cities, unmanned aerial vehicles (UAVs) offer an alternative way of gathering data and delivering products. On the one hand, in congested urban areas UAVs might represent a faster way of performing some operations than employing road vehicles. On the other hand, they are constrained by driving-range limitations. This paper copes with a version of the well-known Team Orienteering Problem in which a fleet of UAVs has to visit a series of customers. We assume that the rewarding quantity that each UAV receives by visiting a customer is a random variable, and that the service time at each customer depends on the collected reward. The goal is to find the optimal set of customers that must be visited by each UAV without violating the driving-range constraint. A simheuristic algorithm is proposed as a solving approach, which is then validated via a series of computational experiments.

**1 INTRODUCTION**

In a supply chain, a transport system is typically defined as a robust set of links that allows a continuous flow of resources such as information, money, and products. This set of links connects suppliers, production locations, retailers, and customers (McKinnon et al. 2015). This concept is nowadays evolving due to the market dynamics. Customers continuously place orders, which must be satisfied over the course of a vehicle route. As a consequence, multi-echelon supply chains emerge, thus delaying response times and amplifying uncertainty in the supply chain. The introduction of new technologies allows for considering real-time data that can be useful in order to identify suitable links at each time. As a result, the European Commission (2016) has proposed different initiatives and some governmental projects, such as CITYLOG, to facilitate the emergence of sustainable and smart cities. This initiative leads the promotion of transport logistics on a modular and temporal system in order to improve the distribution process, especially in urban zones. For example, the efficient management of last-mile deliveries has gained a critical role in urban logistics, which has been reinforced by the incorporation of ‘greener’ vehicles such as bikes, electric vehicles, and unmanned aerial vehicles (UAVs). These transportation means represent potential benefits in terms of delays, traffic congestion, and flexibility in city logistics (Ha et al. 2018). Figure 1 shows a simple example where a heavy vehicle brings the resources close to the urban zone. From there, resources are transferred to lighter vehicles that conduct the pick-up and delivery actions inside the urban area. The last-mile distribution is limited by the payload capacity and the driving range of these vehicles.

The use of UAVs in smart cities is still in an initial stage. However, this potential activity has raised the interest of many businesses due, in part, to the promise of quick responses to dynamic situations (Rao et al. 2016) and even door-to-door deliveries (Goodchild and Toy 2018). Besides, the monitoring of





**Main procedure of code BR-VNS for solving the sustainable MDVRP**

```
public class BR_VNS_SMDVRP {  
public BR_VNS_SMDVRP(Instance instance) {  
  this.instance = instance;  
  this.inputs = instance.getInput();  
  network = inputs.getNetwork();  
  this.parameters = instance.getParameters();  
  priorityList = new LinkedList<>();  
  this.rng = new Random(parameters.getSeed());  
  cache = new RouteCache();  
  this.hatami = true;  
  globalTime = new Time();  
  mappingTime = new Time();  
}  
public Solution[] solve() {  
  /*STAGE 0*/  
  globalTime.start();  
  calculatePriorityLists();  
  
  /*STAGE 1*/  
  Solution[] bestSols = generationOfTopPromisingCustomerDepotMaps()  
    ;  
  
  /*STAGE 2*/  
  intensiveRoutingOfTopSols(bestSols);  
  elapsedTime = globalTime.elapsedTime();  
  return bestSols;  
}
```

```

}
private Solution [] generationOfTopPromisingCustomerDepotMaps () {
mappingTime.start ();
int depots = network.getDepots ().size ();
int minK = (int) (depots * parameters.getMinPercent () /100.0);
minK = Math.max(minK,2);
int maxK = (int) (depots * parameters.getMaxPercent () /100.0);
maxK = Math.max(maxK,3);
maxK = Math.min(maxK, depots);
/* BRCWS*/
Solution baseSol = buildFeasibleSol(network);
if(baseSol == null) return new Solution [0]; //Return empty array,
    no solutions found
Solution bestSol = new Solution(baseSol);
BestSolutions bestSols = new BestSolutions ();
bestSols.update(baseSol);

/* VNS Parameters -Neighborhoods-*/
int k = minK;
double credit = 0.0;
it = 0;
int itK = 0;
int minItK = 1000;
System.out.println("Initial solution:" + baseSol.getCost());
while(mappingTime.elapsedTime () <= parameters.getTimeMax ()) {

    /* Diversification Operator*/
    Solution newSol = shakeDepot(k, baseSol);
    newSol.fastLocalSearch(inputs, cache, parameters);
    double delta = newSol.getCost () - baseSol.getCost ();
    if (delta < -EPSILON) { // newSol is better than baseSol
        credit = -delta;
        if (newSol.getCost () < bestSol.getCost ()) {
            bestSol = newSol;
            System.out.println("K: " + k);
            System.out.println("I improve: " + newSol
                .getCost () + " credit: " + credit);
        }
        newSol.setCpuTime(globalTime.elapsedTime ());
        baseSol = newSol;
        k = minK;
        bestSols.update(bestSol);
    }
}

```

```
/* Acceptance criterion */
else if (hatami && Math.abs(delta) > EPSILON) {
    if (delta <= credit) {
        System.out.println("K: " + k);
        System.out.println("criterion: " + newSol
            .getCost());
        System.out.println("Base sol: "+ baseSol.
            getCost());
        credit = 0.0;
        baseSol = newSol;
        k = minK;
    }
}
++itK;
/* Reset neighborhood */
if(itK > minItK){
    ++k;
    itK = 0;
}
/* Neighborhood increment */
k = k > maxK ? minK : k;
++it;
}
/* Sorted array of top sols*/
return bestSols.getSortedSols();
}
```



Exact model for solving the sustainable MDVRP (GAMS)

```
*Option Optcr=0.0000000000000001;
*option reslim = 20000;

Set i nodos /1*22/;
Set k vehicles /1*10/;
Set c customers /1*20/;
Set a depots /21*22/;

Alias (i,j);
Alias (i,l);
Alias (i,w);

Parameter N number of nodes;
N = card(i);

Parameter P number of vehicles;
P = 5;

Parameter B number of customers;
B = card(c);

Parameter F number of depots;
F = card(a);

Parameter Q capacity;
Q=60;

Scalar a_ij social cost;
a_ij=0.0004;

Scalar Ctime time cost;
Ctime=7.7;

Scalar Cf distance cost;
Cf=1.809;

Scalar Ce;
Ce=0.877;
```

```

Scalar kpl;
kpl=5.56;
Scalar M;
M=100000;
$onecho > datos.txt
par=Dist rng=DT!A1:w23 Cdim=1 Rdim=1
par=Time rng=DT!A26:w48 Cdim=1 Rdim=1
par=d rng=DT!a51:a73 Cdim=0 Rdim=1
par=V rng=DT!A76:B78 Cdim=0 Rdim=1

$offecho
$call gdxrw.exe p01medium.xlsx @datos.txt
Parameter
Time(i,j) time
Dist(i,j) distance
d(j) Demand
V(i)
$gdxin p01medium.gdx
$load Dist,Time, d, V
$gdxin

Variable x(i,j,k) 1 if point i immediately precedes point j on route k 0 otherwise
U(l,k) auxiliary vble sub-tour elimination constraints in route k
Cacum(i,j,k) load vehicle k after visit node i;

Binary variable x, rest_10_11;
Positive variable Cacum, U, TotalDemand;
Variable Z, Cost_time, Cost_dist, Cost_co2, Cost_social, Distance, Tiempo;
Equation DTtotal función objetivo;

```

$$DTotal.. Cost\_time+Cost\_dist+ Cost\_co2+ Cost\_social=e=Z;$$

Equation rest\_2 ;

$$rest\_2(j)\$(ord(j)\leq B)..sum((i,k)\$(ord(i)\leq N), x(i,j,k)) =e= 1;$$

Equation rest\_2b ;

$$rest\_2b(i)\$(ord(i)>B)..sum((j,k)\$(ord(j)\leq B), x(i,j,k)) =l= P;$$

Equation rest\_3 ;

$$rest\_3(k)..sum((i,j), d(j)* x(i,j,k)) =l= Q;$$

Equation rest\_4 ;

$$rest\_4(l,j,k)\$(ord(l)\leq B \text{ and } ord(j)\leq B)..U(l,k)-U(j,k)+B*x(l,j,k) =l= B-1;$$

Equation rest\_5 ;

$$rest\_5(k,i)\$(ord(i)\leq B)..sum((j)\$(ord(j)\leq N), x(i,j,k) - x(j,i,k)) =e= 0;$$

Equation rest\_6a ;

$$rest\_6a(i)\$(ord(i)>B)..sum((j,k)\$(ord(j)\leq B), x(i,j,k)) =g= 1;$$

Equation rest\_6b ;

$$rest\_6b(i,k)\$(ord(i)=21 \text{ and } ord(k)<6)..sum((j)\$(ord(j)\leq B), x(i,j,k)) =l= 1;$$

Equation rest\_6c ;

$$rest\_6c(i,k)\$(ord(i)=21 \text{ and } ord(k)>5)..sum((j)\$(ord(j)\leq B), x(i,j,k)) =e= 0;$$

Equation rest\_6c1 ;

$$rest\_6c1(j,k)\$(ord(j)=21 \text{ and } ord(k)>5)..sum((i)\$(ord(i)\leq B), x(i,j,k)) =e= 0;$$

Equation rest\_6d ;

$$rest\_6d(i,k)\$(ord(i)=22 \text{ and } ord(k)>5)..sum((j)\$(ord(j)\leq B), x(i,j,k)) =l= 1;$$

Equation rest\_6e ;

$$rest\_6e(i,k)\$(ord(i)=22 \text{ and } ord(k)<6)..sum((j)\$(ord(j)\leq B), x(i,j,k)) =e= 0;$$

Equation rest\_6e1 ;

$$\text{rest\_6e1}(j,k)\$(\text{ord}(j)=22 \text{ and } \text{ord}(k)<6)..\text{sum}(\{(i)\$(\text{ord}(i)\leq B), x(i,j,k)\}) =e= 0;$$

Equation rest\_6f ;

$$\text{rest\_6f}(i)\$(\text{ord}(i)>\{B\})..\text{sum}(\{(j,k)\$(\text{ord}(j)\leq B), x(i,j,k)\}) =e= \text{sum}(\{(j,k)\$(\text{ord}(j)\leq B), x(j,i,k)\});$$

Equation rest\_7 ;

$$\text{rest\_7}(i)\$(\text{ord}(i)>\{B\})..\text{sum}(\{(j,k)\$(\text{ord}(j)\leq B), d(j)\} * x(i,j,k)\}) =I= P * Q;$$

Equation rest\_9;

$$\text{rest\_9}(j,k)\$(\text{ord}(j)\leq B)..\text{sum}(\{(i)\$(\text{ord}(i)\leq N), \text{Cacum}(i,j,k)\}) - d(j) * \text{sum}(\{(i)\$(\text{ord}(i)\leq N), x(j,i,k)\}) =e= \text{sum}(\{(i)\$(\text{ord}(i)\leq N), \text{Cacum}(j,i,k)\});$$

Equation rest\_10;

$$\text{rest\_10}(i,j,k)\$(\text{ord}(i) \text{ ne } \text{ord}(j))..\text{Cacum}(i,j,k) =g= d(j) * x(i,j,k);$$

Equation rest\_11;

$$\text{rest\_11}(i,j,k)\$(\text{ord}(i) \text{ ne } \text{ord}(j))..\text{Cacum}(i,j,k) =I= (Q-d(i)) * x(i,j,k);$$

Equation rest\_14;

$$\text{rest\_14}(i,j,k)\$(\text{ord}(i)\leq B \text{ and } \text{ord}(j)>\{B\} \text{ and } \text{ord}(j)\leq N)..\text{Cacum}(i,j,k)=e=0;$$

Equation rest\_Distances Max distance per route;

$$\text{rest\_Distances}(k)..\text{sum}(\{(i,j)\$(\text{ord}(i)\leq N \text{ and } \text{ord}(j)\leq N), \text{Dist}(i,j)\} * x(i,j,k)\})=I=M;$$

Equation Costtime;

$$\text{Costtime}..\text{sum}(\{(i,j,k)\$(\text{ord}(i)\leq N \text{ and } \text{ord}(j)\leq N), \text{Ctime} * \text{Time}(i,j)\} * x(i,j,k)\}) =e= \text{Cost\_time};$$

Equation Costdist;

$$\text{Costdist}..\text{sum}(\{(i,j,k)\$(\text{ord}(i)\leq N \text{ and } \text{ord}(j)\leq N), \text{Cf} * (\text{Dist}(i,j)/\text{kpl})\} * x(i,j,k)\}) =e= \text{Cost\_dist};$$

Equation Costco2;

$$\text{Costco2}..\text{sum}(\{(i,j,k)\$(\text{ord}(i)\leq N \text{ and } \text{ord}(j)\leq N), \text{Ce} * (\text{Dist}(i,j)/\text{kpl})\} * x(i,j,k)\}) =e= \text{Cost\_co2};$$

Equation Costsocial;



```
Costsocial..sum((i,j,k){ord(i)<=N and ord(j)<=N), a_ij*Cacum(i,j,k)*Dist(i,j)) =e=
Cost_social;
```

Equation Distances;

```
Distances..sum((i,j,k){ord(i)<=N and ord(j)<=N), Dist(i,j)*x(i,j,k)) =e= Distance;
```

Equation tiempos;

```
tiempos..sum((i,j,k){ord(i)<=N and ord(j)<=N), Time(i,j)*x(i,j,k)) =e= Tiempo;
```

Equation calcTotalDemand;

```
calcTotalDemand..sum((j), d(j)) =e=TotalDemand;
```

```
model MDVRP /All/;
```

```
solve MDVRP using MIP minimizing Z;
```

```
display Dist, Time, d, V, Q, N, P, B, F, x.l, U.l, TotalDemand.l,Cacum.l, Cost_time.l, Cost_dist.l,
Cost_co2.l, Cost_social.l, Distance.l, Tiempo.l, Z.l;
```



### Main procedure of code SIM-BR-MS for solving the EVRPST

```

public SIM_BR_MS(Test myTest, Inputs myInputs, Random myRng) {
    aTest = myTest;
    inputs = myInputs;
    rng = myRng;
}

public Outputs solve() {
    /* Generates the CWS solution */
    long start = ElapsedTime.systemTime();
    baseSolution = BRCWS.solve(aTest, inputs, rng, false);
    baseSolution.setSafetyStockPercentage(aTest.getSafetyStock());
    double elapsed = ElapsedTime.calcElapsed(start, ElapsedTime.
        systemTime());
    baseSolution.setTime(elapsed);
    StochasticTravelTime.solve(aTest, inputs, baseSolution, aTest.
        getFirstMaxIter());
    bestSol = baseSolution;
    outputs.setbaseSolution(baseSolution);
    RouteCache cache = new RouteCache();
    SolutionSet ss = new SolutionSet(5);
    /* Iterates calls to BRCWS */
    start = ElapsedTime.systemTime();
    elapsed = 0.0;
    double bestCost = Double.POSITIVE_INFINITY;
    while (elapsed < aTest.getMaxTime()) {
        boolean update = false;

```

```

newSol = BRCWS.solve(aTest, inputs, rng, true);
newSol.setSafetyStockPercentage(aTest.getSafetyStock());
newSol = cache.improveRoutesUsingHashTable(newSol);
newSol.setTime(ElapsedTime.calcElapsed(start, ElapsedTime
    .systemTime()));
if (newSol.getCosts() < bestSol.getCosts()) {
    System.err.print("aggiorno det; ");
    bestSol = newSol;
    update = true;
}
elapsed = ElapsedTime.calcElapsed(start, ElapsedTime
    .systemTime());
/* MCS for each new solution */
newSol = StochasticTravelTime.solve(aTest, inputs, newSol
    , aTest.getFirstMaxIter());
newSol.setTime(ElapsedTime.calcElapsed(start, ElapsedTime
    .systemTime()));
ss.add(newSol);
if (newSol.getStochasticCost() < bestCost) {
    bestCost = newSol.getStochasticCost();
    System.err.print("aggiorno stoch");
    update = true;
}
if (update) {
    System.err.println();
}
baseSolution = StochasticTravelTime.solve(aTest, inputs,
    baseSolution, aTest.getSecondMaxIter());
bestSol = StochasticTravelTime.solve(aTest, inputs,
    bestSol, aTest.getSecondMaxIter());
double c = Double.POSITIVE_INFINITY;
for (Solution s : ss) {
    StochasticTravelTime.solve(aTest, inputs, s,
        aTest.getSecondMaxIter());
    if (s.getStochasticCost() < c) {
        solStochastic = s;
        c = s.getStochasticCost();
    }
}
outputs.setStochasticOBSol(solStochastic);
outputs.setbaseSolution(baseSolution);
outputs.setOBSol(bestSol);

```

```
printSolOnScreen(aTest, baseSolution, bestSol,
    solStochastic);
/* 6. Returns the best-found sol. */
return outputs;
}
```



### The diversification and intensification operator for the SIM-BR-ILS algorithm

```
public SIM_BR_ILS(Test myTest, Inputs myInputs, Random myRng) {
    aTest = myTest;
    inputs = myInputs;
    rng = myRng;
}

public static Solution DestructionConstruction(Solution
    baseSolution, Costs costList, ) {
    Sol.updateSolution(costList, aTest);
    Solution currentSol=new Solution(baseSolution);
    boolean balanced=false;
    int iter=0;
    double elapsed=0;
    while(iter<100){
        iter++;
        long start = ElapsedTime.systemTime();
        /* 3. Iterates calls to BRCW */
        currentSol.updateSolution(costList, aTest);
        Solution diffSol=new Solution(currentSol);
        float destPercentage= aTest.getmaxParamDest();
        double nRoutes=Math.round(destPercentage*Sol.getRoutes().
            size());
        if(nRoutes==0) {nRoutes=1;}
    }
}
```

```

diffSol=destructionConstructionR(nRoutes, rng, diffSol,
    costList, aTest, inputs);
diffSol.updateSolution(costList, aTest);
balanced=diffSol.checkBalanceConditions(inputs, costList)
;
elapsed = ElapsedTime.calcElapsed(start, ElapsedTime.
    systemTime());
//TO DO:
if(currentSol.getOptimizationCost()>diffSol.
    getOptimizationCost())
{currentSol=diffSol;}
int nodesSize= inputs.getNumNodes();
diffSol=improvementSolution(nodesSize, rng, new Solution (
    currentSol), costList, aTest, inputs, destPercentage);
diffSol.updateSolution(costList, aTest);
diffSol.LocalSearch(aTest, inputs, rng, costList, true);
balanced=diffSol.checkBalanceConditions(inputs, costList)
;
//TO DO:
//if(currentSol.getOptimizationCost()>diffSol.
    getOptimizationCost() && balanced==true) currentSol=
    diffSol;
if(currentSol.getOptimizationCost()>diffSol.
    getOptimizationCost()) currentSol=diffSol;
currentSol.updateSolution(costList, aTest);}
currentSol.setElapsedTime(elapsed);
return currentSol;}

public static Solution improvementSolution(double n, Random rng,
    Solution newSolBase, Costs costList,
    Test aTest, Inputs inputs, double pNodes) {
Solution newSol = new Solution (newSolBase);
int totalToRemove = (int) (pNodes * n);
LinkedList<Node> selectedNodes= removeFrom(newSol, rng,
    totalToRemove, costList, aTest, inputs);
List<Route> routes = new LinkedList<Route>(newSol.getRoutes());
Node depot = routes.get(0).getEdges().get(0).getOrigin();
for (Node node : selectedNodes) {
    int bestRoute = -1;
    int bestPos = -1;
    double bestDiff = Double.MAX_VALUE;
    int nRoute = 0;
    for (Route sr : routes) {

```



```

List<Node> subNodesOrder = extractNodes(sr);
Node last = depot;
int pos = 0;
if (sr.getDemand() + node.getDemand() <= inputs.
    getVehCap()) {
    for (Node curr : subNodesOrder) {
        double costIncrement = calcArc(
            last, node, curr, inputs,
            costList, aTest);
        if (costIncrement < bestDiff) {
            bestDiff = costIncrement;
            bestPos = pos;
            bestRoute = nRoute;}
        pos++;
        last = curr;}}
    nRoute++;}
if(bestRoute >= 0) {
    Route toInsert = routes.get(bestRoute);
    Route result = insertNodeInPosition(toInsert,
        node, bestPos,inputs, costList, aTest);
    routes.set(bestRoute, result);}
else {LinkedList<Node> nodes = new LinkedList<Node>();
nodes.add(node);
nodes.add(depot);
Route r = createRoute(nodes,inputs, costList, aTest);
routes.add(r);}

Solution reconstructed = new Solution();
addRoutes(reconstructed, routes, aTest, costList, inputs);
reconstructed.updateSolution(costList, aTest);
//TO DO
if(newSolBase.getOptimizationCost() < reconstructed.
    getOptimizationCost() && newSolBase.checkBalanceConditions(
    inputs, costList)) return newSolBase;
//if(newSolBase.getOptimizationCost() < reconstructed.
    getOptimizationCost() ) return newSolBase;
return reconstructed;}

private static Route createRoute(LinkedList<Node> nodes, Inputs
    input, Costs cost, Test test) {
Node last = nodes.getLast();
Route res = new Route();
for (Node n : nodes) {
    Edge edge = input.getEdge(last.getId(), n.getId());

```

```

        edge.setTime(input.getTime(last.getId(), n.getId()));
        edge.calcOptimizationCosts(cost, test);
        res.addEdge(edge);
        res.updateRoute(cost, test);
        last = n;}
return res;}

private static Solution destructionConstructionR( double nRoutes,
        Random rng, Solution diffSol, Costs costList, Test aTest,
        Inputs inputs) {
double routeToremove=nRoutes;
Solution subproblem= new Solution();
int extractedRoutes=0;
while(extractedRoutes<routeToremove){
    int randomRoute=(int)(Math.random()*diffSol.getRoutes().
        size());
    subproblem.getRoutes().add(diffSol.getRoutes().get(
        randomRoute));
    diffSol.getRoutes().remove(diffSol.getRoutes().get(
        randomRoute));
    extractedRoutes++;}
LinkedList<Node> nodesSubproblem= extractnodes(subproblem);
subproblem=BRCW.partialconstruction(aTest,inputs, nodesSubproblem
    ,rng, costList, true);
for(Route r:subproblem.getRoutes()){diffSol.getRoutes().add(r);}
diffSol.updateSolution(costList, aTest);
diffSol.LocalSearch(aTest, inputs, rng, costList, true);
return diffSol;}

```

**Main procedure of code SIM-BR-VNS for solving the TOP**

```

public SIM_BR_VNS(Test myTest, Inputs myInputs, Random myRng) {
    aTest = myTest;
    inputs = myInputs;
    rng = myRng;
}

public Object[] multiSto(){
    Solution newsol = null;
    Object[] BestSols = new Object[2];
    BestSolutions listBestSols = new BestSolutions(); //Best
        solutions customers violated
    BestSolutionsDiff listBestSolsDist = new BestSolutionsDiff(); //
        Best solutions distance violated
    long start = ElapsedTime.systemTime();
    double elapsed = 0.0;
    boolean firstime = true;
    initialSol = new Solution(solve());
    initialSol = Stochastic.simulate(initialSol, aTest.getShortSim(),
        aTest.getRandomStream(), aTest, inputs, -1, 0); //Fast simulation
    baseSol = initialSol;
    bestSol = initialSol;
    bestSolDist = initialSol;
    bestSolVio = initialSol;
    PairBestCust solToAdd = new PairBestCust(initialSol, initialSol.
        getPercentTimesViolated());
    listBestSols.addSolution(solToAdd);
}

```

```

PairBestDist solToAddDist = new PairBestDist(initialSol ,
    initialSol.getDistanceViolated());
listBestSolsDist.addSolution(solToAddDist);
while( elapsed < aTest.getMaxTime() )
{
    newSol = new Solution(solve());
    if(newSol.getTotalScore() > bestSol.getTotalScore()){
        bestSol = new Solution(newSol);
        newSol = Stochastic.simulate(newSol, aTest.
            getShortSim(), aTest.getRandomStream(), aTest,
            inputs, -1, 0); //Fast simulation
        /*Percentage customers violated*/
        if(newSol.getPercentTimesViolated() <= bestSolVio
            .getPercentTimesViolated()){
            bestSolVio = new Solution(newSol);
            bestSolVio.setTime(elapsed);
            solToAdd = new PairBestCust(bestSolVio,
                newSol.getPercentTimesViolated());
            listBestSols.addSolution(solToAdd);
            System.out.println("Mejoro SOL : " +
                bestSolVio.getTotalScore());
        }
        /*Percentage distance violated*/
        if(newSol.getPercentTimesViolated() <=
            bestSolDist.getPercentTimesViolated()){ //
            Percentage customers violated
            bestSolDist = new Solution(newSol);
            bestSolDist.setTime(elapsed);
            solToAddDist = new PairBestDist(
                bestSolDist, newSol.getDistanceViolated
                ());
            System.out.println("Mejoro SOL : " +
                bestSolDist.getDistanceViolated());
            listBestSolsDist.addSolution(solToAddDist
            );
        }
    }

    elapsed = ElapsedTime.calcElapsed(start, ElapsedTime.
        systemTime());
}
/*Final simulation (long simulation)*/
Iterator iterator = listBestSols.getSolutions().iterator();

```

```

int c = 0;
while(iterator.hasNext()){
    PairBestCust pairIter = (PairBestCust) iterator.next();
    Solution sol = pairIter.getKey();
    sol = Stochastic.simulate(sol , aTest.getLongSim(), aTest.
        getRandomStream(), aTest, inputs, 1, c);
    c++;
}
/*Best solutions violated distance*/
BestSolutionsDiff auxListBestSolsDist = new BestSolutionsDiff();
auxListBestSolsDist = listBestSolsDist;
listBestSolsDist = new BestSolutionsDiff();
iterator = auxListBestSolsDist.getSolutions().iterator();
c = 0;
while(iterator.hasNext()){
    PairBestDist pairIter = (PairBestDist) iterator.next();
    Solution sol = pairIter.getKey();
    sol = new Solution(Stochastic.simulate(sol , aTest.
        getLongSim(), aTest.getRandomStream(), aTest, inputs, 2, c
    ));
    PairBestDist newPairIter = new PairBestDist(sol , sol.
        getDistanceViolated());
    listBestSolsDist.addSolution(newPairIter);
}
/*test*/
iterator = listBestSolsDist.getSolutions().iterator();
while(iterator.hasNext()){
    PairBestDist pairIter = (PairBestDist) iterator.next();
    Solution sol = pairIter.getKey();
    System.out.println("Solucion distancia " + sol.
        getDistanceViolated());
}

/*fin test*/
BestSols[0] = listBestSols;
BestSols[1] = listBestSolsDist;
return BestSols;
}

```









## References

- Afsar, H. M. and Nacima, L. (2013). Team orienteering problem with decreasing profits. *Electronic Notes in Discrete Mathematics*, 41:285–293.
- Ahvenniemi, H., Huovila, A., Pinto-Seppä, I., and Airaksinen, M. (2017). What are the differences between sustainable and smart cities? *Cities*, 60:234–245.
- Alizadeh, T. (2017). An investigation of IBM’s Smarter Cites Challenge: What do participating cities want? *Cities*, 63:70–80.
- Angelelli, E., Archetti, C., Filippi, C., and Vindigni, M. (2017). The probabilistic orienteering problem. *Computers & Operations Research*, 81:269–281.
- Arvis, J.-F., Ojala, L., Wiederer, C., Shepherd, B., Raj, Anasuya; Dairabayeva, K., and Kiiski, T. (2018). *Connecting to Compete 2018*. World Bank. Available at: <https://elibrary.worldbank.org/doi/abs/10.1596/29971> (accessed: February 2019).
- Augerat, P., Belenguer, J. M., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1998). Computational results with a branch-and-cut code for the capacitated vehicle routing problem.
- Bashiri, M. et al. (2016). Green vehicle routing problem with safety and social concerns. *Journal of Optimization in Industrial Engineering*, 10(21):93–100.
- Bektaş, T., Ehmke, J. F., Psaraftis, H. N., and Puchinger, J. (2018). The role of operational research in green freight transportation. *European Journal of Operational Research*.

- Bektaş, T. and Laporte, G. (2011). The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250.
- Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287.
- Bibri, S. E. and Krogstie, J. (2017). Smart sustainable cities of the future: an extensive interdisciplinary literature review. *Sustainable Cities and Society*, 31:183–212.
- Bozorgi, A. M., Farasat, M., and Mahmoud, A. (2017). A time and energy efficient routing algorithm for electric vehicles based on historical driving data. *IEEE Transactions on Intelligent Vehicles*, 2(4):308–320.
- Bozorgi-Amiri, A., Jabalameli, M., and Al-e Hashem, S. M. (2013). A multi-objective robust stochastic programming model for disaster relief logistics under uncertainty. *OR Spectrum*, 35(4):905–933.
- Cabrera, G., Juan, A. A., Lázaro, D., Marquès, J. M., and Proskurnia, I. (2014). A simulation-optimization approach to deploy internet services in large-scale systems with user-provided resources. *Simulation*, 90(6):644–659.
- Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. A. (2015). Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):32.
- Calvet, L., Ferrer, A., Gomes, M. I., Juan, A. A., and Masip, D. (2016). Combining statistical learning with metaheuristics for the multi-depot vehicle routing problem with market segmentation. *Computers & Industrial Engineering*, 94:93–104.
- Chao, I.-M., Golden, B. L., and Wasil, E. (1993). A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions. *American Journal of Mathematical and Management Sciences*, 13(3-4):371–406.
- Chao, I.-M., Golden, B. L., and Wasil, E. A. (1996). The team orienteering problem. *European journal of operational research*, 88(3):464–474.
- Chen, L.-W., Sharma, P., and Tseng, Y.-C. (2013). Dynamic traffic control with fairness and throughput optimization using vehicular communications. *IEEE Journal on Selected Areas in Communications*, 31(9):504–512.
- Chibeles-Martins, N., Pinto-Varela, T., Barbosa-Póvoa, A. P., and Novais, A. Q. (2016). A multi-objective meta-heuristic approach for the design and planning of green supply chains-mbsa. *Expert Systems with Applications*, 47:71–84.

- Christofides, N., Mingozzi, A., and Toth, P. (1981). Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282.
- Clarke, G. and Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.
- Contardo, C. and Martinelli, R. (2014). A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146.
- Cordeau, J.-F. and Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033–2050.
- Dabia, S., Demir, E., and Van Woensel, T. (2014). An exact approach for the pollution-routing problem. *Beta Research School for Operations Management and Logistics*.
- Dang, D.-C., Guibadj, R. N., and Moukrim, A. (2013). An effective pso-inspired algorithm for the team orienteering problem. *European Journal of Operational Research*, 229(2):332–344.
- De Armas, J., Juan, A. A., Marquès, J. M., and Pedroso, J. P. (2017). Solving the deterministic and stochastic uncapacitated facility location problem: from a heuristic to a simheuristic. *Journal of the Operational Research Society*, 68(10):1161–1176.
- Delucchi, M. A. and McCubbin, D. R. (2010). External costs of transport in the US. *Institute of Transportation Studies*, pages 1–34.
- Demir, E., Bektaş, T., and Laporte, G. (2011). A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357.
- Demir, E., Bektaş, T., and Laporte, G. (2012). An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359.
- Demir, E., Huang, Y., Scholts, S., and Van Woensel, T. (2015). A selected review on the negative externalities of the freight transportation: modeling and pricing. *Transportation Research Part E: Logistics and Transportation Review*, 77:95–114.
- Desaulniers, G., Errico, F., Irnich, S., and Schneider, M. (2016). Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405.

- Dhingra, S. L., Rao, K., and Tom, V. (2003). Environmental impact assessment for sustainable transport. In *Handbook of Transport and the Environment*, pages 309–329. Emerald Group Publishing Limited.
- Dolinskaya, I., Shi, Z. E., and Smilowitz, K. (2018). Adaptive orienteering problem with stochastic travel times. *Transportation Research Part E: Logistics and Transportation Review*, 109:1–19.
- Dominguez, O., Guimarans, D., Juan, A. A., and de la Nuez, I. (2016a). A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, 255(2):442–462.
- Dominguez, O., Juan, A. A., Barrios, B., Faulin, J., and Agustin, A. (2016b). Using biased randomization for solving the two-dimensional loading vehicle routing problem with heterogeneous fleet. *Annals of Operations Research*, 236(2):383–404.
- Dominguez, O., Juan, A. A., and Faulin, J. (2014). A biased-randomized algorithm for the two-dimensional vehicle routing problem with and without item rotations. *International Transactions in Operational Research*, 21(3):375–398.
- Erdogan, G. and Laporte, G. (2013). The orienteering problem with variable profits. *Networks*, 61(2):104–116.
- Erdogan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.
- Escobar, J. W., Linfati, R., Toth, P., and Baldoquin, M. G. (2014). A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of Heuristics*, 20(5):483–509.
- Eshtehadi, R., Fathian, M., and Demir, E. (2017). Robust solutions to the pollution-routing problem with demand and travel time uncertainty. *Transportation Research Part D: Transport and Environment*, 51:351–363.
- European Commission (1999). MEET-Methodology for Calculating Transport Emissions and Energy Consumption. Available at: <https://trimis.ec.europa.eu/project/methodology-calculating-transport-emissions-and-energy-consumption> (accessed: March2019).
- European Commission (2015). *EU Transport in figures*. Statistical pocketbook (2015th ed.).

- European Transport Safety Council (2011). Tackling fatigue: EU social rules and heavy goods vehicle drivers. Technical Report October, European Transport Safety Council, Brussels. Available at:<http://www.trb.org/Main/Blurbs/166202.aspx> (accessed: February 2019).
- European Transport Safety Council (2017). Reducing deaths in single vehicle collisions. Technical Report April, European Transport Safety Council, Brussels. Available at:<http://etsc.eu/reducing-deaths-in-single-vehicle-collisions-pin-flash-32> (accessed: February 2019).
- Eurostat (2011). Roadmap to a single european transport area: Towards a competitive and resource efficient transport system. *White paper*. Available at: <http://eur-lex.europa.eu/legal-content> (accessed: February 2019).
- Eurostat (2017). Energy, transport and environment indicators. *Publications Office of the European Union, Luxembourg*. Available at: <https://ec.europa.eu/eurostat/web/products-statistical-books/-/KS-DK-17-001> (accessed: February 2019).
- Evers, L., Glorie, K., Van Der Ster, S., Barros, A. I., and Monsuur, H. (2014). A two-stage approach to the orienteering problem with stochastic weights. *Computers & Operations Research*, 43:248–260.
- Faulin, J., Grasman, S., Juan, A., and Hirsch, P. (2018). *Sustainable Transportation and Smart Logistics: Decision-Making Models and Solutions*. Elsevier.
- Faulin, J., Juan, A., Lera, F., and Grasman, S. (2011). Solving the capacitated vehicle routing problem with environmental criteria based on real estimations in road transportation: a case study. *Procedia-social and Behavioral Sciences*, 20:323–334.
- Felipe, Á., Ortuño, M. T., Righini, G., and Tirado, G. (2014). A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research Part E: Logistics and Transportation Review*, 71:111–128.
- Ferone, D., Gruler, A., Festa, P., and Juan, A. A. (2018). Enhancing and extending the classical grasp framework with biased randomisation and simulation. *Journal of the Operational Research Society*, pages 1–14.
- Fisher, M. L. and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124.

- Garrido, P. and Riff, M. C. (2010). Dvrp: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *Journal of Heuristics*, 16(6):795–834.
- Gendreau, M., Ghiani, G., and Guerriero, E. (2015). Time-dependent routing problems: a review. *Computers & Operations Research*, 64:189–197.
- Geurs, K. T., Boon, W., and Van Wee, B. (2009). Social impacts of transport: literature review and the state of the practice of transport appraisal in the netherlands and the united kingdom. *Transport Reviews*, 29(1):69–90.
- Gillett, B. E. and Miller, L. R. (1974). A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 22(2):340–349.
- Global Energy Prices (2018). Electricity prices around the world. last accessed: 22-May-2018.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549.
- Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318.
- Golden, B. L., Wasil, E. A., Kelly, J. P., and Chao, I.-M. (1998). The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In *Fleet Management and Logistics*, pages 33–56. Springer.
- Gonzalez-Neira, E. M., Ferone, D., Hatami, S., and Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 79:23–36.
- Gosavi, A. et al. (2015). *Simulation-based optimization*. Springer.
- Grasas, A., Juan, A. A., Faulin, J., de Armas, J., and Ramalhinho, H. (2017). Biased randomization of heuristics using skewed probability distributions: a survey and some applications. *Computers & Industrial Engineering*, 110:216–228.
- Grasas, A., Juan, A. A., and Lourenço, H. R. (2016). SimILS: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation*, 10(1):69–77.

- Guimarans, D., Dominguez, O., Panadero, J., and Juan, A. A. (2018). A simheuristic approach for the two-dimensional vehicle routing problem with stochastic travel times. *Simulation Modelling Practice and Theory*, 89:1–14.
- Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: a survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.
- Gunawan, A., Ng, K. M., Kendall, G., and Lai, J. (2018). An iterated local search algorithm for the team orienteering problem with variable profits. *Engineering Optimization*, 50(7):1148–1163.
- Hansen, P., Mladenović, N., Brimberg, J., and Pérez, J. A. M. (2019). Variable neighborhood search. In *Handbook of metaheuristics*, pages 57–97. Springer.
- Hansen, P., Mladenović, N., and Pérez, J. A. M. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407.
- Hatami, S., Calvet, L., Fernández-Viagas, V., Framiñán, J. M., and Juan, A. A. (2018). A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory*, 86:55–71.
- Hatami, S., Ruiz, R., and Andrés-Romano, C. (2015). Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. *International Journal of Production Economics*, 169:76–88.
- Hiermann, G., Puchinger, J., Ropke, S., and Hartl, R. F. (2016). The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018.
- Hof, J., Schneider, M., and Goeke, D. (2017). Solving the battery swap station location-routing problem with capacitated electric vehicles using an avns algorithm for vehicle-routing problems with intermediate stops. *Transportation Research Part B: Methodological*, 97:102–112.
- Holland, S. P., Mansur, E. T., Muller, N. Z., and Yates, A. J. (2015). Environmental benefits from driving electric vehicles? Technical report, National Bureau of Economic Research. Available at: <https://www.nber.org/papers/w21291> (accessed: February 2019).
- Ilhan, T., Iravani, S. M., and Daskin, M. S. (2008). The orienteering problem with stochastic profits. *IIE Transactions*, 40(4):406–421.

- International Electro-Technical Commissions (2014). Orchestrating infrastructure for sustainable smart cities executive summary. Technical report. Available at: <https://www.iec.ch/whitepaper/smartcities/> (accessed:February2019).
- Jabbarpour, M. R., Noor, R. M., and Khokhar, R. H. (2015). Green vehicle traffic routing system using ant-based algorithm. *Journal of Network and Computer Applications*, 58:294–308.
- Juan, A., Faulin, J., Grasman, S., Riera, D., Marull, J., and Mendez, C. (2011a). Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. *Transportation Research Part C: Emerging Technologies*, 19(5):751–765.
- Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., and Jorba, J. (2014a). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46:101–117.
- Juan, A. A., Faulin, J., Ferrer, A., Lourenço, H. R., and Barrios, B. (2013a). Mirha: multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems. *Top*, 21(1):109–132.
- Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., and Figueira, G. (2015a). A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62–72.
- Juan, A. A., Faulin, J., Jorba, J., Caceres, J., and Marquès, J. M. (2013b). Using parallel & distributed computing for real-time solving of vehicle routing problems with stochastic demands. *Annals of Operations Research*, 207(1):43–65.
- Juan, A. A., Faulin, J., Jorba, J., Riera, D., Masip, D., and Barrios, B. (2011b). On the use of monte carlo simulation, cache and splitting techniques to improve the clarke and wright savings heuristics. *Journal of the Operational Research Society*, 62(6):1085–1097.
- Juan, A. A., Faulin, J., Pérez-Bernabeu, E., and Jozefowicz, N. (2014b). Horizontal cooperation in vehicle routing problems with backhauling and environmental criteria. *Procedia-Social and Behavioral Sciences*, 111:1133–1141.
- Juan, A. A., Faulin, J., Ruiz, R., Barrios, B., and Caballé, S. (2010). The sr-gcws hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing*, 10(1):215–224.



- Juan, A. A., Goentzel, J., and Bektaş, T. (2014c). Routing fleets with multiple driving ranges: is it possible to use greener fleet configurations? *Applied Soft Computing*, 21:84–94.
- Juan, A. A., Grasman, S. E., Caceres-Cruz, J., and Bektaş, T. (2014d). A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory*, 46:40–52.
- Juan, A. A., Kelton, W. D., Currie, C. S., and Faulin, J. (2018). Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas. In *2018 Winter Simulation Conference (WSC)*, pages 3048–3059. IEEE.
- Juan, A. A., Mendez, C. A., Faulin, J., de Armas, J., and Grasman, S. E. (2016). Electric vehicles in logistics and transportation: a survey on emerging environmental, strategic, and operational challenges. *Energies*, 9(2):86.
- Juan, A. A., Pascual, I., Guimarans, D., and Barrios, B. (2015b). Combining biased randomization with iterated local search for solving the multidepot vehicle routing problem. *International Transactions in Operational Research*, 22(4):647–667.
- Kadziński, M., Tervonen, T., Tomczyk, M. K., and Dekker, R. (2017). Evaluation of multi-objective optimization approaches for solving green supply chain design problems. *Omega*, 68:168–184.
- Kiba-Janiak, M. (2016). Key success factors for city logistics from the perspective of various groups of stakeholders. *Transportation Research Procedia*, 12:557–569.
- Koc, C., Bektaş, T., Jabali, O., and Laporte, G. (2014). The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254.
- Koc, C. and Karaoglan, I. (2016). The green vehicle routing problem: a heuristic based exact solution approach. *Applied Soft Computing*, 39:154–164.
- Kramer, R., Subramanian, A., Vidal, T., and Lucídio dos Anjos, F. C. (2015). A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539.
- Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering*, 59(1):157–165.

- Lee, D.-Y., Thomas, V. M., and Brown, M. A. (2013). Electric urban delivery trucks: energy use, greenhouse gas emissions, and cost-effectiveness. *Environmental science & technology*, 47(14):8022–8030.
- Lei, C., Lin, W.-H., and Miao, L. (2014). A multicut l-shaped based algorithm to solve a stochastic programming model for the mobile facility routing and scheduling problem. *European Journal of Operational Research*, 238(3):699–710.
- Letaifa, S. B. (2015). How to strategize smart cities: revealing the smart model. *Journal of Business Research*, 68(7):1414–1419.
- Li, F., Golden, B., and Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165–1179.
- Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., and Lam, H. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, 41(4):1118–1138.
- Liu, W.-Y., Lin, C.-C., Chiu, C.-R., Tsao, Y.-S., and Wang, Q. (2014). Minimizing the carbon footprint for the time-dependent heterogeneous-fleet vehicle routing problem with alternative paths. *Sustainability*, 6(7):4658–4684.
- Lourenço, H. R., Martin, O. C., and Stützle, T. (2010). Iterated local search: framework and applications. In *Handbook of Metaheuristics*, pages 363–397. Springer.
- Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.
- Marcosig, E. P., Giribet, J. I., and Castro, R. (2017). Hybrid adaptive control for uav data collection: a simulation-based design to trade-off resources between stability and communication. In *2017 Winter Simulation Conference (WSC)*, pages 1704–1715. IEEE.
- Martin, S., Ouelhadj, D., Beullens, P., Ozcan, E., Juan, A. A., and Burke, E. K. (2016). A multi-agent based cooperative approach to scheduling and routing. *European Journal of Operational Research*, 254(1):169–178.
- Matl, P., Hartl, R. F., and Vidal, T. (2017). Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science*, 52(2):239–260.
- McKinnon, A., Browne, M., and Whiteing, A. (2015). *Green logistics: improving the environmental sustainability of logistics*. Kogan Page Publishers.

- 
- Mirabi, M., Ghomi, S. F., and Jolai, F. (2010). Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem. *Robotics and Computer-Integrated Manufacturing*, 26(6):564–569.
- Montoya-Torres, J. R., Muñoz-Villamizar, A., and Vega-Mejía, C. A. (2016). On the impact of collaborative strategies for goods delivery in city logistics. *Production Planning & Control*, 27(6):443–455.
- Muñoz-Villamizar, A., Montoya-Torres, J. R., and Faulin, J. (2017). Impact of the use of electric vehicles in collaborative urban transport networks: A case study. *Transportation Research Part D: Transport and Environment*, 50:40–54.
- Nash, C. (2003). Competitive and sustainable growth program. Technical report, UNITE Partner Organisations, Unite Kingdom.
- Navarro, C., Roca-Riu, M., Furió, S., and Estrada, M. (2016). Designing new models for energy efficiency in urban freight transport for smart cities and its application to the spanish case. *Transportation Research Procedia*, 12:314–324.
- Nealer, R., Reichmuth, D., and Anair, D. (2015). Cleaner cars from cradle to grave: How electric cars beat gasoline cars on lifetime global warming emissions. *Union of Concerned Scientists Report*.
- Niknamfar, A. H. and Niaki, S. T. A. (2016). Fair profit contract for a carrier collaboration framework in a green hub network under soft time-windows: dual lexicographic max–min approach. *Transportation Research Part E: Logistics and Transportation Review*, 91:129–151.
- Ntziachristos, L., Gkatzoflias, D., Kouridis, C., and Samaras, Z. (2011). COPERT: A European Road Transport Emission Inventory Model. *Environmental Science and Engineering*, 3:445–459.
- Panadero, J., de Armas, J., Currie, C. S., and Juan, A. A. (2017). A simheuristic approach for the stochastic team orienteering problem. In *2017 Winter Simulation Conference (WSC)*, pages 3208–3217. IEEE.
- Pecin, D., Pessoa, A., Poggi, M., and Uchoa, E. (2017). Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100.
- Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

- Poggi, M., Viana, H., and Uchoa, E. (2010). The team orienteering problem: formulations and branch-cut and price. In *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
- Ranaiefar, F. and Amelia, R. (2011). *Freight-transportation externalities*. Elsevier Inc.
- Ritzinger, U., Puchinger, J., and Hartl, R. F. (2016). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231.
- Royset, J. O. and Reber, D. N. (2009). Optimized routing of unmanned aerial systems for the interdiction of improvised explosive devices.
- Salhi, S. (2017). *Heuristic search: the emerging science of problem solving*. Springer.
- Santos, G., Behrendt, H., Maconi, L., Shirvani, T., and Teytelboym, A. (2010). Part I: Externalities and economic policies in road transport. *Research in Transportation Economics*, 28(1):2–45.
- Sawik, B., Faulin, J., and Pérez-Bernabeu, E. (2017a). A multicriteria analysis for the green vrp: a case discussion for the distribution problem of a spanish retailer. *Transportation Research Procedia*, 22:305–313.
- Sawik, B., Faulin, J., and Pérez-Bernabeu, E. (2017b). Selected multi-criteria green vehicle routing problems. In *Applications of Management Science*, pages 57–83. Emerald Publishing Limited.
- Schall, D. L. and Mohnen, A. (2015). Incentives for energy-efficient behavior at the workplace: a natural field experiment on eco-driving in a company fleet. *Energy Procedia*, 75:2626–2634.
- Scheuer, S. (2005). *EU environmental policy handbook. A critical analysis of EU environmental legislation: Making it accessible to environmentalists and decision makers*. European Environmental Bureau (EEB), Brussels.
- Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.
- Schneider, M., Stenger, A., and Hof, J. (2015). An adaptive VNS algorithm for vehicle routing problems with intermediate stops. *OR Spectrum*, 37(2):353–387.

- 
- Shao, S., Guan, W., and Bi, J. (2017). Electric vehicle-routing problem with charging demands and energy consumption. *Intelligent Transport Systems*, 12(3):202–212.
- Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). Lectures on stochastic programming: modeling and theory. *Society for Industrial and Applied Mathematics, Philadelphia*.
- Silva, B. N., Khan, M., and Han, K. (2018). Towards sustainable smart cities: a review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society*, 38:697–713.
- Smart Freight Transport Center (2017). Smart freight leadership - a journey to a more efficient and environmentally sustainable global freight sector. Technical report. Available at: <http://www.smartfreightcentre.org/sfl/smart-freight-leadership-publication> (accessed: February 2019).
- Soysal, M., Bloemhof-Ruwaard, J. M., and Bektaş, T. (2015). The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics*, 164:366–378.
- Subramanian, A., Uchoa, E., and Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10):2519–2531.
- Tajik, N., Tavakkoli-Moghaddam, R., Vahdani, B., and Mousavi, S. M. (2014). A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty. *Journal of Manufacturing Systems*, 33(2):277–286.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.
- Taniguchi, E., Noritake, M., Yamada, T., and Izumitani, T. (1999). Optimal size and location planning of public logistics terminals. *Transportation Research Part E: Logistics and Transportation Review*, 35(3):207–222.
- The European Parliament and the Council of the European Union (2013). Regulation (EU) No 1379/2013. Available at: <http://data.europa.eu/eli/reg/2013/1379/oj> (accessed: February 2019).
- The European Parliament and the Council of the European Union (2014). Regulation (EU) No 540/2014. Available at: <https://www.ojec.com/> (accessed: February 2019).
- Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

- Ubeda, S., Arcelus, F. J., and Faulin, J. (2011). Green logistics at eroski: a case study. *International Journal of Production Economics*, 131(1):44–51.
- Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.
- US. Department of transport (2016). Federal Automated Vehicles Policy. Technical report. Available at: <https://www.transportation.gov/AV/federal-automated-vehicles-policy-september-2016>(accessed: March2019).
- Varakantham, P., Kumar, A., Lau, H. C., and Yeoh, W. (2018). Risk-sensitive stochastic orienteering problems for trip optimization in urban environments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(3):24.
- Vega-Mejía, C. A., Montoya-Torres, J. R., and Islam, S. M. (2017). Consideration of triple bottom line objectives for sustainability in the optimization of vehicle routing and loading operations: a systematic literature review. *Annals of Operations Research*, pages 1–65.
- Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, 60(3):611–624.
- Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. *European Journal of Operational Research*, 231(1):1–21.
- Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research*, 234(3):658–673.
- Wang, F., Lai, X., and Shi, N. (2011). A multi-objective optimization for green supply chain network design. *Decision Support Systems*, 51(2):262–269.
- Wang, X., Fan, T., Li, W., Yu, R., Bullock, D., Wu, B., and Tremont, P. (2016). Speed variation during peak and off-peak hours on urban arterials in shanghai. *Transportation Research Part C: Emerging Technologies*, 67:84–94.
- Wang, X., Kopfer, H., and Gendreau, M. (2014). Operational transportation planning of freight forwarding companies in horizontal coalitions. *European Journal of Operational Research*, 237(3):1133–1141.

- 
- World Bank (2015). State and Trends of Carbon Pricing. Technical Report September, World Bank Group, Washington, DC.
- Xiao, Y. and Konak, A. (2015). Green vehicle routing problem with time-varying traffic congestion. In *Proceedings of the 14th Informs Computing Society Conference*, pages 134–148.
- Xie, K., Wang, X., Huang, H., and Chen, X. (2013). Corridor-level signalized intersection safety analysis in shanghai, china using bayesian hierarchical models. *Accident Analysis & Prevention*, 50:25–33.
- Zhang, J., Zhao, Y., Xue, W., and Li, J. (2015). Vehicle routing problem with fuel consumption and carbon emission. *International Journal of Production Economics*, 170:234–242.
- Zhang, S., Lee, C. K. M., Wu, K., and Choy, K. L. (2016). Multi-objective optimization for sustainable supply chain network design considering multiple distribution channels. *Expert Systems with Applications*, 65:87–99.