

FÁBIO NEVES SEABRA DA SILVA MOREIRA



On Improving Supply Chain Performance Through Integrated Vehicle Routing Problems

Submitted to Faculdade de Engenharia da Universidade do Porto in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Engineering and Management, supervised by Bernardo Almada Lobo, Associate Professor of Faculdade de Engenharia da Universidade do Porto and Luís Guimarães, Assistant Professor of Faculdade de Engenharia da Universidade do Porto

DEPARTMENT OF INDUSTRIAL ENGINEERING AND MANAGEMENT
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

2018

This research was partially supported by the PhD grant SFRH/BD/108251/2015 awarded by the Portuguese Foundation for Science and Technology.

“Things don’t have to change the world to be important.”

- Steve Jobs

Acknowledgments

I don't really know how to start something that will always fall short. Despite my exaggerating complaints about almost everything that exists in the world (maybe I forgot one or two things!), the time I spent during my PhD has been wonderful. When I arrived at I201, I was just an introverted kid who loved to play football and could implement some Excel macros. Well, I couldn't do much about the introverted kid... but I've learned a lot about myself, I've travelled a bit, I've met researchers from all around the world, I could teach, I learned things no one cares about, and most importantly (at least for me), I laughed a lot while doing this. I can hardly imagine myself without this puzzle piece that was built and given to me by a group of amazing people I am lucky to have in my life.

First and foremost, I must thank my supervisors.

To Bernardo Almada-Lobo. I would really like to know what are the odds of working with someone like Bernardo just to realize how lucky I am. I have heard stories about other supervisors so I have an idea of how special he is. In the end, he is not a supervisor. He is a life mentor who has contributed to my professional, academic, and personal development. One of the things I must appreciate in Bernardo is the joy and happiness he feels when he looks back and realizes the difference he makes in other people's development. Bernardo, I hope you can have that feeling when looking at my journey too. You always inspired me to be better and my evolution was so big that I will be forever indebted to you.

To Luís Guimarães, who is someone I should be calling brother. I know Luís since he was doing his PhD. During our meetings, I don't even need to talk sometimes, he knows what I am thinking (this is frighteningly true!). He has the gift to turn complex things into simpler ones, making my internal panic go away (I am having a hard time trying to describe this!). Luís, you know I've always been a fan of your work (I usually say you are the one who releases new versions of CPLEX because you know everything about it!) and my PhD would not be possible without your supervision. I thank you for all the meetings, all the reviewed documents, all the ideas, all the software, and all the techniques you shared with me. I may be quiet most of the time but today, I really want to thank you for being the person you are. For being the one who doesn't want to understand that it is impossible to help everyone. For being the one who was always there... and I really mean it. Always!

Secondly, I would like to express a word of appreciation and gratitude to Jean François Cordeau and Raf Jans. Being a visiting researcher at CIRRELT and GERAD (Montreal) for six months was one of the most enriching experiences of my life. When I enrolled in this adventure, I could never anticipate that I would be able to work with two of the best researchers in the transportation field. Furthermore, Jean François and Raf could always find time to fit our weekly meetings and have highly contributed to the quality and development of my PhD. I am really grateful for this once in a lifetime opportunity.

Thirdly, to the ones I have been laughing with during my PhD. Our research group is full of magnificent people which made part of my enriching journey. I have to thank Gonçalo for showing me that innocent, trustworthy, and friendly people still exist. I love the discussions we have on the future of Bitcoin, video games, pedagogy, and science. I would also like to express my gratitude to Professor Rocha for all the lessons about everything. Thanks to him, we take several mini PhDs every day. Additionally, a special thanks goes to the

members of “*Ilha 2*”: I have to thank Mário (Doutor MAL) for being “*mais do que um pai*”, teaching me everything I know about politics, economics, philosophy, and history. He even employed me! I wish to thank Sofia Gomes (Sofs Cruz) for founding Sofs Caffé where most of our conversations and laughs take place. Thanks to Maria João (Pires) for all the stupid gifts she sends from time to time, making me laugh even when I don’t want to. Finally, I wish to thank Álvaro, Eduardo, Cherri, Leandro, Xavier (Doutor Mestrando), Luís Dias (Éder), Pedro Amorim, Elsa, António, José Fernando, Maria Antónia, Miguel Gomes and so many others I can only list a few.

Fourthly, to the amazing people I met while I was a visiting researcher at CIRRELT. A big thanks to Luciano, Diego Pecin, Aurélien Froger, David Koza, Gislaine Melega, Laila Abdelhai, Serena Mantovani, Gianluca Morganti, Marcos Clavier, Bárbara, and Justyna Kulpa. You have been amazing while I was away from Porto. I will never forget how kind you were on Christmas, on New Year’s Eve, and in my birthday.

Fifthly, to my friends outside the research community. I would like to express a word of gratitude to Vítor Sousa, Pedro Carvalho, Tiago Cecílio, Rui Santos, Diogo Garcez, José Nuno, Juan Costa, Mafalda, João Alves (¯_(`´)_/_), and Diogo (Bam!). You may think you have not contributed to this thesis but that is not true. Without you, I would be living in *Magalhães Lemos* already! For now, I am just a neighbour.

Last but not the least, thank you Diana and Ricardo. You may be the ones who observed most of the events that happened to me during these years. There are some things impossible to understand in me but you’ve always been there! For everything. Without asking any questions. I am forever indebted to you for your patience, affection, and love. Nothing would be possible without you. Good to know that this is going to be reprinted. I don’t want you to see my tears on the paper. . . Thank you all.

Os melhores!

Abstract

The joint optimization of sequential activities in supply chain has been proven to yield large cost savings even when applied to simple real-world contexts. The technological advances in the available computational power turned integrated planning into a hot topic in recent years. Typical supply chains comprise production, inventory management, and transportation activities, which are planned at strategic, tactical and operational planning levels. In some, transportation costs assume a significant portion of the global supply chain cost. Notwithstanding, planning transportation at an operational level is a difficult task, as it maps into the classical Vehicle Routing Problem, considered to be one of the most challenging problems in Operational Research. Routing decisions impact and constrain the planning of surrounding supply chain processes. Nonetheless, it is yet to be found an established methodology for integrating routing decisions with interrelated supply chain decisions. Considering this, the present thesis tackles five different Integrated Vehicle Routing Problems (iVRPs), which belong to two research branches. In the first branch, dealing with the integration of supply chain processes at the operational level, we explore a pickup and delivery problem with synchronization, an inventory routing problem, and a production routing problem. In the second branch, dealing with the integration of different planning horizon levels, we explore the time window assignment vehicle routing problem and the consistent vehicle routing problem. The outcome of this research mitigates some research gaps and contributes to the field of iVRPs at three distinct levels: (1) new problems and novel mathematical formulations with increasing levels of realism; (2) new decomposition, exact, and approximation algorithms to efficiently explore the structure of several integrated routing problems which are validated within real-world contexts; (3) managerial insights which emerged from extensive sensitivity analysis to literature and real-world instances. Hence, our contributions are particularly aimed at bridging the gap between scientific research and practical application, adding value to the transportation field by fulfilling the needs of both the scientific community and the practitioners.

Resumo

O planeamento integrado de várias atividades da cadeia de abastecimento possibilita a obtenção de poupanças significativas, mesmo quando realizado em contextos simples do mundo real. Recentemente, com os avanços tecnológicos ao nível do poder computacional, o interesse pelo tema do planeamento integrado aumentou consideravelmente. Tipicamente, as cadeias de abastecimento incluem atividades de produção, gestão de inventário e transportes que são planeadas ao nível estratégico, tático e operacional. Em algumas cadeias de abastecimento, os custos de transporte assumem uma porção muito significativa do custo global. No entanto, o planeamento dos transportes ao nível operacional é uma árdua tarefa que obriga à resolução de um Problema de Roteamento de Veículos, considerado um dos mais complexos na área da Investigação Operacional. Numa cadeia de abastecimento, as decisões de transporte impactam e restringem todos os processos adjacentes. Contudo, ainda não existem metodologias para planear decisões de transporte simultaneamente com outros processos da cadeia de abastecimento, de forma integrada. A presente tese aborda cinco Problemas de Roteamento de Veículos Integrados (PRVis) cujos tipos de integração de desdobram em dois ramos.

No primeiro ramo, cujo objetivo é integrar processos da cadeia de abastecimento ao nível operacional, são explorados os problemas de recolha e entrega com sincronização, roteamento de inventários e roteamento da produção.

No segundo ramo, que visa a integração de níveis de planeamento, são explorados os problemas de alocação de janelas de entrega com roteamento de veículos e roteamento veículos com consistência no condutor e na hora de entrega.

O resultado deste trabalho de investigação mitiga algumas lacunas da literatura e contribui para a área relacionada com PRVis em três níveis distintos: (1) novos problemas e formulações matemáticas mais realistas; (2) novos algoritmos de decomposição, exatos e de aproximação para explorar a estrutura de vários problemas de roteamento integrado eficientemente, validados em contextos reais; (3) conselhos de gestão que emergem de análises de sensibilidade extensivas a instâncias da literatura e reais. Deste modo, as contribuições desta tese são particularmente profícuas para a aproximação entre a investigação científica e a aplicação prática, adicionando valor ao sector dos transportes, satisfazendo as necessidades tanto da comunidade científica como dos profissionais do sector.

Résumé

Il a été prouvé que l'optimisation conjointe des activités séquentielles dans les chaînes d'approvisionnement permet de réaliser des économies importantes, même lorsqu'elles sont appliquées aux contextes simples du monde réel. La planification intégrée est devenue un sujet populaire ces dernières années, grâce aux progrès technologiques dans la puissance informatique. Les chaînes d'approvisionnement typiques comprennent les activités de production, de gestion des stocks et de transport qui sont planifiées aux niveaux de la planification stratégique, tactique et opérationnelle. En général, les coûts de transport représentent une partie importante des coûts de la chaîne d'approvisionnement. Cependant, la planification du transport à un niveau opérationnel fait parti du problème classique de tournées de véhicules, qui est considéré comme l'un des problèmes les plus complexes de la recherche opérationnelle. Les décisions de transport limitent la planification des autres processus de la chaîne d'approvisionnement. Néanmoins, on n'a pas encore trouvé une méthodologie pour intégrer les décisions de transport aux décisions interdépendantes de la chaîne d'approvisionnement. Par suite, cette thèse aborde cinq problèmes différents de tournées de véhicules intégrés (PTVi) qui appartiennent à deux branches de recherche. La première branche traite l'intégration des processus de la chaîne d'approvisionnement au niveau opérationnel. Nous explorons un problème du ramassage et de la livraison avec la synchronisation, un problème de gestion de stocks et tournées de véhicules et un problème de production et tournées de véhicules. La deuxième branche traite l'intégration de différents niveaux d'horizon de planification. Ici, nous explorons le problème de tournées de véhicules avec attribution de fenêtres de temps et le problème de tournées de véhicules consistants. Le résultat de notre recherche atténue certaines lacunes dans la connaissance et contribue au domaine des PTVi à trois niveaux distincts : (1) nouveaux problèmes et nouvelles formulations mathématiques avec des niveaux de réalité croissant ; (2) nouveaux algorithmes de décomposition, exacts et d'approximation pour explorer efficacement la structure de plusieurs problèmes de tournées intégrés qui sont validés dans des contextes réels ; (3) les idées de gestion qui ont émergé d'une analyse de sensibilité approfondie à les instances de la littérature et du monde réel. Ainsi, nos contributions visent particulièrement à combler la lacune entre la recherche scientifique et l'application pratique, en ajoutant de la valeur au secteur des transports en répondant aux besoins des communautés scientifiques et les praticiens.

Contents

1	Motivation and overview	1
1.1	A Brief Review On Integrated Distribution Literature	2
1.2	Research Objectives and Methodology	5
1.3	The Collection Of Papers	9
1.3.1	Supply Chain Process Integration	9
1.3.2	Planning Horizon Level Integration	11
1.4	Final considerations	12
1.5	Further work	13
	Bibliography	16
2	Pickup and delivery with synchronization	19
2.1	Introduction	20
2.2	Literature Review	23
2.2.1	TTRP	24
2.2.2	PDP	26
2.2.3	VRPB	27
2.2.4	RRVRP	28
2.2.5	Full truckloads	29
2.2.6	Precedence constraints	30
2.2.7	Multiple time-windows	30
2.3	Problem Statement	31
2.3.1	Definition of entities	31
2.3.2	Pairing objective	33
2.3.3	Trip types	33
2.3.4	Transfer concept	34
2.3.5	Pre-processing	36
2.3.6	Long-haul example	37
2.4	Mathematical formulation	38
2.4.1	Decision variables	38
2.4.2	Objective function	38
2.4.3	Vehicle flow conservation	39
2.4.4	Travel times consistency	40
2.4.5	Requests flow coherence	40
2.4.6	Transfer times	42
2.4.7	Time windows	42
2.4.8	Practical constraints	42
2.5	Solution Approach: Matheuristic MH1	44
2.6	Computational Experiments	46
2.6.1	Real-world instances	46

2.6.2	Matheuristic settings	46
2.6.3	Solution approach analysis	46
2.7	Conclusions and Future Work	50
	Bibliography	51
2.A	Results Table	56
3	Inventory-Routing	59
3.1	Introduction	60
3.2	Literature Review	62
3.3	Problem Description and Mathematical Formulation	64
3.3.1	Valid Inequalities	67
3.4	Methodology	69
3.4.1	Branch-and-Cut scheme with deterministic demand	69
3.4.2	Rolling Horizon scheme with error prone forecasts	72
3.5	Computational experiments	74
3.5.1	Instance set and test conditions	74
3.6	Conclusions	82
	Bibliography	83
3.A	Summary of computational results for the literature instances	85
4	Production-Routing	89
4.1	Introduction	90
4.2	Literature Review	92
4.3	Problem Description and Formulation	95
4.3.1	Notation	95
4.3.2	Formulation	96
4.4	Solution Approach	98
4.4.1	Size Reduction	99
4.4.2	Initial Solution	102
4.4.3	Solution Improvement	106
4.5	Computational Experiments	112
4.6	Case Study	117
4.6.1	Analysis of the Current Situation	117
4.6.2	Methodology Application	120
4.6.3	Results Analysis	120
4.7	Conclusion	121
	Bibliography	122
4.A	Decomposition strategy base formulations	126
5	Routing for time window assignment	131
5.1	Introduction and Related Work	132
5.2	Problem Description and Mathematical Formulation	136
5.3	Solution Method	138
5.3.1	Route Generation	139

5.3.2	Initial Solution Construction	142
5.3.3	Improvement Matheuristic	143
5.4	Computational Experiments	145
5.4.1	Instance Set	146
5.4.2	Stochastic Optimization versus Average Demand Scenario	147
5.4.3	Operational Models and Change Levels	148
5.5	Conclusion	154
	Bibliography	156
5.A	Other Operational and Business Constraints	158
6	Enforcing driver and time consistency in vehicle routing	161
6.1	Introduction	162
6.2	Literature Review	164
6.2.1	Vehicle Routing Problem With Time Windows	164
6.2.2	Periodic Vehicle Routing Problem	165
6.2.3	Consistent Vehicle Routing Problem	166
6.3	Problem Statement and Mathematical Formulation	168
6.3.1	Mathematical formulation	169
6.4	Solution approach	172
6.4.1	Methodology overview	172
6.4.2	Node grouping	172
6.4.3	Initial solution construction	173
6.4.4	Fix-and-Optimize algorithm	175
6.4.5	Validation of the solution approach	178
6.5	Case study in a pharmaceutical distribution company	179
6.5.1	Case study overview	179
6.5.2	Route planning	180
6.5.3	Simulation model	181
6.6	Results	182
6.6.1	Instance characteristics	182
6.6.2	conVRPSLA results	183
6.6.3	Simulation results	184
6.7	Conclusions and future work	185
	Bibliography	187
6.A	Nomenclature	190
6.B	Comparison between the solver and the matheuristic	192

Motivation and overview

Globalization has fostered a more aggressive competition between companies around the world (Ghani et al. [2005]). The receptiveness for innovative and systematic approaches for supply chain optimization is increasing considerably, in an attempt to foster efficiency during the planning and execution phases. With the recent technological advances and the available computational power, new alternatives became available to fulfil the requirements of decision makers, considering broader sets of decisions simultaneously.

Supply chain activities are generally classified according to a matrix where the two dimensions account for the planning horizon and the supply chain process (Fleischmann et al. [2015]). The interdependency between adjacent planning levels and supply chain processes adds a huge complexity to the inherent planning phases. Sometimes, planners address this interdependency by taking an hierarchical approach. In terms of planning horizon, strategic problems are solved initially and then, the tactical and operational levels are addressed. In terms of supply chain process, problems are generally solved following the materials flow considering procurement, production, distribution and sales, though this is not mandatory. Hierarchical approaches, however, are prone to result in suboptimal solutions for the supply chain as a whole. The decisions made at each planning level in each supply chain process are highly dependent on the quality of the decisions made at other levels. Due to this fact, some integration between different dimensions is likely to be beneficial (Bektas [2017]).

In the foreword of Bektas [2017], Gilbert Laporte describes supply chain distribution processes as core to ensure the contemporary economic system and civilization. The author affirms that to maintain their competitiveness, it has become essential for companies to improve their distribution activities, employing advanced models and algorithms based on Operational Research (OR) techniques. These techniques are considered to be very successful for solving distribution-related problems. Laporte refers three factors to justify this success. First, mathematical programming often shows an unprecedented ability to capture complexity and intricacies of distribution problems. Second, the hardness of these problems makes them suitable for heuristic search. Third, the improvements found by optimization techniques are generally translated into considerable monetary savings and higher efficiency.

The scope of this research is framed by the integration of distribution supply chain processes, particularly at the operational level (i.e, transportation), with interdependent decisions. Hence, the classical Vehicle Routing Problem (VRP), which is at the very heart of this thesis, is integrated with other decisions related to production, inventory, and scheduling activities. This class of problems, combining the VRP with other optimization problems within the broader context of logistics, is known as Integrated Vehicle Routing Problems (iVRPs) (Bektas et al. [2015] and Côté et al. [2017]).

This thesis seeks to establish meaningful breakthroughs both for the scientific community and practitioners as well. First, we introduce new problems and propose novel mathematical formulations with increased levels of realism. Our aim is to address particular real-world transportation contexts, to capture the interest of researchers into new challenges, as well as companies in the transportation sector that may benefit from this research. Second, to efficiently solve the realistic formulations, we aspire to hybridize exact methods with approximation algorithms. We expect to achieve important findings in the recent matheuristics field, which is far from maturity with regards to applications in transportation problems. Matheuristics show an enormous potential for solving real-world integrated problems as they can be easily adapted to cope with model extensions and decomposed subproblems. All the formulations and matheuristics are validated using a case study and should be extensible to other companies operating in the transportation sector, facing similar problems. Third, new managerial insights are derived in each case study. We not only quantify the potential savings for integrating routing activities but also cover several planning aspects that might be faced by real-world companies by means of performing extensive sensitivity analysis on critical parameters.

The remainder of this chapter is composed of four sections. In Section 1.1 we review the literature on integrated vehicle routing. Section 1.2 poses the research questions to achieve our research objectives. Section 1.3 gives guidance to the reader about the organization and subjects covered in this thesis. In Section 1.4 we discuss the main scientific breakthroughs achieved. Finally, Section 1.5 provides some research directions.

1.1. A Brief Review On Integrated Distribution Literature

Recently, the research community has been very active on combining different classical optimization problems. The increasingly interest on these integrated problems has been mainly fostered by two reasons. Firstly, with the technological advances in processing units, more people have access to larger computing power, enabling them to solve more complex problems. Secondly, practitioners are becoming more and more interested on the potential savings achieved not only by OR techniques but also by the integration of supply chain activities. In particular, the vehicle routing community has been researching several classes of integrated problems. Since its seminal paper by Dantzig and Ramser [1959], the VRP has been extensively studied. The literature on vehicle routing has achieved a very mature state, exploring an immense number of extensions and objectives with exact and heuristic solution approaches. The remarkable development of this field, partially described by Laporte [2009], reached a point where the integration of supply chain processes and / or decision levels is attainable by combining state-of-the art solution approaches with efficient decomposition methods. In other words, solving larger than toy instances of iVRPs is now a reality and, as stated by Archetti and Speranza [2014], there is a growing trend towards modelling and solving such problems.

To the best of our knowledge, there is no standardized taxonomy to classify the different types of iVRPs. However, there are three main classes of iVRPs which have been very well studied by the vehicle routing community, under a widely accepted nomenclature:

the Location Routing Problem (LRP), the Inventory Routing Problem (IRP), and the Production Routing Problem (PRP). The LRP integrates two kinds of decisions that need to be made while designing distribution systems. The strategic facility location problem is solved while taking into account operational routing costs. As stated in the survey of [Prodhon and Prins \[2014\]](#), many studies show that if the interdependence between location and routing decisions is ignored, an excessive overall system cost may be incurred. Most approaches to the LRP are heuristics (e.g. [Harks et al. \[2013\]](#)) or lower bounds (e.g. [Albareda-Sambola et al. \[2005\]](#)). The proposed exact and matheuristic approaches are quite recent (e.g. [Belenquer et al. \[2011\]](#) and [Contardo et al. \[2014\]](#)). Additionally, the authors point out the quick growth of the number of articles on this topic, since the survey published by [Nagy and Salhi \[2007\]](#). Another class of problems that integrates distribution with other inter-related supply chain decisions is the IRP. These problems, which were introduced by [Bell et al. \[1983\]](#), integrate the operational inventory management and transportation decisions. Many solution approaches have been proposed, including heuristics (e.g. [Bertazzi et al. \[2002\]](#) and [Archetti and Bertazzi \[2012\]](#)) and exact methods (e.g. [Archetti et al. \[2007\]](#) and [Desaulniers et al. \[2016\]](#)). The number of papers on this subject has increased considerably in the recent years, as it is shown by [Coelho et al. \[2014\]](#). A natural extension to the IRP is to further consider the production-related supply chain processes. The PRP jointly optimizes production, inventory management and routing decisions. The production rate parameter that is used in IRPs is turned into a decision variable, adding a lot-sizing component to the problem. This idea was introduced in the work presented by [Chandra and Fisher \[1994\]](#). Since the PRP is more recent, its literature is not at the same state of maturity as the aforementioned integrated problems. Nonetheless, several heuristics (e.g. [Absi et al. \[2015\]](#) and [Adulyasak et al. \[2014a\]](#)) and exact methods (e.g. [Bard and Nananukul \[2010\]](#) and [Adulyasak et al. \[2014b\]](#)) have been recently proposed. For an overview on the recent works addressing the PRP, the reader is referred to the survey published by [Adulyasak et al. \[2015\]](#).

It can be noted that the aforementioned problem classes demand particular modelling techniques and solution approaches to be tackled. While the LRP integrates different planning horizon levels, strategic and operational, the IRP and the PRP focus on integrating different operational supply chain processes, comprising production planning, inventory management and routing, at the operational level.

In the LRP, decision makers need to take into account an extended time horizon and, in some way, capture the data patterns that may influence the strategic location decisions inherent to the problem. In location-routing decisions, it is common to have different time-period granularity for the location and routing parts. As stated by [Fleischmann and Koberstain \[2015\]](#), one way of integrating these patterns is to consider stochastic optimization with a set of representative scenarios spanning through a sufficiently long horizon and solve the operational routing problem for each scenario. Using the probability of occurrence of each scenario, such approach allows for the definition of the location decisions while computing the expected cost of performing the operational delivery schedules. Hence, the routing problems are solved in order to capture expected costs, guiding the strategic objectives of the LRP.

This is completely different from the type of approaches that are usually adopted in the case

of IRPs and PRPs. These problems focus on a single planning horizon level, operational, and seek to define inventory-routing and production-routing plans that are generally to be executed in the short term. Most publications regarding these two problems consider deterministic data and shorter planning horizons comprising few periods (usually days). Thus, the solution approaches that are generally applied to these operational problems consider information provided by the surrounding processes such as short term demand forecasts, capacity allocations and production rates. The granularity of the data and decision variables is similar across the production, inventory management and routing processes, considering each product, customer, vehicle and period separately. It is less common to use stochastic approaches as the uncertainty is not so severe (i.e. planning horizons are shorter) and re-planning can be an option when new information becomes available (i.e. after executing part of the operations). These information flows between execution and planning iterations can improve supply chain performance significantly (Fleischmann et al. [2015]).

Logically, these examples on how to deal with planning horizon level and supply chain process integration are just a myopic view over the entire iVRP literature. The recent trend towards integrated planning is fostering the introduction of new optimization problems, solution approaches and real-world applications. There is a large number of recent publications where new problems are proposed to integrate vehicle routing with several planning problems in particular. We review recent publications on iVRPs.

With respect to integrating strategic decisions with vehicle routing, the LRP is not the only one dealing with such issue. For instance, You et al. [2011] address the optimization of industrial gas distribution systems, integrating short-term distribution decisions for the vehicle routing with long-term inventory decisions for sizing storage tanks at customer locations. Three case studies considering instances with up to 200 customers are solved to show the performance of two fast computational strategies. This is a good example of a real-world application modelling iVRPs. However, the article focuses on the computational efficiency instead of providing insightful analysis on the underlying trade-offs of the problem. A relatively new extension to location problems is the Hub Location Problem (HLP) ([Farahani et al., 2013]) where hubs need to be located to efficiently consolidate, connect, and switch flows between predefined origins and destinations. Lopes et al. [2016] propose new heuristics for setting hub location-routing decisions, commonly considered in freight transportation and telecommunications. Three heuristics are implemented and tested comprising variable neighbourhood descent algorithms and a biased random-key genetic algorithm. The authors compare the proposed heuristics with a branch-and-cut algorithm and show that the former are efficient to solve medium to large instances.

In the tactical level, there is a large number of planning problems that need to be solved. These are generally solved in order to set some parameters that are taken into account in the subsequent activities. Dalmeijer and Spliet [2018] integrate the tactical assignment of time windows with the operational VRP. The authors propose a branch-and-cut algorithm which is considered to be the state-of-the-art among other exact approaches. Several valid inequalities and precedence inequalities are proposed to strengthen the model. The approach is able to solve small instances comprising 35 customers. However, the authors do not provide any insights on the structure of the integrated problem. Enderer et al. [2017] integrate the operation decisions of assigning trucks to dock-doors and defining vehicle

routes. The objective is to jointly minimize the total material handling and transportation costs. The authors propose a column generation algorithm and a heuristic approach to solve small instances of the problem, showing that significant savings can be achieved even when one of the costs dominates the other. [Dong et al. \[2017\]](#) combine inventory management of cargos with tactical maritime fleet deployment. A time-continuous formulation is proposed and computational experiments are performed on a set of realistically generated test instances. However, the size of instances that are tackled is too small. The authors point some research directions including decomposition methods, new valid inequalities, and advanced heuristics to obtain near optimal solutions for larger instances.

Regarding supply chain process integration at the operational level, [Darvish et al. \[2017\]](#) explore the existing trade-offs between economically optimized versus environmentally friendly solutions in the context of IRPs and PRPs. The authors contribute to the literature by modelling and solving emission minimization integrated distribution problems (with randomly generated instances), optimizing the total cost, distance, and emissions. A counter-intuitive insight states that it is interesting to decrease the average vehicle load during the route to reduce emissions, even if the travelled distance increases. In a working paper on the IRP with pickups and deliveries, [Archetti et al. \[2017\]](#) propose a branch-and-cut algorithm to solve instances considering a single vehicle serving up to 50 customers over 3 periods. The study reveals that the average cost of a non-integrated policy, with decoupled routing and inventory decisions, can be 35% higher than the cost of an integrated one. The authors show that more complex problems can now be tackled by modern technologies and methodologies. [Côté et al. \[2017\]](#) study the value of integrating loading and routing. The authors show that it is worthwhile to jointly optimize strongly interdependent problems even when the monolithic model cannot be solved by exact methods. Indeed, the error generated by heuristic approaches while solving the integrated problem, remains smaller than the cost increase of non-integrated approaches.

Analysing these recent publication and technical reports, it seems that the iVRP literature still lacks publications considering realistic models with real-world applications. The great majority of the papers deals with random generated instances and focuses on analysing the algorithmic efficiency of the integrated approach. We consider that integration is one of the most challenging objectives when it comes to planning operations that are performed without any optimization software support. Therefore, despite the scientific value of such challenge, there is a clear opportunity to make an impact on large organizations by proposing new models, algorithms, and managerial insights regarding real-world iVRPs.

1.2. Research Objectives and Methodology

This research intends to contribute with new insights into the resolution of iVRPs, particularly in real-world contexts. We address a set of integrated problems and propose new models and algorithms, evaluating their efficiency and proneness to real-world applications. Supply chain integration can be achieved over two dimensions: (1) integrating supply chain processes or (2) integrating planning horizon levels. Since they pose distinct difficulties in terms of modelling techniques, this research covers integrated routing problems which fall

into these two dimensions.

Since the vehicle routing problem is NP-hard (Laporte [2009]), the class of problems addressed in this thesis is computationally intensive to solve. Therefore, to tackle the proposed formulations and realistic instances, our solution approaches focus on two aspects. First, the vehicle routing hardness needs to be tackled efficiently as it will be present in all addressed problems. Second, since we address a very particular set of realistic problems, our solution approaches need to be sufficiently flexible and adaptive to several formulations and decomposition strategies. By incorporating exact and heuristic methods, we are able to provide efficient solutions to a set of real-world cases, providing valuable managerial insights. We find the integration of scientific breakthroughs within companies to be fundamental. However, a significant gap between theory and practice, regarding decision support for integrated supply chain planning, is yet to be bridged. Firstly, most approaches do not point out the challenges posed by different types of integration. Secondly, the great majority of iVRP approaches focus on algorithmic issues but not on solving problems regardless of their size (i.e. through efficient decompositions). Thirdly, it is rare to find literature providing useful managerial insights while solving real-world instances.

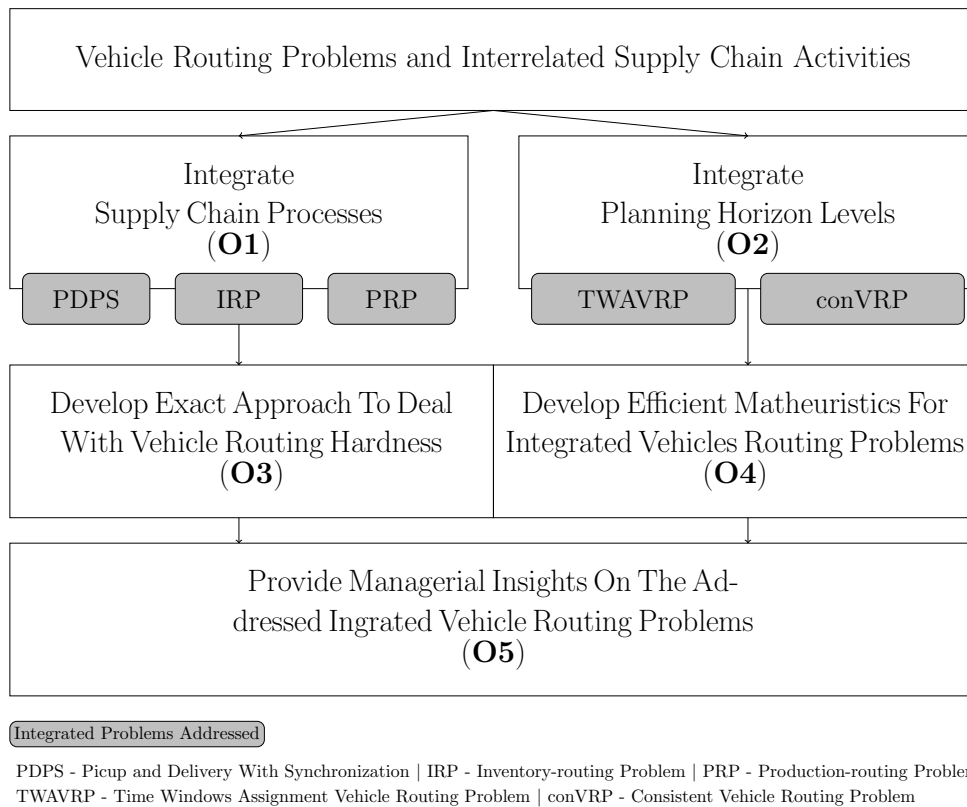


Figure 1.1 – Research canvas and research objectives

Figure 1.1 presents a visual representation of the research objectives of this thesis and their relationship. To achieve them, five research questions provide the underlying guidelines.

Research Questions For Research Objective 1

How to comprehensively address realistic supply chain process integration?

To answer this research question we propose a set of mathematical formulations which model more than one supply chain process. Since our focus is on transportation-related processes, we integrate, extend and propose new models for integrating vehicle routing with other operational decisions. The models cover three different problems comprising pickup and delivery with request synchronization, inventory-routing, and production-routing decisions, and account for different realistic features, such as new operational models, lateral transshipments, and perishability control.

Research Questions For Research Objective 2

How to comprehensively address realistic planning horizon level integration?

This research question is answered by exploring the integration of different planning horizon levels with the operational routing decisions. In this case, we cover two iVRPs. The operational routing decisions are integrated with the tactical time window and driver assignment. Since the planning horizon of both problems is not the same, demand uncertainty is one of the issues that needs to be addressed by the new models. We expect to draw some conclusions regarding transportation planning with non-deterministic demand, providing guidelines for the reader to model, solve, and extract insightful conclusions while integrating different planning horizon levels in vehicle routing problems.

Research Questions For Research Objective 3

What are the challenges posed by integrated routing problems when devising exact approaches? Which instance size can we expect to solve by such approaches?

We seek to understand the challenges of combining vehicle routing problems with other optimization problems while developing exact solution methods. The main solution techniques that are successful in the classical vehicle routing problems are explored in the context of integrated vehicle routing problems. We devise an exact solution approach, embedding several acceleration techniques, expecting to understand if their applicability to integrated problems holds. Furthermore, it is expected that the output of this objective will be useful in the subsequent phases of our research.

Research Question For Research Objective 4

Are innovative hybrid solution methods, namely matheuristics, suitable to solve mathematical formulations related to integrated vehicle routing problems? Where do they stand in comparison to purely exact and heuristic methods?

The attainment of the third research objective is core for pursuing our fourth research objective. Using the developed exact approach to deal with vehicle routing efficiently, we seek to materialize the outcome of this research objective by proposing a set of hybrid solution methods where a set of subproblems is solved to optimality. The idea is to take advantage of the huge flexibility provided by matheuristics to deliver good results in reasonable time (considering the needs imposed in a business context). We aim at contributing to the liter-

ature by showing that several business-based decomposition strategies enable the efficient application of matheuristics to a considerable set of iVRPs.

Research Question For Research Objective 5

What are the main challenges and insights obtained by addressing integrated routing problems in real-world contexts? In scientific and practical terms, what are the impacts of such integration?

All the aforementioned objectives are fundamental to achieve the last research objective of this thesis. We seek to analyse the results provided by the developed models and algorithms to derive enlightening conclusions related to the structure of integrated routing problems as well as valuable managerial insights. First, we expect to come up with general guidelines for dealing with the computational challenges of integrated routing problems. Second, we seek to quantify the potential savings coming from the integration of several supply chain processes as well as different planning horizon decisions. We deal with four real-world integrated routing problems, proving the applicability of our approaches in an attempt to bridge the literature gap related to real-world applications.

Table 1.1 summarizes the work developed in this thesis, revealing the main differences between each chapter. We indicate the supply chain processes and planning horizon levels that are jointly optimized in each chapter, as well as the type of solution approach devised to solve the integrated vehicle routing formulations that serve as a basis for each one. Furthermore, we provide a brief summary of the terms that are considered in each objective function and the businesses in which the provided managerial insights are validated. Finally, we indicate the chapters that are more connected to the answers provided to each research question in the final considerations of this thesis.

Table 1.1 – Chapter summary and relation to each research question.

Chapter	Supply Chain Process	Planning Horizon Level	Solution Approach	Objective Function	Managerial Insights	Research Questions				
						Q1	Q2	Q3	Q4	Q5
2 (PDP)	Transportation + Inventory Synchronization	Operational	MH	Maximize Request Movements	Third Party Logistics Operator	✓			✓	✓
3 (IRP)	Transportation + Inventory Management	Operational	E	Minimize Inventory And Transportation Cost	Theoretical Study On Transshipments	✓		✓		
4 (PRP)	Production + Inventory Management + Transportation	Operational	MH	Minimize Production Inventory And Transportation Cost	Meat Store Chain	✓			✓	✓
5 (TWAVRP)	Time Window Assignment + Transportation	Tactical + Operational	MH	Minimize Fleet And Transportation Cost	Food Retail Operations		✓		✓	✓
6 (conVRP)	Time Window And Driver Assignment + Transportation	Tactical + Operational	MH	Minimize Penalty And Transportation Cost	Pharmaceutical Industry		✓		✓	✓

Legend: MH - Matheuristic | E - Exact Approach | PDP - Pickup and Delivery Problem | IRP - Inventory Routing Problem
 PRP - Production Routing Problem | TWAVRP - Time Window Assignment Vehicle Routing Problem
 conVRP - Consistent Vehicle Routing Problem

1.3. The Collection Of Papers

We have chosen to elaborate this thesis as a collection of papers either already published or under review in international peer-reviewed journals instead of a monograph. Hence, the collection of papers targets the research objectives defined in the previous section. In Figure 1.2 we present the supply chain matrix presented by Fleischmann et al. [2015]. We explore the integration of transportation problems can be with other supply chain processes (horizontally) and other planning horizon levels (vertically).

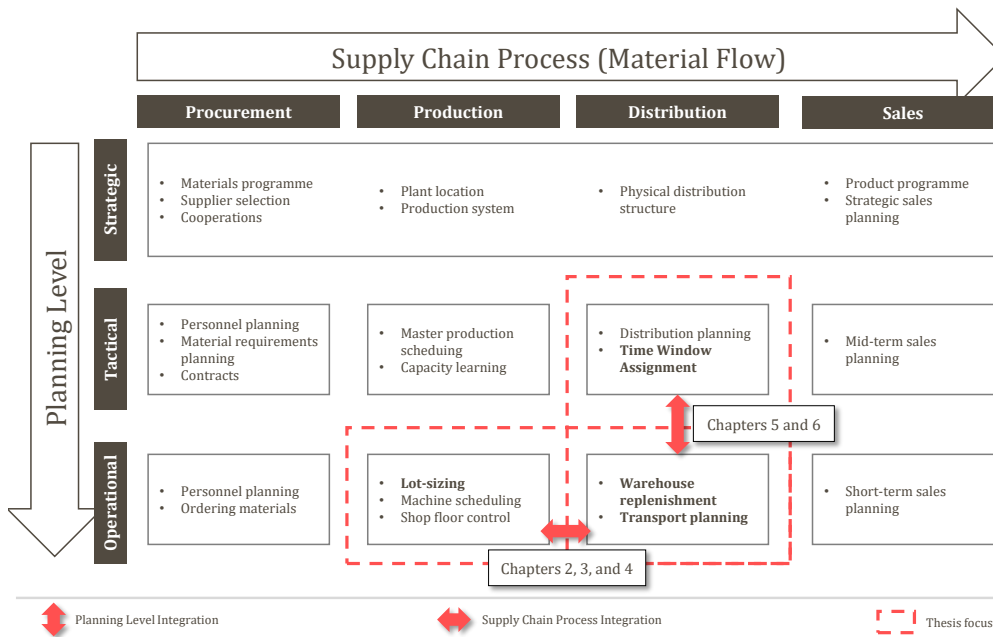


Figure 1.2 – Thesis focus represented in the supply chain matrix adapted from Fleischmann et al. [2015].

Chapters 2, 3, and 4 are aligned with the objectives pursued for integrating routing with other supply chain processes. Chapters 5 and 6 follow the research objectives related to the integration of planning horizon levels. In the remainder of this section we briefly point out the main subjects and contributions presented in each of the papers. Additionally, we indicate the references where this research was originally published or submitted to.

1.3.1 Supply Chain Process Integration

The first three papers in this thesis (Chapters 2, 3, and 4) focus on integrating routing problems with other production and distribution supply chain processes.

1.3.1.1 Pickup and delivery with synchronization

Chapter 2 introduces a new optimization problem which is based on a very unique operational model which is designed to reduce empty trips of a third-party logistics operator. In

this problem, truck semi-trailers need to be moved passing through multiple transshipment locations. Hence, inventories of semi-trailers need to be synchronized through several depots, resulting in a problem comprising routing, inventory, and scheduling aspects. Three main contributions are added to the literature. Firstly, we provide an overview over different parts of the VRP literature. As the paper unfolds, it becomes clear that there is a need for a set of features coming from a broad set of problems. In some cases, the similarities may only be physically noticeable whereas in other cases there are similarities only in the methodologies and modelling techniques used. Secondly, a new operational model for the pickup and delivery problem is presented as a new glance at the possibilities of using transshipment locations to support distribution. Considering that the use of transshipment locations is part of a new logistics paradigm, this work surely presents a fresh approach to vehicle routing and scheduling problems. Thirdly, we contribute to the matheuristics field by proposing a solution approach which outperforms a general-purpose solver, obtaining superior solutions in shorter time. We further validate the approach in a real-world case study and we understood, however, that inventory management should be considered explicitly in the models. One of main innovative features of the proposed model is in the objective function, which attempts to maximize the number of semi-trailers flowing through the network. The reference that serves as basis for this chapter is:

- Fábio Neves-Moreira, Pedro Amorim, Luís Guimarães, and Bernardo Almada-Lobo. A long-haul freight transportation problem: Synchronizing resources to deliver requests passing through multiple transshipment locations. *European Journal of Operational Research*, 248(2):487 – 506, 2016.

1.3.1.2 Inventory-Routing

Chapter 3 explicitly integrates the VRP with inventory management. To explore the value of owned fleet lateral transshipments in the context of the IRP, the paper associated with this chapter provides four main scientific contributions. We first propose a novel IRP formulation considering owned fleet lateral transshipments. This realistic feature is used in business contexts where inventories can be reallocated between retail sites, after some disruptive events introduce unbalances in the inventory management activity. The second contribution is an efficient exact branch-and-cut algorithm which focuses on patching infeasible routing solutions to accelerate the branching process. In the third contribution, we perform a sensitivity analysis over a set of parameters that are crucial for real-world applications of the IRP. We provide results for extensive computational experiments where different operational models are tested with different planning horizons and forecast accuracies. Our fourth and last contribution is translated into new managerial insights comparing the standard IRP with the extended problem, where lateral transshipments are an option. The reference that serves as basis for this chapter is:

- Fábio Neves-Moreira, Luís Guimarães, Bernardo Almada-Lobo. Owned fleet lateral transshipments in the inventory routing problem. *Computers & Operations Research* (Submitted)

1.3.1.3 Production-Routing

Chapter 4 explicitly integrates production, inventory management, and vehicle routing decisions. We tackle a large multi-product PRP with delivery time windows using a matheuristic approach based on the exact approach developed in the third chapter. The main contributions of this chapter are fourfold. First, we propose a novel mathematical formulation for the PRP increasing the realism of the standard models. The model provides solutions for problems considering multiple perishable products, production lines with different specifications and delivery time windows. Second, to solve the proposed model, we present a decomposition approach which divides the PRP into several IRPs and one Capacitated Lot Sizing Problem (CLSP) reducing the size of the original problem. Third, to explore the solution space of large PRPs which are intractable for the best available commercial solvers, we propose an improvement matheuristic. The solution approach proves to be competitive while solving literature instances of the IRP and PRP. In the last contribution, we apply our solution approach to a real-world context, and provide valuable managerial insights using data of a meat store chain. The reference that serves as basis for this chapter is:

- Fábio Neves-Moreira, Bernardo Almada-Lobo, Jean-François Cordeau, Luís Guimarães, Raf Jans. Solving a large multi-product production routing problem with time windows. *Omega* (Submitted)

1.3.2 Planning Horizon Level Integration

The last papers in this thesis (Chapters 5 and 6) take advantage of the approaches developed in the first part of the thesis, to integrate the tactical and operational levels of distribution.

1.3.2.1 Time window assignment based on vehicle routing

In Chapter 5, the vehicle routing problem is jointly optimized with the tactical time window assignment. The idea is to develop a systematic approach for redefining time windows based on delivery schedules that combine pre-generated routes. The main contributions of this paper are the following. First, we propose a novel set-partitioning formulation for the time window assignment vehicle routing problem considering multiple product segments. This formulation is guided by a new objective function which accounts for the cost of fleet requirements. Second, to deal with real-world instances, we propose a matheuristic using business related decomposition strategies for accelerating its convergence. Third, we present extensive computational experiments, testing an average demand scenario approach against our stochastic optimization approach. Three different operational models with different time window settings are compared in terms of potential savings. Furthermore, a sensitivity analysis is performed on a baseline solution provided by a European food retailer to explore the trade-off between implementation effort (number of time windows changed) and total cost. Fourth, we provide business related insights as well as real-world considerations that have been revealed during the development and implementation of our solution approach. The reference that serves as basis for this chapter is:

- Fábio Neves-Moreira, Diogo Pereira da Silva, Luís Guimarães, Pedro Amorim, Bernardo Almada-Lobo. The Time Window Assignment Vehicle Routing Problem With Product Dependent Deliveries. *Transportation Research Part E: Logistics and Transportation Review* (Submitted)

1.3.2.2 Enforcing driver and time consistency in vehicle routing

In the last chapter of this thesis, Chapter 6, we deal with the consistent VRP. We integrate the operational vehicle routing problem with two tactical decisions: time windows and driver assignment. The contributions provided by the paper in the sixth chapter are twofold. Firstly, a new problem is introduced and a formulation is proposed. We extend the consistent VRP by considering service level agreements which induce a different number of visits to each customer in each period of the day. Secondly, we devise a solution approach which uses a Consistent Vehicle Routing Problem (conVRP) formulation to effectively address a real-world problem. Furthermore, the proposed mathematical programming based solution approach is validated with historical data for planning the consistent routes of a pharmaceutical distribution company with over 3.000 daily deliveries. This environment comprises both expected and unexpected customers with uncertain demand. The proposed plans are then simulated and their performance is compared to real-world plans of the case study. The reference that serves as basis for this chapter is:

- Pedro Campelo, Fábio Neves-Moreira, Pedro Amorim, Bernardo Almada-Lobo. Consistent Vehicle Routing Problem with Service Level Agreements *European Journal of Operational Research* (Under Review)

1.4. Final considerations

Integrated vehicle routing is still in its infancy. Although there are some well studied classes of problems, such as location, inventory, and production routing, other classes of problems still have to be explored. This thesis presents five iVRPs and proposes new extended mathematical formulations to model them. Given the complexity of iVRPs, it is very unusual for commercial solvers to solve them out of the box. Therefore, the models are tackled by enhanced exact and hybrid approaches. The latter, are proven to be highly successful in dealing with four real-world applications comprising a third party logistics provider, a food retailer, a meat store chain, and a pharmaceutical distributor. We largely contribute to the recent field of matheuristics, pointing out their enormous flexibility to deal both with computational and business complexity. Furthermore, matheuristics detain three main characteristics we felt to be fundamental for real-world implementation of OR techniques: (1) their mathematical programming base allows for fast integration of new features that are constantly popping up in real contexts; (2) matheuristics' subproblems are extremely easy to tune and flexible enough to undertake business tailored decomposition strategies; (3) although they are not as fast as pure metaheuristics, they usually obtain high quality solutions in reasonable computational times.

The results provided by our solution approaches are able to show-off the power of OR while solving complex logistics problems in the real world. Studying VRPs is extremely important for the evolution of supply chain planning. Most companies still rely on manual and hierarchical planning processes that are very time consuming, very error prone, and generally poor in terms of solution quality. Although planners are usually quite good while dealing with separate problems, supply chain process and planning level integration are extremely difficult to be handled manually. Indeed, integrating decisions comprising various activities with continuous and binary decisions rapidly entraps manual planners in local optima. Therefore, we consider that systematic approaches that model different supply chain processes and planning levels, capturing their interaction, are a requisite for competitive supply chain management.

Furthermore, we recognize that most of the insights presented in this thesis would be very hard (if possible) to be found without proposing systematic and flexible approaches based on our enhanced OR techniques. A very unique set of iVRPs is tackled using both literature and real-world instances. By doing so, we were able to solve different scenarios and test new operational models within the context of the iVRP. This research allowed us to address new logistics paradigms and unveil important findings regarding the structure of a wide variety of challenges. Regarding the Pickup and Delivery Problem (PDP), we show how a new operational model can use short-haul trips to provide long-haul services while reducing truck empty trips. In the context of the IRP, we explore a new operational model using lateral transshipments and show that lateral transshipments can be a useful hedging mechanism in uncertain contexts when holding costs are high. We tackled a large PRP and showed the impact of considering Vendor Managed Inventory (VMI) and different demand visibilities in a real-world case study of a meat store chain. We integrated the tactical time window assignment with vehicle routing, modelling the Time Window Assignment Vehicle Routing Problem (TWAVRP) as a two-stage stochastic optimization problem. We quantify the savings obtained by considering stochastic optimization against using average demands showing that a cost reduction of 5.3% can be achieved in a multiple product segment context. Furthermore, based on the case of a large European food retailer, we provided managerial insights for three operational models which test different time window setting paradigms. In the last research of this thesis, we introduce service level agreements in the conVRP for tackling the real-world case of a pharmaceutical distributor. The application of the developed solution approaches to the real-world data estimates a cost reduction of 12.7% in the costs optimized in the project.

Considering the large savings obtained in every chapter of this thesis, it is undeniable that solving iVRPs using OR techniques can be extremely valuable for practitioners. Despite the enormous increment in complexity, we consider that iVRPs can be used for further enhancing new business models or pushing the efficacy of the current ones.

1.5. Further work

In this thesis, several contributions for the iVRP have been made to support a growing interest in this research area and real-world applications. Nonetheless, a number of interesting

issues remains unaddressed. While focusing on devising solution approaches to address each iVRP tackled in this thesis, some questions were constantly emerging:

- While building initial solutions, which problem should we address in first place?
- Which part of the integrated problem has a higher potential to be improved?
- What is the best decomposition strategy for a certain problem?
- How many periods should be considered in the planning phase?
- Do the previous questions depend on the type of problem and instance?

Few papers aim at answering these questions. However, we still find them really important in the context of integrated planning problems in general. Hence, these questions should be approached in a systematic fashion.

Finally, apart from the aforementioned research gaps, we selected a set of iVRPs and challenges that, to the best of our knowledge, have not been well explored yet. We consider that these problems pose real challenges both for the scientific community and practitioners.

Integrate the strategic and tactical decisions Although the literature has already addressed some problems where strategic and tactical decisions are integrated with vehicle routing (location-routing has been well studied [Prodhon and Prins, 2014]), there are some challenges that deserve to be tackled. During our research, it was quite clear that some solutions need to be found to support other supply chain processes. Despite the importance of location decisions, we point out the importance of defining the type of infrastructure in which a decision maker should invest in. In fact, this decision may largely impact operations, posing several challenges to the subsequent fleet size, fleet mix, and workforce planning problems. For instance, if we design a transportation network where deliveries are to be performed autonomously by the drivers at night (demands large storage areas and unloading equipment), using large vehicles (demands good accessibility), a large investment may have to be made to build the necessary infrastructure. Some issues may be apparently despicable at first sight, but the irreversible character of these decisions may be later translated in large costs for not being considered in the early phases of the network design process. We consider that the literature should expand the type of strategic and tactical decisions addressed in the iVRP literature.

Integrate different Lot Sizing Problem (LSP) variants Although our approach to the PRP adds realism to related literature, we consider that a large set of LSP extensions still needs to be integrated with the VRP. The vehicle routing community has been much more active in developing the PRP than the lot-sizing one, and this may be one of the reasons why the LSP extensions are not so well-studied. We encourage researchers to incorporate into PRP models well-known LSP extensions, such as sequence-dependent setups, setup carry-over, production scheduling and backlogs [Quadt and Kuhn, 2008].

Integrate minor planning problems Minor planning problems may be a good opportunity for further improvements in transportation planning. Although they are not the main concern in the planning phase, they are indispensable for the transportation activity to be conducted. For instance, the importance of refuelling decisions can be largely influenced by the context. There is a large differential in the prices of gas between Portugal and Spain. This fact may create a large impact in delivery schedules, as planners seek to refuel vehicles in Spain. Studying the impact of refuelling decisions in transportation planning in order to take advantage of price differentials is a very interesting problem which has not been properly tackled yet.

Robustness in iVRPs Robustness is a concept that has not even achieved maturity within VRP literature. However, we consider that this is a really challenging issue and practitioners are becoming increasingly interested in robust solutions in a large variety of problems. Robust solutions can be extremely important when the impact of disruptive events is high and corrective actions take time to be effective. This is quite common, particularly when integrating the VRP with non-operational decisions such as location decisions, capacity planning, and fleet sizing.

Consider the final customer Very few papers integrate vehicle routing with sales-related supply chain processes. However, there are new challenges that study the behaviour of the final customer to get further benefits. For instance, last-mile and same-day deliveries are very valued by customers in general and are now being integrated within VRPs. Recently, Archetti et al. [2016] present a very interesting problem which takes advantage of crowdshipping where ordinary drivers drop-off packages en route to their destination. This approach has been considered by some retail giants (such as Walmart and Amazon) and is completely in-line with the new trends seeking for shared economies.

We encourage researchers to pursue further developments on integrated routing problems. The world is now witnessing the rise of few technologies that may disrupt the way we deal with transportation, operational research, and collaboration between different companies. New delivery modes will become available with the deployment of drones and autonomous vehicles. The technological advances in computing power may suffer a huge leap with quantum computing, enabling the possibility for solving larger problems, integrating larger sets of decisions. Furthermore, most integrated problems assume specific collaboration conditions between various stakeholders. Self-executing smart contracts, stored on the blockchain ([Swan, 2015]), will allow companies to collaborate more transparently, trustfully, and efficiently. We foresee that powerful applications may rise from the hybridization of optimization techniques, data science, and blockchain based-governance. This could well be the birth of a whole new set of thesis regarding iVRPs.

Bibliography

- N. Absi, C. Archetti, S. Dauzère-Pérès, and D. Feillet. A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795, 2015.
- Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45, 2014a.
- Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120, 2014b.
- Yossiri Adulyasak, Jean-François Cordeau, and Raf Jans. The production routing problem: A review of formulations and solution algorithms. *Computers & Operations Research*, 55(Supplement C):141 – 152, 2015.
- Maria Albareda-Sambola, Juan A. Díaz, and Elena Fernández. A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research*, 32(3):407 – 428, 2005.
- Claudia Archetti and Luca Bertazzi. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 2012.
- Claudia Archetti and M. Grazia Speranza. The value of integration in logistics. *Proceedings of the 28th European Conference on Modelling & Simulation, ECMS 2014*, pages 691 – 697, 2014.
- Claudia Archetti, Luca Bertazzi, Gilbert Laporte, and Maria Grazia Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, August 2007.
- Claudia Archetti, Martin Savelsbergh, and M. Grazia Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472 – 480, 2016.
- Claudia Archetti, Marielle Christiansen, and M. Grazia Speranza. Inventory routing with pickups and deliveries. 2017.
- Jonathan F. Bard and Narameth Nananukul. A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12): 2202–2217, December 2010.

- T. Bektas. *Freight Transport and Distribution: Concepts and Optimisation Models*. CRC Press, 2017.
- Tolga Bektas, Gilbert Laporte, and Daniele Vigo. Integrated vehicle routing problems. *Computers & Operations Research*, 55:126, 2015.
- José-Manuel Belenguer, Enrique Benavent, Christian Prins, Caroline Prodhon, and Roberto Wolfler Calvo. A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931 – 941, 2011.
- WJ Bell, LM Dalberto, and ML Fisher. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.
- Luca Bertazzi, Giuseppe Paletta, and M. Grazia Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002.
- Pankaj Chandra and Marshall L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503 – 517, 1994.
- Leandro C. Coelho, Jean-François Cordeau, and Gilbert Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.
- Claudio Contardo, Jean-François Cordeau, and Bernard Gendron. A grasp + ilp-based metaheuristic for the capacitated location-routing problem. *Journal of Heuristics*, 20(1): 1–38, Feb 2014.
- J F Côté, G Guastaroba, and M G Speranza. The value of integrating loading and routing. *European Journal of Operational Research*, 257(1):89–105, 2017.
- Kevin Dalmeijer and Remy Spliet. A branch-and-cut algorithm for the time window assignment vehicle routing problem. *Computers & Operations Research*, 89(Supplement C):140 – 152, 2018.
- G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6 (1):80–91, 1959.
- Maryam Darvish, Claudia Archetti, and Leandro C. Coelho. Minimizing emissions in integrated distribution problems. *Technical Report, CIRRELT-2017, Montreal, Canada*, 55(Supplement C):141 – 152, 2017.
- Guy Desaulniers, Jørgen G. Rakke, and Leandro C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3):1060–1076, 2016.

- Bo Dong, Tolga Bektaş, Saurabh Chandra, Marielle Christiansen, and Kjetil Fagerholt. *A New Formulation for the Combined Maritime Fleet Deployment and Inventory Management Problem*, pages 321–335. Springer International Publishing, Cham, 2017.
- Furkan Enderer, Claudio Contardo, and Ivan Contreras. Integrating dock-door assignment and vehicle routing with cross-docking. *Computers & Operations Research*, 88 (Supplement C):30 – 43, 2017.
- Reza Zanjirani Farahani, Masoud Hekmatfar, Alireza Boloori Arabani, and Ehsan Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096 – 1109, 2013.
- Bernhard Fleischmann and Achim Koberstain. *Strategic Network Design*, pages 101–122. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- Bernhard Fleischmann, Herbert Meyr, and Michael Wagner. *Advanced Planning*, pages 71–95. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.
- Gianpaolo Ghiani, Gilbert Laporte, and Roberto Musmanno. *Introducing Logistics Systems*, pages 1–23. John Wiley & Sons, Ltd, 2005.
- Tobias Harks, Felix G König, and Jannik Matuschke. Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22, 2013.
- G. Laporte. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.
- Mauro Cardoso Lopes, Carlos Eduardo de Andrade, Thiago Alves de Queiroz, Mauricio G. C. Resende, and Flávio Keidi Miyazawa. Heuristics for a hub location-routing problem. *Networks*, 68(1):54–90, 2016.
- Gábor Nagy and Saïd Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649 – 672, 2007.
- Caroline Prodhon and Christian Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1 – 17, 2014.
- Daniel Quadts and Heinrich Kuhn. Capacitated lot-sizing with extensions: a review. *4OR*, 6(1):61–83, Mar 2008.
- M. Swan. *Blockchain: Blueprint for a New Economy*. O’Reilly Media, 2015.
- Fengqi You, Jose M. Pinto, Elisabet Capón, Ignacio E. Grossmann, Nikhil Arora, and Larry Megan. Optimal distribution-inventory planning of industrial gases. i. fast computational strategies for large-scale problems. *Industrial & Engineering Chemistry Research*, 50(5): 2910–2927, 2011.

Pickup and delivery with synchronization

A long-haul freight transportation problem: Synchronizing resources to deliver requests passing through multiple transshipment locations

Fábio Neves-Moreira* · Pedro Amorim* · Luís Guimarães* · Bernardo Almada-Lobo*

Published in *European Journal of Operational Research*, 2015

Abstract This research aims at tackling a real-world long-haul freight transportation problem where tractors are allowed to exchange semi-trailers through several transshipment points until a request reaches its destiny. The unique characteristics of the considered logistics network allow for providing long-haul services by means of short-haul jobs, drastically reducing empty truck journeys. A greater flexibility is achieved with faster responses. Furthermore, the planning goals as well as the nature of the considered trips led to the definition of a new problem, the long-haul freight transportation problem with multiple transshipment locations. A novel mathematical formulation is developed to ensure resource synchronization while including realistic features, which are commonly found separately in the literature. Considering the complexity and dimension of this routing and scheduling problem, a mathematical programming heuristic (matheuristic) is developed with the objective of obtaining good quality solutions in a reasonable amount of time, considering the logistics business context. We provide a comparison between the results obtained for 79 real-world instances. The developed solution method is now the basis of a decision support system of a Portuguese logistics operator.

Keywords Transportation · Tractor-and-trailer · Resource Synchronization · Fix-and-optimize · Case Study

*INESC TEC and Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

2.1. Introduction

Setting an efficient and flexible logistics network and defining its planning and operational processes is one of the most complex challenges one can find in the transportation sector. In the last two decades, we have witnessed a considerable effort towards the creation of a new generation of transportation systems [Crainic et al., 2009] which have to fit the value proposals of each company, focusing on distinct strengths such as quality, speed, reliability or cost.

Recently, cost pressures fostered a modification in the logistics paradigm and transshipment points are being deployed for long-haul freight transportation, introducing more flexibility into logistics networks. Usually, logistics operators (LO) find different solutions for the challenges imposed by long distance trips, such as using more than one driver per vehicle. However, adopting a transshipment-based distribution process may be truly advantageous. From a global point of view, the transportation network becomes much more flexible as it provides additional possibilities to perform the transportation of freight. Since the tasks assigned to each resource are much shorter, points in time when a resource becomes available can be rationally spread along the entire area occupied by the customers and will happen more frequently. In what concerns real-world cases, knowing that the variance of the planning variables may drastically affect timings, this is an advantage to ensure the execution of a transportation plan. In fact, this flexibility will be reflected in an increment of the service level as the network offers more solutions per unit of time. Furthermore, this flexibility not only accounts for the aforesaid facts, but also may yield solutions with less and shorter empty truck paths, which are a major concern among LO.

Logically, these advantages come at a cost. On one hand, the complexity of a planning problem considering transshipment points is much larger. On the other hand, the operational complexity at each facility is also likely to increase due to the difficulties imposed by the need to synchronize resources. This also means that it is extremely difficult to obtain a feasible solution manually.

In this research we develop a systematic approach on top of a real transportation network that has both the intermediate facilities and the human expertise to perform transshipment operations. Since in Mitrovic-Minic and Laporte [2006] the authors state that transshipment points have shown to be very useful in clustered instances (typical found in long-haul transportation) and that the advantages of using those nodes increases with problem size (real-world instances are large), we consider that our challenge is a valuable research topic. In order to provide an example of the aforementioned networks and context, Figure 2.1 is shown. In this novel problem, a set of tractors (Figure 2.2a) is located in each depot and the objective is to pickup requests from a certain location and deliver them to other locations. A request consists in the transportation of an entire semi-trailer (Figure 2.2b) and thus, considering that each tractor can only pull one semi-trailer at a time, we are in the presence of a full truckload case. Whenever a customer submits a request, both a pickup and a deliver time-window have to be defined, meaning that loading and unloading operations have to occur during those intervals. It is also possible to request each operation to be made in different days and for that reason, semi-trailers can be temporarily stored in a transshipment location (note that every depot also has a transshipment location). Additionally, a service to

be provided by a tractor can only occur during the period in which its base depot is opened. Tractors can only travel inside their reach radius which is defined by the maximum number of hours that drivers are allowed to work.

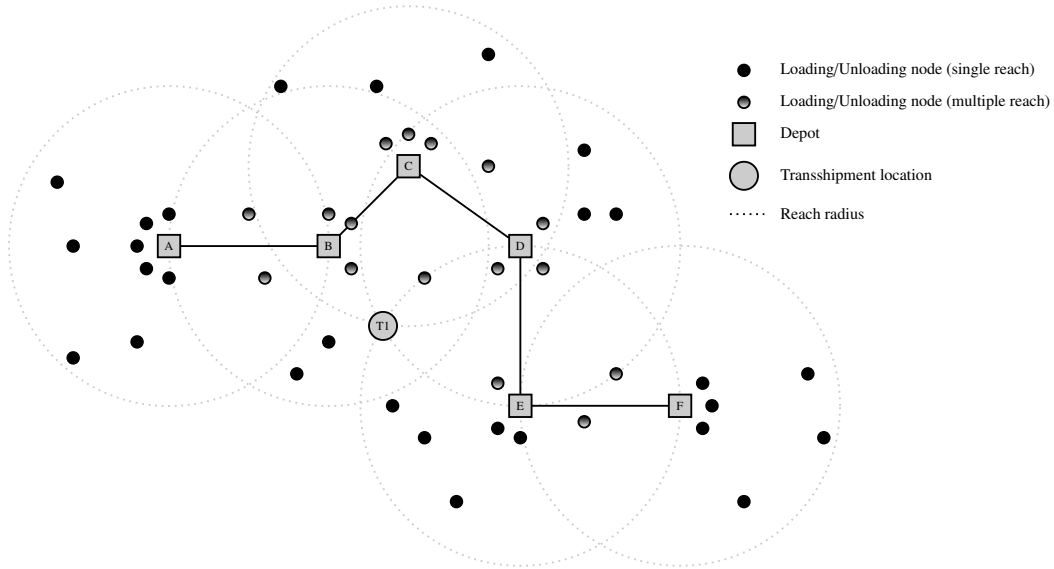


Figure 2.1 – Long-haul transportation network

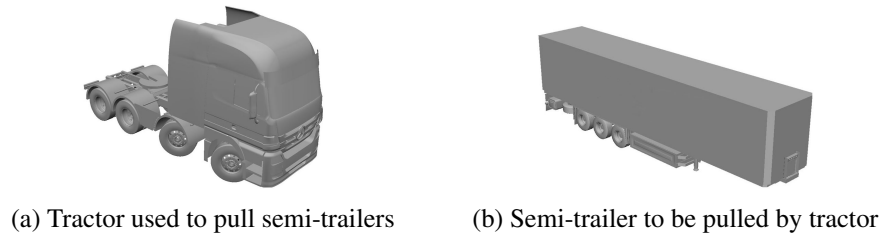


Figure 2.2 – Resources that compose a vehicle

This real-world challenge includes a combination of conditions that are rare in the literature.

Firstly, in consideration of the company's desire to maintain a certain level of comfort among its drivers, the maximum working time (including driving and other activities) is assumed to be 9 hours. Note that tractors are obliged to return to their base depot in the end of a workday. Therefore, drivers have a limited reach radius of approximately 4.5 hours which clusters the customers around each depot. In sum, we may assume that policies found in short-haul transportation activities are preferred.

Secondly, since drivers are confined to a limited region around their base depot, the necessity of executing multiple transshipments with the same request may be imposed in some services. Thus, in order to provide long-haul services by means of short-haul jobs, the company allows for the possibility of performing transshipments at certain locations. In fact, different regions may only communicate in these locations where two vehicles are

able to exchange freight. This is strictly necessary, otherwise it would not be possible to send requests between every combination of sender/receiver, while ensuring compliance with the law and with the policies of the company. It is now clear that if a request is to be picked in the zone of a certain depot and to be delivered in the zone of another depot, it is mandatory that at least one transshipment is going to happen. Additionally, a transfer is only possible if the compatibility between tractors and semi-trailers is assured (the terms “transfer” and “transshipment” are used interchangeably in this paper).

Thirdly, tractors are only able to leave a certain depot if they are to return pulling a semi-trailer, meaning that the delivery of a request must always be paired with the pickup of another request in the same trip (throughout the paper we use the term *path* to describe the movement of a vehicle from one location to another and the term *trip* to refer to a set of paths performed by a vehicle). Specifically, a tractor is not allowed both to leave or to arrive at its base depot without pulling a semi-trailer. In fact, if this condition is verified in every trip, the distance travelled without pulling semi-trailers will be minimized since the only possibility to execute such trips is reserved to the case when a tractor travels from an unloading location to a loading location. In reality, if the truck unloads a semi-trailer and picks another semi-trailer in the same location (despite being represented by different nodes), trips including empty truck paths can be fully extinguished from the transportation plans emerging from this pairing strategy. Additionally, there is no room for complex itineraries because most of the requests have different pickup and delivery zones, meaning that the semi-trailers will most likely be transported directly from their initial location to a depot or between depots before reaching their final destination.

Lastly, given the latter conditions, one may conclude that in long-haul requests there is a portion of the total journey to be crossed by the request that is defined *a priori*, specifically, the depots through which the request has to pass in order to be moved from one zone to another are known in advance. The necessary transshipment, pickup and delivery paths are defined in the moment a request is posted. Therefore, the objective of minimizing the travelled distance was not considered to be critical when analysing movements between zones. However, if a tractor unloads a semi-trailer in a certain location and has to return without pulling a semi-trailer, an empty truck path is incurred. Since these empty truck paths are undesirable among the transportation sector, although they are essential to continuing operations as stated by Crainic [1998], the company decided to follow the strategy of maximizing the number of pairings which are the trips where a tractor delivers and picks a semi-trailer. Although this pairing maximization objective is not the most common, if we consider that the arcs to be traversed are roughly defined in the beginning of the planning phase, we conclude that the routing part is not the main concern of this problem and thus it is admissible not to focus the attention in the objective of minimizing the travelled distance. Still, one does not have an idea of the exact moment when the trips are to be executed and the tractor that will execute them. Additionally, in requests with pickup and delivery time-windows to be made in different days, a minimum number of transfers must be accomplished in order to ensure that a feasible solution exists in the future. This is due to the fact that the length of the considered planning horizon may be shorter than the difference between the pickup and delivery time-windows of a given request.

To the best of our knowledge such conditions were never addressed in a systematic man-

ner before. Furthermore, being motivated by a real-world case, this research has to consider practical aspects which confer an additional complexity to the planning problem that hampers the computation of a feasible solution manually, even for planners with years of experience. We developed a novel mathematical formulation which models transshipment locations based on the approach presented by Cortés et al. [2010]. In order to address the necessary practicalities of the real-world problem, some enhancements had to be made resulting in a more complex formulation, which is only capable of solving small instances. Aiming at the resolution of larger instances and with the objective of obtaining better quality solutions in a reasonable amount of time, a matheuristic was developed in order to iteratively decompose the set of integer variables in the original MIP and explore simpler sub-problems.

The remainder of this paper is as follows. In Section 2.2 a literature review of problems with similar features is performed. The problem description is detailed in Section 2.3. In Section 2.4 the novel mathematical formulation capable of producing real-world applicable solutions is introduced. Section 2.5 is devoted to the proposed solution method and Section 2.6 reports numerical experiments that were performed on real-world instances. Finally, in Section 2.7, the paper summarizes the main achievements and future works that may arise from this research.

2.2. Literature Review

Since its introduction by Dantzig and Ramser [1959], the research community has been studying extensively different vehicle routing problem (VRP) variants and applications. Expectedly, the complexity of the addressed challenges also suffered a massive increase since the value of the optimization techniques captured the interest of most competitive LO. In addition, the computational power that is available nowadays enables the scientific community to further develop new mathematical models as well as the necessary methods to solve them. This remarkable and logical evolution is described by Laporte [2009].

Regarding the VRP variants, this paper is devoted to a practical application that is mainly similar to the pickup and delivery vehicle routing problem (PDVRP), which was introduced by Dumas et al. [1991]. In the PDVRP vehicles have to pickup goods from one location and deliver them to another location. This variant is considered to be the most flexible routing problem and its applicability to real-world problems, which has sparked an enormous interest among companies, is motivating the introduction of novel features. The reader is invited to explore the surveys made by Mitrovic-Minic [1998], Parragh et al. [2008a], Parragh et al. [2008b], Berbeglia et al. [2007] and Berbeglia et al. [2010] in order to obtain an overview of the principal paradigms that rule pickup and delivery problems. Considering the characteristics presented in the previous section, the interest of this paper also goes to problems including transshipment activities and using tractors and semi-trailers to perform deliveries. In order to capture important aspects about these features we studied the truck and trailer routing problem (TTRP), which allows for a different kind of transshipment where vehicles can park trailers and reload the truck, to study cases where the trailer may be separated from the rest of the vehicle. We also studied pickup and delivery problems

(PDP) as our challenge includes loading and unloading operations at different locations. Since some semi-trailers are stored at depots for later consideration, we consider that it is also important to review the literature on vehicle routing problems with backhauls (VRPB). Additionally, we review some roll-on roll-off vehicle routing problems (RRVRP) because, similarly to our case, it is common to model different trip types to enable vehicles to perform their activities correctly. Finally, given that we consider full truckloads, precedence constraints and multiple time-windows, some papers about these topics are also reviewed. None of these problems includes all the features addressed in our problem, nevertheless they share important points. To clarify our positioning, we define our problem as a full truckload a pickup and delivery problem with trailers and multiple transshipped requests (FT-PDPTmTR).

2.2.1 TTRP

Most of the literature dealing with trailers is concerned with the TTRP (trailers are not exchanged). Villegas et al. [2010] present a GRASP/VND and multi-start evolutionary local search for the TTRP. Although there is the possibility of detaching the trailer, the authors only address the single-vehicle case which is a significant simplification compared to the objective of this paper. Additionally, the goods are located at the depot eliminating the necessity of picking them up. In Villegas et al. [2011], the authors propose a GRASP with evolutionary path relinking. Although they consider an heterogeneous fleet, the truck and trailer concept is only used to respect the accessibility constraints showed by some customers. Therefore, each vehicle needs to leave the trailer in a designated location and has the possibility of delivering goods and picking up more goods from the parked trailer. Drex1 [2011] consider variable and fixed costs for trailers, time-windows and pure transshipment locations simultaneously. The author tests his branch-and-price and heuristic column generation approaches on randomly generated instances structured to resemble real-world situations and TTRP benchmark instances from the literature. The results show that with a heuristic column generation approach, real-world general TTRP instances can be solved in short time with high solution quality. However, the results for the benchmark instances are not so successful due to the low ratio of customer supplies to vehicle capacity in these instances. Recently, Derigs et al. [2013b] develops some simple heuristics that are able to compete with complex approaches from the literature. The heuristic is capable of solving TTRPs with/without load transfer as well as with/without time-windows. Additionally, the author considers the benchmark instances to be unrealistic since they do not include real-world features such as coupling and un-coupling costs and time consumption. Villegas et al. [2013] also tackle the same problem by means of a matheuristic that uses the routes of the local optima of a hybrid GRASP/ILS as columns in a set-partitioning formulation. This approach outperforms state-of-the-art methods both in terms of solution quality and computing time. The authors also present a table summarizing the most relevant approaches to the TTRP, providing a short description of the problem along with the proposed solution methods. Prodhon and Prins [2014] present a section dedicated to the TTRP which provides a very good overview of recent approaches to this problem. Considering the papers cited in the latter survey, it is clear that this type of problem contains

features that are quite similar to our challenge. Although using the same acronym, TTRP, Derigs et al. [2011] tackle a slightly different problem which is the tractor and trailer routing problem. In this problem, vehicles are not allowed to transport load without a trailer. The authors study a real-world situation in which EU-Regulations and compatibilities between trips and trailers have to be taken into consideration. The authors clearly state that this problem has not been studied neither in literature nor from a practical point of view. Analysing the aforementioned papers, it is necessary to point out that none of them considers the possibility of picking up requests from one customer and delivering them to another customer. Indeed, vehicles are only allowed to pick up additional load from their parked trailers which is something that does not happen in our case. Furthermore, their concept of transshipment is completely different from the one that is of interest for this paper in which two tractors may have the possibility of exchanging trailers. Although these problems are slightly different from ours, they include the main aspects regarding the binomen tractor/trailer that is found in the challenge to be addressed. In a recent survey concerning the applications of the VRPTT, an extension where trailers may be exchanged between vehicles, Drex1 [2013] makes a reference to the pickup and delivery problem with trailers and transshipments (PDPTT). This problem seems quite similar to ours as it includes pickup and delivery operations, trailers and transshipments. Drex1, which is the only one mentioning this pickup and delivery problem type, states that when transshipments are allowed, the set of vehicle routes of a solution is no longer sufficient to describe the path taken by each request. Drex1 [2007] also refers the difficulty of considering complex synchronization requirements at transshipment locations. In fact, the necessity of assuring that a specific set of resources is at a certain place in a given time is something extremely challenging, specially because it influences a considerable portion of the surrounding variables. Although Drex1 considers that such problems constitute a promising application for VRPTT, we were not able to find a single application or paper devoted to this version of the pickup and delivery problem which is, in fact, quite similar to the problem we aim to solve (except for the multi-transshipment flow of requests). In order to better understand the main features in which the aforementioned references are related to our problem, we provide Table 2.1.

Table 2.1 – Problem features comparison for cited papers considering TTRPs

Ref	Depot		Fleet				Transshipment Locations			Requests					Time windows		Practical aspects		
	Multi	Single	Heterogeneous	Homogeneous	Truck and trailer	Tractor and trailer	Single Transfers	Multiple Transfers	Load transfers between truck and trailer	Split deliveries	Pickups	Deliveries	Backhauls	Full truckload	Single	Multiple	Driving Legislation	Max waiting times	Accessibility constraints
Villegas et al. (2010)	•		•		•				•			•							•
Derigs et al. (2011)			•			•		•				•			•		•		
Drexel (2011)		•	•		•				•			•							•
Villegas et al. (2011)		•	•		•				•			•							•
Villegas et al. (2013)		•			•				•			•					•		•
Derigs et al. (2013b)		•		•	•				•			•			•		•		•
Our problem	•			•		•		•		•	•	•	•	•		•	•	•	•

2.2.2 PDP

On the subject of the pickup and delivery problem without considering trailers or semi-trailers, only a few papers addressed transshipments. Note that this transshipment concept assumes that there is an exchange of load between different vehicles, which in our case may be a swap of the entire semi-trailer between two tractors. One of the problems which makes use of this transshipment concept is the dial-a-ride problem. This problem is concerned with the transportation of people from one place to another and is tackled by Cortés et al. [2010] with a formulation of the transshipment nodes where people have the possibility to change from one vehicle to another. The problem is described with a mathematical formulation and solved by means of a branch-and-cut method. Mitrovic-Minic and Laporte [2006] address a pickup and delivery problem with time-windows allowing for the exchange of loads between vehicles. The main reasoning behind this transfer opportunity lies in the idea of keeping the drivers near their domicile, which is something quite similar to the depot zones considered in our case. A heuristic procedure is presented and tested in small instances with the objective of proving the utility of performing transshipments. Similarly, Qu and Bard [2012] tackle the pickup and delivery problem with transshipments with an implementation of a GRASP with adaptive large neighbourhood search (ALNS). Considering the previous three approaches, their main drawback is related to the number of requests that each can address. Indeed, this continues to be the greatest disadvantage of formulations and solutions methods to solve transportation problems with transshipments. When considering problems with less features such as the pickup and delivery problem with time-windows (PDPTW), researchers are able to obtain good results in larger instances. Masson et al. [2013b] develop an efficient feasibility test for inserting requests in pickup and delivery problems with transfers. Since requests are strongly interdependent due to the transshipment option, heuristic operators may constantly introduce infeasibilities which need to be detected and eliminated. The authors achieve a constant

time feasibility checker which is based on the forward time slack principle presented in [Savelsbergh, 1992]. This feasibility checking procedure is later extended by Grangier et al. [2014] to be faster when tackling their problem. Masson et al. [2013a] implement an ALNS for the PDP with transfers introducing new neighbourhood operators. The algorithm improves the results obtained in [Mitrovic-Minic and Laporte, 2006]. When the PDP deals with people, it is called the Dial-a-Ride problem. Masson et al. [2014] tackle a Dial-a-Ride problem where vehicles have to pickup and deliver people who have the possibility of changing vehicles in the middle of a trip. A large neighbourhood search algorithm solves real-life and generated instances, obtaining savings of up to 8% due to the use of transfers. The reader is also referred to the work of Dumas et al. [1991], Nanry and Wesley Barnes [2000], Bent and Hentenryck [2006], Ropke and Pisinger [2006a] and Liu et al. [2013] for an overall perspective of the existent solutions for this type of problems. Heuristic methods have been the most successful. A comparison between these PDPs and our problem is presented in Table 2.2.

Table 2.2 – Problem features comparison for cited papers considering PDPs

Ref	Depot		Fleet				Transshipment Locations				Requests				Time windows		Practical aspects			
	Multi	Single	Heterogeneous	Homogeneous	Truck and trailer	Tractor and trailer	Single Transfers	Multiple Transfers	Load transfers between truck and trailer	Split deliveries	Pickups	Deliveries	Backhauls	Full truckload	Single	Multiple	Driving Legislation	Max waiting times	Accessibility constraints	Real data
Dumas et al. (1991)	●		●								●	●			●		●			
Nanry & Wesley Barnes (2000)		●		●							●	●			●			●		
Mitrovic-Minic & Laporte (2006)	●			●			●				●	●			●					
Bent & Hentenryck (2006)		●		●							●	●			●					
Ropke & Pisinger (2006a)	●		●								●	●			●				●	
Cortés et al. (2010)		●		●			●				●	●			●			●		
Qu & Bard (2012)		●		●			●				●	●			●					
Grangier et al. (2014)		●	●				●				●	●			●					
Masson et al. (2013a)		●		●			●				●	●			●			●		●
Masson et al. (2014)		●		●			●				●	●			●			●		●
Liu et al. (2013)		●		●							●	●			●		●			
Our problem	●			●	●			●			●	●	●	●		●	●	●		●

2.2.3 VRPB

Since backhauls are a necessity arising from the fact that certain requests may not be satisfied during the considered planning horizon (1 day in our case), it is worth referring some works related to the vehicle routing problem with backhauls (VRPB). The concept

of pickup and delivery with backhauls is not common in the literature. In our case, requests may return to a depot, perform several inter-depot trips and be sent from a depot to the linehaul customer (final destination). However, if the requests are not able to reach its destination within the given planning horizon, we consider that a backhaul is incurred and the request has to be considered in the next planning horizon. Generally, we can only find vehicle routing problems with backhauls (not pickup and delivery problems) because the customers who demand a pickup want only to send something back to one of the depots, that is, once the load enters a depot it stays there. In order to obtain some insights on this type of feature, the reader is referred to the papers of [Duhamel et al. \[1997\]](#), [Toth and Vigo \[1997\]](#), [Zhong and Cole \[2005\]](#), [Ropke and Pisinger \[2006b\]](#) and [Liu and Chung \[2008\]](#).

2.2.4 RRVRP

There is still a class of problems which shares several characteristics our problem: the rollon-rolloff vehicle routing problem (RRVRP). In the RRVRP, tractors move large trailers between locations and a disposal facility. [Bodin and Mingozzi \[2000\]](#) address this problem providing a mathematical programming formulation, two lower bounds and four heuristic algorithms, testing the results on 20 different problems. The authors modelled 4 trip types in a problem with one depot and one disposal facility which did not include time-windows on the servicing of the trailer. A multiple disposal facilities with multiple depots (inventory locations) version is considered by [Baldacci et al. \[2006\]](#). The authors model the RRVRP as a time constrained vehicle routing problem on a multigraph. They describe an exact method based on a set-partitioning formulation. Recently, interesting hybridizations and heuristic procedures are also presented for the RRVRP by [Derigs et al. \[2013a\]](#), [Wy and Kim \[2013\]](#) and [Wy et al. \[2013\]](#). The main objectives of the research community when considering the RRVRP are to include new real-world constraints and to solve larger instances in an efficient manner. Likewise, since our attention goes for a real application, solving large and realistic instances is one of our first priorities.

Table 2.3 gives an overview of the similarities between the cited VRPBs and RRVPRs and our challenge.

Table 2.3 – Problem features comparison for cited papers considering VRPBs and RRVRPs

Ref	Depot		Fleet				Transshipment Locations			Requests					Time windows		Practical aspects			
	Multi	Single	Heterogeneous	Homogeneous	Truck and trailer	Tractor and trailer	Single Transfers	Multiple Transfers	Load transfers between truck and trailer	Split deliveries	Pickups	Deliveries	Backhauls	Full truckload	Single	Multiple	Driving Legislation	Max waiting times	Accessibility constraints	Real data
Duhamel (1997)		•		•							•	•	•		•		•			
Toth & Vigo (1997)		•		•							•	•	•							
Zhong & Cole (2005)		•		•							•	•	•		•					
Ropke & Pisinger (2006b)	•		•								•	•	•		•					
Liu & Chung (2008)		•		•							•	•	•							
Bodin & Mingozzi (2000)		•	•			•					•	•		•			•			
Baldacci et al. (2006)		•		•		•					•	•		•			•			
Wy & Kim (2013)		•		•							•	•		•			•			
Derigs et al. (2013a)		•		•		•					•	•		•			•			
Wy et al. (2013)		•		•		•					•	•		•	•		•			•
Our problem	•		•	•	•		•			•	•	•	•	•	•	•	•	•	•	•

2.2.5 Full truckloads

Most of the referenced papers deal with less-than-truckload requests as opposed to the problem we aim to solve. Considering this fact, our research also included problems where containers need to be moved from one point to another. Zhang et al. [2010] tackle a truck scheduling problem for container transportation with multiple depots and multiple terminals including containers as a resource. Four types of movements are used to describe different transportation activities and the objective is to reduce the total operating time of the fleet. A mathematical model is presented being addressed as a multiple travelling salesman problem with time-windows. In Zhang et al. [2011], a container drayage problem with resource constraints is studied. Although drayage operations are usually short-haulage container transportation between terminals and shippers/receivers, some of the its challenges are also found in our problem. The authors state that the problem becomes extremely complicated when different resources are regarded separately. The problem is described as a multiple travelling salesman problem with time-windows and additional constraints are tackled with a meta-heuristic based on reactive tabu-search. Arunapuram et al. [2003] develop a branch-and-bound for solving an integer-programming formulation of a VRP with full truckloads. The algorithm takes into consideration time-window constraints and waiting costs. This problem is simpler than ours since it does not include transfers. The work-flow in container transportation is also analyzed by Chung et al. [2007]. The authors develop mathematical models integrating the operating and design characteristics of con-

tainers. Additionally, they present heuristic algorithms to solve the models and report an example problem in order to explain how to apply the models to real-world cases. Imai et al. [2007] address a VRP that arises in picking up and delivering full container loads transporting them between intermodal terminals. Substantial cost and time savings are expected arising from an efficient linkage between pickup and delivery tasks taking into account temporal constraints. A two sub-problem heuristic is developed in which the classical assignment problem and the generalized assignment problem are solved. Considering these full truckload articles, we notice that none of them could address every aspect contained in the challenge we aim to solve. Having requests flowing through multiple transshipment locations augments the complexity of our full truck load routing problem and to the best of our knowledge this is the first time that such problem is addressed.

2.2.6 Precedence constraints

One of the most distinctive characteristics of the problem we aim to solve is the necessity to meet precedence constraints. This is due to the sequence of moves that each request has to perform. These request moves may be seen as activities which need resources that may be available or not in a certain moment, the vehicles. In Schmid et al. [2009], the authors effectively integrated optimization and heuristic techniques in order to solve a ready-mixed concrete delivery problem which includes several constraints that impose precedence between events. For example, some orders require vehicles with special equipment to be present for the delivery of concrete, something that requires a synchronization between resources. Moreover, some vehicles need to arrive first and remain at construction sites until the complete order has been fulfilled as they have some tools that are necessary during the entire process. Interesting modelling techniques were applied in order to ensure the applicability of the model to a real-world problem.

2.2.7 Multiple time-windows

This last subsection is concerned with the concept of multiple time-windows. Although few papers were found addressing this feature, in real-world problems this is frequent since it is difficult to maintain a steady availability particularly during work peaks or lunch time, for example. In Tricoire et al. [2010], the multi-period orienteering problem with multiple time-windows (MuPOPTW) is presented as a new routing problem combining objective and constraints of the orienteering problem (OP) and team orienteering problem (TOP), standard VRPs, and the original constraints form a real-world application. In this problem, sales representatives have to visit customers on a regular basis in certain periods. Customers may have up to two different time-windows per day. The authors develop an exact algorithm to check route feasibility and solve some instances using an efficient variable neighborhood search algorithm. Doerner et al. [2008] develop a model and several heuristic procedures to solve a problem motivated by a project carried out with the Austrian Red Cross which demands blood pickup services in certain periods of the planning horizon. A mathematical model for the VRP with multiple time-windows is presented by Bitao and Fei [2010]. The authors tackle the problem with a hybrid algorithm which combines the

ant colony system with 2-opt, a cell algorithm and a mutation operator. The proposed algorithm shows very good performance in terms of solution quality and robustness. This problem is also addressed by Belhaiza et al. [2013] in an implementation of a hybrid variable neighborhood search-tabu search heuristic. Additionally, a minimum backward time slack algorithm is developed.

Finally, for the latter 3 subsections, we present an analogy between the considered references and the problem that is tackled in this paper.

Table 2.4 – Problem features comparison for cited papers considering full truckloads, precedence constraints or multiple time-windows

Ref	Depot		Fleet				Transshipment Locations			Requests					Time windows		Practical aspects			
	Multi	Single	Heterogeneous	Homogeneous	Truck and trailer	Tractor and trailer	Single Transfers	Multiple Transfers	Load transfers between truck and trailer	Split deliveries	Pickups	Deliveries	Backhauls	Full truckload	Single	Multiple	Driving Legislation	Max waiting times	Accessibility constraints	Real data
Arunapuram et al. (2003)	•			•		•					•	•		•	•					
Chung et al. (2007)	•		•			•					•	•		•	•		•			•
Imai et al. (2007)		•	•			•					•	•		•			•			
Zhang et al. (2010)	•			•		•					•	•		•	•					
Zhang et al. (2011)		•		•		•					•	•		•	•					
Schmid et al. (2009)	•		•							•		•			•			•		•
Doerner et al. (2008)		•		•							•		•			•		•		•
Tricoire et al. (2010)		•		•								•				•				•
Bitao & Fei (2010)		•		•								•				•				
Belhaiza et al. (2013)	•		•									•				•	•			
Our problem	•			•		•		•			•	•	•	•		•	•	•	•	•

2.3. Problem Statement

In this section we describe the full truckload pickup and delivery problem with trailers and multiple transshipped requests (FT-PDPTmTR). We consider a planning horizon of one to two days. Some requests may need to be considered in more than one planning iteration, as their pickup and delivery time windows can be separated by more than two days.

2.3.1 Definition of entities

Requests The main objective of this challenging problem is to serve a set $R = \{1, \dots, |R|\}$ of customers' requests for transporting semi-trailers from one point to another. A request r is defined by two nodes, r^+ and r^- , corresponding to pickup and delivery

destinations, respectively. Also, denote the set of pickup nodes by $P = \{1^+, \dots, r^+\}$ and the set of delivery nodes by $D = \{1^-, \dots, r^-\}$. The union between these two sets, $N = P \cup D$, includes all the nodes where loading or unloading operations are to be made, i.e., customer sites. Another specification of each request is related to the moments when the deliveries may be executed. Customers are allowed to define multiple time windows l for each node $i \in N$, with limits denoted by $[a_{il}, b_{il}]$, within which the picking and delivering operations can be executed. Additionally, in each planning horizon it is necessary to define a minimum number of moves that a request has to perform in order to ensure that it can be delivered in time. For instance, if the pickup and delivery time-windows of a certain request are separated by three days and it needs to perform three one-day trips, it will be obliged to perform one trip per day, otherwise it will not reach its final destination in time.

Trips and paths The requests are to be transported by tractors, actually, semi-trailers need to be pulled by a tractor. Thus, in each planning horizon, tractors have to perform a set of trips $K = \{1, \dots, |K|\}$. Each trip k comprises several paths where a tractor leaves the depot and returns to the same depot.

Transportation network Each tractor is assigned to one of the base depots $s \in S = \{1, \dots, |S|\}$ considered in the planning horizon. A depot is defined by two nodes, s^+ and s^- , corresponding to starting and finishing nodes, and it has a time window $[an_s, bn_s]$ indicating the period during which it is opened. The set of starting nodes is $I = \{1^+, \dots, s^+\}$ and the set of finishing nodes is denoted by $F = \{1^-, \dots, s^-\}$. The union between these two sets, $I \cup F$, is W . Regarding transshipment locations, they are represented using four nodes, t^{us} , t^{uf} , t^{ls} and t^{lf} . We define a set of upper start transfer nodes $T^{us} = \{1^{us}, \dots, t^{us}\}$, a set of upper finish transfer nodes $T^{uf} = \{1^{uf}, \dots, t^{uf}\}$, a set of lower start transfer nodes $T^{ls} = \{1^{ls}, \dots, t^{ls}\}$ and a set of lower finish transfer nodes $T^{lf} = \{1^{lf}, \dots, t^{lf}\}$. The entire set of transfer nodes, which defines a transshipment location, is the union $T = T^{us} \cup T^{uf} \cup T^{ls} \cup T^{lf}$. The relevance of the four sets of transfer nodes will be clarified in Section 2.4. Generally, these nodes are positioned on the same exact location of depots although this is not mandatory, meaning that transshipment locations can be defined in other places, rather than depots. These locations are only able to store semi-trailers as opposed to depots where tractors can also be parked. In Figure 2.3, the nodes of a depot with a transshipment location are depicted.

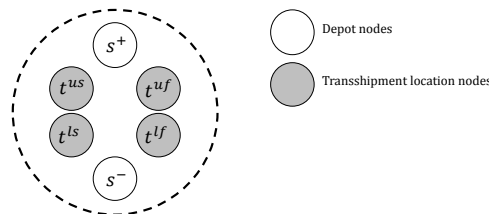


Figure 2.3 – Node representation of a depot with transshipment location

Each trip k has a specific set $N_k = P_k \cup D_k$, $W_k = I_k \cup F_k$ and $T_k = T_k^{us} \cup T_k^{uf} \cup T_k^{ls} \cup T_k^{lf}$. Given so, we define a network $G_k = (V_k, A_k)$ for each trip. The set $V_k = N_k \cup W_k \cup T_k$ includes all the reachable pickup and delivery nodes, the origin and sink nodes and the necessary transshipment nodes. The set of arcs A_k is obtained in a pre-processing procedure that is described in Section 2.3.5. Travel times and distances between two different nodes $i, j \in A_k$, are considered to be identical for every trip and are denoted by tt_{ij} and td_{ij} , respectively.

2.3.2 Pairing objective

One of the most distinctive features of this problem is related to the pairing objective that was mentioned previously. Indeed, this objective demands a different mindset when solving transportation problems. In our problem it is not necessary to serve all the customers in a certain planning horizon and the travel distance is not taken into account. Note that after imposing that a tractor can only leave and enter a depot while pulling a semi-trailer, a single trip will always move at least two requests, otherwise it cannot be executed. Considering that the itinerary to be crossed by each request is roughly defined *a priori* (as we know the depots and transshipment locations through which it will pass), every move that is accomplished means that we are closer to deliver a pair of requests. Provided that the company is not able to deliver all the requests in one planning iteration, we choose to maximize the number trips moving pairs of requests (pairings) as a proxy to the number of requests that we will be able to service in future planning iterations. Hence, maximizing the number of pairings turns to be logical in this case as the company wants to deliver the maximum number of requests. The greatest challenge of this planning problem is to synchronize all the resources while ensuring the accomplishment of the requested services.

2.3.3 Trip types

The pairing objective cuts several possibilities regarding path combinations. In order to describe the possible trips to be performed in this transportation problem, we assume that there exists a stock of semi-trailers in each depot, otherwise tractors would not be allowed to initiate a trip. Given so, a vehicle is allowed to execute one out of three existing trip types:

Zone trips A vehicle delivers a request to an unloading location, picks up a different request from a location inside the same zone and returns to the starting point (Figure 2.4a). Zone trips are suitable to finish a delivering of a request (since the vehicle leaves the depot to an unloading location) and to start a long-haul service (since the vehicle returns to the depot pulling a semi-trailer).

Depot trips A vehicle delivers a semi-trailer to a depot or transshipment location and returns pulling a semi-trailer of another request (Figure 2.4b). Depot trips allow for a transportation between zones, something that is mandatory in this problem as the reach of one vehicle would not be sufficient to provide long-haul services. These trips are defined *a priori* minimizing the distance to go from one zone to another.

Mixed trips A vehicle delivers a request to an unloading location, picks up another request from a different depot or transshipment location and returns to the starting point (Figure 2.4c). Mixed trips also support the transportation of freight between zones and are specially advisable in situations where the loading point is in between two depots. Some gains are achieved since the freight always travels in direction to its delivering point.

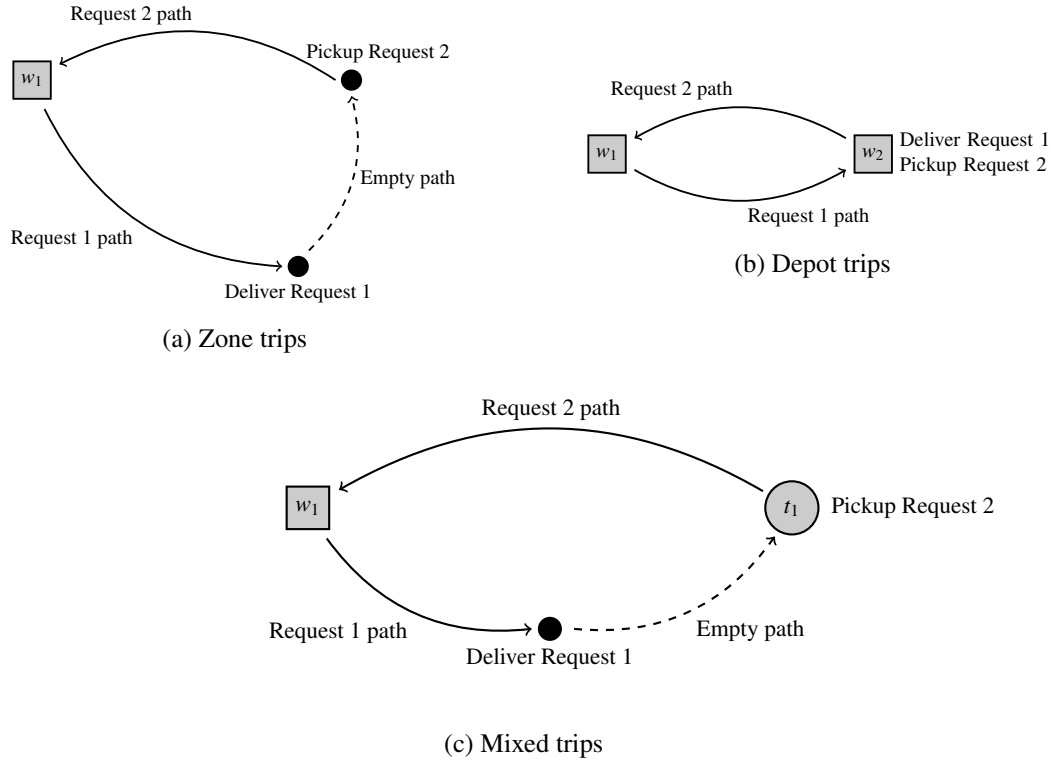


Figure 2.4 – Types of trips allowed

These three types of trips are flexible enough to provide the service that a long-haul transportation network is supposed to offer. Although we present the most basic trip type possibilities, some variants may also be applied. Analysing the moves represented in Figure 2.4, one concludes that tractors are always pulling a semi-trailer both when they leave and when they arrive at a depot or a transshipment location. In sum, whenever a vehicle delivers a request, it has to pickup another one before returning to its starting point.

2.3.4 Transfer concept

Considering the definition of request in this problem, it is not possible to split the contents of a given semi-trailer. Accordingly, whenever a semi-trailer needs to be transported between different zones, a transfer must be executed in a location reachable by the involved zones. Transshipment locations are then necessary not only to park semi-trailers but also

to provide transfers. Figure 2.5 shows an example of a transfer between two tractors and a semi-trailer that was already in the transshipment location.

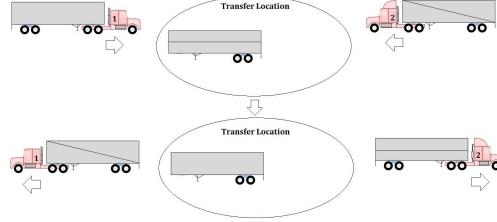


Figure 2.5 – Vehicles exchanging semi-trailers at a transshipment location

In terms of nodes, in order to allow multiple transshipments in a given transfer point for the same vehicle, transfer operations have to follow some rules which are the main reason to have four nodes representing a transshipment location. Transfers may occur in each of the three trip types that have been presented before. In the following examples we consider depots that are also transshipments locations.

In order to perform a zone trip (Figure 2.6), a tractor needs to leave its source node s^+ to load a semi-trailer in an upper starting transfer node t^{us} . When the tractor leaves this node in direction of its finishing transfer node t^{uf} , it incurs in a service time st . In the next moves, the tractor delivers the semi-trailer to a customer and loads another semi-trailer which is returned when it enters the lower starting transfer node t^{ls} . The tractor unloads the semi-trailer and returns to its sink node s^- .

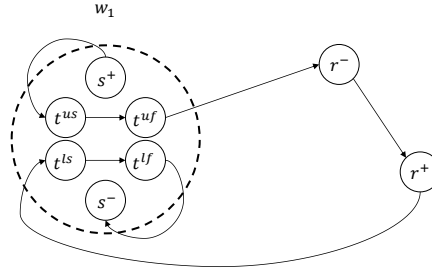


Figure 2.6 – Zone trip node paths

In case a depot trip is to be executed (Figure 2.7), a tractor initiates at a starting node s^+ which is located at its base depot w_1 . After initiating the trip, the tractor loads a semi-trailer at the upper starting transfer node f^{us} and incurs in a service time st during its fictitious path to the upper finishing transfer node t^{uf} . After loading the semi-trailer, the tractor pulls it to depot w_2 and unloads it after reaching the lower starting transfer node t^{ls} . Given that the considered tractor does not belong to depot w_2 , it needs to load another semi-trailer at the lower finishing transfer node t^{lf} in order to return to its base depot. The moment it arrives to its base depot, the tractor enters the node t^{ls} to unload the semi-trailer that it picked from w_2 . Another service time st is incurred in the fictitious path to t^{lf} and, because

it is in its base depot, the tractor has to finish its trip in the finishing node s^- .

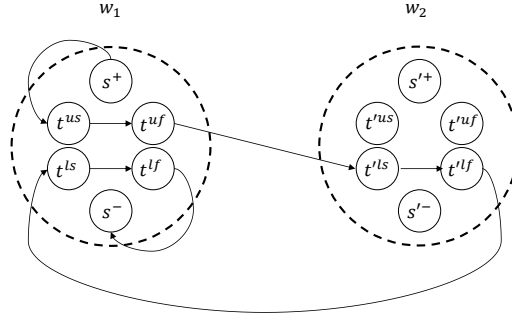


Figure 2.7 – Depot trip node paths

When a mixed trip is necessary, the paths to be executed are quite similar to the latter trip types, as it is shown in Figure 2.8. The same loading path is used and the tractor leaves the upper finishing transfer node t^{us} delivering the semi-trailer to a customer. In its next move, the tractor travels to a different depot without pulling a semi-trailer. The tractor enters the lower starting transfer node t^{ls} of depot w_2 and it loads a semi-trailer that is to be returned to w_1 . The return path and the unloading process is exactly the same as it was presented before.

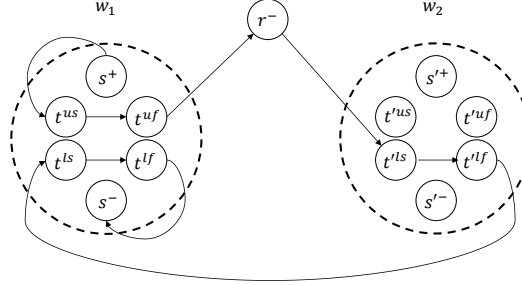


Figure 2.8 – Mixed trip node paths

2.3.5 Pre-processing

Pre-processing the data is a crucial step to tackle this planning problem. Since the trips to be made by each request are roughly defined *a priori*, as well as their corresponding paths, the set of arcs to be considered is much smaller. In order to define the set of possible arcs to be made in each trip k , we analyse the origin and destination of each request and insert the possible and logical arcs that are necessary into A_k . This phase is able to comprise a considerably large number of real-world constraints that can be addressed *a priori* and do not have to be included in mathematical models.

Necessary arcs A request can only be transported throughout arcs that it explicitly needs in order to be delivered. This means that only the arcs that are comprised in the pos-

sible *zone trips*, *depot trips* and *mixed trips* need to be added. Note that the resultant arc set has to be in concordance with all the aforementioned rules that were described above. Additionally, it is possible to define certain tractors to perform *depot trips* or special arcs for subcontracted vehicles, for example. This pre-processing step largely reduces the number of possible arcs.

Request/Tractor Compatibility In case we have more than one type of tractor and semi-trailer, the compatibility between the resources can be ensured by assigning the right trips to the right resources. This simplifies the model to tackle our problem and output solutions that are applicable to real-world problems.

Minimum number of nodes The aforementioned minimum number of nodes that each request needs to visit in a certain planning iteration is also computed in this pre-processing procedure. The number of trips and their duration needs to be taken into account in order to define this number of moves.

Depot requests When a request from a latter planning iteration needs to be considered in the current planning iteration, its pickup node will appear at one of the depots (where it was left). In this case, fictitious pickup locations need to be created in the pre-processing phase. Therefore, when backhauls are incurred, it is necessary to keep track of the stock of semi-trailers that was left at the depot at the end of each planning iteration.

2.3.6 Long-haul example

With the objective of explaining the remaining characteristics of the problem, we present an example of a long-haul request that passes through five transshipment locations.

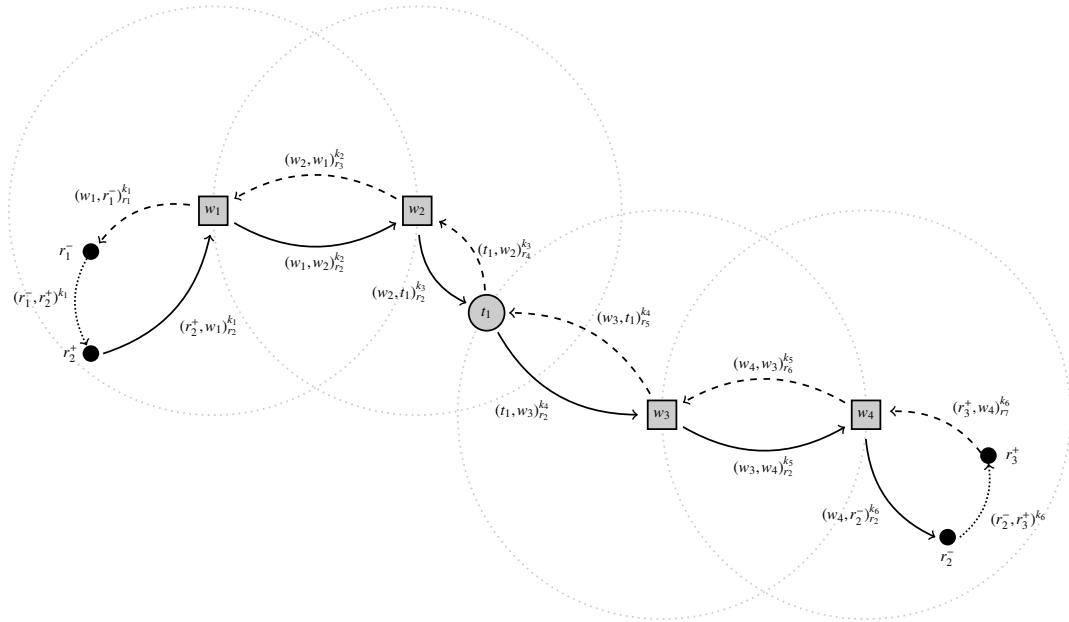


Figure 2.9 – Delivering a long-haul request

The example presented in Figure 2.9 shows the itinerary to be executed by a request which has to be picked from a loading location r_2^+ and delivered to an unloading location r_2^- . Although we are particularly interested in focusing on request r_2 , it is necessary to represent other movements which are executed by other vehicles. Therefore, the figure shows paths where a tractor is pulling the semi-trailer r_2 to be delivered to r_2^- (solid lines), paths where a tractor is finishing the rest of the trip dealing with other requests (dashed lines) and paths where a tractor is not pulling a semi-trailer (dotted lines), which happens in zone trips. The label of each path indicates its origin and destination. The vehicle which crosses a path is superscript and the request that is transported is subscript. In the beginning, a tractor k_1 , located at depot w_1 , executes a zone trip where it delivers a request r_1 to its unloading location r_1^- , picks up request r_2 from its loading location r_2^+ and returns to depot w_1 because r_2 has to be transported to another zone. At this point, the request r_2 needs to be transported to the next zone and thus a depot trip has to occur. In order to execute this depot trip a tractor k_2 , which has to move a request from w_2 to w_1 , will move the request r_2 from w_1 to w_2 in its return. Here, request r_2 , which is at depot w_2 , needs to perform 3 more depot trips in order to be moved to the zone to which its unloading location belongs (reachable from w_4). When it arrives to depot w_4 , another zone trip has to be executed to finish this long-haul transportation. Hence, a tractor k_6 pulls the semi-trailer requested by r_2 from w_4 to r_2^- , picks up another request from r_3^+ and returns to w_4 .

During this procedure, the freight requested by r_2 had to perform six moves that were executed by 6 different tractors. Although this may seem inefficient, if we analyse the cases in which a tractor was travelling empty, we conclude that the distance driven in this condition is minimal. In fact, empty paths were only executed during zone trips when traversing arcs between loading and unloading locations.

2.4. Mathematical formulation

2.4.1 Decision variables

The mathematical model works with the following binary variables. X_{ij}^k is equal to 1 if and only if location j immediately follows location i on trip k . Z_{ir}^k is equal to 1 if and only if request r visits location i on trip k . U_{il} is equal to 1 if and only if time window l is used for location i . S_{ir} is equal to 1 if and only if final stock of request r exists at transfer node i . Additionally, continuous variables W_i^k are used to represent the time when trip k passes at loading or unloading location i . WD_i^k and WA_i^k are used to represent the time when trip k departs and arrives at location $i \in T \cup W$.

Let $\delta^+(i) = \{j : (i, j) \in A\}$ and $\delta^-(j) = \{i : (i, j) \in A\}$ denote the set of successors and predecessors of i and j , respectively. The arcs that are included in this set are generated in the pre-processing procedure that was described previously.

2.4.2 Objective function

Considering that it is not mandatory to deliver every single request, we needed to induce movements only if the request is picked. We want to maximize the number of moves

which can be measured in visits to pickup or delivery locations by counting the Z variables. In the moment a certain request is picked up, the model has to make sure it reaches a favourable and feasible position in the network in order to be delivered in a later planning iteration. This is particularly important to ensure a connection between different periods (days) since the requests that are not delivered in a certain planning iteration are afterwards tackled in a rolling horizon approach. Objective function (2.1) simultaneously maximizes the movements performed by each request and minimizes the time that is necessary to execute the plan. These two criteria are weighted by α and β , respectively, which should be defined in collaboration with the person who is going to use the model (decision maker).

$$\max \alpha \cdot \sum_{i \in N} \sum_{r \in R} \sum_{k \in K} Z_{ir}^k - \beta \cdot \sum_{k \in K} \sum_{(i,j) \in A_k} tt_{ij} X_{ij}^k \quad (2.1)$$

2.4.3 Vehicle flow conservation

Constraints (2.2) ensure that only one arc exits from the starting node s^+ of the pre-processed arc set A_k whereas constraints (2.3) guarantee that only one arc enters to the finishing node s^- of the same arc set. In case a trip k is not to be executed, its vehicle may fictitiously traverse an arc directly from its starting node to its finishing node.

$$\sum_{j \in \delta^+(s_k^+)} X_{s_k^+, j}^k = 1, \quad \forall k \in K, (s^+, j) \in A_k. \quad (2.2)$$

$$\sum_{i \in \delta^-(s_k^-)} X_{i, s_k^-}^k = 1, \quad \forall k \in K, (i, s^-) \in A_k. \quad (2.3)$$

Loading and unloading nodes are visited at most one time with constraints (2.4).

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} X_{i, j}^k \leq 1 \quad \forall i \in N. \quad (2.4)$$

Flow conservation is ensured by constraints (2.5) which establish that a tractor executing a path entering a loading or unloading node must also leave that node.

$$\sum_{j \in \delta^+(i)} X_{ij}^k - \sum_{j \in \delta^-(i)} X_{ji}^k = 0 \quad \forall (i, j) \in A_k, k \in K. \quad (2.5)$$

Transfer nodes also have rules to be addressed. Constraints (2.6) to (2.9) define the flow of upper and lower transfers which have to start at a given starting transfer node i^{*s} and finish

at a finishing transfer node t^{*f} of the same transfer location t .

$$\sum_{i \in \delta^-(t^{us})} X_{i,t^{us}}^k = X_{t^{us},t^{uf}}^k \quad \forall t \in T, k \in K. \quad (2.6)$$

$$\sum_{i \in \delta^-(t^{ls})} X_{i,t^{ls}}^k = X_{t^{ls},t^{lf}}^k \quad \forall t \in T, k \in K. \quad (2.7)$$

$$\sum_{j \in \delta^+(t^{uf})} X_{t^{uf},j}^k = X_{t^{us},t^{uf}}^k \quad \forall t \in T, k \in K. \quad (2.8)$$

$$\sum_{j \in \delta^+(t^{lf})} X_{t^{lf},j}^k = X_{t^{ls},t^{lf}}^k \quad \forall t \in T, k \in K. \quad (2.9)$$

2.4.4 Travel times consistency

For all types of movements, it is necessary to enforce coherence in terms of travel times. When traversing a certain arc/path (i, j) , a service time st_i and a travel time tt_{ij} have to be taken into account. Consequently, constraints (2.10) have to be defined for every single arc that is used in the solution. Only arcs that belong to the pre-processed arc sets A_k need to be considered.

$$W_i + st_i + tt_{ij} \leq M(1 - X_{ij}^k) + W_j \quad \forall (i, j) \in A_k, k \in K. \quad (2.10)$$

2.4.5 Requests flow coherence

When a trip enters in an upper starting transfer node t^{us} , its tractor cannot be pulling a semi-trailer. Likewise, when a trip enters its sink node s^- , its tractor has to be empty. This means that $Z_{t^{us},r}^k$ and $Z_{s^-,r}^k$ have to be 0 and thus we do not need to instantiate these variables for each trip k and request r .

The requests coherence throughout trips is ensured by constraints (2.11) and (2.12). If a trip traverses an arc, its tractor has to be pulling the same semi-trailer r in both nodes of the arc or it is pulling it in neither. Moreover, each move from one transfer node to another one belonging to a different depot has to carry the same request from the departure depot to the destination depot (constraints (2.13)).

$$Z_{ir}^k - Z_{jr}^k \leq 1 - X_{ij}^k \quad \forall (i, j) \in A_k : i \notin I \cup T^{us} \cup T^{ls}, r \in R, k \in K. \quad (2.11)$$

$$Z_{jr}^k - Z_{ir}^k \leq 1 - X_{ij}^k \quad \forall (i, j) \in A_k : i \notin I \cup T^{us} \cup T^{ls}, r \in R, k \in K. \quad (2.12)$$

$$2X_{ij}^k \leq \sum_{r \in R} (Z_{ir}^k + Z_{jr}^k) \quad \forall (i, j) \in A_k : i \in T^{uf} \cup T^{lf}, j \in T^{us} \cup T^{ls}, k \in K. \quad (2.13)$$

Constraints (2.14) ensure that movements for a given request can only happen after its respective loading. After leaving a loading node the tractor of a trip has to be loaded with the respective request (constraints (2.15)) and after leaving an unloading node the tractor has to be empty (constraints (2.16)). Moreover, a tractor performing a trip can only enter an unloading node in case it is carrying the respective request (constraints (2.17)).

$$\sum_{k \in K} \sum_{i \in N} Z_{ir}^k \leq M \sum_{k' \in K} \sum_{j \in \delta^+(r^+)} X_{r^+, j}^{k'} \quad \forall r \in R. \quad (2.14)$$

$$Z_{jr}^k \geq X_{r^+, j}^k \quad \forall j \in V_k, r \in R, k \in K. \quad (2.15)$$

$$Z_{jr}^k \leq 1 - X_{r^-, j}^k \quad \forall j \in V_k, r \in R, k \in K. \quad (2.16)$$

$$Z_{r^-, r}^k \geq \sum_{j \in \delta^-(r^-)} X_{j, r^-}^k \quad \forall r \in R, k \in K. \quad (2.17)$$

A request is not allowed to enter a transfer node and leave it within the same vehicle (constraints (2.18)) as the vehicle could be traveling to a region where it does not belong. Furthermore, with constraints (2.19) each request is not allowed to be loaded at the same transfer node more than once. Additionally, a request may only be carried to a node if there is actually a trip passing in this node (constraints (2.20)).

$$Z_{t^{ls}, r}^k + Z_{t^{lf}, r}^k \leq 1, \quad \forall t \in T, r \in R, k \in K. \quad (2.18)$$

$$\sum_{k \in K} Z_{jr}^k \leq 1, \quad \forall j \in T^{us} \cup T^{ls}, r \in R. \quad (2.19)$$

$$Z_{jr}^k \leq \sum_{i \in \delta^-(j)} X_{ij}^k, \quad \forall j \in V_k, r \in R, k \in K. \quad (2.20)$$

2.4.6 Transfer times

The transfer time coherence for each request is assured by constraints (2.21) and (2.22). Constraints (2.21) are related to transfers that leave the transfer node by its upper finishing node and constraints (2.22) are active for trips that perform a transfer leaving the transfer node by its lower finishing node. These constraints ensure that the trip k' may only load a request that has already arrived in a different trip k .

$$W_{t^{ls}}^k + st_{t^{ls}} \leq W_{t^{uf}}^{k'} + M(2 - Z_{t^{ls},r}^k - Z_{t^{uf},r}^{k'}), \quad \forall r \in R, t \in T, k \in K, k' \in K, k \neq k'. \quad (2.21)$$

$$W_{t^{ls}}^k + st_{t^{ls}} \leq WD_{t^{lf}}^{k'} + M(2 - Z_{t^{ls},r}^k - Z_{t^{lf},r}^{k'}), \quad \forall r \in R, t \in T, k \in K, k' \in K, k \neq k'. \quad (2.22)$$

2.4.7 Time windows

Multiple time windows have to be respected at each loading and unloading node (constraints (2.23)).

$$\sum_{l \in L} a_{il} U_{il} \leq W_i^k \leq \sum_{l \in L} b_{il} U_{il}, \quad \forall i \in N, k \in K. \quad (2.23)$$

Moreover, it is only possible to use one of the available time windows (constraints (2.24)).

$$\sum_{l \in L} U_{il} = 1, \quad \forall i \in N. \quad (2.24)$$

Time windows also have to be taken into account in the origin and sink nodes (constraints (2.25)). These correspond to the opening and closing times of the depots. Additionally, the arriving time of each vehicle needs to be larger than the departure time, as forced by constraints 2.26.

$$an_i^k \leq W_i^k \leq bn_j^k \quad \forall i \in I, k \in K. \quad (2.25)$$

$$W_{s^+}^k \leq W_{s^-}^k \quad \forall \{s^+, s^-\} \in V_K, k \in K. \quad (2.26)$$

2.4.8 Practical constraints

Constraints (2.27) put a limit of wt on the time elapsed in a path between an unloading node and a loading node. The time taken by a path between nodes i and j is defined by tt_{ij} . These constraints come from a practical standpoint, as the company does not want to have

its resources too much time waiting and they also want to control the time that tractors are traveling without pulling a semi-trailer.

$$W_j^k - tt_{ij} - W_i^k - M(1 - \sum_k X_{ij}^k) \leq wt \quad \forall (i, j) \in A_k, k \in K. \quad (2.27)$$

Moreover, due to the working time limitation of the drivers, a trip cannot last beyond a fixed parameter mT (constraints (2.28)).

$$W_{s^-}^k - W_{s^+}^k \leq mT \quad \forall \{s^+, s^-\} \in V_K, k \in K. \quad (2.28)$$

To ensure that the decisions made in a planning period are not completely blind in relation to the next planning period, constraints (2.29) set a minimum number of transfers that need to occur in case a request is loaded. For instance, if a certain request needs to be delivered in two days but the total distance takes more than a workday to be crossed, part of the trip to be made by this request needs to be executed in the first day. Otherwise, it will not be possible to deliver the request in time. Therefore, if the model decides to pickup a request, the number of times that this request passes through transfer nodes plus its zone trip has to be greater or equal to a minimum number of moves defined *a priori*.

$$\sum_{k \in K} \sum_{j \in T^{us} \cup T^{ls}} Z_{jr}^k + \sum_{k \in K} \sum_{j \in A_k} X_{jr^-}^k \geq nM_r \sum_{k \in K} \sum_{j \in A_k} X_{r^+, j}^k \quad \forall r \in R. \quad (2.29)$$

Intermediate time-windows are also important to be enforced through the paths crossed by a request, in order to ensure that the next planning period is able to deliver requests on time (constraints (2.30) - (2.32)). We introduce intermediate time-windows $[ap_{t,r}, bp_{t,r}]$ for each transfer node t and request r in order to ensure that each request arrives at transshipment locations on time.

$$W_{t^{ls}}^k \leq bp_{t^{ls}, r} + M(1 - Z_{t^{ls}, r}^k) \quad \forall t \in T, r \in R, k \in K. \quad (2.30)$$

$$W_{t^{uf}}^k \geq ap_{t^{uf}, r} - M(1 - Z_{t^{uf}, r}^k) \quad \forall t \in T, r \in R, k \in K. \quad (2.31)$$

$$W_{t^{lf}}^k \geq ap_{t^{lf}, r} - M(1 - Z_{t^{lf}, r}^k) \quad \forall t \in T, r \in R, k \in K. \quad (2.32)$$

Integrality and binary conditions are defined by conditions (2.33).

$$X_{ij}^k, Z_{ir}^k, U_{il}, S_{ir} \in \{0, 1\}; \quad W_i^k \geq 0. \quad (2.33)$$

Finally, since we may need to reconsider some requests in a future planning iteration, we need to keep track of the remaining stock of each request r at transfer location t , S_{tr} . If a request starts a transfer, either it leaves the transfer by the upper finishing transfer nodes, the lower finishing transfer nodes or it stays in stock to be carried in the next planning period.

$$S_{tr} = \sum_{k \in K} Z_{t^{ls},r}^k - \sum_{k \in K} Z_{t^{uf},r}^k - \sum_{k \in K} Z_{t^{lf},r}^k, \quad \forall r \in R, t \in T.$$

2.5. Solution Approach: Matheuristic MH1

The matheuristic designed for the problem is a neighborhood search improvement heuristic and is inspired by the ‘exchange’ improvement heuristic of Pochet and Wolsey [2006] and the fix-and-optimize version of Helber and Sahling [2010]. Essentially, this heuristic explores the idea that most of the computational burden comes from the existence of a large number of integer variables and that small instances of the MIP can be solved efficiently. Hence, the heuristic proceeds iteratively by decomposing the set of integer variables in the original MIP to create easier MIP subproblems to re-optimize.

The matheuristic, *MH1*, works on the set \mathcal{Z} defined by all Z variables, since they control the assignment of requests to trips and nodes, and therefore most of the problem’s decisions. Simultaneously, the remaining integer variables of the problem denoted by the set C have their values limited by the definition of Z . At each iteration of the matheuristic, the original MIP problem is decomposed into a MIP subproblem by selecting a subset \mathcal{Z}' of the Z variables corresponding to the variables to be re-optimized in the next iteration, while the value of the Z variables not present in this set is fixed to the best solution found in the previous iterations. As this is an improvement heuristic it requires an initial feasible solution, which in the context of our problem can be easily obtained by setting all Z to zero. The subset \mathcal{Z}' is determined by defining a set of requests for which all the associated Z variables will be ‘freed’. i.e. re-optimized in the next iteration. To avoid local minima entrapment the selection of \mathcal{Z}' is guided through a neighborhood search algorithm and using an ordered finite set of user-defined neighborhood structures N_n , ($n = 1, \dots, n_{max}$), where n denotes the n^{th} neighborhood structure. Neighborhood structures are defined by the number of requests E to be optimized setting the size of the set of ‘freed’ variables and the potential complexity of the MIP subproblem to solve. Neighborhoods contain all possible combinations of \mathcal{Z}' of a given cardinality which are denoted as neighbors. Since our neighbor evaluation (solving a MIP) is a computational expensive process a full evaluation of the neighborhoods is unpractical. Therefore, a stochastic process controls neighbor selection to conduct a partial neighborhood search. After solving a subproblem from the current neighborhood structure the new solution objective value is compared with the previous best solution value. In case of an improvement, the search restarts at the first neighborhood structure ($n = 1$). Otherwise, the number of failed attempts within the current neighborhood structure is increased. A limited number of failures within a given neighborhood is allowed before switching to the next neighborhood structure in the ordered set.

Based on the latter ideas, *MH1* tries to find path related requests in order to set similar

conditions during the procedure. In order to improve the algorithm efficiency, the selection of the R requests is biased. Given that the tractor will always leave and arrive pulling a semi-trailer, some combinations of requests are more advantageous than others based on the complementarity of the requests paths. Thus, before selecting any request the complementarity among requests is assessed by calculating a compatibility score $\tau(r_i, r_j)$. To this purpose we introduce function $\sigma((i, j), p)$ which equals one if the inverse arc (j, i) is found in request path p and zero otherwise. The compatibility score of two requests can be computed using:

$$\tau(r_i, r_j) = \tau_0 + \sum_{(i,j) \in p_{r_i}} \sigma((i, j), p_{r_j})$$

When creating the subset \mathcal{Z}' only the first request is chosen randomly. The following selections are made considering the last request r_i appended to the subset \mathcal{Z}' and according to probabilities $prob(r_j) = \frac{\tau(r_i, r_j)}{\sum_k \tau(r_i, r_k)}$, until the number of desired requests has been selected. The term τ_0 is added to every score to ensure that any request can be chosen, but is sufficiently low to bias the search towards more desirable combinations (τ_0 was set to 0.25 in our tests).

To illustrate how the selection is performed consider the following example. Suppose that at the first iteration of the algorithm two requests are to be selected from a set composed of r_1, r_2, r_3 and r_4 with the associated paths $p_1 = \{1, 2, 3\}$, $p_2 = \{3, 2, 4\}$, $p_3 = \{3, 2, 1\}$ and $p_4 = \{4, 2, 3, 1\}$. The first request chosen randomly is r_1 , the selection of the second request is based on scores $\tau(r_1, r_2) = 1.25$, $\tau(r_1, r_3) = 2.25$, $\tau(r_1, r_4) = 0.25$ and the consequent probabilities $prob(r_2) = 33\%$, $prob(r_3) = 60\%$ and $prob(r_4) = 7\%$.

The algorithm ends according to the following stopping criteria: (1) the maximum running time allowed has been achieved, or (2) the maximum number of neighbors without improvement has been achieved in all neighborhood structures. The pseudo-code for the described matheuristic is given in Algorithm 1.

Algorithm 1 Matheuristic_MH1 ($n_{max}, noimp_{max}, tlimit$)

```

1: Calculate  $\tau$  for all possible request combinations
2:  $stop \leftarrow false, n \leftarrow 1, noimp \leftarrow 0, solution_{best} \leftarrow 0$ 
3: while not  $stop$  do
4:   Create subset  $\mathcal{Z}'$  with  $n$  requests
5:    $solution \leftarrow$  solve subMIP with  $\mathcal{Z}' \in \{0, 1\}$ 
6:   if  $solution > solution_{best}$  then
7:      $solution_{best} \leftarrow solution, n \leftarrow 1, noimp \leftarrow 0$ 
8:   else
9:      $noimp \leftarrow noimp + 1$ 
10:  if  $noimp = noimp_{max}$  then
11:     $n \leftarrow n + 1, noimp \leftarrow 0$ 
12:  if  $n > n_{max}$  or  $time > tlimit$  then
13:     $stop \leftarrow true$ 
return  $solution_{best}$ 

```

2.6. Computational Experiments

In this section we present the results obtained by the matheuristic as well as the limitations shown by the mathematical model when it was used independently. Our algorithm was coded in C++ and every time we needed to solve a mathematical model, *CPLEX 12.4* was used. The tests were performed on Intel @ 2.40 GHz processing units with 4 GB of random access memory running under the Linux operating system.

2.6.1 Real-world instances

The instances that were used to test the solution approaches were collected during the daily distribution of a real-world LO and comprise orders placed during March (03) and April (04) of the year 2014. Furthermore, we include instances from two different contexts (C1 and C2), corresponding to two different regions, one in the south and the other in the north of the country. Overall, 79 real-world instances were solved. A summary of the characteristics of the realistic instances faced by the company is presented in Table 2.5. The column *Depot Requests* concerns the number of requests that start at a depot and do not have to be picked from a remote location.

Table 2.5 – Description of the instances considered

	Requests	Depot Requests	Vehicles	Physical Locations	Nodes	Depots	Transfer Locations
Max	96.00	16.00	62.00	196.00	216.00	5.00	5.00
Average	60.23	3.67	59.97	124.94	147.42	4.49	4.49
Min	35.00	0.00	58.00	75.00	100.00	4.00	4.00

2.6.2 Matheuristic settings

Considering Algorithm 1, we need to set some parameters for testing the algorithm. The maximum number of iterations is set to a large number because time is a priority in our case. Therefore, n_{max} was set to 1000 iterations. Regarding the number of consecutive iterations in the same neighbourhood without improving, $noimp_{max}$ was set to 3. The neighbourhood structures are defined by varying the number of requests to be optimized in each iteration. The initial neighbourhood structure reoptimizes 10 requests and it is increased by two every $noimp_{max}$ consecutive fails to improve. These values were defined after a careful tuning that was performed during the preliminary tests of our matheuristic. The computational time is limited to 15 minutes as required by the company.

2.6.3 Solution approach analysis

With the objective of analysing the efficiency of matheuristic *MHI*, we make a comparison with an implementation of the mathematical model (*MathMod*) in *IBM CPLEX 12.4*. Figure 2.10 shows the average results for tests comprising the aforementioned contexts and months of data. It is possible to conclude that, in a large portion of the instance set, the matheuristic obtains better results as its objective values are above the line that represents the direct solution of the mathematical model. Moreover, as the size of the instances

increases, the differences between the two approaches also increase (larger instances are likely to have larger objective values in this case).

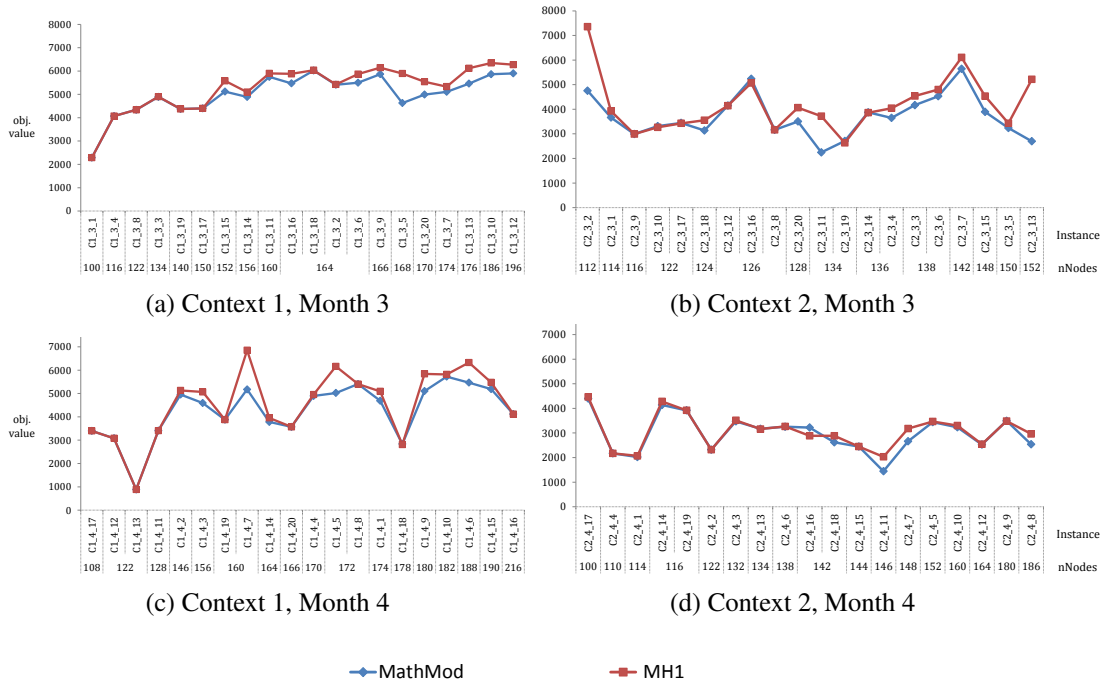


Figure 2.10 – Comparison between the average values obtained by the mathematical model and the matheuristic while maximizing the objective function (10 runs, 15 minutes)

The aggregated results per context and month can be observed in Figure 2.11. This figure depicts the ranges, median and average (crosshair) values for the relative different between the mathematical model *MathMod* and the implemented matheuristic *MH1*. It is clear that the matheuristic obtains superior results in every case since the relative difference is always positive. It is also important to note that the intervals are mostly located on the positive side.

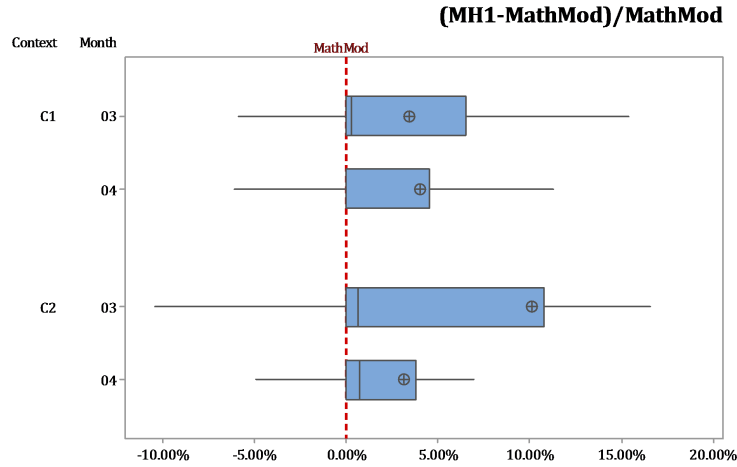


Figure 2.11 – Box plots depicting the relative difference between matheuristic *MH1* and the mathematical model *MathMod* (average objective values)

According to the ideas presented in [Dolan and Moré, 2002], an overall assessment comparing the mathematical model against our matheuristic is performed in Figure 2.12. The chart depicts the cumulative probability for each algorithm to obtain a solution with a relative gap smaller than or equal to τ . The relative gap is computed in relation to the best-known solution for each instance p . For each method s the relative gap $r_{p,s}$ is computed. The performance probability $\rho_s(\tau)$ is then for each τ defined according to expression:

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\}$$

n_p is the total number of instances and τ is the threshold for the relative gap based on the best solution found among all methods. Chart 2.12 allows us to understand that there is a probability of around 80% for the matheuristic *MH1* to find the best-known solution (for $\tau = 0$), whereas this probability corresponds to slightly more than 30% in the case of the mathematical model *MathMod*. Larger probabilities are therefore preferred. We conclude that for short periods of time, our matheuristic has a larger probability to find the best-known solution, which is completely in line with the initial objective of efficiently providing a good solution approach to be applied in a business context.

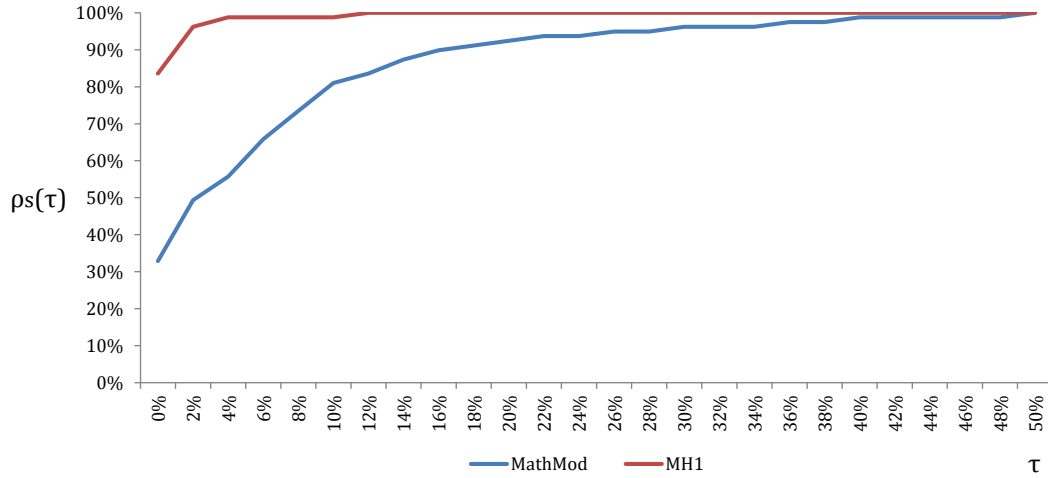
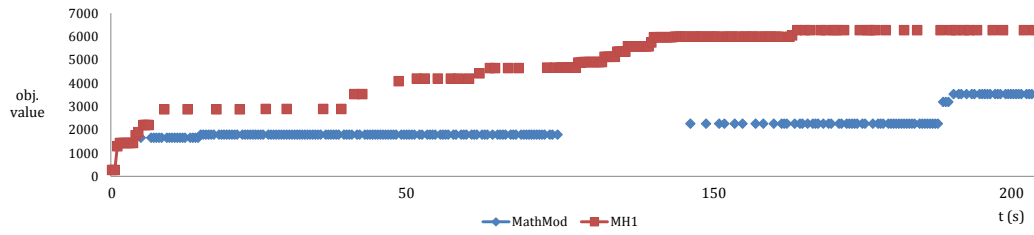
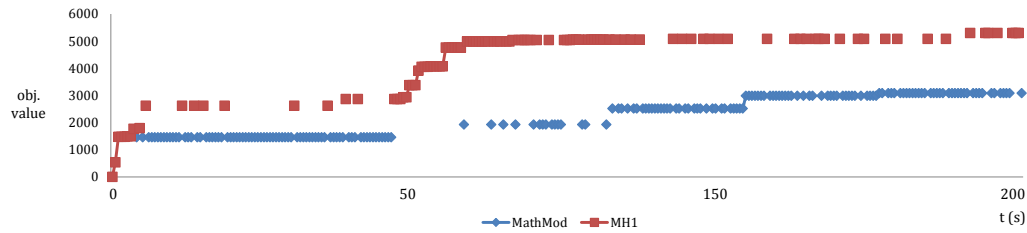


Figure 2.12 – Performance profiles for average objective values, 10 runs with a running time of 15 minutes

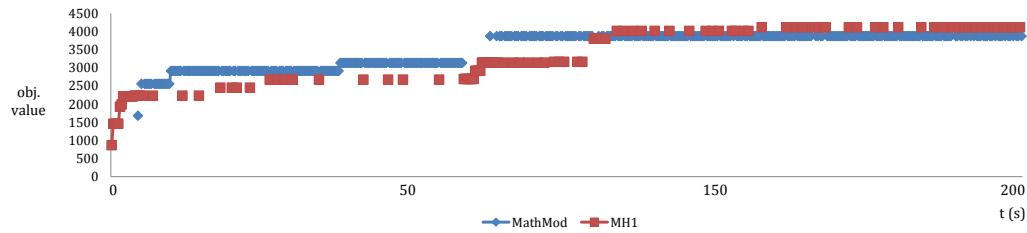
The time of convergence is also important. Indeed, we decided to test the mathematical model and the matheuristic for 15 minutes, but with the latter charts it is not possible to understand if the total running time is needed. Therefore, in Figure 2.13, we present an example of a run for the 3 largest instances of the considered instance set. It is possible to observe that the mathematical model never achieves a better solution when compared with the matheuristic (at the end of the 15 minutes). Furthermore, note that the 15 minutes may be unnecessary for the matheuristic to find good quality solutions, as it stabilizes early in these runs. Indeed, for these three instances, the best solution was obtained in less than 200 seconds, meaning that our approach is specifically suited for a business context where different solutions may be needed at a fast pace.



(a) Instance C1_3_12 - Context 1, Month 3, Day 12



(b) Instance C1_4_15 - Context 1, Month 4, Day 15



(c) Instance C1_4_16 - Context 1, Month 4, Day 16

Figure 2.13 – Scatter plots depicting the evolution of a run for the 3 largest instances

Additionally, Table 2.6 (2.A) presents the results obtained for all instances.

2.7. Conclusions and Future Work

In this paper the long-haul freight transportation problem with requests flowing through multiple transshipment points is presented. A novel mathematical formulation is proposed in order to obtain solutions for a real-world problem faced by a Portuguese LO. Additionally, with the objective of obtaining good quality solutions in short periods of time, we propose a fix-and-optimize matheuristic, proving its superiority compared to solutions obtaining by solving the mathematical model with a commercial solver.

The main contribution of this research is threefold. Firstly, a new operational model for the pickup and delivery problem is presented. Although this is a very specific case of a full truckload vehicle and scheduling problem, this is a new glance at the possibilities of using transshipment locations to support distribution. Considering that the use of transshipment

locations is part of a new logistics paradigm, this work surely presents a fresh approach to vehicle routing and scheduling problems. Additionally, we unified the best of long-haul and short-haul policies, resulting in a broader concept which can offer more service types without jeopardizing the comfort and working conditions of the drivers as well as the compliance with the legislation. Secondly, we provide an overview over different parts of the VRP literature. As this paper unfolds, it becomes clear that this problem includes a set of features that is present in several problems. In some cases, the similarities may only be physically noticeable whereas in other cases there are similarities only in the methodologies and modelling techniques used. Thirdly, we provide a solution approach that is able to beat one the best commercial solvers on the market. In fact, the developed matheuristic is able to obtain better solutions and converges more rapidly.

Regarding real-world results, although we are not allowed to reveal detailed information, it is worth mentioning that the company obtained a reduction of 6% in the cost per shipment for the requests served in the year after the implementation of the transshipment paradigm. This reduction is computed relatively to the year before the implementation, where the requests were delivered following a direct shipment strategy.

Finally, taking into account the lack of variety of papers addressing the tractor and trailer concept, we consider that tackling problems with the concept of modular vehicle parts may be an interesting field of research since these ideas can provide greater flexibility, occupation and efficiency. Furthermore, it would be interesting to extend this study to the case where request paths are not given in advance. In this extended version, the mathematical model would have to make additional decisions. The routing part becomes more complex and the synchronization possibilities are more diverse. Therefore, it would be interesting to test if our solution approach would be able to efficiently explore this new search space as it is larger and more difficult to be explored.

Acknowledgements

This work is partially financed by the ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "FCOMP-01-0124-FEDER-037281".

Bibliography

- Arunapuram, S., Mathur, K., and Solow, D. (2003). Vehicle Routing and Scheduling with Full Truckloads. *Transportation Science*, 37(2):170–182.
- Baldacci, R., Bodin, L., and Mingozzi, A. (2006). The multiple disposal facilities and multiple inventory locations rollon–rolloff vehicle routing problem. *Computers & Operations Research*, 33(9):2667–2702.
- Belhaiza, S., Hansen, P., and Laporte, G. (2013). A hybrid variable neighborhood tabu

- search heuristic for the vehicle routing problem with multiple time windows. *Computers & Operations Research*, pages 1–13.
- Bent, R. and Hentenryck, P. V. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15(1):1–31.
- Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15.
- Bitao, P. and Fei, W. (2010). Hybrid intelligent algorithm for vehicle routing problem with multiple time windows. *2010 International Forum on Information Technology and Applications*, pages 181–184.
- Bodin, L. and Mingozzi, A. (2000). The rollon–rolloff vehicle routing problem. *Transportation Science*, 34(3):271–288.
- Chung, K. H., Seong, C., Young, J., and Hwang, H. (2007). Development of mathematical models for the container road transportation in Korean trucking industries. *Computers & Industrial Engineering*, 53:252–262.
- Cortés, C. E., Matamala, M., and Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, 200(3):711 – 724.
- Crainic, T. (1998). *A survey of optimization models for long-haul freight transportation*.
- Crainic, T. G., Gendreau, M., and Potvin, J.-Y. (2009). Intelligent freight-transportation systems: Assessment and the contribution of operations research. *Transportation Research Part C: Emerging Technologies*, 17(6):541–557.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- Derigs, U., Kurowsky, R., and Vogel, U. (2011). Solving a real-world vehicle routing problem with multiple use of tractors and trailers and eu-regulations for drivers arising in air cargo road feeder services. *European Journal of Operational Research*, 213(1):309–319.
- Derigs, U., Pullmann, M., and Vogel, U. (2013a). A short note on applying a simple ls/lrs-based metaheuristic to the rollon–rolloff vehicle routing problem. *Computers & Operations Research*, 40(3):867–872.
- Derigs, U., Pullmann, M., and Vogel, U. (2013b). Truck and trailer routing—problems, heuristics and computational experience. *Computers & Operations Research*, 40(2):536–546.

- Doerner, K. F., Gronalt, M., Hartl, R. F., Kiechle, G., and Reimann, M. (2008). Exact and heuristic algorithms for the vehicle routing problem with multiple interdependent time windows. *Computers & Operations Research*, 35(9):3034–3048.
- Dolan, E. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.
- Drexl, M. (2007). *On Some Generalized Routing Problems*. RWTH Aachen / 08 Fakultät für Wirtschaftswissenschaften. Deutsche Post Lehrstuhl für Optimierung von Distributionsnetzwerken (N.N.) [813210].
- Drexl, M. (2011). Branch-and-price and heuristic column generation for the generalized truck-and-trailer routing problem. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, (12).
- Drexl, M. (2013). Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, 227(2):275 – 283.
- Duhamel, C., Potvin, J.-Y., and Rousseau, J.-M. (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science*, 31(1):49–59.
- Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22.
- Grangier, P., Gendreau, M., Lehuédé, F., and Rousseau, L.-M. (2014). An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. Research Report CIRRELT 2014-33, CIRRELT.
- Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, pages 1–22.
- Imai, A., Nishimura, E., and Current, J. (2007). A Lagrangian relaxation-based heuristic for the vehicle routing with full container load. *European Journal of Operational Research*, 176(1):87–105.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- Liu, R., Xie, X., Augusto, V., and Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, 230(3):475–486.
- Liu, S.-C. and Chung, C.-H. (2008). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing*, 20(1):29–42.
- Masson, R., Lehuédé, F., and Péton, O. (2013a). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.

- Masson, R., Lehuédé, F., and Péton, O. (2013b). Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers. *Operations Research Letters*, 41(3):211–215.
- Masson, R., Lehuédé, F., and Péton, O. (2014). The dial-a-ride problem with transfers. *Computers & Operations Research*, 41:12–23.
- Mitrovic-Minic, S. (1998). Pickup and delivery problem with time windows: A survey. *SFU CMPT TR*, 12:1–43.
- Mitrovic-Minic, S. and Laporte, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR : information systems and operational research*, 44(3):217–228.
- Nanry, W. P. and Wesley Barnes, J. (2000). Solving the pickup and delivery problem with time windows using reactive tabu search. *Transportation Research Part B: Methodological*, 34(2):107–121.
- Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008a). A survey on pickup and delivery problems part i: Transportation between customers and depot. *Journal fur Betriebswirtschaft*, 58(1):21–51.
- Parragh, S. N., Doerner, K. F., and Hartl, R. F. (2008b). A survey on pickup and delivery problems part ii: Transportation between pickup and delivery locations. *Journal fur Betriebswirtschaft*, 58(2):81–117.
- Pochet, Y. and Wolsey, L. (2006). *Production planning by mixed integer programming*.
- Prodhon, C. and Prins, C. (2014). A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17.
- Qu, Y. and Bard, J. F. (2012). A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers & Operations Research*, 39(10):2439–2456.
- Ropke, S. and Pisinger, D. (2006a). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, (August):1–30.
- Ropke, S. and Pisinger, D. (2006b). A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775.
- Savelsbergh, M. W. P. (1992). The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4:146–154.
- Schmid, V., Doerner, K. F., Hartl, R. F., Savelsbergh, M. W. P., and Stoecher, W. (2009). A Hybrid Solution Approach for Ready-Mixed Concrete Delivery. *Transportation Science*, 43(1):70–85.

- Toth, P. and Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science*.
- Tricoire, F., Romauch, M., Doerner, K. F., and Hartl, R. F. (2010). Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research*, 37(2):351–367.
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2010). Grasp/vnd and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. *Engineering Applications of Artificial Intelligence*, 23(5):780–794.
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2011). A grasp with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, 38(9):1319–1334.
- Villegas, J. G., Prins, C., Prodhon, C., Medaglia, A. L., and Velasco, N. (2013). A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, 230(2):231–244.
- Wy, J. and Kim, B.-I. (2013). A hybrid metaheuristic approach for the rollon–rolloff vehicle routing problem. *Computers & Operations Research*, 40(8):1947–1952.
- Wy, J., Kim, B.-I., and Kim, S. (2013). The rollon–rolloff waste collection vehicle routing problem with time windows. *European Journal of Operational Research*, 224(3):466–476.
- Zhang, R., Yun, W. Y., and Kopfer, H. (2010). Heuristic-based truck scheduling for inland container transportation. *OR Spectrum*, 32(3):787–808.
- Zhang, R., Yun, W. Y., and Moon, I. K. (2011). Modeling and optimization of a container drayage problem with resource constraints. *International Journal of Production Economics*, 133(1):351–359.
- Zhong, Y. and Cole, M. H. (2005). A vehicle routing problem with backhauls and time windows: a guided local search solution. *Transportation Research Part E: Logistics and Transportation Review*, 41(2):131–144.

Appendix 2.A Results Table

Table 2.6 – Average values for 10 runs of 15 minutes

Instance	Mean running time (secs)		Mean Objective Function		Mean Deviation to best	
	MathMod	MH1	MathMod	MH1	MathMod	MH1
C1_3_1	903.1	900.5	2289.87	2289.87	-	-
C1_3_2	917.3	901.1	5413.13	5425.14	-	-
C1_3_3	907.4	901.3	4890.55	4900.55	-	-
C1_3_4	901.9	902.0	4067.88	4070.08	-	-
C1_3_5	912.8	902.1	4628.66	5892.06	-0.21	-
C1_3_6	911.2	901.2	5500.27	5867.04	-0.06	-
C1_3_7	923.2	901.0	5107.60	5332.06	-0.04	-
C1_3_8	924.9	904.6	4337.18	4337.18	-	-
C1_3_9	902.5	901.2	5865.00	6146.00	-0.05	-
C1_3_10	912.9	900.4	5862.04	6346.01	-0.08	-
C1_3_11	909.9	901.0	5751.55	5900.04	-0.03	-
C1_3_12	901.1	901.6	5896.58	6272.54	-0.06	-
C1_3_13	914.7	902.4	5459.38	6118.55	-0.11	-
C1_3_14	911.7	901.4	4888.92	5097.61	-0.04	-
C1_3_15	940.4	902.6	5118.18	5580.14	-0.08	-
C1_3_16	910.0	901.7	5470.61	5879.58	-0.07	-
C1_3_17	910.0	901.3	4399.63	4399.63	-	-
C1_3_18	914.1	900.6	6019.84	6036.03	-	-
C1_3_19	915.2	900.5	4374.10	4374.10	-	-
C1_3_20	908.8	901.4	4992.10	5542.07	-0.10	-
C1_4_1	912.7	901.0	4688.27	5092.16	-0.08	-
C1_4_2	923.3	900.8	4954.00	5131.04	-0.03	-
C1_4_3	905.9	901.0	4593.11	5067.07	-0.09	-
C1_4_4	910.4	901.2	4895.12	4953.13	-0.01	-
C1_4_5	904.4	900.8	5022.14	6160.04	-0.18	-
C1_4_6	912.5	900.9	5464.56	6326.99	-0.14	-
C1_4_7	920.6	901.5	5175.00	6853.56	-0.24	-
C1_4_8	919.5	901.8	5404.58	5404.59	-	-
C1_4_9	919.3	901.7	5102.53	5842.61	-0.13	-
C1_4_10	909.9	901.5	5723.05	5819.05	-0.02	-
C1_4_11	20.6	900.6	3408.31	3408.31	-	-
C1_4_12	21.2	900.8	3079.77	3079.75	-	-
C1_4_13	2.4	901.0	884.88	884.88	-	-
C1_4_14	916.8	901.7	3783.00	3955.08	-0.04	-
C1_4_15	907.4	901.0	5188.41	5475.06	-0.05	-
C1_4_16	917.8	902.1	4119.70	4119.70	-	-
C1_4_17	74.2	710.8	3394.83	3394.83	-	-

Continued on next page

Table 2.6 – *Continued from previous page*

Instance	Mean running time (secs)		Mean Objective Function		Mean Deviation to best	
	MathMod	MH1	MathMod	MH1	MathMod	MH1
C1_4_18	914.3	901.7	2834.44	2809.69	-	-0.01
C1_4_19	900.7	901.9	3877.18	3877.18	-	-
C1_4_20	909.5	900.9	3569.68	3569.68	-	-
C2_3_1	907.2	904.1	3664.81	3930.89	-0.07	-
C2_3_2	914.1	902.0	4753.18	7352.75	-0.35	-
C2_3_3	908.4	904.6	4167.33	4536.84	-0.08	-
C2_3_4	924.0	902.2	3647.03	4042.80	-0.10	-
C2_3_5	912.4	901.6	3234.35	3429.60	-0.06	-
C2_3_6	930.3	904.0	4527.36	4800.66	-0.06	-
C2_3_7	910.7	901.1	5643.82	6104.81	-0.08	-
C2_3_8	913.8	906.4	3162.43	3162.43	-	-
C2_3_9	913.4	904.3	2990.36	2990.36	-	-
C2_3_10	915.2	904.2	3312.34	3263.45	-	-0.01
C2_3_11	906.7	907.2	2244.68	3714.88	-0.40	-
C2_3_12	932.5	904.4	4149.87	4144.77	-	-
C2_3_13	925.6	902.4	2697.43	5214.63	-0.48	-
C2_3_14	925.8	904.5	3873.85	3858.55	-	-
C2_3_15	900.9	902.6	3891.36	4534.85	-0.14	-
C2_3_16	926.1	903.6	5237.35	5064.90	-	-0.03
C2_3_17	930.1	903.1	3442.88	3426.38	-	-
C2_3_18	914.8	904.0	3137.41	3551.09	-0.12	-
C2_3_19	932.5	907.4	2709.90	2633.99	-	-0.03
C2_3_20	944.2	904.5	3502.39	4063.38	-0.14	-
C2_4_1	920.7	908.0	2020.94	2071.03	-0.02	-
C2_4_2	913.6	910.0	2320.93	2318.18	-	-
C2_4_3	908.1	902.7	3472.60	3514.40	-0.01	-
C2_4_4	953.9	908.0	2169.91	2169.91	-	-
C2_4_5	954.0	903.1	3441.85	3467.35	-0.01	-
C2_4_6	930.3	903.3	3252.35	3263.75	-	-
C2_4_7	925.5	903.9	2661.91	3178.90	-0.16	-
C2_4_8	905.9	903.3	2535.40	2958.89	-0.14	-
C2_4_9	914.4	902.8	3494.87	3483.87	-	-
C2_4_10	931.5	906.2	3230.90	3302.40	-0.02	-
C2_4_11	914.7	904.5	1445.92	2030.38	-0.29	-
C2_4_12	916.2	905.7	2529.64	2545.14	-0.01	-
C2_4_13	918.2	903.5	3169.89	3158.84	-	-
C2_4_14	927.7	904.4	4138.62	4281.62	-0.03	-
C2_4_15	906.3	902.2	2447.42	2447.42	-	-
C2_4_16	901.1	904.3	3219.91	2883.92	-	-0.10
C2_4_17	915.7	919.0	4402.43	4462.38	-0.01	-

Continued on next page

Table 2.6 – *Continued from previous page*

Instance	Mean running time (secs)		Mean Objective Function		Mean Deviation to best	
	MathMod	MH1	MathMod	MH1	MathMod	MH1
C2_4_18	915.0	903.8	2620.66	2887.40	-0.09	-
C2_4_19	913.4	904.6	3918.81	3918.81	-	-

Avg	871.54	900.58	3999.76	4293.64	-0.102	-0.021
Min	2.40	710.80	884.88	884.88	-0.489	-0.104
Max	954.00	919.00	6019.84	7352.75	-0.007	-0.005
StDev	194.83	21.66	1174.60	1339.86	0.10	0.02

Inventory-Routing

Owned Fleet Lateral Transshipments In Inventory Routing Problems

Fábio Neves-Moreira* · Luís Guimarães* · Bernardo Almada-Lobo*

Submitted to *Computers & Operations Research*, 2017

Abstract The purpose of this paper is to assess the value of owned fleet lateral transshipments in the context of the Inventory Routing Problem (IRP). This approach for transferring products between retail sites is adopted in various business contexts, such as the automatic-teller-machine and the vending-machine sectors. A novel mathematical formulation considers lateral transshipments performed by vehicles that are based at a central supplier. The model is solved by means of an exact branch-and-cut algorithm with a patching heuristic that explores routing-infeasible solutions by turning them feasible and performing local search before injecting them back into the branching process. Our solution approach efficiently solves small to medium sized instances available in the literature, both for instances with and without owned fleet lateral transshipments. We further study the case where the IRP is solved on a rolling horizon scheme with non-deterministic demands. A sensitivity analysis is performed on the length of the planning horizon, forecast accuracy, and proportion of the inventory costs over the total costs. The results show that for the literature instances with deterministic demand, owned fleet lateral transshipments allow for an average cost reduction of 1.1% with 2.4% of the total demand being transshipped. In a context with non-deterministic demand, the rolling horizon scheme leads to significant cost reduction when the inventory cost assumes a large portion of the total cost. Clearly, owned fleet lateral transshipments can constitute a hedging mechanism that eradicates inventory unbalances, which otherwise would remain in the system for several time periods.

Keywords lateral transshipments · inventory-routing · rolling horizon · branch-and-cut

*INESC TEC and Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

3.1. Introduction

Organizations are constantly seeking for added value opportunities that may foster better service levels or cost reductions. The Vendor Managed Inventory (VMI) policy centralizes inventory management decisions in a single entity, the vendor or supplier, which owns inventories and defines the inventory levels for a set of retail sites where the final customer buys the product. VMIs target solutions where inventory management is coordinated with distribution decisions to achieve lower global activity costs. Despite its advantages, inventory-distribution coordination implies a very challenging optimization problem called the Inventory Routing Problem (IRP). In this problem, a supplier holds a central inventory and decides the replenishment process to a set of retail sites which also have a storage area. Therefore, the supplier needs to decide: (1) when to serve a certain retail site, (2) how much to deliver of each product, and (3) which delivery routes should be performed in each day. Lateral transshipments correspond to inventory movements that are performed between locations of the same echelon. In a VMI system, the supplier corresponds to one echelon and the retail sites constitute a second echelon. Therefore, whenever two retail sites exchange inventory in order to satisfy their requirements (i.e. avoiding stockouts or stock in excess), they are performing lateral transshipments. Figure 3.1 shows an example of a supplier which controls a central inventory to be delivered to retail sites. Lateral transshipments are allowed between them.

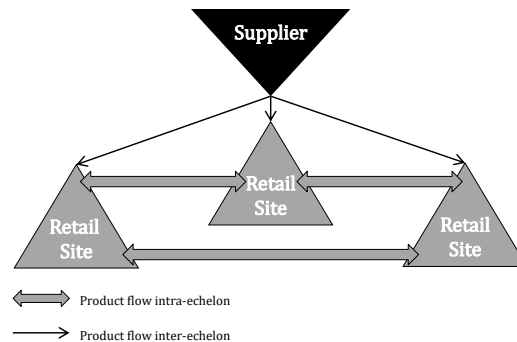


Figure 3.1 – Possible product flows in an IRP with lateral transshipments. Retail sites are allowed to exchange inventories helping each other to solve inventory unbalances.

Entities have several motivations to perform lateral transshipments. Additional operational flexibility allows for lower stock levels, reducing global inventory cost. In some situations, inventory unbalances may be solved more rapidly by reallocating inventory from surrounding retail sites than from the central warehouse. Consequently, better service levels are likely to be achieved in supply chains using lateral transshipments [Paterson et al., 2011]. Paterson et al. [2011] classify the timing of lateral transshipments as proactive and reactive. Proactive lateral transshipments are conducted periodically at predetermined points in time whereas reactive lateral transshipments occur in stock-out (or when a reorder point triggers an order) situations at any instant. The first type is more suited to situations where the events triggering a lateral transshipment, such as inventory unbalances left from latter period or predicted demand peaks, are somehow noticeable in advance. On the contrary,

the second type may have greater fit with very uncertain scenarios where the decision of transshipping is made after unbalances are created. Logically, reactive lateral transshipments are preferred when the costs of transshipment are relatively low compared to the costs incurred by maintaining higher inventory levels and by lost sales.

In this work, we are particularly interested in a situation where a central supplier deals with an IRP with multiple products, having the possibility to perform proactive lateral transshipments using owned fleet. In non-deterministic environments, this type of inventory re-allocation mechanism allows for a consistent reduction of the mismatch between supply and demand essentially caused by forecast errors. The idea is to increase robustness while minimizing the global cost. An example of an application of the IRP with owned fleet lateral transshipments is the vending-machine industry where a supplier has to deliver a mix of products to its machines (which are retail sites). In this business, it is common to re-allocate products from one machine to another either because the inventory is unbalanced or because some products are close to perish and could be sold faster in another machine. Figure 3.2 shows a schematic representation of the planning process faced by a decision maker.

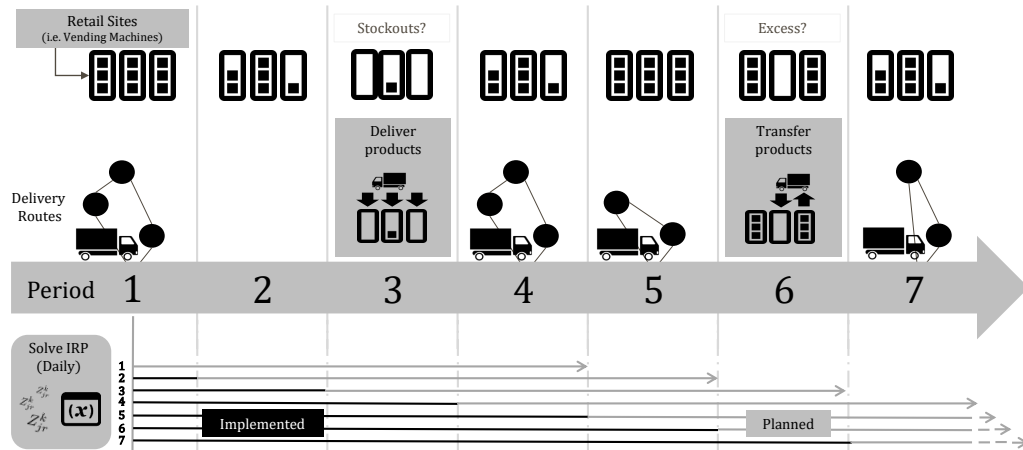


Figure 3.2 – Stockouts are solved by delivering more products (period 3) while stock surplus is mitigated solved by transferring products between retail sites (period 6). IRPs are solved in the beginning of each period, after recoding demands and updating forecasts of the previous period.

In every period, an IRP considering some periods ahead (four in the example) is usually solved on a rolling horizon basis. The decisions of the first period are implemented and it is likely that new information (real demand figures) becomes available at the end of that period. Planners use the new information to perform a new planning iteration, rolling forward the forecasts. The accuracy of the forecasts deteriorates with the number of periods considered ahead. Therefore, stockouts (period 3) or excesses of stock (period 6) are likely to occur on some of the vending machines. In a scenario where lateral transshipments are not possible, these events can have a high impact on lost sales and inventory costs as well as shrinkage costs. In fact, without transferring stock between vending-machines the

quantities to be replenished in out of stock machines can only be provided by the supplier, which is likely to have its capacity fully occupied and predestined. Additionally, if the demand of a certain machine is over-forecasted and an excess of stock is created, large inventory costs will be incurred if there exists no mechanism to re-allocate inventories. After such events, the initial inventory of each machine will not be balanced. Without considering lateral transshipments, unbalances could only be fixed several periods ahead. Despite the potential savings arising from lateral transshipments, it is worth mentioning that the IRP with owned fleet lateral transshipments that is not well-studied in the literature, mostly due to its inherent complexity. For that reason, this paper seeks to extend the IRP literature and provide four main scientific contributions:

- A new IRP formulation that addresses two versions of the problem (the standard Inventory-Routing Problem Without Transshipments (IRPWOT) and the extended Inventory-Routing Problem With Owned Fleet Transshipments (IRPWT)).
- A branch-and-cut procedure which improves the number of optimal solutions as well as the average gap of the multi-product multi-vehicle instances available in the literature (proposed by [Coelho and Laporte \[2013\]](#)).
- A sensitivity analysis over a set of parameters that are crucial for real-world implementations of planning processes based on the IRP.
- New managerial insights are derived from the value of owned fleet lateral transshipments by comparing the IRPWOT against the IRPWT for both the deterministic and non-deterministic demand cases.

The remainder of this document is divided as follows. Section 3.2 reviews the main relevant literature regarding extensions of the IRP focusing on lateral transshipments. In Section 3.3 we introduce the IRPWT and present the novel mathematical model for the problem. Section 3.4 describes the solution methodology developed for the deterministic and non-deterministic versions of the IRPWT. The value of lateral transshipments is discussed in Section 3.5. Finally, in Section 3.6, we present the main conclusions of this work and suggest future research directions.

3.2. Literature Review

Since the introduction of the single item IRP by [Bell et al. \[1983\]](#), the research community has been quite active while developing new solution approaches for the standard version of the IRP. For more than 30 years [[Coelho et al., 2014](#)], researchers have been focused on increasing the efficiency of mathematical formulations which could only find optimal solutions for very small IRP instances with few customers. Maybe because the complexity of the standard problem is already challenging enough, IRP extensions received less attention. Indeed, it seems that the potential gains of extending the IRP were not sufficient to induce the research community to explore and tackle more complex IRPs.

Although some IRP extensions were already tackled, there are not many papers regarding each type of extension. Usually, the IRP models available in the literature extend the standard problem with only one new feature. Hence, there are no mathematical formulations capable of addressing a multiple set of features simultaneously. The existent extensions deal with features such as pickups and deliveries [Anholt et al., 2013, Ramkumar et al., 2012], backhauls [Mes et al., 2014, Liu and Chung, 2008], split deliveries [Papa-georgiou et al., 2014, Cordeau et al., 2015], time-windows [Liu and Lee, 2011, Rusdiansyah and Tsao, 2005], and perishable products [Diabat et al., 2016, Coelho and Laporte, 2014]. All these extensions provide reasonable flexibility to apply the IRP to various real-world contexts including the vending-machine business, cash replenishment of automatic-teller-machines, distribution of fuel, distribution of automotive components, and maritime transportation. However, the inherent complexity found when integrating inventory management and transportation activities still poses enormous difficulties when the size of the instances goes beyond few customers, vehicles, and products. Since the literature is still scarce in terms of real-world applications, we consider important to measure the value of each extension related to the standard IRP in order to further foment the use of integrated models.

In this paper, we are particularly interested in the case where the standard operational model of the IRP is extended with the possibility of transferring products between entities of the same echelon, using owned fleet. This type of inventory reallocation is commonly referred as lateral transshipments in the inventory management literature [Paterson et al., 2011]. In order to cope with demand variability, companies may use lateral transshipments so as to rebalance stocks whenever mismatches between supply and demand exist. Many papers highlight the benefits of transshipments in systems where inventory management is centralized [Salameh and Jaber, 1997, Tiacchi and Saetta, 2011].

Regarding these stock movements, Coelho et al. [2012] tackle an extension of the basic IRP which includes lateral transshipments. The authors introduce transshipment variables to model quantities to be transferred between retail sites by a subcontracted fleet. The problem is solved by an Adaptive Large Neighbourhood Search (ALNS) heuristic and the results indicate that cost reductions are possible when considering subcontracted transshipments. Instances with up to 50 retail sites and three periods, and 30 retail sites and six periods are solved considering both the Order-up-to Level (OU) and the Maximum Level (ML) replenishment policies with and without transshipments. It is concluded that transshipments are profitable when the cost of outsourcing the delivery of ten units does not exceed the cost of transporting one unit with the owned fleet. Note that the considered transshipments are slightly different from the ones we aim to study. In fact, since the author considered subcontracted fleet, transshipment variables do not directly impact the routing decisions.

Shen et al. [2011] address an IRP in crude oil transportation. This problem is based on a real-world application which includes many different entities and business-tailored constraints. They formulate the problem as a mixed-integer program and use a Lagrangian relaxation method to find near optimal solutions. Multiple transportation modes and various logistics costs are considered. The solution approach is tested on instances with a 12-period planning horizon. Along with the fact of being a maritime transportation problem,

the transshipments executed in this problem are also slightly different from ours. There are no product flows between retail sites (customer harbours), specifically, input and output ports are used to perform lateral transshipments. This means that if a product is already at the retail site, it has to return to an output port and after that, it can be delivered to a different retail site. Considering this specification, the presented mathematical model is not applicable to the problem we addressed here. Jemai et al. [2013] consider a particular case of an IRP with static routings and show that transshipments allow for a better optimization of global indicators of the supply chain. A comparison between three different scenarios considering the total distance and the number of visits is performed. Recently, Mirzapour Al-e-hashem and Rekik [2014] address the IRP with transshipments considering the concept of "green logistics". The objective is to solve an IRP while selecting the appropriate vehicle by considering greenhouse emission levels, capacities and transportation costs. In fact, vehicles need to visit pickup locations and deliver the goods to a plant. The considered transshipments allow for the possibility to temporarily store products at every node. This means that vehicles can consolidate demands at a certain node in a given period and deliver the products to the plant in another period. The authors develop a mathematical formulation and use a commercial solver to obtain the solution to instances with up to 16 nodes, five vehicle types and 12 periods. Logically, this problem differs from the one we aim to tackle, however it is one of the few papers where stock reallocations between entities of the same echelon are executed in the context of the IRP.

3.3. Problem Description and Mathematical Formulation

To introduce the IRPWT, consider a complete graph $G = (N, E)$, where the set of vertices $N = \{0, 1, \dots, n\}$ is partitioned into vertex 0, the supplier which acts as the vehicle depot, and vertices $\{1, \dots, n\}$, corresponding to n retail sites to be served. Edges $(i, j) \in E$ are associated with a travelling cost c_{ij} . Let $P = \{1, \dots, p\}$ be the set of products that can be sold by each retail site and stored in their warehouse with an associated periodic holding costs of h_i^p . All inventories need to be maintained between a lower and an upper limit defined by the interval $[il_i^l, il_i^u]$. The supplier owns a fleet of vehicles K with capacity vc^k which is used both to deliver and to transship products. We assume that lateral transshipments are performed at no cost. Note that this assumption is reasonable in a real context, since the driver may only incur in a larger service time to unload and load the desired products. In each period $t \in T$ a certain quantity p^{pt} of product p is made available at the supplier and each retail site needs to satisfy a demanded quantity d_i^{pt} . Figure 3.3 shows an example of the IRPs to be considered in this research.

The figure shows one route to be performed by a single vehicle on a given period t of the IRPWT. The pickups and deliveries are described below each retail site. In the example, the vehicle performs two deliveries and three pickups. Note that the last pickup allows the vehicle to return one unit to the depot so it can be used at a later period.

To model the IRPWT, we propose a novel mathematical formulation which allows for owned fleet lateral transshipments while routing inventories. The replenishment policy considered in this research is the ML policy. We use binary decision variables X_{ij}^{kt} , Z_i^{kt} ,

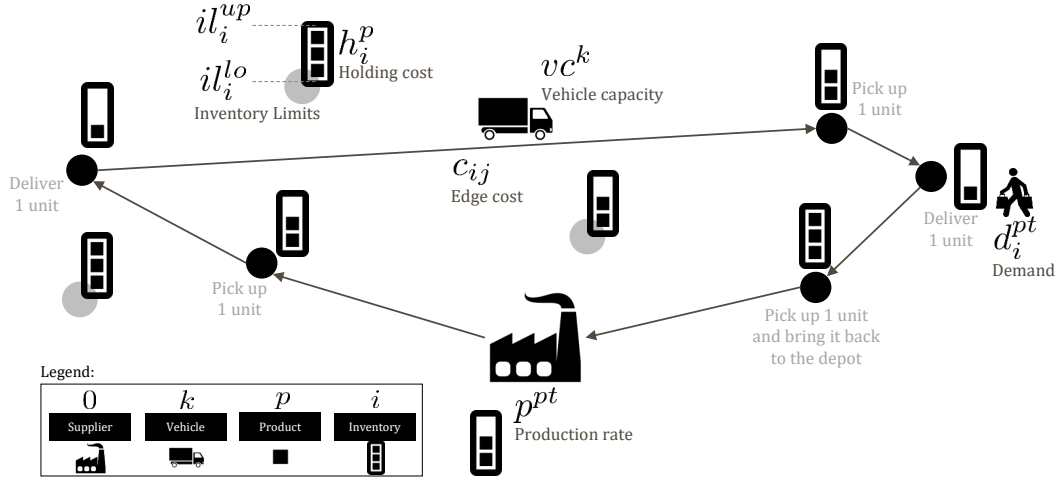


Figure 3.3 – Example of one period of the considered IRP with owned fleet lateral transshipments representing the entity sets and parameters involved.

BD_i^{pt} , BP_i^{pt} for routing decisions, and continuous variables D_i^{pt} , P_i^{pt} , I_i^{pt} , and L_{ij}^{pt} for dealing with quantities delivered, picked up, stocked, and flowing in each arc. Let X_{ij}^{kt} be the binary variables to indicate whether an edge (i, j) is traversed by vehicle k in period t . Z_i^{kt} indicate if retail site i is visited by vehicle k in period t . Binary variables BD_i^{pt} assume value one if product p is delivered to retail site i in period t , while continuous variables D_i^{pt} indicate the quantity that is delivered. Binary variables BP_i^{pt} detect if a pickup of product p is made at retail site i in period t while continuous variables P_i^{pt} indicates the quantity that is picked up. Auxiliary continuous variables I_i^{pt} define the inventories of each product p held in each retail site i in each period t . Finally, continuous variables L_{ij}^{pt} control the quantity of product p that flows through each edge (i, j) in each period t . The proposed formulation reads as follows:

(IRPWT):

$$\text{minimize } f_{WT} = \sum_{i \in N} \sum_{p \in P} \sum_{t \in T} h_i^p \cdot I_i^{pt} + \sum_{(i,j) \in E} \sum_{k \in K} \sum_{t \in T} c_{ij} \cdot X_{ij}^{kt} \quad (3.1)$$

s.t.

$$I_0^{pt} = I_0^{pt-1} + p^{pt} + \sum_{j \in N} L_{j0}^{pt} - \sum_{j \in N} L_{0j}^{pt} \quad \forall p \in P, t \in T \quad (3.2)$$

$$I_i^{pt} = I_i^{pt-1} + D_i^{pt} - P_i^{pt} - d_i^{pt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.3)$$

$$il_0^{lo} \leq \sum_{p \in P} I_0^{pt} \leq il_0^{up} \quad \forall t \in T \quad (3.4)$$

$$il_i^{lo} \leq \sum_{p \in P} (I_{ip}^{t-1} + D_i^{pt}) \leq il_i^{up} \quad \forall i \in N \setminus \{0\}, t \in T \quad (3.5)$$

$$\sum_{j \in N} L_{ji}^{pt} - D_i^{pt} + P_i^{pt} = \sum_{j \in N} L_{ij}^{pt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.6)$$

$$\sum_{p \in P} L_{ij}^{pt} \leq \sum_{k \in K} vc^k \cdot X_{ij}^{kt} \quad \forall (i, j) \in E, t \in T \quad (3.7)$$

$$D_i^{pt} \leq \sum_{j \in N} L_{ji}^{pt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.8)$$

$$P_i^{pt} \leq I_i^{pt-1} - d_i^{pt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.9)$$

$$\sum_{j \in N} L_{0j}^{pt} \leq I_0^{pt-1} + p^{pt} \quad \forall p \in P, t \in T \quad (3.10)$$

$$\sum_{j \in N} X_{ij}^{kt} = Z_i^{kt} \quad \forall i \in N, k \in K, t \in T \quad (3.11)$$

$$\sum_{j \in N} X_{ji}^{kt} = Z_i^{kt} \quad \forall i \in N, k \in K, t \in T \quad (3.12)$$

$$\sum_{k \in K} Z_i^{kt} \leq 1 \quad \forall i \in N, t \in T \quad (3.13)$$

$$\sum_{j \in N} X_{0j}^{kt} \leq 1 \quad \forall k \in K, t \in T \quad (3.14)$$

$$D_i^{pt} \leq il_i^{up} \cdot BD_i^{pt} \quad \forall i \in N \setminus \{0\}, t \in T \quad (3.15)$$

$$P_i^{pt} \leq il_i^{tp} \cdot BP_i^{pt} \quad \forall i \in N \setminus \{0\}, t \in T \quad (3.16)$$

$$BP_i^{pt} + BD_i^{pt} \leq \sum_{k \in K} Z_i^{kt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.17)$$

$$\sum_{i \in O} \sum_{j \in O} X_{ij}^{kt} \leq \sum_{i \in O} Z_i^{kt} - Z_g^{kt} \quad \forall O \subseteq V, k \in K, t \in T, \text{ for some } g \in O \quad (3.18)$$

$$X_{ij}^{kt}, Z_i^{kt}, BD_i^{pt}, BP_i^{pt} \in \{0, 1\}; \quad I_i^{pt}, L_{ij}^{pt}, D_i^{pt}, P_i^{pt} \geq 0. \quad (3.19)$$

The objective function f_{WT} (3.1) minimizes the total cost of the system, comprising inventory and transportation costs. Constraints (3.2) and (3.3) define the inventories of each product in each period both for the supplier and retail sites, respectively. In our model back-hauls (i.e. returns to the depot) are allowed and given by $\sum_{j \in N} L_{j0}^{pt}$. At each retail site, stock may flow out due to external demand (d_i^{pt}) or lateral transshipments (allowed by pickup variables P_i^{pt}). Constraints (3.4) and (3.5) ensure that the inventory levels of the supplier and retail sites are maintained in-between their minimum and maximum levels $[il_i^{lo}, il_i^{up}]$. Note that, since constraints (3.5) consider the remaining inventory $t - 1$ and the deliveries performed in t , we assume that the retail site capacity is never violated in each period. The network commodity flows are ensured by constraints (3.6). Constraints (3.7) ensure that vehicles' capacities are respected. Constraints (3.8) limit delivery quantities to the amount transported by the vehicle while constraints (3.9) limit pickup quantities. Note that stocks to be picked in a certain period were stocked at the end of the previous period. Constraints (3.10) limit quantities available to be shipped by the supplier. The vehicle flow conservation is ensured by constraints (3.11) and (3.12). Constraints (3.13) force each retail site to receive a single visit per period. Each vehicle can only leave the depot once in each period, as enforced by constraints (3.14). Constraints (3.15) and (3.16) capture the binary decisions which define whether a product is delivered or picked up, respectively. A product can only be delivered or picked up if the retail site is visited, as imposed by constraints (3.17). Hence, if a product is delivered, it cannot be picked up. Constraints (3.18) are the so-called subtour elimination constraints, used to prevent disconnected cycles or paths. Finally, the non-negativity constraints and variable bounds are defined by expressions (3.19).

3.3.1 Valid Inequalities

To strengthen the IRPWT model we add three groups of constraints to the original formulation. The first group focuses on symmetry breaking and on connecting X_{ij}^{kt} to Z_i^{kt} variables.

$$Z_i^{kt} \leq \sum_{j \in N} Z_j^{k-1t} \quad \forall i \in N, k \in K \setminus \{0\}, t \in T \quad (3.20)$$

$$\sum_{i \in N} Z_i^{kt} \leq \sum_{i \in N} Z_i^{k-1t} \quad \forall k \in K \setminus \{0\}, t \in T \quad (3.21)$$

$$Z_i^{kt} \leq Z_0^{kt} \quad \forall i \in N \setminus \{0\}, k \in K, t \in T \quad (3.22)$$

$$X_{ij}^{kt} \leq Z_i^{kt} \quad \forall i, j \in N, k \in K, t \in T \quad (3.23)$$

Given that the fleet is homogeneous, symmetry breaking constraints (3.20) are added to ensure that the vehicles with smaller index k are used in first place. Constraints (3.21) also cut symmetries by ordering the vehicles based on the number of retail sites visited. Constraints (3.22) only allow a vehicle to visit a retail site if it leaves the depot. Similarly, constraints (3.23) allow a vehicle to traverse an edge leading to a retail site only if this retail site is visited.

The second group of constraints is adapted from the work proposed by other authors.

$$\left\lceil \left(\sum_{g=1}^{t-1} D_i^{pg} - I_i^{p0} \right) / i l_i^{up} \right\rceil \leq \sum_{k \in K} \sum_{g=1}^t Z_i^{kg} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.24)$$

$$I_i^{pt-g} \geq \left(1 - \sum_{k \in K} \sum_{t'=t-g+1}^t Z_i^{kt'} \right) \left(\sum_{t'=t-g+1}^t d_i^{pt'} \right) \quad \forall i \in N \setminus \{0\}, p \in P, t \in T, g \in T, g < t \quad (3.25)$$

To tighten the bounds on the number of visits to each retailer, we adapt the valid inequalities proposed by [Coelho and Laporte \[2013\]](#) to our formulation and add constraints (3.24). These constraints discount the initial stock and compute the deliveries that are made to each retail site of each product in each set of subsequent periods. By doing so, and taking into account the maximum inventory level allowed by each retail site, a lower bound on the number of visits is computed and used to guide the definition of the Z_i^{kt} variables. Additionally, we add constraints (3.25) which adapt the valid inequalities presented by [Archetti et al. \[2014\]](#) for the case where a certain retail site is not visited between periods g and t . The inventory held by a retail site in period $t - g$ needs to be sufficient to serve the total demand between periods g and t . Otherwise, the retail site has to be visited at least once, which is guaranteed in case the summation of the Z_i^{kt} variables is positive. Finally, we propose a third group of constraints to cut off solutions where the same product

leaves and returns to the depot, which is clearly suboptimal.

$$\sum_{i \in N \setminus \{0\}} L_{0i}^{pt} \leq \sum_{i \in N \setminus \{0\}} D_i^{pt} \quad \forall p \in P, t \in T \quad (3.26)$$

$$\sum_{i \in N \setminus \{0\}} L_{i0}^{pt} \leq \sum_{i \in N \setminus \{0\}} P_i^{pt} \quad \forall p \in P, t \in T \quad (3.27)$$

Inequalities (3.26) impose that it is not possible to load the vehicles with a larger quantity than the one that is going to be delivered. Inequalities (3.27) impose that the supplier only receives products (i.e. products returned to the depot) if they have been previously picked up. Considering the in and out flows of the depot, the expression

$$\sum_{i \in N \setminus \{0\}} L_{0i}^{pt} - \sum_{i \in N \setminus \{0\}} D_i^{pt} + \sum_{i \in N \setminus \{0\}} P_i^{pt} = \sum_{i \in N \setminus \{0\}} L_{i0}^{pt}$$

holds for each product p and period t . For the sake of simplicity, we rewrite this expression as

$$q^- - d + p = q^+, \quad (3.28)$$

where q^- is the quantity of product leaving the depot, d is the total quantity delivered, p is the total quantity picked up, and q^+ is the quantity returning to the depot. Clearly, it is not necessary to load the vehicles with a larger quantity than the one to be delivered, which is expressed by requirements (3.26). Hence, $q^- \leq d$ holds. Rearranging the terms of (3.28) and considering that $q^- - d \leq 0$ for any optimal solution, then $q^+ - p \leq 0$, proving that inequalities (3.27) are valid.

3.4. Methodology

The main objective of this paper is to provide insights on the value of lateral transshipments IRPs. To do so, we test two different contexts. Firstly, the instances available in the literature are solved considering the whole planning horizon with deterministic demand. Secondly, a rolling horizon scheme is used and several planning iterations are performed to construct a solution for the entire problem considering non-deterministic demands (i.e. forecasts with an associated error). Both cases are later compared with the case where lateral transshipments are not allowed.

3.4.1 Branch-and-Cut scheme with deterministic demand

The IRP with lateral transshipments contains the Vehicle Routing Problem (VRP) as a special case. Therefore, the formulation presented in Section 3.3 is particularly challenging as it incorporates a few complicating constraints. Indeed, the subtour elimination constraints (3.18) are quite demanding as their number grows in an exponentially with the number of

retail sites considered. This means that, in case we include all these constraints, the formulation is only able to provide solutions for toy instances, with a very limited number of retail sites, products, vehicles and periods. However, for medium-sized problems (up to 50 customers) the formulation can be solved exactly if the complicating constraints (3.18) are properly tackled. In order to provide solutions for larger instance sizes, it is necessary to drop the subtour elimination constraints (3.18) and solve a simpler relaxed problem. The violated subtour elimination constraints are dynamically added to the problem on the fly. The solution approach starts by quickly creating an initial solution. This solution consists in a decoupled inventory-routing plan, meaning that the delivery quantities and inventory levels of each retail site are defined first using a lot-sizing formulation. The routing sequences are defined afterwards by means of a random cheapest insertion heuristic and local search based on 2-OPT moves. This decoupled solution may already provide a high quality upper bound if the link between inventory management and routing activities is not too strong for a particular instance. Figure 3.4 provides an overview of the devised branch-and-cut scheme.

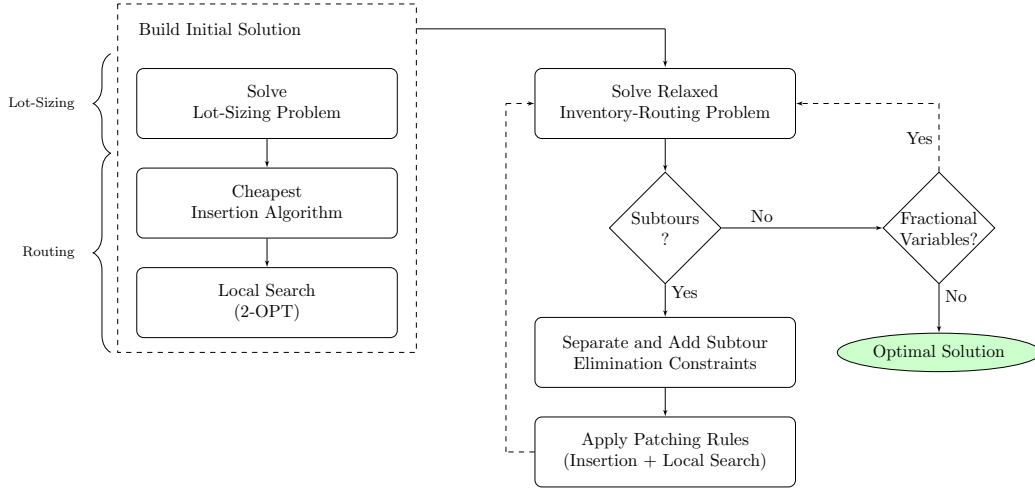


Figure 3.4 – The solution approach for the IRP with deterministic demand embeds a procedure to build an initial solution followed by a branch-and-cut procedure. During the branching process, the routing part of the solution is improved by several simple heuristics.

The first step is to solve an adapted Capacitated Lot Sizing Problem (CLSP) which adds a penalty to the objective function whenever a retail site receives a delivery. This penalty is proportional to the distance between the supplier and the retail site receiving the deliveries. This penalty, equal to the distance of a direct trip from the depot to the served retail site i , is given by $c_{0i} + c_{i0}$. To model the adapted CLSP, a different set of continuous variables is used to define delivery quantities. Let D_i^{pkt} represent the quantity of product p that needs to be delivered to retail site i using vehicle k in period t . Maintaining all the previous decision variables and parameters, the formulation of the adapted CLSP reads as follows:

(ACLSP):

$$\text{minimize } \sum_{i \in K} \sum_{p \in P} \sum_{t \in T} h_i^p \cdot I_i^{pt} + \sum_{i \in N \setminus \{0\}} \sum_{k \in K} \sum_{t \in T} (c_{0i} + c_{i0}) \cdot Z_i^{kt} \quad (3.29)$$

s.t.

(3.4) and (3.13),

$$I_0^{pt} = I_0^{pt-1} + p^{pt} - \sum_{k \in K} D_i^{pkt} \quad \forall p \in P, t \in T \quad (3.30)$$

$$I_i^{pt} = I_i^{pt-1} + \sum_{k \in K} D_i^{pkt} - d_i^{pt} \quad \forall i \in N \setminus \{0\}, p \in P, t \in T \quad (3.31)$$

$$il_i^{lo} \leq \sum_{p \in P} (I_{ip}^{t-1} + \sum_{k \in K} D_i^{pkt}) \leq il_i^{up} \quad \forall i \in N \setminus \{0\}, t \in T \quad (3.32)$$

$$D_i^{pkt} \leq \min(vc^k, il_i^{up}) \cdot Z_i^{kt} \quad \forall i \in N, p \in P, k \in K, t \in T \quad (3.33)$$

$$\sum_{i \in N \setminus \{0\}} \sum_{p \in P} D_i^{pkt} \leq vc^k \cdot Z_0^{kt} \quad \forall k \in K, t \in T \quad (3.34)$$

$$Z_i^{kt} \in \{0, 1\}; \quad D_i^{pkt}, I_i^{pt} \geq 0. \quad (3.35)$$

The objective function (3.29) minimizes the total inventory cost and the penalties that try to roughly estimate the routing cost. Other constraints are adapted to use the new set of variables D_i^{pkt} . Constraints (3.30) and (3.31) define the inventories at the supplier and retail sites, respectively. Constraints (3.4) and (3.32) impose lower and upper limits in these inventories. Constraints (3.33) connect continuous variables D_i^{pkt} to the integer variables Z_i^{kt} , capturing the retailers that need to be visited. Constraints (3.34) are vehicle capacity constraints. To impose at most one visit to each location in each period, constraints (3.13) are added to the model.

After solving this CLSP (it is not necessary to solve it to optimality) using a general-purpose solver, we obtain a set of deliveries that need to be performed in each period. We use a cheapest insertion algorithm which randomly chooses retail sites and inserts them in the cheapest position of an opened route until the capacity of the vehicle is fulfilled. When all the retail sites are inserted, a fast local search, composed of 2-OPT moves, is applied to the routes to finalize the initial solution construction.

The initial solution is then used in the root node of a branch-and-cut process. In each node

of the search tree, a relaxed Mixed-Integer Program (MIP), defined with constraints (3.1) - (3.25) except (3.18), is solved. The solutions provided by this formulation may have subtours. We separate the necessary subtour elimination constraints, which are a subset of constraints (3.18), and add them to the current MIP. At this point we use some patching rules to turn the infeasible solutions (with respect to the routing part) into feasible ones. In these patching rules, the routing part of each infeasible solution is completely redefined by the same route construction algorithm used in the initial solution phase and fed into the current MIP. This simple procedure is able to find feasible solutions and allows for quick improvements in the upper bound. The program is then reoptimized and this process is repeated until no more cuts are added. Here, branching on fractional variables occurs until the optimal solution is reached.

3.4.2 Rolling Horizon scheme with error prone forecasts

This section aims at developing an approach to simulate the hypothetical situation of a supplier that needs to solve an IRP on a regular basis. This scenario can be described with a rolling horizon planning scheme with non-deterministic demand. Consider a planning horizon with T time periods which are planned taking into account L periods in each planning iteration. The horizon rolls forward F periods in each iteration. We consider the case where only the first period of each planning iteration is implemented ($F = 1$). Based on the ideas presented by Clark [2005], we assume that the accuracy of demand forecasts improves as we approach the actual demand event. This fact is usually observed due to the incorporation of new information into demand forecasts. Figure 3.5 presents an example with the necessary steps to simulate demand forecasts. The rationale of the procedure is described below.

In this approach, a parameter α is used to quantify the degree of uncertainty in the demand forecasts ($\alpha = 0$ corresponds to perfect forecast). Note that we do not want to recreate a certain forecast process, we only assume that a generic forecast process has an inherent error proportional to α (similarly to Clark [2005]). For each retail site, product, and period, we define the base value VT for the first forecast T periods ahead of the actual demand $V0$. VT is computed as a function of the actual demand value ($V0$), a random factor (α), and a random variable r following a standardized random distribution:

$$VT = \max\{0, V0(1 + T\alpha r)\}. \quad (3.36)$$

The larger the value of α , the more likely it is for VT to be further away from $V0$. The updated forecast F_t is computed as a function of the interpolated base value $V_t = V0 + \frac{t}{T}(VT - V0)$, a random factor that is proportional to $t = \text{DemandPeriod} - \text{PlanningIteration}$, α , and a random variable r following a standardized random distribution, similarly to what is done in expression (3.36):

$$F_t = \max\{0, V_t(1 + t\alpha r)\} \quad t = T, T - 1, \dots, 1, 0. \quad (3.37)$$

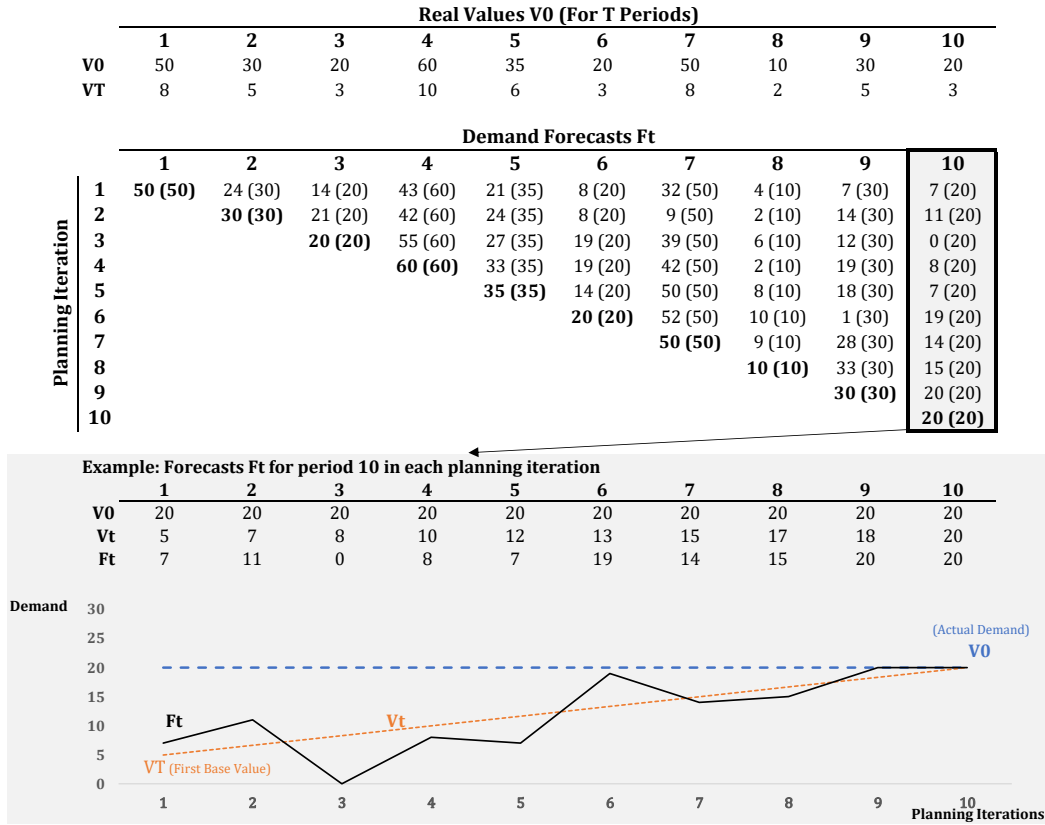


Figure 3.5 – Demand forecasts for a set of planning iterations. The real demand for each period is V_0 .

On the top of Figure 3.5, we present the values for V_0 and V_T that allow us to compute the forecasts F_t presented in the matrix in the middle of the figure. This matrix shows the forecasts F_t that are considered in each planning iteration and the actual demand values of each period (inside the parenthesis). For example, the second column of the first row tells that in the first planning iteration, the model will consider a forecast of 24 units ($t = 1$) but the actual demand will be 30 units. In the second planning iteration, the model will see this demand as 30 units which is exactly the same as the forecast ($t = 0$). The chart in the bottom of Figure 3.5 illustrates how the forecasts of the demand of period 10 evolve until the tenth planning iteration. V_t ensures the convergence of the forecast F_t to the actual demand V_0 . However, the random element is much stronger in the first planning iteration, resulting in a larger volatility of F_t . Thus, this forecast simulation method incorporates a random element while accounting for the accuracy improvement as we approach each actual demand period. Although real-world demands are stochastic, deterministic subproblems arise when solving stochastic IRPs in a rolling horizon framework Coelho et al. [2012]. Therefore, we use demand forecasts for some periods ahead as approximations of the unknown demand in order to guide and align decisions to be made in the present period. Only the most immediate decisions are implemented although the whole plan is built for all periods within a certain time horizon, based on available information [Sethi and Sorger, 1991]. Taking the

aforementioned ideas into account, we simulate a realistic rolling horizon planning process using the procedure depicted in Figure 3.6.

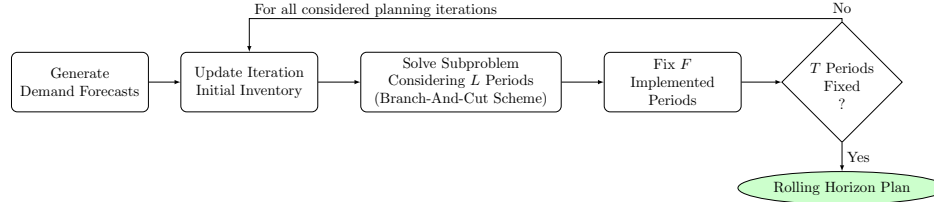


Figure 3.6 – The rolling horizon scheme start by computing demand forecasts, *apriori*. Afterwards, a series of subproblems are solved considering L periods. In each iteration, F periods are fixed and the procedure is complete when all the periods are fixed.

The proposed rolling horizon scheme starts by computing the demands to be used in each IRP subproblem (according to the example in Figure 3.5). Afterwards, a series of T planning iterations is solved considering subproblems with a planning horizon of L periods. In each planning iteration, the initial inventory is loaded (the remaining inventories of the previous iteration) and our branch-and-cut scheme is invoked to solve the inventory-routing subproblem. When the stopping criteria of the branch-and-cut scheme are met, F periods are implemented by fixing the values of their variables. If all the periods have been implemented, the algorithm outputs the rolling horizon inventory-routing plan. Otherwise, the planning horizon rolls over F periods and the process is repeated until all the T periods are implemented.

3.5. Computational experiments

This section presents the computational experiments performed using both schemes described in Section 3.4. First, we validate and assess the efficiency of our branch-and-cut scheme on a set of instances available in the literature. We compare the performance of our exact approach against the approach proposed by Coelho and Laporte [2013]. Second, using the same set of instances, we provide insights on the additional savings achieved by using an operational model with owned fleet lateral transshipments. Third, the instances are adapted to introduce demand uncertainty in order to test the rolling horizon scheme in a realistic environment. We aim at deriving conclusions regarding the use of owned fleet lateral transshipments in such context.

3.5.1 Instance set and test conditions

The considered set of instances is based on the MMIRP (multi-product multi-vehicle IRP) instances proposed by Coelho and Laporte [2013] (<http://www.leandro-coelho.com/instances/inventory-routing/>). We selected instances with multiple products, vehicles, and periods to fully evaluate our problem structure. The set contains five instances for each possible combination of retail sites (10, 20, 30, 40, 50, 100), products (1, 3, 5), vehicles (1, 3, 5) and periods (3, 5, 7). In the computational experiments related to the

developed branch-and-cut scheme, the instances are used in their original form. For the experiments regarding the rolling horizon scheme, each instance was extended by copying the same demand in the same order until the desired number of periods was achieved. All numerical tests were conducted on Intel®Xeon®E5-2650 processing units at 2.00GHz and 16GB memory. A single thread was used. The algorithms were implemented in C++ (Visual Studio 2015) and CPLEX 12.6 is used as the mathematical model solver.

To compare different algorithms, we rely on the ideas presented by Dolan and Moré [2002]. The authors introduce the concept of performance profile, a chart depicting the cumulative probability for each algorithm to obtain a solution with a relative gap smaller than or equal to τ . The relative gap is computed in relation to the best-known solution for each instance p . For each approach s the relative gap $r_{p,s}$ is computed. The performance probability $\rho_s(\tau)$ is then defined, for each τ , according to expression

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\},$$

where n_p is the total number of instances and τ is the threshold for the relative gap based on the best solution found among all methods.

3.5.1.1 Patching B&C efficiency assessment

To assess the efficiency of our exact branch-and-cut algorithm, we compare its results with the results presented by Coelho and Laporte [2013]. We refer to the authors' branch-and-cut as B&C and to our patching branch-and-cut as P-B&C. The P-B&C is tested in a context without owned fleet lateral transshipments (WOT) and in a context where these inventory reallocations are allowed (WT). We solve 675 literature instances and compare the average relative gap ($\frac{UB-LB}{UB}$), time, and number of optimal solutions between the three approaches. The time limit for these tests was 43 200 seconds, similarly to Coelho and Laporte [2013]. The summarized results of the computational experiments are presented in Table 3.1.

Table 3.1 – Summary of computational results for the MMIRP instances presented by Coelho and Laporte [2013]. The P-B&C is able to improve the average relative gaps for the standard problem (WOT) and for the extended problem considering owned fleet lateral transshipments (WT).

	B&C	P-B&C WOT	P-B&C WT
Optimally Solved	268	280	266
Avg Gap (%)	18.71	16.77	17.84
Avg Time (s)	26 358	27 119	28 432

Considering the case without transshipments (WOT), our P-B&C algorithm proves to be efficient compared to the B&C. Although the tests show larger average computational times, the number of instances that are optimally solved by the proposed approach is larger and the average gap is improved by 1.94%. When solving the IRP with owned fleet lateral transshipments (WT), the P-B&C obtains slightly larger gaps and less optimal solutions. This

was expected due to the additional complexity added by the lateral transshipments, which implies a larger solution space. The performance profiles comparing the three algorithms are depicted in Figure 3.7.

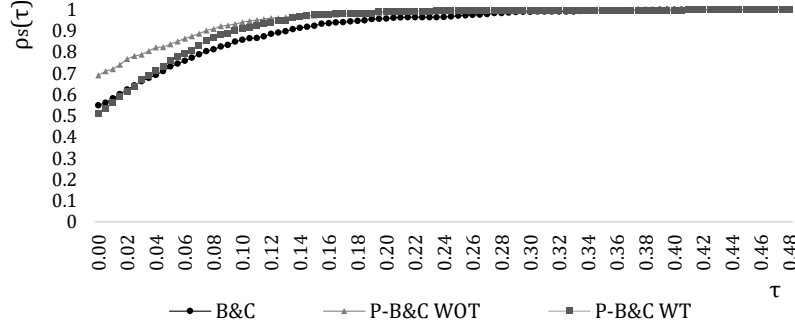


Figure 3.7 – The performance profiles show that the patching branch-and-cut algorithm is superior for solving IRP instances without owned fleet lateral transshipments. When owned fleet lateral transshipments are allowed, the heuristic is also competitive, although the number of optimal solutions found is the smallest.

The superiority of the P-B&C is clear when no lateral transshipments are considered, meaning that the probability of solving a certain problem with a lower relative gap is higher with our approach. When owned fleet lateral transshipments are considered, our approach finds less optimal solutions but for $\tau \in [0.03, 0.34]$ it shows a larger probability of obtaining smaller relative gaps. We provide more detailed results of our efficiency assessment tests in 3.A.

3.5.1.2 The value of owned fleet lateral transshipments in literature instances

Our proposed formulation extends the standard IRP by allowing vehicles to transship products between retail sites. All the solutions that are valid in the formulation without transshipments (WOT) are also valid in the formulation with transshipments (WT). Therefore, $f_{WT}^* \leq f_{WOT}^*$ holds, when f_{WT}^* and f_{WOT}^* are the optimal objective values of each formulation, respectively. Although this relation is trivial, the magnitude of the potential savings that can be obtained by considering owned fleet lateral transshipments in the IRP has not been measured. This analysis is different from the one presented in Coelho et al. [2012] as the lateral transshipments are performed by vehicles that are based at the supplier location. From the entire set of instances that have been tested, we compare the results of 264 runs that were able to achieve the optimal solution both in the WOT and in the WT case. Owned fleet transshipments have been performed in 83% of the instances. We compute the transshipped quantity over the total delivered quantity $\gamma = \sum P_i^{pt} / \sum D_i^{pt}$ to measure the percentage of deliveries that are served by transshipped products. Although transshipments are not used in 45 instances, the average value for the transshipped percentage γ is 2.40%. The maximum transshipped percentage achieved in an instance was 15.55%. In Table 3.2 we present the potential savings achieved by owned fleet lateral transshipments.

Table 3.2 – Summary of the results obtained in the 264 instances optimally solved by both models (WOT and WT). The table shows the minimum, maximum, and average savings ($\frac{WOT-WT}{WOT}$), obtained for every cost component.

	Total Cost	Transportation Cost	Supplier Inventory Cost	Retail Site Inventory Cost	Transshipped percentage γ
Max	13.25%	14.06%	9.92%	20.26%	15.55%
Avg	1.05%	0.57%	-0.47%	2.08%	2.40%
Min	0.00%	-18.53%	-14.61%	-24.31%	0.00%

In terms of objective function, the owned fleet lateral transshipments allowed for an average reduction of 1.05% in the total cost. These average savings are essentially due to a reduction in the inventories cost of the retail sites, given that the supplier inventory costs and the transportation costs are not largely affected. The largest total cost saving achieved in an instance was 13.25%. However, in the worst cases, there are instances where the transportation cost increases by 18.53% and the retail site inventory cost increases by 24.31%. Note that despite these large cost increases in each component separately, the total cost will never be larger in the WT case, as other cost components will certainly compensate (given that $f_{WT}^* \leq f_{WOT}^*$). This is confirmed since the minimum total cost saving achieved in an instance is zero.

We further explore the structure of the solutions obtained, depending on the type of instance considered. To do so, we ordered the instances by transshipped percentage γ and plotted bubble charts where each bubble is positioned according to an instance dimension (retail sites, vehicles, products or periods) and its size measures the saving percentage. Therefore, instances where owned fleet lateral transshipments have been largely used and large savings achieved are represented with a large bubble in the right side of the chart. In Figure 3.8 we position each bubble in the y-axis according to the number of retail sites of each instance.

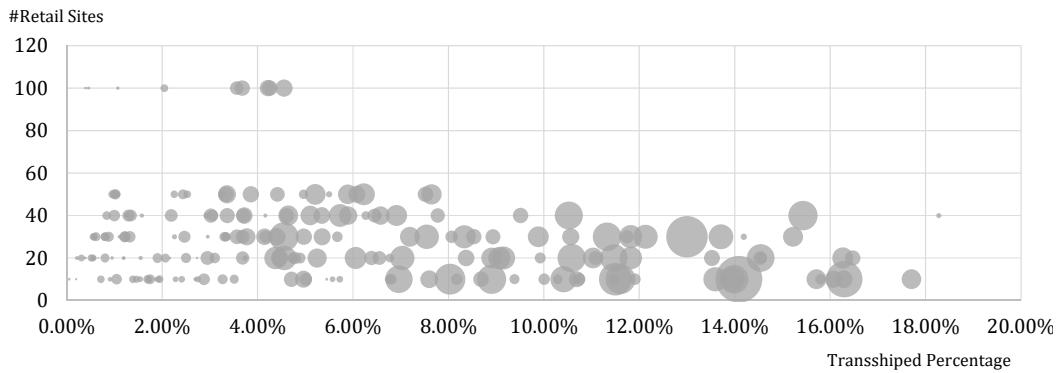


Figure 3.8 – The y-axis shows the number of retail sites of each instance. In the x-axis, instances are ordered according to the transshipped percentage γ obtained in the optimal solution to the P-B&C WT. The size of the bubble is proportional to the savings obtained relatively to the optimal solution obtained to the P-B&C WOT.

As expected, in instances where larger quantities are transshipped, larger saving percentages were obtained. Although larger savings are obtained for instances with less retail sites, we should not ignore the fact that less instances are solved to optimality when the number

of retail sites increases.

Figure 3.9 presents similar information as Figure 3.8, but now emphasizing the number of vehicles of each instance.

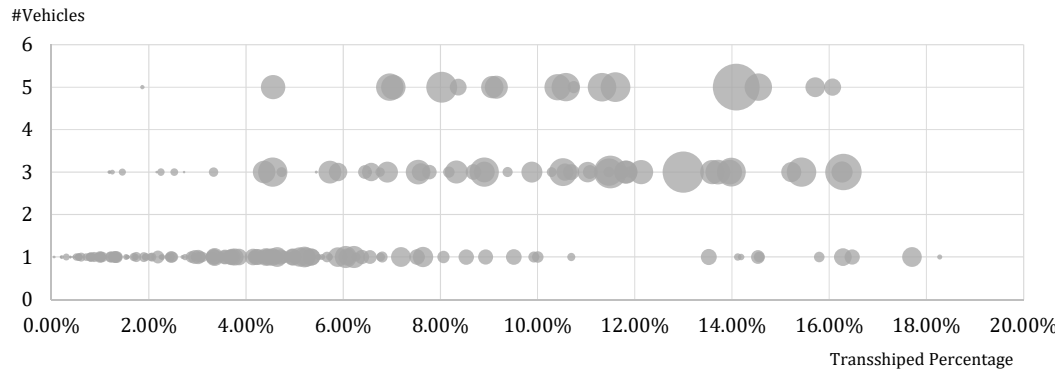


Figure 3.9 – The y-axis shows the number of vehicles of each instance (x-axis and bubble as in Figure 3.8).

Results seem to indicate that for a larger number of vehicles, the bubbles are larger and most frequently in the right part of the chart. In other words, the number of vehicles is positively correlated both with the transshipped percentage and savings percentage. With respect to the number of products, Figure 3.10 provides additional information.

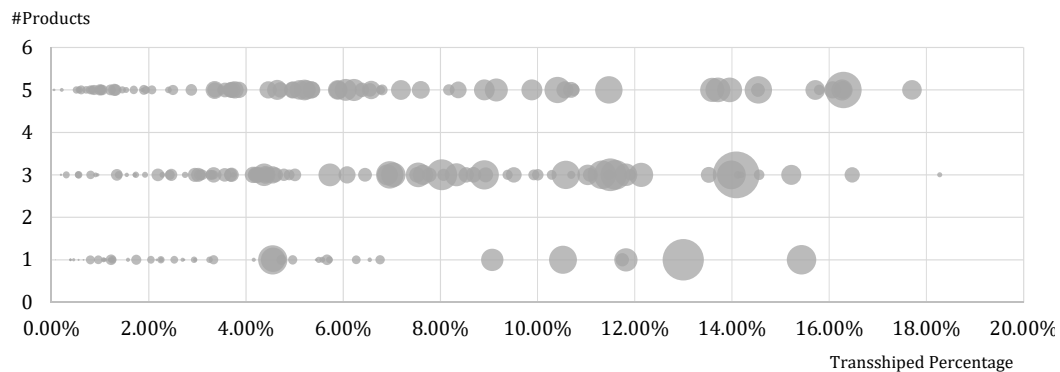


Figure 3.10 – The y-axis shows the number of products of each instance (x-axis and bubble as in Figure 3.8).

The conclusions we can derive from the number of products are similar to the ones taken for the number of vehicles. Indeed, the larger savings are found on instances with more than one product. It is interesting to note (despite not having a justification for it) that our approach was able to optimally solve more instances containing more than one product than with just one product.

Finally, the number of periods of each instance is represented in the y-axis of Figure 3.11.

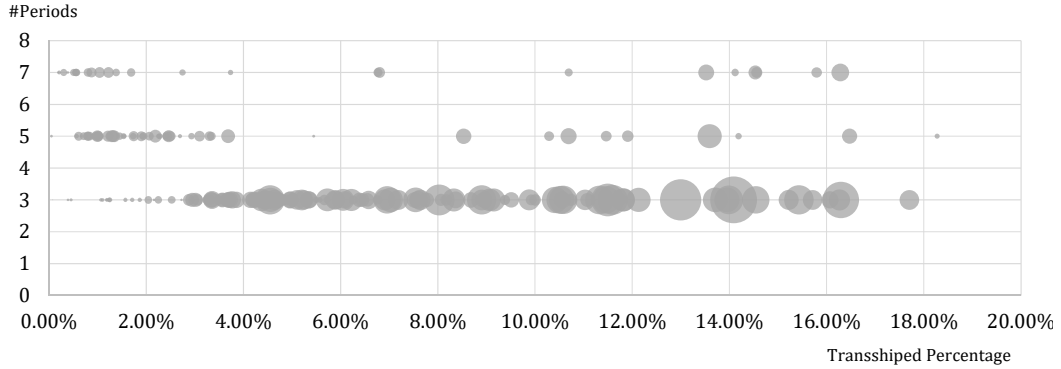


Figure 3.11 – The y-axis shows the number of periods of each instance (x-axis and bubble as in Figure 3.8).

The number of periods where more instances have been solved to optimality was three. Additionally, larger quantities are transshipped in these instances. This behaviour is likely to be related to the nature of our formulation. Since we do not define any rules for remaining inventories, it seems that in instances with longer planning horizons it is not profitable to reallocate inventories as they will eventually be needed later.

3.5.1.3 The value of owned fleet lateral transshipments in a rolling horizon planning context

The results presented in Section 3.5.1.2 show that owned fleet lateral transshipments are not largely used across all instances. The question of whether owned fleet lateral transshipments are useful in realistic setting stills remains unanswered. Our last group of experiments consists in simulating a realistic planning process to provide managerial insights on the IRP with owned fleet lateral transshipments. To do so, we test it in a rolling horizon planning context, performing a sensitivity analysis over three dimensions:

Planning periods Defining the number of periods to be considered in each planning iteration is not trivial. There is a trade-off between computational complexity and solution quality that needs to be taken into account when solving IRPs in a rolling horizon planning process. We test demand visibility L equal to 2, 3, and 4 periods.

Forecast errors In the real world, since demands are usually uncertain, inventory-routing planning is based on demand forecasts. The accuracy of these forecast deteriorates as the number of periods ahead to which the forecasts are made increases. Hence, considering the concepts presented in Section 3.4.2, we test the α parameter with the values 0.05, 0.10, and 0.20.

Holding cost magnitude The results obtained in the latter section showed that using lateral transshipments can lead to a total cost reduction of 1.05% on the tested literature instances. However, inventory costs are a small portion of the total cost in these instances. Therefore, to test different holding cost magnitudes, we adapt the original

holding costs of the literature instances by multiplying them by a factor m . In our tests, the holding cost multiplier m assumes the values 1 (original holding cost), 10, 20, and 30.

In each run, the plans need to satisfy the demand of ten periods ($T = 10$) coming from the adapted instances that have been extended for the rolling horizon case. For this test, we consider all the instances with 10 retail sites, 3 products and 3 vehicles. The instances are solved for the WOT and WT cases, using our P-B&C algorithm, and the improvement ($\frac{f_{WOT} - f_{WT}}{f_{WOT}}$) achieved from using owned fleet lateral transshipments is computed. The results of our computational experiments are presented in Table 3.3.

Table 3.3 – Summary of the computational results for the rolling horizon setting. The table shows the average improvement ($\frac{f_{WOT} - f_{WT}}{f_{WOT}}$) in the total cost, transportation cost, and inventory cost (supplier and retail sites), for each combination of holding cost multiplier, demand visibility periods, and α .

Holding Cost Multiplier (m)	#Periods (L)	$\alpha = 0.05$			$\alpha = 0.1$			$\alpha = 0.2$		
		TC	TRSP	IC	TC	TRSP	IC	TC	TRSP	IC
1	2	-7.57%	-8.27%	-1.06%	-4.00%	-4.49%	0.62%	-5.01%	-5.68%	1.32%
	3	-0.72%	-1.14%	2.08%	-1.83%	-2.65%	3.35%	0.87%	0.00%	5.21%
	4	-5.21%	-6.50%	1.56%	-4.31%	-5.20%	0.42%	-3.16%	-4.51%	3.95%
10	2	-3.31%	-7.67%	1.24%	-4.98%	-8.01%	-1.80%	-2.72%	-6.14%	0.94%
	3	0.95%	-2.68%	4.27%	1.65%	-1.28%	4.43%	1.21%	-1.77%	4.00%
	4	-1.68%	-7.06%	3.27%	-2.21%	-7.30%	2.54%	-1.64%	-8.12%	4.49%
20	2	3.13%	-8.17%	9.87%	2.84%	-8.05%	9.48%	3.71%	-7.96%	10.76%
	3	4.92%	-7.43%	12.12%	4.69%	-8.08%	12.19%	4.42%	-7.70%	11.67%
	4	2.06%	-7.80%	7.88%	1.85%	-5.93%	6.67%	1.92%	-6.69%	7.22%
30	2	3.39%	-9.19%	8.92%	6.32%	-7.81%	12.51%	5.35%	-8.85%	11.66%
	3	4.81%	-5.97%	9.62%	4.66%	-7.89%	10.26%	6.52%	-6.78%	12.44%
	4	4.19%	-7.03%	9.17%	4.98%	-5.82%	9.88%	4.88%	-5.89%	9.78%

Legend: TC - Total Cost TRSP - Transportation Cost IC - Inventory Cost

The results show that in a rolling horizon planning process under uncertain demand, owned fleet lateral transshipments are more beneficial when the inventory cost is high and when demand uncertainty is more severe. For the original holding costs ($m = 1$), inventory costs are generally reduced. When demand uncertainty increases (for larger values of α) larger inventory cost savings are achieved. However, in terms of total cost, the WT case only outperforms the WOT case when demand visibility is 3 periods and $\alpha = 0.2$ under $m = 1$. In our tests, we have not set any value for the final inventory, thus in each planning iteration the model tries to minimize inventory as much as possible, without considering the future iterations. In the WT case, it seems that the model is myopic and tries to anticipate some routing decisions that turn to be not so good in the later planning iterations. Analysing the results for the remaining values of m (10, 20, and 30), we observe that the total cost savings of WT over WOT increase proportionally to the value of m . It is not clear whether the number of periods impacts the magnitude of lateral transshipments. In most cases, the impact from considering the WT case is more beneficial with a visibility of 3 periods rather than with 4. The α parameter is not determinant as the lateral transshipments add almost no benefit when compared to the WOT case. Note that when m is greater than 10 the

WT operational model outperforms the WOT operational model in every case. Although the transportation costs of the WT operational model are larger, the inventory cost savings always compensate. Given so, these results suggest that owned fleet lateral transshipments are more likely to be beneficial in a rolling horizon planning process when the ratio $\frac{IC}{TC}$ is high. In Figure 3.12, we present the total cost savings achieved by the WT operational model in each instance. Instances are ordered first by the holding cost multiplier and then by the ratio $\frac{IC}{TC}$.

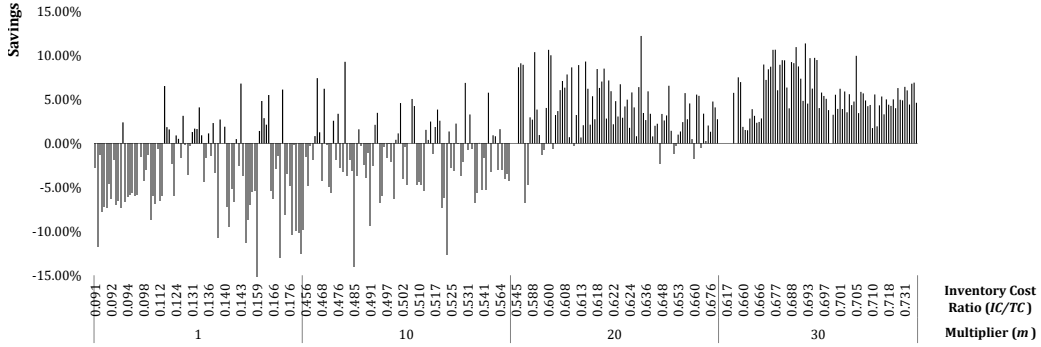


Figure 3.12 – The savings obtained by the WT operational model are larger when the total inventory cost assumes a large proportion of the total cost.

According to the results obtained for this set of instances, using lateral transshipments when $m \leq 10$ is not likely to be beneficial, as the lateral transshipments induced in the first planning iterations turn out to be expensive compared to the future savings on the inventory cost. When the inventory cost ratio $\frac{IC}{TC}$ is slightly larger than 0.6, it is more likely to achieve cost savings by performing owned fleet lateral (proactive) transshipments. This is in-line with what is stated by Paterson et al. [2011]. Reactive transshipments are more suitable to environments where the transshipment costs are low compared to the inventory costs. The authors refer the case of spare parts environments such as the semi-conductor sector. Finally, we present some routing indicators in Figure 3.13 to further compare the type of transportation solutions obtained by each operational model.

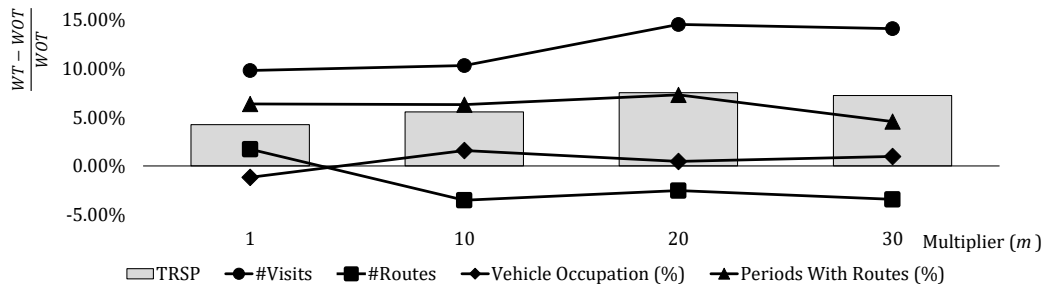


Figure 3.13 – The figure represents the average difference $\left(\frac{WT-WOT}{WOT}\right)$ of the number of visits per route, number of routes, vehicle occupation, and percentage of periods with routes (at least one vehicle is used) for each m . The transportation cost (TRSP) is also represented.

To analyse this figure, keep in mind that the magnitude of the savings obtained in the WT case increases proportionally to the value of m . The number of periods with routes increases by around 5% for every value of m . The vehicle occupation is similar in every m . Hence, it seems that the larger savings obtained by the WT case are not related to these indicators. The main difference between the results obtained for each value of m is in the reduction of the number of routes and in the increment on the number of visits of each route. Despite the increase in the transportation costs, in the WT case, longer routes performing lateral transshipments allow for better inventory cost. We conclude that the more important and frequent the necessity to reallocate inventories is (when holding costs are large), the more important it is to transship products between retail sites efficiently (with longer routes and higher vehicle occupation).

3.6. Conclusions

In this paper, the value of owned fleet lateral transshipments is studied for of the Inventory Routing Problem (IRP). A novel mathematical formulation generates integrated inventory and distribution plans that are able to reallocate inventory between entities of the same echelon. To solve the formulation, we propose a new exact branch-and-cut algorithm which repairs non-feasible routing solutions and performs routing local search procedures during the branching process. We improve the average gap of the best results available in the literature for multi-product multi-vehicle instances without considering transshipments, proving the effectiveness of our approach. Furthermore, we provide managerial insights on the IRP with owned fleet lateral transshipments both for deterministic demand, and for non-deterministic demand in rolling horizon planning setting. In the former, the average results obtained show that when 2.40% of the demand is served by lateral transshipments, we achieve a reduction of 1.05% on the total cost. In some instances, a total cost reduction of 13.25% is obtained and 15.55% of the total demand is transshipped. Hence, it is clear that some instances show a larger proneness to lateral transshipments. In the rolling horizon planning setting with non-deterministic demand, a set of instances with 10 retail sites, 3 vehicles, and 3 products is tested. For the original holding costs from the literature instances, the lateral transshipments do not reduce the total cost in the great majority of the instances. The performance of the model with lateral transshipments improves as the demand uncertainty increases, yet the weight of the holding costs is so small in the literature instances that lateral transshipments are not fundamental. For instances with larger holding costs, lateral transshipments allow for longer routes and better inventory reallocations. Usually, the transportation costs are higher but large savings in the inventory cost result in substantial savings in the total cost. Clearly, owned fleet lateral transshipments are more beneficial when the total inventory cost represents more than 60% of the total cost.

Acknowledgements

This research was partly supported by the PhD grant SFRH/BD/108251/2015, awarded by the Portuguese Foundation for Science and Technology (FCT) and by the project TEC4Growth

- Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020, financed by the North Portugal Regional Operational Program (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This support is gratefully acknowledged.

Bibliography

- Anholt, R. V., Coelho, L., Laporte, G., and Vis, I. (2013). An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines. Technical Report November, CIRRELT-2013-71, Monreal, Canada.
- Archetti, C., Bianchessi, N., Irnich, S., and Speranza, M. G. (2014). Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374.
- Bell, W., Dalberto, L., and Fisher, M. (1983). Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23.
- Clark, A. R. (2005). Rolling horizon heuristics for production planning and set-up scheduling with backlogs and error-prone demand forecasts. *Production Planning & Control*, 16(1):81–97.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548.
- Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.
- Coelho, L. C. and Laporte, G. (2013). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23-24):7156–7169.
- Coelho, L. C. and Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, 47(Supplement C):42 – 52.
- Cordeau, J.-F., Laganà, D., Musmanno, R., and Vocaturro, F. (2015). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55(Supplement C):153 – 166.
- Diabat, A., Abdallah, T., and Le, T. (2016). A hybrid tabu search based heuristic for the periodic distribution inventory problem with perishable goods. *Annals of Operations Research*, 242(2):373–398.
- Dolan, E. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

- Jemai, Z., Rekik, Y., and Kalaï, R. (2013). Inventory routing problems in a context of vendor-managed inventory system with consignment stock and transshipment. *Production Planning & Control*, 24(8-9):671–683.
- Liu, S.-C. and Chung, C.-H. (2008). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing*, 20(1):29–42.
- Liu, S.-C. and Lee, W.-T. (2011). A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications*, 38(10):13223–13231.
- Mes, M., Schutten, M., and Rivera, A. P. (2014). Inventory routing for dynamic waste collection. *Waste management (New York, N.Y.)*, 34(9):1564–76.
- Mirzapour Al-e-hashem, S. M. J. and Rekik, Y. (2014). Multi-product multi-period inventory routing problem with a transshipment option: A green approach. *International Journal of Production Economics*, 157(Supplement C):80 – 88.
- Papageorgiou, D. J., Keha, A. B., Nemhauser, G. L., and Sokol, J. (2014). Two-stage decomposition algorithms for single product maritime inventory routing. *INFORMS Journal on Computing*, 26(4):825–847.
- Paterson, C., Kiesmüller, G., Teunter, R., and Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. *European Journal of Operational Research*, 210(2):125–136.
- Ramkumar, N., Subramanian, P., Narendran, T. T., and Ganesh, K. (2012). Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *Opsearch*, 49(4):413–429.
- Rusdiansyah, A. and Tsao, D.-b. (2005). An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering*, 70(3):421–434.
- Salameh, M. K. and Jaber, M. Y. (1997). Reserve stock and transshipment in hierarchical inventory systems. *Production Planning & Control*, 8(5):469–474.
- Sethi, S. and Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, 29(1):387–415.
- Shen, Q., Chu, F., and Chen, H. (2011). A lagrangian relaxation approach for a multi-mode inventory routing problem with transshipment in crude oil transportation. *Computers & Chemical Engineering*, 35(10):2113–2123.
- Tiacci, L. and Saetta, S. (2011). Reducing the mean supply delay of spare parts using lateral transshipments policies. *International Journal of Production Economics*, 133(1):182–191.

Appendix 3.A Summary of computational results for the literature instances

Table 3.4 – Summary of computational results for the instances with 1 product.

#Vehicles	#Periods	#Customers	B&C			P-B&C WOT			P-B&C WT		
			Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)
1	3	10	0.00%	5	0	0.00%	5	0	0.00%	5	0
		20	0.00%	5	2	0.00%	5	1	0.00%	5	1
		30	0.00%	5	13	0.00%	5	10	0.00%	5	10
		40	0.00%	5	102	0.00%	5	151	0.00%	5	111
		50	0.00%	5	300	0.00%	5	190	0.00%	5	219
	5	10	0.00%	5	1	0.00%	5	1	0.00%	5	4
		20	0.00%	5	20	0.00%	5	38	0.00%	5	112
		30	0.00%	5	182	0.00%	5	1 116	0.00%	5	1 500
		40	0.00%	5	9 465	0.00%	5	8 978	0.00%	5	13 257
		50	0.56%	4	20 713	0.31%	4	14 806	0.81%	3	26 045
	7	10	0.00%	5	28	0.00%	5	80	0.00%	5	244
		20	0.00%	5	3 336	0.00%	5	8 439	1.00%	4	19 685
		30	3.06%	2	28 529	2.19%	3	25 958	3.45%	3	29 730
		40	6.03%	0	43 200	3.47%	0	43 200	5.10%	0	43 200
		50	9.97%	0	43 188	5.66%	1	39 508	10.30%	0	43 200
	3	10	0.00%	5	7	0.00%	5	15	0.00%	5	21
		20	0.00%	5	184	0.00%	5	143	0.00%	5	312
		30	0.00%	5	5 113	0.00%	5	3 135	0.00%	5	4 105
		40	2.92%	3	28 222	1.59%	4	17 428	0.00%	5	11 848
		50	19.44%	0	43 200	1.47%	3	24 464	2.93%	2	35 679
	5	10	0.22%	4	12 555	2.59%	3	28 446	3.48%	2	29 502
		20	17.04%	0	41 683	13.71%	0	43 200	16.84%	0	43 200
		30	31.16%	0	43 181	26.08%	0	43 200	29.39%	0	43 200
		40	42.11%	0	43 189	43.61%	0	43 200	42.91%	0	43 200
		50	48.12%	0	43 197	38.02%	0	43 200	43.88%	0	43 200
	7	10	14.40%	0	28 173	10.42%	0	33 580	14.09%	0	43 200
		20	27.67%	0	41 297	31.95%	0	43 200	32.95%	0	43 200
		30	33.60%	0	43 171	39.92%	0	43 200	40.40%	0	43 200
		40	61.12%	0	43 200	52.67%	0	43 200	52.61%	0	43 200
		50	62.17%	0	43 200	53.06%	0	43 200	54.73%	0	43 200
3	3	10	0.00%	5	120	0.00%	5	235	0.00%	5	279
		20	4.09%	2	29 770	3.27%	2	28 984	2.67%	3	25 446
		30	15.53%	0	43 200	13.64%	0	43 200	11.07%	0	43 200
		40	16.20%	0	43 190	14.59%	1	41 148	14.48%	1	42 779
		50	32.22%	0	43 189	28.49%	0	43 200	17.26%	0	43 200
	5	10	12.82%	1	35 354	12.81%	0	43 200	16.75%	0	43 200
		20	43.33%	0	43 200	48.07%	0	43 200	49.01%	0	43 200
		30	43.84%	0	43 197	49.34%	0	43 200	50.18%	0	43 200
		40	51.82%	0	43 198	52.33%	0	43 200	50.01%	0	43 200
		50	65.98%	0	43 200	58.19%	0	43 200	60.64%	0	43 200
	7	10	27.67%	0	40 106	24.82%	0	26 240	28.54%	0	43 200
		20	46.28%	0	43 175	55.50%	0	43 200	55.10%	0	43 200
		30	61.35%	0	43 191	55.76%	0	40 907	61.01%	0	42 537
		40	71.09%	0	43 200	53.37%	0	43 200	61.82%	0	43 200
		50	71.61%	0	40 688	60.79%	0	43 200	63.19%	0	43 200

Table 3.5 – Summary of computational results for the instances with 3 products.

#Vehicles	#Periods	#Customers	B&C			P-B&C WOT			P-B&C WT		
			Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)
1	3	10	0.00%	5	2	0.00%	5	0	0.00%	5	0
		20	0.00%	5	1	0.00%	5	2	0.00%	5	3
		30	0.00%	5	1	0.00%	5	16	0.00%	5	23
		40	0.00%	5	5	0.00%	5	52	0.00%	5	265
		50	0.00%	5	11	0.00%	5	153	0.00%	5	293
	5	10	0.00%	5	3	0.00%	5	12	0.00%	5	30
		20	0.00%	5	46	0.00%	5	122	0.00%	5	2 168
		30	0.00%	5	506	0.00%	5	2 489	0.00%	5	5 669
		40	0.00%	5	6 968	0.96%	4	23 803	1.18%	3	33 460
		50	2.72%	1	40 243	1.29%	2	34 381	1.62%	2	43 200
	7	10	0.00%	5	24	0.00%	5	91	0.00%	5	130
		20	0.00%	5	6 173	0.00%	5	4 207	0.00%	5	13 840
		30	5.41%	0	43 200	1.17%	3	25 152	2.05%	2	29 893
		40	11.04%	0	43 200	3.78%	0	43 200	4.65%	0	43 200
		50	15.15%	0	43 200	4.36%	0	43 200	3.96%	0	43 200
3	3	10	0.00%	5	6	0.00%	5	14	0.00%	5	32
		20	0.00%	5	323	0.00%	5	1 835	0.47%	4	9 197
		30	0.00%	5	3 575	0.00%	5	14 037	0.33%	4	17 470
		40	0.26%	4	12 953	0.00%	5	14 922	0.34%	4	21 876
		50	2.27%	2	35 502	6.27%	1	43 200	7.79%	1	40 206
	5	10	0.00%	5	8 008	1.34%	4	17 503	1.30%	3	25 584
		20	17.05%	0	42 202	11.77%	0	43 200	17.41%	0	43 200
		30	20.67%	0	43 196	23.78%	0	43 200	30.76%	0	43 200
		40	23.57%	0	43 189	24.83%	0	43 200	29.28%	0	43 200
		50	45.54%	0	18 555	32.91%	0	43 200	33.82%	0	43 200
	7	10	21.94%	0	27 184	13.15%	0	43 200	17.16%	0	43 200
		20	29.36%	0	43 198	27.82%	0	43 200	33.17%	0	43 200
		30	30.89%	0	43 198	34.54%	0	43 200	38.37%	0	43 200
		40	38.35%	0	43 200	42.38%	0	51 840	37.74%	0	43 200
		50	54.69%	0	43 200	43.04%	0	43 200	38.40%	0	43 200
5	3	10	0.00%	5	102	0.00%	5	385	0.00%	5	710
		20	2.46%	3	21 424	3.96%	2	29 007	2.46%	3	26 014
		30	3.81%	2	36 037	7.43%	1	38 343	4.03%	2	34 234
		40	2.36%	1	42 242	10.72%	0	43 200	5.29%	1	39 165
		50	9.45%	0	43 200	23.54%	0	43 200	17.43%	0	43 200
	5	10	19.39%	0	41 232	15.34%	0	41 582	17.30%	0	43 200
		20	26.04%	0	43 200	28.99%	0	43 200	32.96%	0	43 200
		30	37.85%	0	43 181	48.63%	0	43 200	47.40%	0	43 200
		40	51.09%	0	37 074	43.74%	0	43 200	48.02%	0	43 200
		50	47.08%	0	21 928	46.99%	0	43 200	47.77%	0	43 200
	7	10	30.18%	0	42 541	27.26%	0	37 858	28.30%	0	43 200
		20	41.00%	0	43 195	46.35%	0	43 200	48.63%	0	43 200
		30	60.65%	0	43 190	47.79%	0	43 200	53.28%	0	43 200
		40	61.09%	0	34 767	47.66%	0	43 200	52.32%	0	43 200
		50	56.43%	0	43 200	49.79%	0	43 200	47.86%	0	43 200

Table 3.6 – Summary of computational results for the instances with 5 products.

#Vehicles	#Periods	#Customers	B&C			P-B&C WOT			P-B&C WT		
			Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)	Gap (%)	#Solved	Time (s)
1	3	10	0.00%	5	5	0.00%	5	0	0.00%	5	1
		20	0.00%	5	1	0.00%	5	4	0.00%	5	9
		30	0.00%	5	2	0.00%	5	19	0.00%	5	49
		40	0.00%	5	6	0.00%	5	70	0.00%	5	333
		50	0.00%	5	19	0.00%	5	427	0.00%	5	2 618
	5	10	0.00%	5	7	0.00%	5	7	0.00%	5	14
		20	0.00%	5	95	0.00%	5	622	0.00%	5	1 057
		30	0.00%	5	5 670	0.00%	5	3 929	0.21%	4	19 747
		40	2.11%	2	32 089	0.13%	5	13 999	1.73%	3	19 368
		50	7.49%	0	43 194	1.88%	2	32 666	5.09%	2	38 046
	7	10	0.00%	5	54	0.00%	5	140	0.00%	5	256
		20	0.33%	4	18 774	0.00%	5	10 253	0.51%	4	15 104
		30	9.39%	0	43 195	1.23%	3	29 736	1.68%	2	36 070
		40	12.30%	0	43 200	2.40%	1	42 690	5.09%	0	43 200
		50	17.65%	0	43 195	5.99%	0	43 200	11.00%	0	43 200
3	3	10	0.00%	5	5	0.00%	5	6	0.00%	5	13
		20	0.00%	5	520	0.00%	5	3 120	0.00%	5	543
		30	0.81%	3	17 739	2.04%	3	19 614	0.86%	4	17 096
		40	0.71%	4	14 205	0.69%	4	20 258	2.47%	2	35 638
		50	1.14%	2	33 966	3.61%	2	43 200	5.58%	0	43 200
	5	10	5.10%	3	16 428	3.42%	3	21 483	3.89%	3	23 325
		20	15.28%	0	43 200	11.58%	0	43 200	18.60%	0	43 200
		30	22.49%	0	43 189	19.29%	0	43 200	25.78%	0	43 200
		40	30.40%	0	43 195	25.38%	0	43 200	31.78%	0	43 200
		50	45.56%	0	42 173	31.12%	0	43 200	30.17%	0	43 200
	7	10	22.77%	0	26 783	15.00%	0	43 200	15.37%	0	43 200
		20	30.34%	0	43 200	28.47%	0	43 200	27.51%	0	43 200
		30	30.84%	0	43 200	35.32%	0	43 200	34.94%	0	43 200
		40	62.96%	0	43 192	34.12%	0	43 200	36.67%	0	43 200
		50	55.65%	0	43 200	37.81%	0	43 200	32.03%	0	43 200
5	3	10	0.00%	5	64	0.00%	5	176	0.00%	5	310
		20	0.60%	4	9 813	1.65%	4	15 177	0.94%	4	24 624
		30	3.13%	2	37 412	7.04%	0	43 200	6.84%	1	39 597
		40	5.55%	0	43 199	18.64%	0	43 200	24.86%	0	43 200
		50	12.14%	0	41 010	25.77%	0	43 200	34.28%	0	43 200
	5	10	17.73%	0	43 122	11.13%	0	43 200	14.77%	0	43 200
		20	30.16%	0	43 200	29.53%	0	43 200	34.19%	0	43 200
		30	39.07%	0	43 200	39.13%	0	43 200	42.52%	0	43 200
		40	54.65%	0	43 200	42.10%	0	43 200	43.52%	0	43 200
		50	54.34%	0	41 345	44.33%	0	43 200	42.48%	0	43 200
	7	10	29.58%	0	42 114	21.67%	0	43 200	22.78%	0	43 200
		20	40.03%	0	43 200	42.02%	0	43 200	47.76%	0	43 200
		30	44.05%	0	43 185	48.31%	0	43 200	47.29%	0	43 200
		40	54.70%	0	43 200	47.25%	0	43 200	54.47%	0	43 200
		50	55.06%	0	43 200	46.65%	0	43 200	51.33%	0	43 200

Production-Routing

Solving a large multi-product production-routing problem with delivery time windows

Fábio Neves-Moreira* · Bernardo Almada-Lobo* · Jean-François Cordeau† · Luís Guimarães† · Raf Jans†

Under review in *Omega*, 2017

Abstract Even though the joint optimization of sequential activities in supply chains is known to yield significant cost savings, the literature concerning optimization approaches that handle the real-life features of industrial problems is scant. The problem addressed in this work is inspired by industrial contexts where vendor-managed inventory policies are applied. In particular, our study is motivated by a meat producer whose supply chain comprises a single meat processing centre with several production lines and a fleet of vehicles that is used to deliver different products to meat stores spread across the country. A considerable set of characteristics, such as product family setups, perishable products, and delivery time windows, needs to be considered in order to obtain feasible integrated plans. However, the dimensions of the problem make it impossible to be solved exactly by current solution methods. We propose a novel three-phase methodology to tackle a large Production Routing Problem (PRP) combining realistic features for the first time. In the first phase, we attempt to reduce the size of the original problem by simplifying some dimensions such as the number of products, locations and possible routes. In the second phase, an initial PRP solution is constructed through a problem decomposition comprising several inventory-routing problems and one lot-sizing problem. In the third phase, the initial solution is improved by different mixed-integer programming models which focus on small parts of the original problem and search for improvements in the production, inventory management and transportation costs. Our solution approach is tested both on simpler instances available in the literature and on real-world instances containing additional details, specifically developed for a European company's case study. By considering an integrated approach, we achieve global cost savings of 21.73% compared to the company's solution.

Keywords production-routing · time windows · perishability · matheuristic · case study

*INESC TEC and Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

†HEC Montréal and CIRRELT, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, Canada

4.1. Introduction

Competitiveness is intimately connected with the ability to be prepared, to predict market changes, and to be ready for adopting fast changes in various operations. Today, competitiveness is being considered at a supply chain level because no single organizational unit is solely responsible for the competitiveness of its products and services [Stadtler and Kilger, 2010]. Provided that companies are aware of the latter facts, the receptiveness for systematic and optimized planning processes has been steadily increasing in the last few years. A recent trend in operations research is to integrate and coordinate various planning problems in order to obtain better plans, trading-off pros and cons from a much wider and inclusive perspective [Pochet and Wolsey, 2006]. Indeed, in the past, most optimization processes comprised a number of problems that were solved sequentially. In most cases, the effects caused by decisions fixed in previous stages of the planning process were not even measured. Therefore, most entities were not aware that optimizing a certain problem could prevent the achievement of better solutions for subsequent problems and, consequently, a better global solution for the whole planning process.

The literature has been reporting various experiments in which considerable cost savings arise from planning integration [Chandra and Fisher, 1994, Thomas and Griffin, 1996, Fumero and Vercellis, 1999]. Activities are optimized with a monolithic model, as opposed to sequentially optimizing parts of the global problem (Figure 4.1). Logically, the impressive results obtained by monolithic models contributed to augment the interest in integrated planning processes. However, the enormous set of decisions and factors to be considered often results in intractable problems [Schmid et al., 2013].

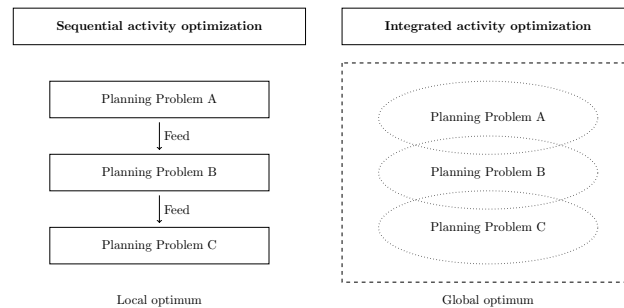


Figure 4.1 – Sequential activity optimization and integrated activity optimization planning processes. Integrated planning allows for lower global cost but poses a more difficult problem to solve.

The Production Routing Problem (PRP) is an integrated planning problem which jointly optimizes production, inventory management and transportation decisions. The integration of these activities is particularly interesting in a Vendor Managed Inventory (VMI) context where inventories held at retail sites are managed by a single entity, usually the supplier of the products. The supplier produces a set of products deciding whether it sends them directly to be stocked at retail sites or if it stocks them in its own warehouse in order to distribute them at later periods. Therefore, as it is shown in Figure 4.2, typical decisions comprised by the PRP include (1) when and how much to produce; (2) when and how much

to deliver to each retail site; (3) how to route vehicles such that production, inventory and transportation costs are minimized while meeting retail sites' demand. These decisions are usually to be made during a planning horizon composed of several days.

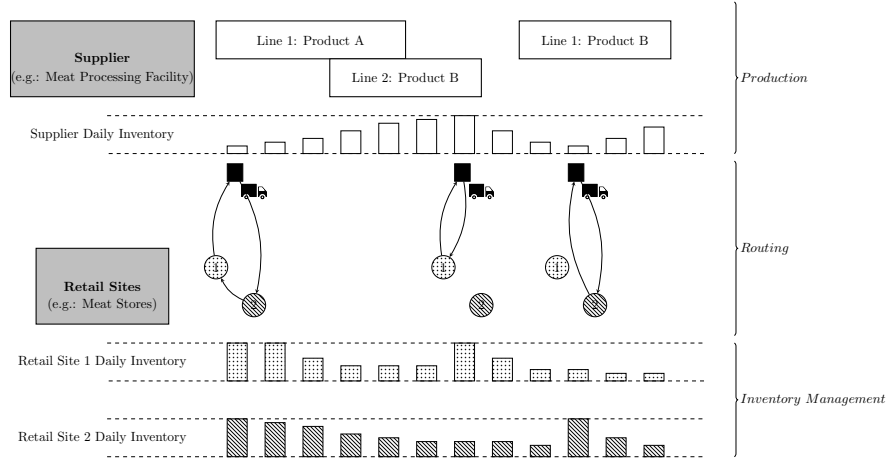


Figure 4.2 – Decisions to be made in a PRP. Productions, Inventory management, and transportation are coordinated to deliver cost-effective solutions from a monolithic perspective.

This problem is extremely challenging as it integrates two classic optimization problems, the Lot Sizing Problem (LSP) and the Vehicle Routing Problem (VRP), which were proposed by [Wagner and Whitin \[1958\]](#) and by [Dantzig and Ramser \[1959\]](#), respectively. Similarly to its particular case where production is not taken into account, the Inventory Routing Problem (IRP), the PRP is NP-hard, as it demands the solution of several VRPs [[Adulyasak et al., 2014c](#)]. However, [Chandra and Fisher \[1994\]](#) reported 3 to 20% cost savings coming from integration, which is a strong motivation to study the PRP in the context of realistic and large planning problems.

We aim to propose a novel mathematical formulation considering product and family setups when producing perishable products that are to be delivered to retail sites within certain time windows. The set of features comprised in the model allows great flexibility, which enables the possibility to apply it to several process industries. However, the inherent complexity of such a PRP requires the development of advanced solution methods in order to efficiently explore its large and heterogeneous structure. The proposed solution approach consists of three main phases comprising a size reduction phase, an initial solution construction phase, and an improvement phase based on a fix-and-optimize (F&O) approach.

The developed algorithm is first tested on standard PRP instances available in the literature and then on real-world instances considered in the case study of a European meat store chain that motivated the underlying research. The real-world challenge clearly shows the necessity of considering sets of realistic features in the context of the PRP where any kind of extension is rare to be found in the literature. Additionally, the current practice of the company is based on decoupled plans considering demand for a single period ahead. Therefore, we devise an integrated approach so as to show its advantages with respect to

various practical dimensions.

The contributions of this paper are fourfold:

- We propose a novel mathematical formulation for the PRP with realistic features including multiple vehicles performing routes with time windows, multiple perishable products, and multiple production lines with different specifications.
- We present a decomposition approach for large PRPs where available data are used to reduce the problem size and divide the original problem into several tractable subproblems. The decomposition requires the solution of several IRPs and one LSP.
- We provide an improvement matheuristic based on a F&O scheme, capable of exploring the solution space of large PRPs which are intractable for the best general-purpose solvers available.
- The algorithm is tested on multi-product multi-vehicle instances available in the literature and validated with real-world instances belonging to the case study of a European vertical meat store chain.

The remainder of this paper is organized as follows. In Section 4.2 a literature review is presented, focusing on different extensions and real-world applications of planning problems integrating production, inventory management and distribution. Section 4.3 gives a description of the general PRP to be tackled in this work. Section 4.4 presents the proposed solution approach to efficiently solve large PRPs. The computational experiments obtained in the literature instances are discussed in Section 4.5. Section 4.6 details the case study of a meat store chain. Finally, Section 4.7 summarizes the main achievements and conclusions, pointing out future research opportunities.

4.2. Literature Review

In this research, we are particularly interested in the PRP, which integrates production, inventory management and routing decisions [Adulyasak et al., 2014c]. Chandra and Fisher [1994] are amongst the first attempts to integrate production decisions with distribution, reporting considerable gains coming from activity integration. Since then, several authors addressed supply chain coordination [Thomas and Griffin, 1996, Fumero and Vercellis, 1999, Bard and Nananukul, 2008, Boudia et al., 2007, 2008, Boudia and Prins, 2009] and there are some works reporting case studies where large gains were obtained [Çetinkaya et al., 2009, Brown et al., 2001]. For a review on the origins of the PRP, the reader is referred to the work of Sarmiento and Nagi [1999]. Although there are some PRP variants, we focus on problems where lot sizing, inventory management, and explicit routing decisions are integrated at an operational level.

Lei et al. [2006] study a production, inventory, and distribution routing problem and propose a two-phase solution approach. In the first phase, only direct shipments are considered and in the second phase, an associated consolidation problem is solved in order to eliminate direct shipment inefficiencies. Unlike most uncoupled approaches, this one

does not completely separate production from distribution. Bard and Nananukul [2009] present a comparative analysis of various heuristics for the PRP. Since optimal solutions could not be achieved with exact methods, a two-step procedure first allocates daily delivery quantities and solves a VRP afterwards. They show that the IRP component can be solved efficiently with a branch-and-price framework. Absi et al. [2015] propose a heuristic which iteratively focuses on production and distribution decisions. The production part is modelled as a Capacitated Lot Sizing Problem (CLSP) whereas the daily distribution is modelled as a Travelling Salesman Problem (TSP) or a VRP. A single-item PRP under the Maximum Level (ML) policy is studied. Both the single and multiple vehicle versions are explored. Computational results show that this solution approach outperforms previously proposed methods. Armentano et al. [2011] use a combination of Tabu Search (TS) and a Path Relinking (PR) procedure and test it on instances with up to 10 products. They introduce a mathematical model which is able to consider multiple vehicles and products but resort to a phased approach where a production problem is solved before an unlimited fleet distribution problem. Instances with up to 15 retail sites, 14 periods and five products are solved with less than 2% gaps based on CPLEX optimal solutions. While CPLEX needs more than four hours to compute the solutions, the heuristic only needs less than 30 seconds. Adulyasak et al. [2014b] introduce an Adaptive Large Neighbourhood Search (ALNS) heuristic for the PRP which handles setup variables by an enumeration scheme and routing variables with upper-level search operators. The continuous variables are computed in a network flow subproblem. The proposed algorithm outperforms existing approaches, computing superior quality solutions in short amounts of time. Computational tests were performed on instances with up to 200 retail sites, 13 vehicles and 20 periods. Recently, Qiu et al. [2017] address a pollution PRP considering carbon emissions. The authors propose a mathematical formulation and solve it by means of a branch-and-price heuristic. Managerial insights are provided for instances with 14 retail sites and 6 periods, showing that it is possible to reduce carbon emissions and operational costs simultaneously.

The above-mentioned papers consider heuristic methods to tackle the integrated planning problems. Given the complexity of integrated planning, these are still the most suitable methods to address large instances of this problem. Exact approaches are quite rare in the context of the PRP and the size of the problems that are possibly solved is obviously smaller. Their complexity seems to have repelled researchers from studying them. Indeed, few exact methods are found in the literature and the problems they address are usually simplified in the number of entities or in the type of features considered. Solyali and Süral [2009] propose a mathematical formulation and a Lagrangian relaxation based approach to solve a single-vehicle, single-product PRP. Although the supplier is able to decide order quantities, the problem is simplified because the production activity is uncapacitated. The authors tackle instances with up to 50 retail sites and 30 periods with constant demands. Bard and Nananukul [2009, 2010] propose a branch-and-price algorithm for a PRP. This algorithm is a combination of a Dantzig-Wolfe (D-W) decomposition with traditional Branch-and-Bound (B&B). Different methods to obtain initial solutions, branching rules and rounding schemes are tested. After tuning, their approach is able to solve instances with up to 50 retail sites and 8 periods. Ruokokoski et al. [2010] present efficient formulations and a branch-and-cut algorithm for the PRP. They consider a problem

with uncapacitated plants and vehicles and test new formulations strengthened by valid inequalities. Randomly generated instances with up to 40 customers and 15 periods or with 80 customers and eight periods are solved to optimality. A heuristic algorithm is also tested, obtaining solutions with an average cost increase of 0.33% within less than 1% of the time required by the exact approach. Archetti et al. [2011] develop a hybrid heuristic for the PRP with ML policy which quickly obtains high quality solutions by solving the production and distribution problems sequentially. The authors compare the single-vehicle case with an exact solution method based on a branch-and-cut scheme. Note that the latter two approaches only create delivery routes separately, after deciding the quantities to be delivered to each retail site. Adulyasak et al. [2014a] tackle the multi-vehicle PRP. The authors present vehicle and non-vehicle index formulations strengthening them using symmetry breaking constraints and several cuts, respectively. An ALNS heuristic is used to build initial solutions which are improved by a branch-and-cut algorithm. Instances with around 30 customers, 3 vehicles, and 3 periods are solved both for the Order-up-to Level (OU) and ML policies. Adulyasak et al. [2015] tackle a PRP with uncertain demands. Production setups and customer visit schedules are defined in the first stage while the production and delivery quantities are determined in subsequent stages. A Benders decomposition approach using lower-bound lifting inequalities, scenario cuts and Pareto-optimal cuts is implemented. Besides this paper, few other papers deal with PRPs with uncertainty. Adulyasak et al. [2014c] provide a review of formulations and suggest future research directions regarding the PRP.

The importance of considering realistic constraints is undeniable in several industries. However, the literature is still scarce when dealing with realistic integrated planning at an operational level. Furthermore, despite their increasing popularity, matheuristics for the PRP are still rare. We provide a summary of the algorithms proposed in the last decade in Table 4.1.

Table 4.1 – Main recent PRP approaches

Authors	Production			Inventory		Routing		Solution Method	
	#Plants	#Prod	C.	Policy	C.	#Vehicles	C.	Type	Approach
Lei et al. [2006]	Multiple	Single	✓	ML		Limited (Het)	✓	H	Decomposition
Boudia et al. [2007]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	GRASP
Boudia et al. [2008]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Decomposition
Bard and Nananukul [2008]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Tabu search
Boudia and Prins [2009]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H	Memetic algorithm
Solyali and Süral [2009]	Single	Single		OU	✓	Limited (Hom)	✓	H/L	Lagrangian relaxation
Bard and Nananukul [2009, 2010]	Single	Single	✓	ML	✓	Limited (Hom)	✓	H/L	Branch-and-price
Ruokokoski et al. [2010]	Single	Single		ML		Single		E	Branch-and-cut
Armentano et al. [2011]	Single	Multiple	✓	ML	✓	Limited (Hom)	✓	H	Tabu search / Path relinking
Archetti et al. [2011]	Single	Single		ML/OU	✓	Single	✓	E/H	Branch-and-cut / MIP heuristic
Adulyasak et al. [2014b]	Single	Single	✓	ML	✓	Multiple (Hom)	✓	H	ALNS
Adulyasak et al. [2014a]	Single	Single	✓	ML/OU	✓	Multiple (Hom)	✓	E/H	Branch-and-cut / ALNS
Absi et al. [2015]	Single	Single	✓	ML/OU	✓	Multiple (Hom)	✓	H	Iterative MIP heuristic
Adulyasak et al. [2015]	Single	Single	✓	ML	✓	Multiple (Hom)	✓	E	Benders-based branch-and-cut
Our approach	Single	Multiple	✓	ML	✓	Multiple (Het)	✓	MH	Fix-And-Optimize

Legend:

Prod. - Products | C. - Capacitated | Hom. - Homogeneous | Het. - Heterogeneous | H - Heuristic | E - Exact | MH - Matheuristic | L - Lower bound computation

4.3. Problem Description and Formulation

In this section, we first define the problem in general terms. We start by describing the relevant entities as well as the parameters that are related to them. Afterwards the constraints to capture production and routing decisions are presented, as well as the linking constraints to connect both dimensions of the problem.

4.3.1 Notation

The PRP considered in this paper can be defined on a complete directed graph $G = (V, A)$ where V represents a set of locations including a supplier and retail sites indexed by $i \in \{0, \dots, n\}$, and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the set of arcs. The supplier manages the set of retail sites $\mathcal{V}' = V \setminus \{0\}$ applying a VMI policy and it owns a set of production lines $\mathcal{M} = \{0, \dots, |\mathcal{M}|\}$ which are used to produce a set of product families $\mathcal{F} = \{1, \dots, |\mathcal{F}|\}$ over a finite horizon $\mathcal{T} = \{1, \dots, |\mathcal{T}|\}$. Each family is composed of several products which belong to the entire set of products $\mathcal{P} = \{1, \dots, |\mathcal{P}|\}$ to be delivered by the supplier. Each line m can only produce some products belonging to the set \mathcal{P}_m . Deliveries are made by a fleet of vehicles $\mathcal{K} = \{1, \dots, |\mathcal{K}|\}$ that is owned by the supplier. Each vehicle is able to perform one route in each shipping period $t \in \mathcal{T}$. The quantity of each product available in each retail site is used to satisfy the demand of each consumption period $l \in \mathcal{T}$. The parameters and decision variables included in our formulation are the following:

Parameters	Production
	u_{mg} production capacity of line m in production period g (time);
	a_{fm} setup time for family f on line m (time);
	b_{pm} setup time for product p on line m (time);
	p_{pm} processing time of product p on line m (time);
	Inventory
	\bar{c}_i capacity of the warehouse at location i ;
	h_{ip} holding cost of product p at location i ;
	d_{ipl} demand for product p at location i in consumption period l ;
	Routing
	v_k capacity of vehicle k ;
	t_{ij} travel time between i and j ;
	c_{ij} travel cost between i and j ;
	$[a_i, b_i]$ service time window of location i .

Decision Variables	Production
A_{mfg}	equal to 1 if line m is set up to produce family f in production period g , 0 otherwise;
B_{mpg}	equal to 1 if line m is set up to produce product p in production period g , 0 otherwise;
P_{pmgt}	quantity of product p produced on line m in production period g to be delivered in shipping period t ;
Routing	
X_{ijkt}	equal to 1 if arc (i, j) is traversed by vehicle k in shipping period t , 0 otherwise;
Z_{ikt}	equal to 1 if location i is visited by vehicle k in shipping period t , 0 otherwise;
W_{ikt}	arrival time of vehicle k at location i in shipping period t ;
D_{ipklt}	quantity of product p to be delivered to location i by vehicle k in shipping period t to be consumed in consumption period l .

4.3.2 Formulation

We assume an ML policy and that quantities produced in period g are ready to be delivered in period $g + 1$. Quantities received at each retail site are ready to be consumed in the period they are shipped. M is a big number. Given this, the multi-product PRP with time windows can be formulated as follows:

(MPRPTW):

$$\begin{aligned}
\text{minimize } & \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{g \in \mathcal{T}} \sum_{h=0}^g \sum_{t=g+1}^{|\mathcal{T}|} h_{0p} \cdot P_{pmht} \\
& + \sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} \sum_{h=0}^t \sum_{l=t+1}^{|\mathcal{T}|} h_{ip} \cdot D_{ipkhl} \\
& + \sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} c_{ij} \cdot X_{ijkt}
\end{aligned} \tag{4.1}$$

s.t.

$$B_{mpg} \leq A_{mfg} \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, g \in \mathcal{T} \tag{4.2}$$

$$\sum_{t=g+1}^{|\mathcal{T}|} P_{pmgt} \leq M \cdot B_{mpg} \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, g \in \mathcal{T} \tag{4.3}$$

$$\sum_{f \in \mathcal{F}_m} a_{fm} \cdot A_{mfg} + \sum_{p \in \mathcal{P}_m} (b_{pm} \cdot B_{mpg} + \sum_{t=g+1}^{|\mathcal{T}|} p_{pm} \cdot P_{pmgt}) \leq u_{mg} \quad \forall m \in \mathcal{M}, g \in \mathcal{T} \quad (4.4)$$

$$\sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{h=0}^g \sum_{t=g+1}^{|\mathcal{T}|} P_{pmht} \leq \bar{c}_0 \quad \forall g \in \mathcal{T} \quad (4.5)$$

$$\sum_{g=0}^{t-1} \sum_{m \in \mathcal{M}} P_{pmgt} = \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} \sum_{l=t}^{|\mathcal{T}|} D_{ipktl} \quad \forall p \in \mathcal{P}, t \in \mathcal{T} \quad (4.6)$$

$$\sum_{t=0}^l \sum_{k \in \mathcal{K}} D_{ipktl} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, l \in \mathcal{T} \quad (4.7)$$

$$\sum_{v \in \mathcal{K}} \sum_{p \in \mathcal{P}} \sum_{h=0}^t \sum_{l=t}^{|\mathcal{T}|} D_{ipkhl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \quad (4.8)$$

$$\sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{l=t}^{|\mathcal{T}|} D_{ipktl} \leq v_k \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \quad (4.9)$$

$$D_{ipktl} \leq \min\{d_{ipl}, v_k\} \cdot Z_{ikt} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T}, l \in \mathcal{T} \quad (4.10)$$

$$\begin{aligned} \sum_{j \in \mathcal{V}} X_{ijkt} &= Z_{ikt} \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \\ \sum_{j \in \mathcal{V}} X_{jik} &= Z_{ikt} \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \end{aligned} \quad (4.11)$$

$$\sum_{k \in \mathcal{K}} Z_{ikt} \leq 1 \quad \forall i \in \mathcal{V}, t \in \mathcal{T} \quad (4.12)$$

$$W_{ikt} + t_{ij} \leq W_{jkt} + M \cdot (1 - X_{ijkt}) \quad \forall i \in \mathcal{V}', j \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \quad (4.13)$$

$$a_i \leq W_{ikt} \leq b_i \quad \forall i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T} \quad (4.14)$$

$$X_{ijkt}, Z_{ikt}, A_{mfg}, B_{mpg} \in \{0, 1\}; \quad P_{pmt}, D_{ipktl}, W_{ikt} \geq 0. \quad (4.15)$$

The objective function (4.1) minimizes the holding cost at the supplier and retail sites as well as the routing cost. Constraints (4.2) - (4.5) model the lot sizing part of the problem. Constraints (4.2) and (4.3) impose that each production line can only produce if the family and product set up operations have been performed. Constraints (4.4) model the time capacity for each production line. Constraints (4.5) impose the maximum capacity for the inventory at the supplier. Constraints (4.6) ensure that the quantities produced of each product satisfy the deliveries made in each shipping period to each retail site, linking the production and routing decisions. Constraints (4.7) - (4.14) define the routing part of the problem. Constraints (4.7) ensure the demand satisfaction of each retail location. Constraints (4.8) are inventory management constraints to impose the maximum capacity for the inventory at each retail site. Constraints (4.9) are added to the formulation to impose vehicle capacities. Constraints (4.10) ensure that a location can only receive products if it is visited. Constraints (4.11) are the so-called vehicle flow conservation constraints. The degree constraints (4.12) impose at most one visit per location in each period. In order to define arrival times at each location for each vehicle, constraints (4.13) are added. Finally, these constraints also serve as subtour elimination constraints. Time windows are imposed by constraints (4.14). This formulation has been tested for toy instances but, when the number of locations, vehicles, and products grows, the problem becomes intractable. In fact, just enumerating the set of D_{ipktl} variables is already a very time-consuming task. For that reason, and considering the ultimate objective of solving real-world instances, the next section details a new matheuristic approach for large PRPs with realistic constraints.

4.4. Solution Approach

The proposed solution approach was designed by considering the dimension and complexity of the instances to be solved in a real-world context. Indeed, various decomposition mechanisms are added to ensure that the problem can be solved, regardless of its size.

The first phase (**Size Reduction**) attempts to decrease the granularity of the problem to largely reduce computational times in later phases (or make the problem tractable). Reduced subproblems are derived to serve as inputs for the following procedures. Retail sites with similar geographic location and time windows are clustered into hypernodes (a hypernode is a set of retail sites to be visited in a given sequence). Afterwards, the obtained set of hypernodes is partitioned into regions (sets of hypernodes) to obtain tractable subproblems. A reduced set of routes (visiting hypernodes) is created for each region, taking advantage

of a smaller number of locations (hypernodes) to reduce the complexity of the routing decisions. Products with similar production specifications and low demand are aggregated in hyperproducts (sets of products), as it is assumed that smaller priority should be given to them.

In the second phase (*Initial Solution*), the algorithm solves a set of subproblems. Each subproblem corresponds to an IRP where delivery quantities and routes are defined for each region. When a solution for the IRP subproblems is obtained, we also obtain a delivery schedule for the original problem. Afterwards, an LSP is solved to define the setup decisions and production quantities at the supplier, taking the delivery schedule as an input. This procedure builds an initial decoupled production-routing plan.

The third phase (*Improvement*) aims at improving the initial solution. Since the initial solution is obtained by partitioning the original problem, we seek to find solution improvements by relating variables describing entities that have been initially aggregated or separated in different subproblems. To do so, Mixed-Integer Program (MIP) formulations are used to improve different parts of the PRP solution.

The three phases are described in detail in the following subsections.

4.4.1 Size Reduction

4.4.1.1 Node Clustering And Region Decomposition

We propose a node clustering algorithm (Algorithm 1) to create hypernodes. The algorithm mainly focuses on geographical and time aspects. The order in which nodes are inserted into the hypernode defines the order in which each node should be served when a vehicle visits the hypernode. Hence, each hypernode has an associated path. The procedure needs two parameters: a maximum distance d_{max} and a maximum duration t_{max} . Nodes are first ordered by their earliest service time a_i and then by their latest service time b_i . From this ordered list of candidates, we take the first element to be visited, the hypernode seed, and open a new hypernode. Afterwards, a list of candidates to join the currently opened hypernode is created. A node is a candidate if, after its insertion, the following criteria are met:

1. the travel distance of the hypernode path (following the insertion order) is less than d_{max} ;
2. the travel time of the hypernode path (following the insertion order) is less than t_{max} ;
3. the time windows of all nodes in the hypernode are respected.

From the list of candidates obtained by applying these criteria, the nearest candidate is selected. Nodes are added to the hypernode until no candidates are available. At that point, a new hypernode is opened with the first element of the remaining ordered nodes.

Algorithm 1 Node Clustering

```

1: procedure CLUSTERNODES( $Nodes, d_{max}, t_{max}$ )
2:    $hn \leftarrow$  Open a new hypernode
3:   while  $Nodes \neq \emptyset$  do
4:      $n \leftarrow$  Get a candidate from the set  $Nodes$ , considering the hypernode  $hn$ ,  $d_{max}$ , and  $t_{max}$ 
5:     if  $n \neq \emptyset$  then
6:        $hn \leftarrow$  Add the candidate node  $n$  to the currently opened hypernode  $hn$ 
7:       Remove the added node  $n$  from the set  $Nodes$ 
8:     else
9:        $Hypernodes \leftarrow$  Add the hypernode  $h$  to the set  $Hypernodes$ 
10:       $hn \leftarrow$  Open a new hypernode
11:   return  $Hypernodes$ 

```

Algorithm 1 defines a set of hypernodes that will be used in the initial solution phase. This set of hypernodes can still be intractable in the following phase, depending on its size. For that reason, the entire set of hypernodes can be decomposed into subsets called regions. To define the set of regions, we use a K-means algorithm [Hartigan and Wong, 1979] based on the distance between the centroids of each hypernode. Since we incorporate some routing aspects in the node clustering and region decomposition steps, the set of routes to be considered, visiting hypernodes, can only be generated after these steps. This is an advantage as the resulting set of routes of each region becomes more compact.

4.4.1.2 Route Generation

In order to create a new set of routes serving hypernodes, we use a simple procedure (Algorithm 2), as further route improvements can be performed later. The goal of this step is to generate a set of routes that is sufficiently rich to build good feasible solutions yet small enough to be solved using a general-purpose solver to tackle a set-partitioning formulation. The procedure uses four parameters: a set of depots $Depots$, a set of locations to be visited $Vertices$ (i.e., hypernodes), the maximum number of visits per route $maxVisits$, and the maximum duration of each route $maxDuration$. For each hypernode, the $n_{Nearest} = maxVisits - 1$ nearest neighbours are selected (line 4, Algorithm 2). Then, we create every possible combination containing the considered hypernodes. This means that we will create sets with 1 to $maxVisits$ hypernodes. Afterwards, for each depot and each set of hypernodes, we create a route using a cheapest insertion algorithm (line 12, Algorithm 2). After randomly choosing a hypernode to be the seed, the algorithm analyses each hypernode in the set and tries to insert it in the cheapest position. If it is not possible to insert the hypernode, the next hypernode is analysed. All the generated routes obey the time windows and maximum route duration constraints.

Algorithm 2 Route Creator

```

1: procedure CREATEROUTES(Depots, Vertices, maxVisits, maxDuration)
2:    $nNearest \leftarrow maxVisits - 1$ 
3:   for all  $i \in Vertices$  do
4:      $NeighboursSet \leftarrow$  Create a set with vertex  $i$  and its  $nNearest$  nearest neighbours
5:      $Subsets \leftarrow$  Create all distinct subsets of  $NeighboursSet$  with 1 to  $maxVisits$  vertices
6:      $AllSubsets \leftarrow$  Save the subsets of vertices created considering vertex  $i$ 
7:    $AllSubsets \leftarrow$  Clear repeated subsets that are found in  $AllSubsets$ 
8:   for all  $d \in Depots$  do
9:     for all  $S \in AllSubsets$  do
10:       $r \leftarrow$  Open a new route starting and finishing at depot  $d$ 
11:      for all  $i \in S$  do
12:         $r \leftarrow$  Insert vertex  $i$  in the cheapest position considering route  $r$  (respecting time windows)
13:      if  $r.duration \leq maxDuration$  then
14:         $RouteSet \leftarrow$  Add route  $r$  to the set  $RouteSet$ 
15:   return  $RouteSet$ 

```

4.4.1.3 Product Clustering

We assume that higher priority should be given to products with higher demand, which implies that these should be represented with a higher level of detail when approximations are performed. Products within the same family have similar production constraints (i.e., setup times and production lines are similar) thus aggregating them in a hyperproduct is reasonable. The product clustering procedure uses three parameters: the set of demands *Demands*, a maximum quantity of products allowed inside each hyperproduct *maxQuantity*, and the set of product families *Families*. Within each family, we aggregate low demand products in hyperproducts. The aggregated quantity of each hyperproduct should not exceed *maxQuantity* (line 9, Algorithm 3). Note that if the aggregated quantities are small, their impact in the production and transportation capacities is not considerable. We sort products by total demand and aggregate the products corresponding to the bottom of the list (usually around 80% of the products are aggregated). For each family, this clustering procedure ends up with a reduced set of products (which are not clustered because their demand is large), and a set of hyperproducts (including products with small demand). Algorithm 3 details the aforementioned product clustering procedure.

Algorithm 3 Product Clustering

```

1: procedure CLUSTERPRODUCTS(Demand, maxQuantity, Families)
2:   for all  $f \in Families$  do
3:      $D \leftarrow$  From the set of demands Demand get the demands for products belonging to family  $f$ 
4:      $ClassABC \leftarrow$  Classify each demand in  $D$  according to its quantity (A - large, B - medium, or C - small)
5:      $ToCluster \leftarrow$  Get the demands to be clustered (i.e., class B or C, according to  $ClassABC$ )
6:      $hp \leftarrow$  Open a new hyperproduct  $hp$ 
7:     while  $ToCluster \neq \emptyset$  do
8:        $p \leftarrow$  Get the product  $p$  with the lowest demand from the set of products  $ToCluster$ 
9:       if  $hp.total\_quantity + p.demand \leq maxQuantity$  then
10:         $hp \leftarrow$  Add product  $p$  to the currently opened hyperproduct  $hp$ 
11:         $ToCluster \leftarrow$  Remove product  $p$  from the set  $ToCluster$ 
12:       else
13:         $Hyperproducts \leftarrow$  Add the currently opened hyperproduct  $hp$  to the set  $Hyperproducts$ 
14:         $hp \leftarrow$  Open a new hyperproduct  $hp$ 
15:   return  $Hyperproducts$ 

```

The product clustering is the final step in the **Size Reduction** phase. At this point, a set of hyperproducts, and a set of generated routes visiting hypernodes inside each region are available to serve as input to the next phase.

4.4.2 Initial Solution

In the beginning of the second phase, the outputs provided by the first phase will serve as inputs to be used in the construction of the initial solution. These inputs comprise one IRP instance for each region (including a set of routes visiting hypernodes and a set of hyperproducts) and the data related to the LSP part of the problem. Figure 4.3 presents the procedure to obtain the initial solution.

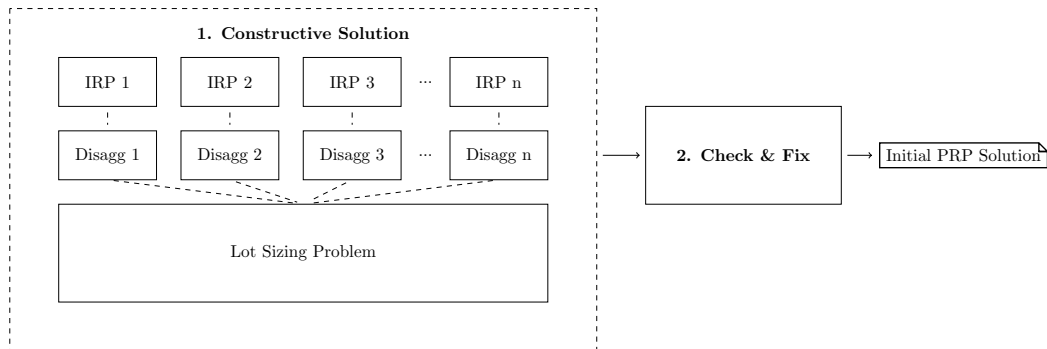


Figure 4.3 – To build the initial solution an IRP is solved for each region. The shipping quantities demanded by each region are then disaggregated and used to solve an LSP to define production quantities. Finally, infeasibilities are fixed in the Check & Fix procedure.

In order to construct the initial solution for the PRP, the algorithm starts by solving the IRP subproblem corresponding to each region defined in the first phase. We assume infinite production capacity. Each of these IRPs is solved, using a set-partitioning formulation

based on path variables, corresponding to the routes created in the first phase. The path variables are used to decide which routes of the reduced route set will be selected to make the deliveries in each period. After solving all IRP subproblems, the aggregated quantities of each hyperproduct to be delivered by each vehicle to each hypernode are known and are used to define the demand for the LSP formulation. The LSP formulation is solved by a general-purpose solver, completing the solution to the original problem. We name this step *Constructive Solution* because this initial decoupled PRP solution may be infeasible. Considering this fact, in the second step, some repair procedures are applied in order to obtain a feasible initial PRP solution. In the following subsections, we detail the steps of the second phase.

4.4.2.1 Constructive Solution

Regional delivery schedule definition

The algorithm starts by solving the IRP subproblem corresponding to each region. To do so, an IRP set-partitioning formulation is used. Consider the decision variables Θ_{rvt} which are equal to 1 if route r is performed by a vehicle of type v in period t , and Q_{ipvlt} which define the quantity of product p delivered to retail site i by a vehicle of type v in period t , to be consumed in period l . We define parameters o_{ir} to indicate if location i is visited by route r , c_r for the cost of performing route r , and cap_v for the capacity of a vehicle of type v . \mathcal{R} is the set including only the previously generated routes and K_v is the available number of vehicles of type v . The set of vehicle types is given by K^{type} . The remaining parameters are retained from the *MPRPTW* formulation. The considered set-partitioning formulation can be defined as follows:

(SPIRP):

$$\text{minimize } \sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{v \in K^{type}} \sum_{t \in \mathcal{T}} \sum_{h=0}^t \sum_{l=t+1}^{|\mathcal{T}|} h_{ip} \cdot D_{ipvhl} + \sum_{r \in \mathcal{R}} \sum_{v \in K^{type}} \sum_{t \in \mathcal{T}} c_r \cdot \Theta_{rvt} \quad (4.16)$$

s.t.

$$\sum_{v \in K^{type}} \sum_{t=0}^l Q_{ipvlt} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, l \in \mathcal{T} \quad (4.17)$$

$$\sum_{v \in K^{type}} \sum_{p \in \mathcal{P}} \sum_{h=0}^t \sum_{l=t}^{|\mathcal{T}|} D_{ipvhl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \quad (4.18)$$

$$\sum_{v \in K^{type}} \sum_{r \in \mathcal{R}} o_{ir} \cdot \Theta_{rvt} \leq 1 \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \quad (4.19)$$

$$\sum_{r \in \mathcal{R}} \Theta_{rvt} \leq K_v \quad \forall v \in K^{type}, t \in \mathcal{T} \quad (4.20)$$

$$Q_{ipvlt} \leq d_{ipl} \cdot \sum_{r \in \mathcal{R}} o_{ir} \cdot \Theta_{rvt} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in K^{type}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l \quad (4.21)$$

$$\sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{l=t}^{|\mathcal{T}|} Q_{ipvlt} \leq cap_v + M \cdot (1 - \Theta_{rvt}) \quad \forall r \in \mathcal{R}, v \in K^{type}, t \in \mathcal{T} \quad (4.22)$$

$$\begin{aligned} \Theta_{rvt} &\in \{0, 1\} \quad \forall r \in \mathcal{R}, v \in K^{type}, t \in \mathcal{T} \\ Q_{ipvlt} &\geq 0 \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in K^{type}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l. \end{aligned} \quad (4.23)$$

The objective function comprises two terms. If a location i stocks a unit of product p in period t , it incurs an inventory holding cost of h_{ip} . Additionally, if route r is made by a vehicle of type v in period t , a transportation cost c_r is incurred. Constraints (4.17) make sure that the delivered quantities satisfy the demand. Constraints (4.18) ensure that the inventory capacity is respected for each retail site. Constraints (4.19) impose at most one visit per retail site in each period. Constraints (4.20) impose a maximum number of vehicles of each type to be used in each period. Constraints (4.21) ensure that retail sites may only receive deliveries if they are visited. Constraints (4.22) are vehicle capacity constraints. Note that what is called retail sites and products in this formulation, are actually hypernodes and hyperproducts that have been aggregated in the previous phase. Therefore, the solutions obtained with this formulation include aggregated deliveries (including various products) that are made to aggregated nodes (including various retail sites).

Disaggregation

Since the IRP subproblems are created using hypernodes and hyperproducts provided by the *Size Reduction* phase, it is necessary to disaggregate the delivered quantities to match the original problem. Note that at this point the inventories and delivery quantities of each individual retail site are not known. Further holding cost optimization may be achieved by assigning delivery quantities at a disaggregated level. A simple linear programming problem is solved, minimizing the holding cost of the disaggregated inventories. Here, the indices i are individual retail sites, p are individual products, $hn \in \mathcal{HV}'$ are hypernodes, and $hp \in \mathcal{HP}$ are hyperproducts. The model works with continuous variables Q_{ipvlt} , which define the quantity of product p delivered to retail site i by a vehicle of type v in period t , to be consumed in period l . The quantities $q_{hn, hp, vlt}$ are parameters provided by the solutions obtained by solving the IRPs of each region. The remaining parameters are maintained

from the *MPRPTW* formulation. The disaggregation linear programming program can be defined as follows:

(Disagg):

$$\text{minimize } \sum_{i \in \mathcal{V}'} \sum_{p \in \mathcal{P}} \sum_{v \in K^{type}} \sum_{t \in \mathcal{T}} \sum_{h=0}^t \sum_{l=t+1}^{|\mathcal{T}|} h_{ip} \cdot Q_{ipvhl} \quad (4.24)$$

s.t.

$$\sum_{i \in \mathcal{V}'_{hn}} \sum_{p \in \mathcal{P}_{hp}} Q_{ipvhl} = q_{hn, hp, v, tl} \quad \forall hn \in \mathcal{H}\mathcal{V}', hp \in \mathcal{H}\mathcal{P}, v \in K^{type}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l \quad (4.25)$$

$$\sum_{v \in K^{type}} \sum_{t=0}^t Q_{ipvhl} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, t \in \mathcal{T} \quad (4.26)$$

$$\sum_{v \in K^{type}} \sum_{p \in \mathcal{P}} \sum_{h=0}^t \sum_{l=t}^{|\mathcal{T}|} Q_{ipvhl} \leq \bar{c}_i \quad \forall i \in \mathcal{V}', t \in \mathcal{T} \quad (4.27)$$

$$Q_{ipvhl} \geq 0 \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, v \in K^{type}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l. \quad (4.28)$$

The objective function minimizes the holding cost incurred by all retail sites. Constraints (4.25) impose that the sum of the disaggregated quantities must be equal to the aggregated quantities of each hypernode and hyperproduct. Constraints (4.26) make sure that the disaggregated demand of each retail site is satisfied. Constraints (4.27) impose the inventory capacities for each retail site. Constraints (4.28) are the non-negativity constraints.

Lot sizing problem

After disaggregating the quantities of each hyper entity, the demand d_{pt} of each product p , for each shipping period t to be satisfied by the supplier can be defined. In order to define how the supplier should satisfy this demand, a CLSP formulation is used (see [Karimi et al. \[2003\]](#) for a review). This formulation works with continuous variables P_{pmgt} to decide the quantity of product p that must be produced on production line m in production period g to be shipped in shipping period t . Additionally, the model uses the binary variables A_{mfg} equal to one if production line m is set up to produce products of family f in production period g , and binary variables B_{mpg} equal to 1 if the production line m is set up to produce product p in period g . The formulation is defined as follows:

(CLSP):

$$\text{minimize } \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{g \in \mathcal{T}} \sum_{h=0}^g \sum_{t=g+1}^{|\mathcal{T}|} h_{0p} \cdot P_{pmht} \quad (4.29)$$

s.t. (4.2), (4.3), (4.4), (4.5),

$$\sum_{m \in \mathcal{M}} \sum_{g=0}^{t-1} P_{pmgt} = d_{pt} \quad \forall p \in \mathcal{P}, t \in \mathcal{T} \quad (4.30)$$

$$P_{pmgt} \geq 0 \quad \forall p \in \mathcal{P}, m \in \mathcal{M}, g \in \mathcal{T}, t \in \mathcal{T}, g < t. \quad (4.31)$$

The objective function minimizes the total holding cost incurred by the supplier. Constraints (4.30) impose that the produced quantities must meet the demand $d_{pt} = \sum_{i,v,l} Q_{ipvtl}$ for each product p in each shipping period t . Note that quantities produced in a production period g only become available to be shipped in the next period. In order to define the necessary conditions to model production line capacities, setups, and the warehouse capacity of the supplier, we add constraints (4.2), (4.3), (4.4), and (4.5) from the original *MPRPTW* formulation presented in Section 4.3.

4.4.2.2 Check & Fix

As indicated in Figure 4.3, after solving the LSP, a constructive solution for the PRP becomes available by joining every IRP solution of each region with the LSP solution. However, since an a priori vehicle assignment is made in the *Size Reduction* phase, it is possible that the created solution exceeds the number of available vehicles. In that case, a procedure is used in order to merge routes and lower the number of vehicles in the problematic periods. At the end of the Check & Fix step, an initial PRP solution is found.

4.4.3 Solution Improvement

The improvement phase of the algorithm relies on the F&O heuristic ([Helber and Sahling, 2010], [Chen, 2015]). Different decomposition strategies are followed to define different subproblems, which allow for a flexible exploration of the search space. The global solution is iteratively improved by analysing the trade-offs that are typically inherent to a PRP. As opposed to most F&O approaches, the variables and constraints related to the entire formulation of our problem are never loaded. In fact, the complete PRP formulation is not used in our approach, as the large number of products and retail sites requires several minutes just to build the model. Therefore, we decided to avoid the case where a single formulation is used, contrasting with the work of other researchers (e.g., the IRP tackled by Larraín et al. [2017]). Here, in each decomposition approach, a subinstance is created and solved using a decomposition-tailored formulation. Afterwards, the values of the variables are updated in the global PRP solution. The size of each subproblem is controlled, ensuring

that the best possible local decisions are taken for the considered entities, while solving them to optimality. With such a matheuristic approach, it is also easier to find a compromise between solution quality and running time.

4.4.3.1 Improvement Matheuristic

In general terms, our matheuristic selects a decomposition strategy $\omega \in \Omega$ and explores it until the criteria to proceed to the next decomposition strategy are met. A decomposition strategy works with an associated mathematical formulation plus a subset of entities \mathcal{ST}^{opt} , which are used to create a subinstance, smaller than the original problem. The size of the subset of entities (and subinstance) is controlled by a parameter ω_size . For instance, in a decomposition strategy based on the routing part of the problem, a pair (k, t) may represent a route performed by vehicle k in period t . A subset of pairs $(k, t) \in \mathcal{ST}^{opt}$ can be defined to select a set of routes across several periods to be optimized. The number of routes in the subset \mathcal{ST}^{opt} is defined by the parameter ω_size . The algorithm explores various decomposition strategies in the set Ω until the stopping criteria are met. All subproblems are solved using a general-purpose solver. However, if the formulation comprises subtour elimination constraints, a simple branch-and-cut procedure is called, which dynamically adds the necessary cuts for violated constraints. Figure 4.4 gives a schematic view of the improvement phase.

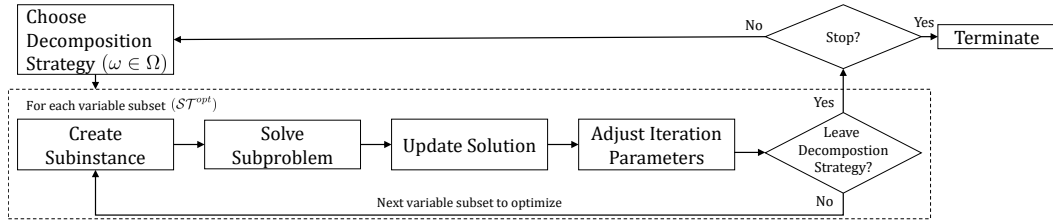


Figure 4.4 – In the improvement phase, decomposition strategies are selected in order to explorer different decisions inherent to PRPs. In each iteration, a subset of entities is selected to create a subproblem to be solved using a decomposition-tailored formulation. The global solution, maintained outside, is updated whenever improvements are found.

After selecting a decomposition strategy ω , the idea is to analyse a small set of related entities \mathcal{ST}^{opt} at a time. In each iteration, we define \mathcal{ST}^{opt} with ω_size entities and create a subinstance. The subinstance is solved using the mathematical formulation associated with the current decomposition strategy. The time it takes to solve each subproblem must be within the interval $[t_{mip}^{lower}, t_{mip}^{upper}]$. If it is below the lower limit, ω_size is increased by one. If it is above the upper limit, ω_size is decreased by one. This allows the algorithm to consider larger sets of entities without maintaining the runtime of each iteration within a defined interval. After solving a subproblem, if the relative gap is excessively large, say larger than a parameter max_gap , the algorithm solves the subinstance with a period-oriented decomposition strategy (i.e., each period is solved separately when the subinstance is multi-periodic). By doing so, we achieve tractable subproblems and ensure that the runtime spent with this iteration is not completely lost. After exploring a decomposition

strategy, the percentage of iterations where improvements were found is computed. If the improvement percentage is larger than $ni\%$, the decomposition strategy is explored again. If the improvement percentage is smaller than $ni\%$, the algorithm advances to the next decomposition strategy ($\omega = \omega + 1$). In the last decomposition strategy ($\omega = |\Omega|$) if the improvement percentage is smaller than $ni\%$, the algorithm terminates. Otherwise, it needs to decide if it repeats the same decomposition strategy or if it is worth to go back to the first ($\omega = 1$). If the percentage of improvement is larger than a parameter $reload\%$, it means the solution may have changed considerably and it may be worth to restart from the first decomposition strategy. If the percentage of improvement is not high but still larger than $ni\%$ (note that $ni\% \leq reload\%$), the algorithm re-explores the same decomposition strategy. The pseudo-code of the algorithm is presented in Algorithm 4 below.

Algorithm 4 Improvement Matheuristic

```

1: procedure IMPROVE( $curr\_sol, max\_gap, t_{max}, t_{mip}^{lower}, t_{mip}^{upper}, ni\%, reload\%$ )
2:    $\omega \leftarrow 1; best\_sol \leftarrow curr\_sol;$ 
3:   while  $curr\_t \leq t_{max}$  and  $\omega \leq |\Omega|$  do
4:     for each entities related to the current decomposition strategy  $\omega$  do
5:       Define the subset  $\mathcal{ST}^{opt}$  with  $\omega\_size$  surrounding entities to create a subinstance
6:        $curr\_sol \leftarrow$  Solve the subinstance with a general-purpose solver
7:       if  $relative\_gap \geq max\_gap$  then
8:          $curr\_sol \leftarrow$  Solve the current subproblem with a periodic F&O procedure
9:       if  $curr\_sol \leq best\_sol$  then
10:         $best\_sol \leftarrow curr\_sol$ 
11:       if  $t_{last\_it} \notin [t_{mip}^{lower}, t_{mip}^{upper}]$  then
12:         $\omega\_size \leftarrow$  Update the parameter  $\omega\_size$  to adjust the runtime of the next iteration
13:       Choose decomposition strategy  $\omega \in \Omega$  depending on  $ni\%$  and  $reload\%$ 
14:   return  $best\_sol$ 

```

4.4.3.2 Decomposition Strategies

We developed three different decomposition strategies in order to achieve a flexible improvement phase, focusing on different solution attributes. We start the search with a decomposition strategy ($\omega = 1$) focused on the routing part of each period. Its base formulation is called *Daily VRP*. The second decomposition strategy ($\omega = 2$) focuses on the lot sizing part of the problem and we call its base formulation *Partial LSP*. The objective of the third decomposition strategy ($\omega = 3$) is to integrate all the decisions inherent to the PRP. Hence, we call its base formulation *Local PRP*. Table 4.2 summarizes the costs to be improved by each decomposition strategy as well as the entities defining the corresponding sets of entities \mathcal{ST}^{opt} related to the decisions to be optimized.

We detail the base formulation and the procedure to build subinstances of each decomposition strategy below.

Daily VRP ($\omega = 1$) A subinstance is defined for each combination of vehicle and period $k \in \mathcal{K}, t \in \mathcal{T}$. The ω_size routes that are nearest to route k in period t (based on their centroids in the incumbent solution) are inserted into \mathcal{ST}^{opt} to define the *Daily VRP* subproblem.

Table 4.2 – Description of the three developed decomposition strategies. Each strategy focuses on and integrates distinct sets of decisions. Only the *LocalPRP* formulation is able to improve all types of cost (for a subset of retail sites).

ω	Base Formulation	Entities defining ST^{opt}	Number of subinstances	Cost Improvements		
				Supplier Inventory	Retail Sites Inventory	Routing
1	<i>Daily VRP</i>	$k = 1, \dots, K, t = 1, \dots, T$	$K \cdot T$			✓
2	<i>Partial LSP</i>	$i = 1, \dots, V$	V	✓	✓	
3	<i>Local PRP</i>	$i = 1, \dots, V$	V	✓	✓	✓

The objective of the *Daily VRP* formulation (details are given in 4.A.0.1) is exclusively to improve the routing cost. This can be interesting due to the fact that the solution that is initially constructed by the algorithm may not be optimal regarding the routing cost. There are two main reasons for this fact. First, it is likely that the route generator does not provide a set containing the optimal routes considering the nodes of a certain region. Thus, it is necessary to transform the path variables from the initial IRP set-partitioning formulation, (4.16) - (4.23), into the routing variables to be included in the *Daily VRP*, which is an arc-based formulation considering every possible arc between two locations. With these new variables, we seek to find inter-route and intra-route improvements. Second, since the problem may have been decomposed into regions, it is probable that some interesting visiting sequences (containing nodes of different regions) are still available. The *Daily VRP* model allows for an analysis of routes that were part of different regional IRPs, trying to find inter-route improvements, and mixing nodes that could not communicate in the formulation with path variables (see Figure 4.5).

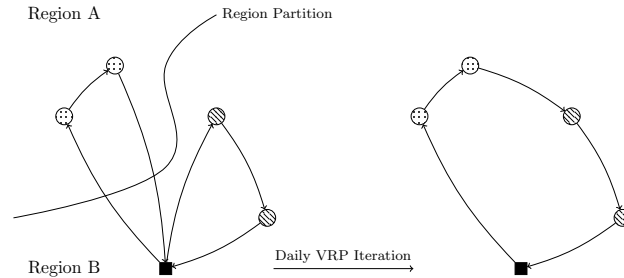


Figure 4.5 – The path variables considered in the initial solution (considering hypernodes) phase are disaggregated into arc variables (between nodes) during the improvement phase. This allows for inter-region, inter-route, and intra-route improvements.

Partial LSP ($\omega = 2$) A subinstance is defined for each retail site $i \in \mathcal{V}'$. The ω_size nearest neighbours of i (surrounding entities) are inserted into the set ST^{opt} . The production and delivery quantities related to the selected retail sites are optimized considering the entire planning horizon.

Considering that the initial solution is composed by decoupled decisions coming from one LSP and several IRP subproblems, further improvements regarding production and inventory management variables may still be found. Furthermore, dur-

ing the improvement phase, most solution changes are focused on small subproblems (particularly when exploring decomposition strategies with routing decisions, which are more complex) and it is necessary to integrate supplier and retail site decisions at a larger scale. The *Partial LSP* base formulation (details are given in 4.A.0.2) aims at improving the production setup variables, the production quantities, and the delivery quantities. Since routing variables are not loaded into this model, larger sets of retail sites can be analysed simultaneously. Nevertheless, the size and complexity of the model can be adjusted, considering the computing time of each iteration. If the model takes too long to solve, each iteration may be simplified either by fixing some production setups or by loading smaller sets of retail sites. Figure 4.6 gives a schematic overview of the possible decisions to be made with this formulation, showing the entities and variables to be loaded in each time period.

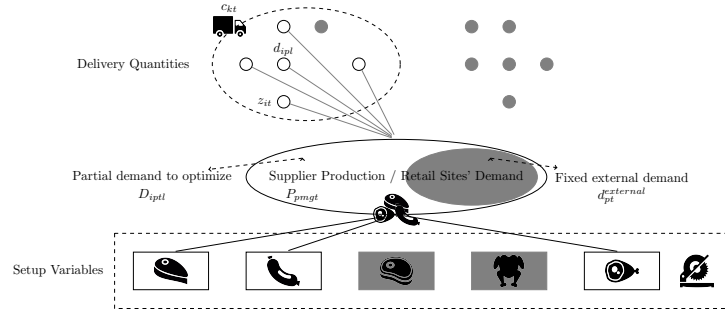


Figure 4.6 – In the Partial LSP, the production and delivery quantities of a subset of retail sites are optimized. All routing decisions are maintained. Only some production flows are allowed (white setup variables, in the bottom) and some shipping flows are allowed (white nodes visited by the vehicle, on the top).

In Figure 4.6, the entities in white are the ones that can be optimized. This means that only some setup, production, and delivery quantity decisions can be improved. The entities represented in gray are fixed and are not to be loaded into the model. The production line shown at the bottom, can produce (or provide flow for) three products. The retail sites that were loaded into the *Partial LSP* (in white) have some scheduled visits fixed in the current solution. These visits are represented by a parameter z_{it} , which is equal to one if the retail site i is visited in period t in the current solution. If the retail site is visited, the delivered quantities of each product p to satisfy its demand in period l , D_{iptl} can be optimized and the supplier can also adapt the production quantities to better satisfy each retail site included in the model (white part of the ellipse). This means that the supplier can decide not only its own inventories but also the inventories to be held at each retail site in the subinstance. The retail sites that were not loaded into the model (in gray) demand a fixed quantity, $d_{pt}^{external}$, for each product p , in each period t (gray ellipse) and their deliveries must remain as they are in the incumbent solution. Therefore, the supplier can only decide if these quantities are to be made-to-stock or made-to-order, which means that the inventories of the unloaded retail sites are not changed. Note that in the incumbent solution,

different vehicles visit different sets of locations (in Figure 4.6, a set is represented by the dashed ellipse). Therefore, it is necessary to keep track of the deliveries of each day to make sure that the capacity of the vehicles c_{kt} and the maximum inventory levels \bar{c}_i of the retail sites continue to be respected.

Local PRP ($\omega = 3$) A subinstance is defined for each retail site $i \in \mathcal{V}'$. The ω_size nearest neighbours of i (surrounding entities) are inserted into the set \mathcal{ST}^{opt} . These retail sites are considered through the whole planning horizon. For each period t we define the set of vehicles \mathcal{K}_t visiting any of the retail sites belonging to \mathcal{ST}^{opt} (in the incumbent solution). Additionally, all the locations visited by the vehicles in \mathcal{K}_t and not in \mathcal{ST}^{opt} are inserted into \mathcal{V}_t^{out} , for each period t . The set of retail sites $\mathcal{V}_t = \mathcal{ST}^{opt} \cup \mathcal{V}_t^{out}$ and vehicles in each subinstance is different in each time period t .

The models presented in the two previous subsections either deal with routing or holding costs separately. Although they can be called iteratively, their solutions will never be able to trade-off holding costs against routing costs. For this reason, a third model is developed. The objective of the *Local PRP* base formulation (details are given in 4.A.0.3) is to enable the solution approach to perform the aforementioned trade-off analysis, allowing for fully integrated decisions considering production, inventory management and routing aspects (at least locally). Figure 4.7 provides a schematic representation of the variables to be loaded in each *Local PRP*.

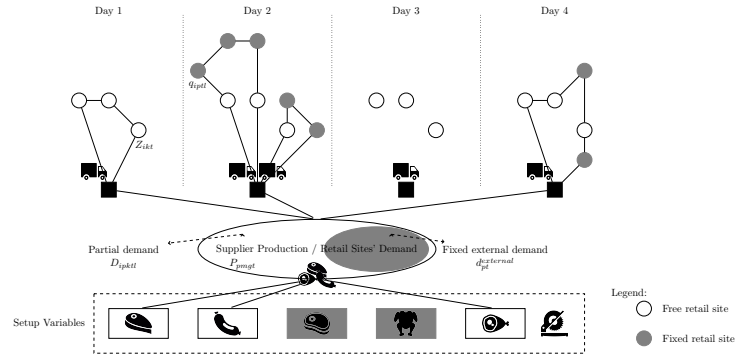


Figure 4.7 – Example showing a Local PRP instance for 3 selected retail sites in the first period

In this model, the set of retail sites \mathcal{ST}^{opt} (white nodes) is considered through the whole planning horizon of the global PRP. In the incumbent solution, these retail sites are visited by a set of vehicles which also visit other retail sites which do not belong to \mathcal{ST}^{opt} (gray nodes in periods 2 and 4). These additional retail sites will be different in each period and are represented by the sets \mathcal{V}_t^{out} . The deliveries to these external retail sites have to be maintained, therefore they will be fixed in the model. The only decision that can actually change is the vehicle that will perform these visits. In some periods, retail sites of the set \mathcal{ST}^{opt} may not be visited in the incumbent solution (as in period 3). However, they are still loaded into the model

in conjunction with vehicles that are not used in that period or with vehicles that are passing nearby. This enables the algorithm to eliminate and start routes. The idea is to create subPRPs allowing for integrated decisions considering only the retail sites of the set \mathcal{ST}^{opt} . The LSP part (represented in the lower part of Figure 4.7) is completely loaded, thus deliveries that are internal and external to the subinstance are both considered in the production plan. Note that it is necessary to ensure that these PRPs are tractable and can be loaded quickly. In order to adjust the running time of these small PRPs, one can fix setup decisions and some routing decisions such as arcs or visiting variables if necessary.

4.5. Computational Experiments

Our solution approach was developed to solve large multi-product PRPs with time windows including complex production activities. However, the literature does not provide instances considering such complexity. For this reason, we compare our algorithm using instances that are proposed both in the IRP and the PRP literature.

The conditions for all the experiments performed by our matheuristic ($F\&O$) are the following:

- Runs of 3600 seconds;
- CPLEX 12.6.1 is used to solve subproblems;
- Intel Core processors running at @ 2.4 GHz;
- A single thread is used during the improvement phase;
- Regions are created using a k-means algorithm based on centroid distances (regions should have less than 50 nodes to be handled by the general-purpose solver);
- Initial route sets are created using a nearest neighbour heuristic.

The comparison of our matheuristic with other approaches is based on the ideas presented by [Dolan and Moré \[2002\]](#). The authors present the concept of performance profile, a chart depicting the cumulative probability for each algorithm to obtain a solution with a relative gap smaller than or equal to τ . The relative gap is computed in relation to the best-known solution for each instance p . For each approach s the relative gap $r_{p,s}$ is computed. The performance probability $\rho_s(\tau)$ is then defined, for each τ , according to expression

$$\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in P : r_{p,s} \leq \tau\},$$

where n_p is the total number of instances and τ is the threshold for the relative gap based on the best solution found among all methods.

Coelho and Laporte, MMIRP instances

In order to compare the performance of our matheuristic we consider the most similar

instances available in the literature, which consider a multi-product and multi-vehicle setting. These instances were proposed by Coelho and Laporte [2013] in a work related to the IRP and can be accessed in the first author's website (www.leandro-coelho.com/instances). Despite disregarding production decisions, we consider that the IRP is closely related to the PRP and this is still an interesting comparison. We compare our *F&O* approach against the *B&C* approach proposed by Coelho and Laporte [2013] using the performance profiles presented in Figure 4.8.

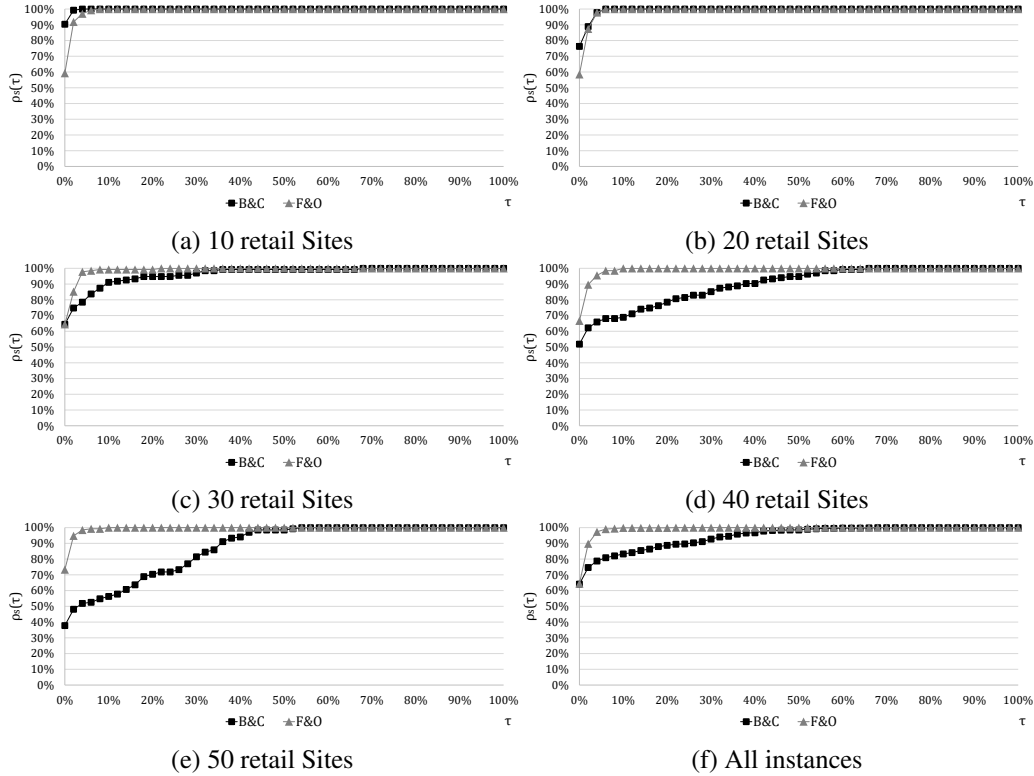


Figure 4.8 – Performance profiles related to Coelho and Laporte [2013] solutions, MMIRP instances

We conclude that for the smaller instances with 10 and 20 retail sites (Figures 4.8a and 4.8b), Coelho and Laporte's exact approach presents a larger probability to find best-known solutions (for $\tau = 0$). For instances with 30, 40, and 50 (Figures 4.8c, 4.8d, and 4.8e) retail sites our approach outperforms Coelho and Laporte's B&C as it is able to find a larger number of best-known solutions (i.e., τ is closer to 0) and presents a larger probability to find solutions with smaller gaps to the best-known solution.

In the performance profile including all the instances (Figure 4.8f) (excluding instances with 100 retail sites that are not solved by Coelho and Laporte's exact approach) we conclude that our F&O never finds solutions with a gap that is larger than 10% whereas Coelho and Laporte's B&C obtains some solutions with a gap larger than 50% for instances with 40 and 50 retail sites. Note that despite being an exact method, Coelho and Laporte's B&C experiments were run for 42 000 seconds. Therefore, we conclude that with a smaller com-

puting time, our matheuristic has a larger probability of finding best-known solutions. Additionally, we present the average improvement in the best-known solutions in Table 4.3.

Table 4.3 – Average upper bound improvement, F&O vs B&C, MMIRP instances (Coelho and Laporte [2013])

Instances	#Retail Sites						Avg Imp	Avg Time (s)
	10	20	30	40	50	100		
mmirp-1-1-3	0.00%	0.00%	0.00%	0.00%	0.00%	-	0.00%	122.84
mmirp-3-1-3	0.00%	-0.05%	0.00%	0.00%	-0.11%	-	-0.03%	97.64
mmirp-5-1-3	-0.12%	0.00%	-0.02%	-0.02%	-0.03%	-	-0.04%	104.64
mmirp-1-3-3	0.00%	0.00%	-0.05%	-0.62%	-1.21%	-	-0.38%	1120.92
mmirp-3-3-3	-0.35%	-0.34%	1.72%	8.15%	27.78%	-	7.39%	1998.60
mmirp-5-3-3	0.00%	0.00%	-0.11%	-0.78%	-0.84%	-	-0.35%	1840.36
mmirp-1-5-3	-0.34%	-1.35%	1.65%	-1.05%	12.61%	-	2.30%	2202.00
mmirp-3-5-3	0.00%	-2.07%	-2.27%	-1.68%	2.25%	-	-0.75%	2703.84
mmirp-5-5-3	0.00%	-0.30%	-0.52%	-1.50%	6.10%	-	0.75%	2686.56
mmirp-1-1-5	0.00%	-0.13%	-0.38%	-0.41%	0.13%	-	-0.16%	371.80
mmirp-3-1-5	-0.24%	-0.43%	-0.45%	-0.71%	-0.65%	-	-0.50%	628.60
mmirp-5-1-5	-0.59%	-0.11%	0.00%	-0.25%	0.12%	-	-0.17%	509.16
mmirp-1-3-5	-0.39%	-0.01%	2.74%	12.53%	21.26%	-	7.23%	2753.96
mmirp-3-3-5	-0.33%	0.66%	-2.49%	-1.01%	25.45%	-	4.46%	3088.20
mmirp-5-3-5	0.00%	0.00%	0.00%	0.65%	10.35%	-	2.20%	3232.76
mmirp-1-5-5	-1.26%	-0.07%	4.00%	20.99%	33.18%	-	11.37%	3165.08
mmirp-3-5-5	-1.94%	-1.64%	4.98%	26.02%	14.00%	-	8.28%	3583.56
mmirp-5-5-5	-0.74%	2.20%	10.93%	35.82%	34.30%	-	16.50%	3600.00
mmirp-1-1-7	-0.36%	-1.90%	-1.06%	-1.07%	-0.64%	-	-1.01%	749.00
mmirp-3-1-7	-0.59%	-1.09%	-0.18%	0.48%	0.71%	-	-0.14%	1597.40
mmirp-5-1-7	-0.02%	-0.67%	-0.58%	-0.12%	2.09%	-	0.14%	1746.04
mmirp-1-3-7	-2.52%	-0.76%	0.34%	29.10%	26.71%	-	10.58%	3087.48
mmirp-3-3-7	-0.36%	0.91%	-1.17%	6.69%	25.40%	-	6.29%	3530.16
mmirp-5-3-7	-0.48%	-0.19%	0.77%	41.73%	34.08%	-	15.18%	3600.00
mmirp-1-5-7	-0.23%	-1.22%	15.60%	31.20%	29.22%	-	14.92%	3480.72
mmirp-3-5-7	-1.10%	2.10%	30.59%	27.65%	14.33%	-	14.71%	3600.00
mmirp-5-5-7	0.30%	0.80%	4.75%	18.84%	14.15%	-	7.77%	3600.00
Avg	-0.43%	-0.21%	2.55%	9.28%	12.25%	-	4.69%	2177.83
	-0.32%		8.03%					

The results suggest that our matheuristic provides better results for instances with larger size. While for smaller instances (#Retail Sites ≤ 20) we provide solutions with an average deviation of -0.32% to the best-known solution, for larger instances (#Retail Sites ≥ 30) we find an average improvement of 8.03% . We provide 377 new best solutions, which account for 46% of the entire instance set. This includes 135 instances (the ones with 100 retail sites) which had no solution available in the literature. The average runtime of these tests was 2177.83 seconds.

Region decomposition analysis *F&O*, MMIRP instances with 100 retail sites

Since Coelho and Laporte [2013] did not report any result for instances with 100 retail sites, we consider that these instances are more challenging. Therefore, we tested our matheuristic for the cases where the initial solution is created without regional decomposition against

the cases where the problem is decomposed into 2 to 4 regions. The results are presented in Table 4.4.

Table 4.4 – Average objective value comparison of regional decompositions, MMIRP instances with 100 retail sites

Periods	Vehicles	Products	Initial solution objective (Instances solved)				Final solution objective			
			No Decomposition	2 Regions	3 Regions	4 Regions	No Decomposition	2 Regions	3 Regions	4 Regions
3	1	1	12970.84 (5)	10972.42 (5)	10815.78 (5)	10738.82 (5)	10563.04	10563.04	10563.04	10563.04
3	1	3	20683.38 (5)	18370.70 (5)	18344.86 (5)	18312.24 (5)	18188.98	18188.98	18188.98	18188.98
3	1	5	27919.62 (5)	25811.02 (5)	25697.92 (5)	25659.78 (5)	25522.64	25522.10	25522.88	25521.78
3	3	1	13768.00 (5)	14414.96 (5)	15527.16 (5)	15344.48 (5)	10866.48	10885.70	10904.86	11126.60
3	3	3	21694.44 (5)	21891.26 (5)	22773.44 (5)	24055.70 (5)	19132.68	19253.38	19784.70	19564.20
3	3	5	29223.30 (5)	28966.26 (5)	30041.06 (5)	30641.42 (5)	26609.82	26168.16	26484.06	26505.40
3	5	1	15608.24 (5)	14540.94 (5)	15028.74 (5)	16235.54 (5)	11690.06	11388.92	11701.62	11701.70
3	5	3	23216.08 (5)	22007.88 (5)	22742.08 (5)	23638.68 (5)	19658.44	19576.88	19797.52	19795.58
3	5	5	31274.80 (5)	29756.44 (5)	29909.64 (5)	30417.04 (5)	27823.64	26531.16	26646.34	26813.86
5	1	1	19032.84 (5)	17102.76 (5)	16271.80 (5)	17061.68 (5)	15605.06	15742.84	15589.42	15665.28
5	1	3	30639.06 (5)	30931.04 (5)	30313.82 (5)	32474.90 (5)	27972.18	28402.06	29097.38	28702.44
5	1	5	43436.58 (5)	47693.66 (5)	47069.48 (5)	56131.92 (5)	38935.94	38893.60	39035.06	39009.36
5	3	1	26565.96 (5)	23281.58 (5)	25061.92 (5)	23241.24 (5)	19381.26	18982.96	18338.22	18739.22
5	3	3	38675.46 (5)	46211.84 (5)	39113.20 (5)	41295.10 (5)	33138.68	38224.50	32116.44	32396.98
5	3	5	50038.72 (5)	64979.92 (5)	54824.44 (5)	51498.48 (5)	45407.74	47205.90	42845.76	43540.44
5	5	1	32585.06 (5)	27607.10 (5)	27603.22 (5)	28324.92 (5)	22923.06	20921.00	20941.68	20945.12
5	5	3	48592.72 (5)	40485.50 (5)	39867.44 (5)	42587.80 (5)	38294.22	34089.00	33725.78	34644.96
5	5	5	62724.86 (5)	49403.10 (5)	49219.82 (5)	52328.88 (5)	51161.62	44429.22	43874.78	45457.62
7	1	1	26812.10 (5)	38001.98 (5)	53033.32 (5)	43965.76 (5)	22176.50	21780.94	22053.54	22367.70
7	1	3	45248.04 (5)	82047.56 (5)	69747.78 (5)	67159.28 (5)	40630.30	41712.42	41442.56	40822.90
7	1	5	60317.36 (5)	114063.60 (5)	112091.00 (5)	106659.10 (5)	55080.28	56279.94	55608.98	55482.66
7	3	1	37075.93 (4)	44608.50 (4)	41849.78 (4)	40040.78 (4)	26896.30	25041.78	26196.00	26291.50
7	3	3	68574.28 (4)	79927.10 (4)	86745.18 (4)	73519.25 (4)	56826.73	52481.08	61993.08	52529.73
7	3	5	73996.94 (5)	80079.32 (5)	97818.48 (5)	96876.50 (5)	64271.64	58845.36	66494.46	67406.94
7	5	1	45519.00 (3)	41347.47 (3)	40239.80 (3)	45359.13 (3)	39297.50	37337.17	30288.57	33779.03
7	5	3	77951.60 (1)	62796.30 (1)	58904.20 (1)	58669.60 (1)	60807.30	52508.80	51595.90	52215.50
7	5	5	105180.40 (2)	71640.10 (2)	67168.50 (2)	91626.25 (2)	79515.55	63972.80	62483.30	66300.00
Average (Count)			37278.97 (124)	40898.11 (124)	41071.95 (124)	41173.27 (124)	31433.83	30407.24	30435.28	30453.64
			Average Time To Best (s)				2363.09	2115.98	2102.51	2275.78

Regional decomposition is particularly useful when the set-partitioning formulation is not able to provide an initial solution, preventing the approach from solving an instance. When the initial solution has a large relative gap, it is also a sign that regional decomposition may be beneficial. Furthermore, cases where the initial solution takes too much time to be found may also be addressed with regional decomposition, as the time available for the improvement phase becomes short. These are the main reasons for the results presented in the *No Decomposition* columns of Table 4.4. When no regional decomposition is applied, 11 of the 135 instances with 100 retail sites are not solved. When regional decomposition is applied (from 2 to 4 regions) all the instances are solved, though we only compare the 124 instances that are solved in all cases. The quality of the initial solution deteriorates as the number of regions increases. However, the time to obtain an initial solution with more regions is usually shorter, providing more time for the improvement phase. The additional time for improving the solution proves beneficial as the average objective value of the final solution is smaller. Figure 4.9 shows that, on average, the algorithm spends 22.36% (size of the bubble) of the time to compute the initial solution when no decomposition is performed, whereas the decomposed instances only need around 7% of the total time to obtain the initial solution. The average runtime to achieve the best solution is smaller with 2 and 3 regions.

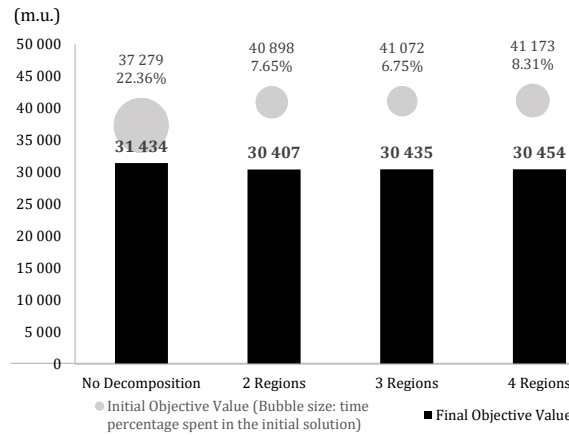


Figure 4.9 – The initial objective value is better if fewer regions are created. However, larger computational times are required to obtain an initial solution. When the original problem is decomposed into more regions, the initial solution is obtained faster. The additional time to use in the improvement phase allows for better final objective values.

One important aspect that needs to be addressed when solving these instances is the feasibility of the fleet. The *Check & Fix* procedure is usually called when the deliveries scheduled for each region cannot be performed by the available fleet. Therefore, this procedure is always called when the number of regions is larger than the number of vehicles (since some vehicles will be repeated in some regions). In larger instances the *Check & Fix* procedure may take more time to repair these infeasibilities as seen in the initial solutions of the instances with 7 periods and 1 vehicle.

Archetti et al., MPRP instances

Finally, we compare the performance of our algorithm with PRP instances considering a single product and multiple vehicles, presented by Archetti et al. [2011] and available on the authors' website (<http://or-brescia.unibs.it/instances>). This algorithm is tailored for a particular PRP with constant demands and an uncapacitated production facility. Some changes were made in the models used in our matheuristic in order to match the same assumptions (e.g.: the authors consider that production becomes available to be shipped as soon as it is produced). Archetti et al. [2011] consider four classes of instances with 19, 50 and 100 retail sites. Since the authors allow for an unlimited fleet, we adapted the instances and considered the maximum number of vehicles that is used in their tests for each number of retail sites. Table 4.5 shows the number of vehicles that are used in our tests, the average deviation between our solutions and those of Archetti et al. [2011] for the four instance classes, and average runtimes.

The results suggest an improvement in the average objective function of the instances with 19 and 50 retail sites. For instances with 100 retail sites, the objective value is 0.68% worse on average. Considering the total set of instances, we improve the average difference over the best solution by 0.11%. Note that the problem considered in Archetti et al. [2011] is slightly different from our context. However, the results show that our matheuristic is also competitive on this instance set. Figure 4.10 shows the performance profiles for each set of

Table 4.5 – Average deviation from Archetti et al. [2011] solutions and runtime

Retail Sites (Vehicles)	I	II	III	IV	Dev	Time (s)
19 (1)	-0.83%	-0.12%	-1.15%	-0.24%	-0.58%	241.22
50 (7)	-0.38%	-0.10%	-0.65%	-0.53%	-0.41%	1733.89
100 (14)	0.25%	0.05%	1.35%	1.07%	0.68%	3599.20
Dev	-0.32%	-0.05%	-0.15%	0.10%	-0.11%	1861.30

instances. For instances with 19 and 50 retail sites, our matheuristic finds a larger number of best solutions. For instances with 100 retail sites, the approach of Archetti et al. [2011] shows a larger probability to find solutions with a deviation smaller than or equal to 5% from the best-known solution. The average runtime of these tests was 1861.30 seconds.

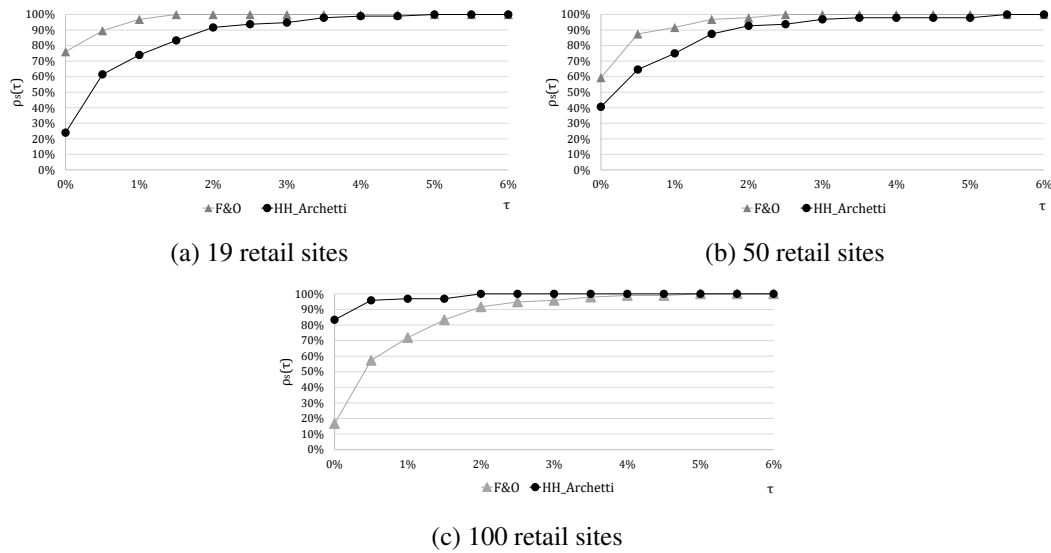


Figure 4.10 – Performance profiles related to Archetti et al. [2011] solutions, MPRP instances

Considering the afore presented perspectives, our matheuristic is considered to be competitive in terms of solution quality. Indeed, the flexibility provided by our approach is one of its greatest strengths as matheuristics considering problem extensions are still scarce in the IRP and PRP literature.

4.6. Case Study

4.6.1 Analysis of the Current Situation

This section details the case study of a European meat store chain which owns a Meat Processing Center (MPC) where several meat products are processed and delivered to several meat stores. The characteristics of the problem include 13 productions lines, 175 perishable products, 185 meat stores with delivery time windows and 35 heterogeneous vehicles.

With the objective of detailing the challenge faced by the company, a simplified overview of the problem is presented in Figure 4.11 and explained below (the numbers in the text correspond to the numbers in the figure).

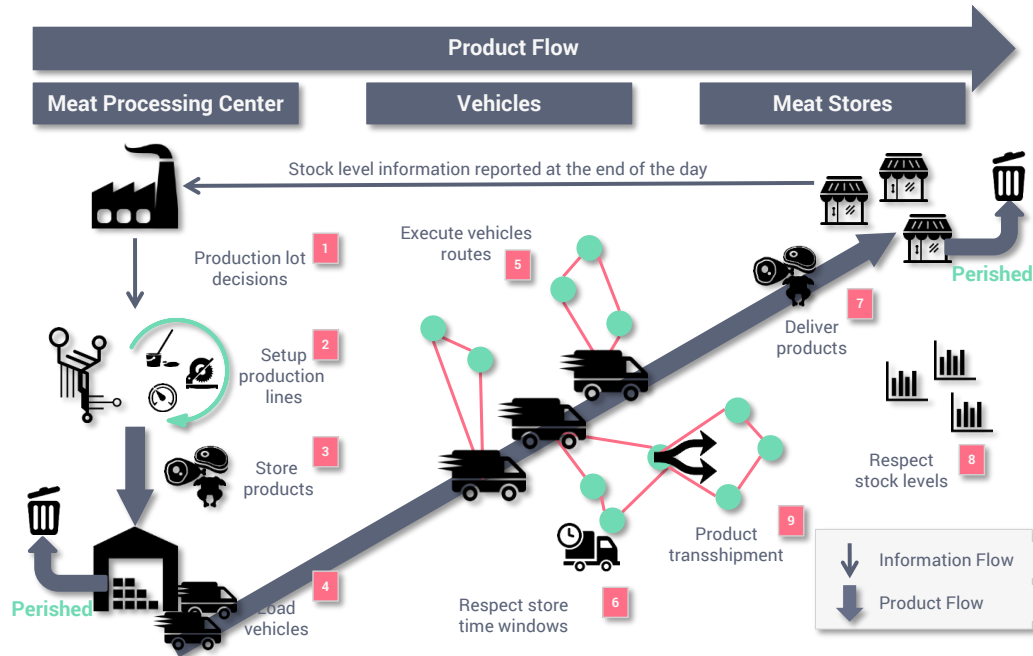


Figure 4.11 – Representation of the realistic multi-product PRP with time windows

The company manages a single MPC where several cutting lines, with particular specifications, are able to make specific cuts to produce different sets of product families. Each family comprises a set of products which are packaged in different sizes. The MPC is the entity which decides the production schedule (1) by applying a VMI policy. In order to produce a certain family, a major setup operation (usually around 1 hour) must be performed on the cutting line. Additionally, in order to set up a cutting line to produce a certain meat product within a product family, a minor setup (usually around 15 minutes) must also be performed. These setup times are usually necessary to clean the machines, to set cutting speeds and thickness, as well as to set the labelling and packaging processes (2). Each time a family is set up on a cutting line, a minimum lot needs to be produced. The MPC works during two work shifts of 8 hours, with a break of 1 hour between shifts. Processed meat products may be directly loaded into vehicles to be delivered or they can be stored at the MPC's refrigerated warehouse to be delivered in a later period. Most of the meat products have a shelf life of around 7 days. Different holding costs per product are incurred when the refrigerated warehouse of the MPC is used (3).

When the products are to be delivered to the meat stores, the fleet is loaded (4), respecting the capacity of each vehicle. Each driver performs a different vehicle route in order to deliver the necessary products and quantities to satisfy the demands of each meat store (5). Note that in this realistic context, each meat store may only receive deliveries during

a certain time window (6). The delivery sizes (7) are defined by the MPC, and have to respect the capacity of the warehouse of each meat store (8). Each store can also stock products incurring a holding cost that is different per store and product. The holding costs are incurred on a daily basis.

Since all the vehicle routes need to start from and finish at the MPC (supplier and depot), it is impossible to perform all the deliveries in the north region, given that the legislation prohibits drivers to work more than 9 hours in a day. Therefore, the MPC also needs to make deliveries to a Transshipment Facility (TF) located in the north of the country. A larger vehicle needs to deliver the products to be distributed by the fleet that is based at the TF (9). This transshipment operation also needs to be performed during a certain time window.

Finally, in the meat stores, the customer demands need to be satisfied on a daily basis. Since we are dealing with food, meat products may perish either at the warehouse of the MPC or at the warehouses of the meat stores.

The current planning process of the company considers all these decisions in a decoupled approach. This means that the events that are triggered by changes both in the information and product flows are not received and interpreted in the same manner by each of the comprised entities (MPC and meat stores). This leads us to four main reasons to investigate an integrated approach:

Capacity issues Currently, each store makes orders without having the information about the MPC capacity. Usually, meat stores order more than they need as they are aware that sometimes the MPC is not able to fulfill all the demand (a clear contributor to the bullwhip effect).

Lack of visibility The MPC only has visibility of the demand of the stores for 1 or 2 periods ahead. This fact is clearly hampering the planning activity of the MPC, as it ends up producing almost every product every day. Therefore, a significant part of the available time capacity is spent setting up the cutting lines to produce small lots of each product.

Improve cost One of the objectives of this research is to quantify the savings obtained by an integrated approach, compared to the decoupled approach where each store is making orders individually. As proven by numerous papers in the past (considering simpler versions of the PRP), we expect to improve the global activity cost.

Deal with complexity Since the planning process currently is built manually by experienced planners, it is important to devise a systematic approach. Although the plans obtained by the planners may already present good quality, the process is still very time consuming and non-scientific. Additionally, when unexpected events occur, it may be quite difficult for the planners to react in a timely manner, thus having an algorithm to aid the planning process is valuable.

4.6.2 Methodology Application

In order to apply the developed solution approach to the case study, some additional constraints need to be taken into account. In fact, these constraints are not common in the context of the PRP literature and need to be adapted to be included in our approach. Table 4.6 presents the additional constraints as well as the specific parts of our approach that have to be adapted.

Table 4.6 – Case study additional constraints

Label	Act.	Constraint / Description	Mathematical form	Adaptation
(a)	Pro	Minimum Family Lot Size	$\sum_{p \in P, t \in T} P_{pmgt} \geq \min Lot_{fm} \quad \forall f \in F, m \in M, g \in T$	IS, PLSP, LPRP
(b)	Pro	Retail Site Perishability	$X_{ipkt} = 0 \quad \forall i \in V, p \in P, k \in K, t \in T, l \in T, t + \max ConsumptionDays < l$	IS, PLSP, LPRP
(c)	Pro	Supplier Perishability	$P_{pmgt} = 0 \quad \forall p \in P, m \in M, g \in T, t \in T, g + \max ShippingDays < t$	IS, PLSP, LPRP
(d)	Pro	Compatible Product/Line	$B_{mpt} \leq comp_{mp} \quad \forall m \in M, p \in P, t \in T$	IS, PLSP, LPRP
(e)	Pro	Time Capacitated Lines	$\sum_f f \cdot st_{fm} \cdot A_{mft} + \sum_p p \cdot st_{pm} \cdot B_{mpt} + \sum_p p \cdot t_{pm} \cdot P_{pmgt} \leq cap_{mt} \quad \forall m \in M, t \in T$	IS, PLSP, LPRP
(f)	Inv	Safety Stocks	$I_{ipt} \geq ss_{ip} \quad \forall i \in V, p \in P, t \in T$	IS, PLSP, LPRP
(g)	Rou	Delivery Time Windows	$a_i \leq W_{it} \leq b_i \quad \forall i \in V, t \in T$	DVRP, LPRP
(h)	Rou	Maximum Route Visits	$\sum_{i \in V} Z_{ikt} \leq \max Vis \quad \forall k \in K, t \in T$	DVRP, LPRP
(i)	Rou	Maximum Route Duration	$\sum_{i, j \in V} H_{ij} \cdot X_{ijkt} \leq \max Dur \quad \forall k \in K, t \in T$	DVRP, LPRP
(j)	Rou	Transshipment Facility	$Z_{ikt} \leq T_t \quad \forall i \in V, k \in K_{TF}, t \in T$	DVRP, LPRP

Legend:

Act. - Activity | Pro - Production | Inv - Inventory | Rou - Routing | IS - Initial Solution | PLSP - Partial LSP | DVRP - Daily VRP | LPRP - Local PRP

To impose a minimum lot size $\min Lot_{fm}$ per family f in each cutting line m , constraints (a) are added. Constraints (b) and (c) impose a maximum time both on the time to ship products after production and on the time to consume products after reception at the meat stores. Constraints (d) ensure that the cutting lines are adequate to cut certain types of meat. For instance, to process minced meat, a different type of cutting line is needed. The parameter $comp_{mp}$ is equal to one if cutting line m is able to process meat product p . Constraints (e) ensure that the time spent on family setups, product setups and processing does not exceed the time capacity cap_{mt} of each line m in each period t . Constraints (f) impose the safety stocks ss_{ip} agreed with each meat store i and product p . This is one method to avoid inadequate stock levels at the end of each planning iteration. Additionally, it allows a better absorption of forecast errors when implementing the solutions in real world. Constraints (g) impose the delivery time window at each store. Constraints (h) and (i) model a maximum number of visits $\max Vis$ and a maximum duration $\max Dur$ for each route. Finally, Constraints (j) model the transshipment facility. We consider that the set of vehicles based at the transshipment facility K_{TF} can only be used if a transshipment operation is performed.

4.6.3 Results Analysis

To provide managerial insights regarding the integration of the planning process of the considered meat store chain, we performed a set of experiments. We recreated the planning process of the company by devising a rolling horizon approach with only two periods of demand visibility. Note that currently the company does not integrate production decisions

with the decisions made by each store (following a Retailer-Managed Inventory (RMI) policy). Therefore, we assume that the company solution can be obtained by using the initial solution of our approach in a rolling horizon approach considering two periods.

We tested the case where all the decisions are integrated while solving the problem, which mimics a VMI policy, and the case where a larger demand visibility is assumed, which would require stores to forecast demands further in time. Table 4.7 shows the results obtained in the experiments comparing them to the company solution. These results are obtained for instances created using the company data corresponding to the month of June 2015, performing the rolling horizon approach for the entire month.

Table 4.7 – Case study results

	Company Solution	Integrated Production	Increased Visibility	Integrated Production Increased Visibility
Cost	2RMI	2VMI	7RMI	7VMI
Routing Cost	170633.00 €	149435.00 €	125793.00 €	118802.00 €
Total Transshipment Cost	7182.00 €	7182.00 €	8208.00 €	8208.00 €
Retailers Holding Cost	29445.18 €	29164.15 €	30124.64 €	30160.04 €
Suppliers Holding Cost	959.56 €	869.60 €	13265.29 €	5799.17 €
Cost Per Delivery	69.69 €	63.47 €	84.35 €	78.58 €
Total Cost	208219.80 €	186650.70 €	177390.93 €	162969.21 €
Cost Reduction		21 569.10 €	30 828.87 €	45 250.59 €
Percentage Savings		10.36%	14.81%	21.73%
Other KPIs				
Number of Routes	395	328	320	291
Number of Visits	2988	2941	2103	2074
Total Driving Time	3345.73 h	2993.40 h	2388.51 h	2277.71 h
Setup Time	292.92 h	306.30 h	286.58 h	257.08 h

We present cost related indicators and other operational Key Performance Indicators (KPIs) as well. In Table 4.7, the company solution corresponds to the case where each planning iteration considers two periods and a RMI policy. We achieve a cost reduction of 10.36% by integrating the production decisions (VMI policy) maintaining the demand visibility of two periods. When we increase demand visibility for seven periods and apply a RMI policy, the cost is reduced by 14.81%. Finally, increasing demand visibility and applying a VMI policy results in a cost reduction of 21.73%

An interesting behaviour is shown by other indicators. Part of the total cost is transferred to the supplier. However, this transfer allows for a large reduction in the routing cost. Furthermore, the cost per delivery increases when demand visibility is increased. However, inventory and routing integrated decisions allowed for a large reduction in the number of visits to the stores.

4.7. Conclusion

In this paper, a large multi-product PRP with time windows is addressed by means of a F&O based matheuristic. A novel mathematical formulation was proposed in order to provide integrated production, inventory, and routing plans for a vertical meat store chain. The large-size instances result in intractable problems which have to be tackled by efficient so-

lution methods. This fact motivated us to propose a novel size reduction and decomposition technique to the PRP allowing for the construction of good quality initial solutions regardless of the size of the problem. Since these solutions are built by several IRP solutions plus a CLSP they do not integrate all decisions (only inventory management and routing decisions are jointly optimized). Therefore, these initial solutions are very good for problems where the link between the production and routing activities is not strong.

We devised a F&O based matheuristic for the PRP so as to integrate all decisions. Our approach iteratively integrates production, inventory management and routing decisions by solving different MIPs with variable size and scope. Matheuristics are still quite rare both in the IRP and PRP literature. Furthermore, to best of our knowledge, adjusting the size of the subproblems based on the runtime of previous iterations is a fresh contribution to the literature related to the F&O heuristic.

It was shown that the algorithm is efficient for solving large-sized instances both for the IRP and PRP. New best solutions are provided for several instances in shorter runtime, compared to state-of-the-art branch-and-cut implementations. Furthermore, we test our region decomposition approach with large IRP instance and show that the algorithm benefits from it.

Additionally, we presented a case study considering a European meat store chain. A set of more complex instances was solved in order to validate the ideas proposed in this paper within a real-world context. The challenge considers additional constraints which include multiple production lines with different specifications and one transshipment facility. Although it is necessary to introduce new constraints, the extensions are trivial and enforce the value of matheuristic approaches that are strongly based on mathematical formulations. After increasing demand visibility and considering a VMI policy, our solution approach achieves savings of 21.73% compared to the company's solution.

Acknowledgements

This research was partly supported by the PhD grant SFRH/BD/108251/2015, awarded by the Portuguese Foundation for Science and Technology (FCT), and by the Fonds de recherche du Québec - Nature et technologies. This support is gratefully acknowledged.

Bibliography

- Absi, N., Archetti, C., Dautère-Pérès, S., and Feillet, D. (2015). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). Optimization-based adaptive large

- neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.
- Adulyasak, Y., Cordeau, J. F., and Jans, R. (2014c). The production routing problem: A review of formulations and solution algorithms. *Computers and Operations Research*, 55:141–152.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015). Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867.
- Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731–1746.
- Armentano, V., a.L. Shiguemoto, and Løkketangen, a. (2011). Tabu search with path re-linking for an integrated production–distribution problem. *Computers & Operations Research*, 38(8):1199–1209.
- Bard, J. F. and Nananukul, N. (2008). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, 12(3):257–280.
- Bard, J. F. and Nananukul, N. (2009). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57(3):713–723.
- Bard, J. F. and Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202–2217.
- Boudia, M., Louly, M. A. O., and Prins, C. (2007). A reactive grasp and path relinking for a combined production-distribution problem. *Computers & Operational Research*, 34(11):3402–3419.
- Boudia, M., Louly, M. A. O., and Prins, C. (2008). Fast heuristics for a combined production planning and vehicle routing problem. *Production Planning & Control*, 19(2):85–96.
- Boudia, M. and Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem. *European Journal of Operational Research*, 195(3):703–715.
- Brown, G., Keegan, J., Vigus, B., and Wood, K. (2001). The kellogg company optimizes production, inventory, and distribution. *Interfaces*, 31(6):1–15.
- Çetinkaya, S., Üster, H., Easwaran, G., and Keskin, B. B. (2009). An integrated outbound logistics model for frito-lay: Coordinating aggregate-level production and distribution decisions. *Interfaces*, 39(5):460–475.
- Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503 – 517.

- Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56:25 – 36.
- Coelho, L. C. and Laporte, G. (2013). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23-24):7156–7169.
- Dantzig, G. B. and Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6(1):80–91.
- Dolan, E. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.
- Fumero, F. and Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation Science*, 33(3):330–340.
- Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.
- Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2):247 – 256.
- Karimi, B., Fatemi Ghomi, S. M. T., and Wilson, J. M. (2003). The capacitated lot sizing problem: A review of models and algorithms. *Omega*, 31(5):365–378.
- Larrain, H., Coelho, L. C., and Cataldo, A. (2017). A variable mip neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines. *Computers & Operations Research*, 85:22–31.
- Lei, L., Liu, S., Ruszczyński, A., and Park, S. (2006). On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):955–970.
- Pochet, Y. and Wolsey, L. (2006). *Production planning by mixed integer programming (Springer Series in Operations Research and Financial Engineering)*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA.
- Qiu, Y., Qiao, J., and Pardalos, P. M. (2017). A branch-and-price algorithm for production routing problems with carbon cap-and-trade. *Omega*, 68:49–61.
- Ruokokoski, M., Solyali, O., Cordeau, J.-F., Jans, R., and Sural, H. (2010). Efficient Formulations and a Branch-and-Cut Algorithm for a Production-Routing Problem. *GERAD Technical Report G-20 10-66, HEC Montreal Canada*, pages 1–43.
- Sarmiento, A. M. and Nagi, R. (1999). A review of integrated analysis of production–distribution systems. *IIE Transactions*, 31(11):1061–1074.
- Schmid, V., Doerner, K. F., and Laporte, G. (2013). Rich routing problems arising in supply chain management. *European Journal of Operational Research*, 224(3):435–448.

- Solyalı, Q. and Süral, H. (2009). A Relaxation Based Solution Approach for the Inventory Control and Vehicle Routing Problem in Vendor Managed Systems. *Modeling, Computation and Optimization*, (2000):171–189.
- Stadtler, H. and Kilger, C. (2010). *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. Springer Publishing Company, Incorporated, 4th edition.
- Thomas, D. J. and Griffin, P. M. (1996). Coordinated supply chain management. *European Journal of Operational Research*, 94(1):1 – 15.
- Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1):89–96.

Appendix 4.A Decomposition strategy base formulations

Before presenting the base formulations of each decomposition strategy we comment on the order in which they are called. Since the largest expected size reduction (obtained in the first phase) usually comes from the routing part, and as it is commonly the largest portion of the total cost, the first decomposition strategy starts by exploring the *Daily VRP* formulation which focuses on routing cost. The second decomposition strategy, which uses the *Partial LSP* formulation, is not so promising at this point because all the inventories have already been optimized, taking into account the delivery schedule of the initial solution. Though large gains are not expected, improvements can still be achieved by jointly optimizing inventories of the supplier and of the retail sites, particularly in the cases where the problem is divided into regions. Furthermore, the production line setups may also be changed to better redistribute the production quantities needed to satisfy the shipping quantities in each period. In case the problem considers setup costs, these can also be improved. In fact, the order in which the first two decomposition strategies are explored is irrelevant as the decisions to be analysed by each of them are completely independent. However, in the third decomposition strategy all the decisions inherent to the PRP are taken into account locally. Integrating most of the decisions taken at the supplier with the routing decisions of a set of retail sites results in difficult subproblems. For this reason, we decided to follow this decomposition strategy in the last place (after reaching a local optimum regarding the routing part of the solution). The following subsections describe the three base formulations of the decomposition strategies comprised in our F&O approach.

4.A.0.1 Daily VRP base formulation

The inputs to this model comprise a single period t , a set of vehicles \mathcal{K} , a set \mathcal{V} including the retail sites visited by these vehicles in the incumbent solution, and the delivery quantities made to each retail site, denoted by d_i . The deliveries are needed because the fleet is not homogeneous (different capacities vc_k), thus their quantities have to be taken into account in the vehicle capacity constraints. Additionally, vehicles must respect the time windows $[a_i, b_i]$ of each location i . The time needed for a vehicle to traverse an arc (i, j) is given by tt_{ij} . Note that it is not necessary to load more than one period, as there are no dependencies between periods in this formulation, given that the delivery quantities are fixed. In fact, this formulation is similar to the routing part of the PRP formulation presented in Section 4.3 but the index t is dropped. The model works with the binary variables X_{ijk} , which are equal to one if vehicle k traverses arc (i, j) . Continuous variables W_{ik} define the time at which vehicle k arrives at location i .

(DailyVRP):

$$\text{minimize } \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} c_{ij} \cdot X_{ijk} \quad (4.32)$$

s.t.

$$\sum_{j \in \mathcal{V}} \sum_{k \in \mathcal{K}} X_{ijk} = 1 \quad \forall i \in \mathcal{V}' \quad (4.33)$$

$$\sum_{j \in \mathcal{V}} X_{ijk} - \sum_{j \in \mathcal{V}} X_{jik} = 0 \quad i \in \mathcal{V}, k \in \mathcal{K} \quad (4.34)$$

$$\sum_{i \in \mathcal{V}'} \sum_{j \in \mathcal{V}} X_{ijk} \cdot d_i \leq vc_k \quad \forall k \in \mathcal{K} \quad (4.35)$$

$$W_{ik} + tt_{ij} \leq W_{jk} + M \cdot (1 - X_{ijk}) \quad \forall i \in \mathcal{V}, j \in \mathcal{V}, k \in \mathcal{K} \quad (4.36)$$

$$a_i \leq W_{ik} \leq b_i \quad \forall i \in \mathcal{V}, k \in \mathcal{K} \quad (4.37)$$

$$\begin{aligned} X_{ijk} &\in \{0, 1\} \quad \forall i, j \in \mathcal{V}, k \in \mathcal{K} \\ W_{ik} &\geq 0 \quad \forall i \in \mathcal{V}, k \in \mathcal{K}. \end{aligned} \quad (4.38)$$

The objective function (4.32) of the *Daily VRP* formulation simply accounts for the routing cost to perform the deliveries that are currently scheduled in the solution and were loaded to be optimized. Constraints (4.33) ensure that the scheduled deliveries in the incumbent solution must continue to be performed after solving this subproblem. Constraints (4.34) are the so called flow conservation constraints to ensure that if a vehicle visits a node, it has to leave that node. Constraints (4.35) are to ensure that vehicle capacities are respected. Constraints (4.36) define the time at which each retail site is visited by a vehicle. This time is used to define constraints (4.37), where the time windows $[a_i, b_i]$ of each retail site must be respected. M is a big number which is at least the duration of a day.

4.A.0.2 Partial LSP base formulation

The *PartialLSP* formulation uses the same variables presented before but in this case, index k is dropped from delivery quantities D_{ipl} as the visits are assumed to be fixed:

(PartialLSP):

$$\begin{aligned}
 \text{minimize} \quad & \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{g \in \mathcal{T}} \sum_{h=0}^g \sum_{t=g+1}^{|\mathcal{T}|} h_{0p} \cdot P_{pmht} \\
 & + \sum_{i \in \mathcal{V}'_t} \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \sum_{h=0}^t \sum_{l=t+1}^{|\mathcal{T}|} h_p \cdot D_{iphl}
 \end{aligned} \tag{4.39}$$

s.t. (4.2), (4.3), (4.4), (4.5), (4.8),

$$\sum_{i \in \mathcal{V}'_{kt}} \sum_{p \in \mathcal{P}} \sum_{l \in \mathcal{T}} D_{iptl} \leq c_{kt} \quad \forall k \in \mathcal{K}, t \in \mathcal{T} \tag{4.40}$$

$$\sum_{g=0}^t \sum_{m \in \mathcal{M}} P_{pmgt} = d_{pt}^{\text{external}} + \sum_{i \in \mathcal{V}'} \sum_{l=t}^{|\mathcal{T}|} D_{iptl} \quad \forall p \in \mathcal{P}, t \in \mathcal{T} \tag{4.41}$$

$$\sum_{t=0}^l z_{it} \cdot D_{iptl} = d_{ipl} \quad \forall i \in \mathcal{V}', p \in \mathcal{P}, l \in \mathcal{T} \tag{4.42}$$

$$P_{pmtl}, D_{iptl} \geq 0. \tag{4.43}$$

The objective function (4.39) is similar to the one presented in the LSP formulation used in the initial solution. However, the inventory holding costs are now accounted for each location and not only for the supplier. In order to define the necessary conditions to model production line capacities, setups, and the warehouse capacity of the supplier, constraints (4.2), (4.3), (4.4), and (4.5) are added from the *MPRPTW* formulation presented in Section 4.3. Constraints (4.8), from the *MPRPTW* formulation, are also added to ensure that warehouse capacities are respected for each retail site. However, the index k is now dropped. The vehicles visiting the selected retail sites in a certain period may also visit other retail sites that were not loaded into the model. Accordingly, we define a partial capacity c_{kt} for each vehicle k in each shipping period t . Constraints (4.40) ensure that the deliveries performed to the set of visited retail sites \mathcal{V}'_{kt} (visited by vehicle k in shipping period t) do not exceed the capacity of the vehicle. Constraints (4.40) ensure that the deliveries performed to the set of visited retail sites \mathcal{V}'_{kt} (visited by vehicle k in shipping period t) do not exceed the capacity c_{kt} of the vehicle in each shipping period. Constraints (4.41) force the production quantities to satisfy both the demands of loaded and unloaded retail sites. Constraints (4.42) ensure the demand satisfaction of each loaded retail site. Finally, the bounds of all variables are defined by constraints (4.43).

4.A.0.3 Local PRP base formulation

The *LocalPRP* formulation includes most of the constraints presented in the *MPRPTW* formulation and it is defined as follows:

(LocalPRP):

$$\begin{aligned}
 \text{minimize} \quad & \sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}} \sum_{g \in \mathcal{T}} \sum_{h=0}^g \sum_{t=g+1}^{|\mathcal{T}|} h_{0p} \cdot P_{pmht} \\
 & + \sum_{i \in \mathcal{V}_t} \sum_{p \in \mathcal{P}} \sum_{k \in \mathcal{K}_t} \sum_{t \in \mathcal{T}} \sum_{h=0}^t \sum_{l=t+1}^{|\mathcal{T}|} h_p \cdot D_{ipkhl} \\
 & + \sum_{i \in \mathcal{V}_t} \sum_{j \in \mathcal{V}_t} \sum_{k \in \mathcal{K}_t} \sum_{t \in \mathcal{T}} c_{ij} \cdot X_{ijk t}
 \end{aligned} \tag{4.44}$$

s.t. (4.2), (4.3), (4.4), (4.5),

$$\sum_{g=0}^t \sum_{m \in \mathcal{M}} P_{pmgt} = d_{pt}^{ext} + \sum_{i \in \mathcal{V}_t} \sum_{k \in \mathcal{K}_t} \sum_{l=t}^{|\mathcal{T}|} D_{ipk t l} \quad \forall p \in \mathcal{P}, t \in \mathcal{T} \tag{4.45}$$

$$\sum_{k \in \mathcal{K}_t} D_{ipk t l} = q_{iptl} \quad \forall i \in \mathcal{V}_t^{out}, p \in \mathcal{P}, t \in \mathcal{T}, l \in \mathcal{T}, t \leq l \tag{4.46}$$

$$\sum_{k \in \mathcal{K}_t} Z_{ikt} = 1 \quad \forall i \in \mathcal{V}_t^{out}, t \in \mathcal{T}. \tag{4.47}$$

The objective function minimizes both the supplier and retail sites' holding costs, and the routing cost. From the *MPRPTW* formulation, constraints (4.2), (4.3), (4.4), and (4.5) are added to model the family setups, the product setups, the time capacity of each production line, and the warehouse capacity constraints of the supplier, respectively. Constraints (4.45) impose that the produced quantities must satisfy both the internal and external demand considered in the model. Note that each period considers a different set of vehicles \mathcal{K}_t , which includes the vehicles performing the routes in the incumbent solution visiting some retail site belonging to \mathcal{V} . The sets of nodes \mathcal{V}_t includes all the retail sites that can be visited in a period (white and grey nodes in Figure 4.7). Constraints (4.46) force the delivery quantities to the fixed retail sites (gray nodes), belonging to the set \mathcal{V}_t^{out} , to be equal to the delivery quantities q_{iptl} defined by the incumbent solution. For each retail site in these sets, the delivery quantities will be fixed, thus we also now know that their warehouse capacity is not violated. However, the vehicle that performs the delivery can still change. Constraints (4.47) ensure that a visit is still performed to the fixed retail sites of each period. The remaining constraints belong to the *MPRPTW* formulation and are added with proper changes in the sets to be considered. For the selected retail sites, constraints (4.7) ensure demand satisfaction of the free retail sites. In constraints (4.9) and

(4.10), which model vehicle capacity and delivered quantities (respectively), the set \mathcal{V}' is replaced by the sets $\mathcal{V}_t \setminus \{0\}$ and the set \mathcal{K} is substituted by the sets \mathcal{K}_t , in each period t . For the remaining constraints regarding the routing part of the problem, namely (4.11) - (4.15), the set \mathcal{V}' is substituted by the sets \mathcal{V}_t and the set \mathcal{K} by the sets \mathcal{K}_t , in each period t .

Routing for time window assignment

The time window assignment vehicle routing problem with product dependent deliveries

Fábio Neves-Moreira* · Diogo Pereira da Silva[†] · Luís Guimarães^{*†} · Pedro Amorim^{*†}
· Bernardo Almada-Lobo^{*†}

Submitted to *Transportation Research Part E: Logistics and Transportation Review*, 2017 (under review)

Abstract In food retail contexts, the delivery time windows to stores must comply with operational constraints imposed by employee expertise, equipment requirements and product characteristics. Here, routing operations need to be efficiently synchronized with resource availability. To tackle this problem, we formulate a two-stage stochastic optimization problem which is solved by means of a fix-and-optimize based matheuristic. The first stage assigns product dependent time windows while the second stage defines delivery schedules. Our approach outperforms average demand scenario approaches, decreasing the average cost by 5.3%. Moreover, significant savings are also achieved using real-world data provided by a large European food retailer.

Keywords retail operations · vehicle routing · time window assignment · multiple product deliveries · fix-and-optimize · stochastic optimization

*INESC TEC and Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

[†]LTP Labs, Rua Doutor Júlio de Matos, 4200-355 Porto, Portugal

5.1. Introduction and Related Work

Most distribution networks need to assign regular time windows to its delivery locations. Actually, in food retail, it is quite rare to find applications with complex unloading operations where no time windows are stipulated. This is due to the importance of the delivery process in managing physical space, product inventories, and personnel [Spliet and Gabor, 2015]. Food retailers face several planning challenges related to warehouse, fleet, and retail site operations in order to provide good quality service to their final customer [Sternbeck and Kuhn, 2014]. An example of a food retailer, including the involved stakeholders, is depicted in Figure 5.1.

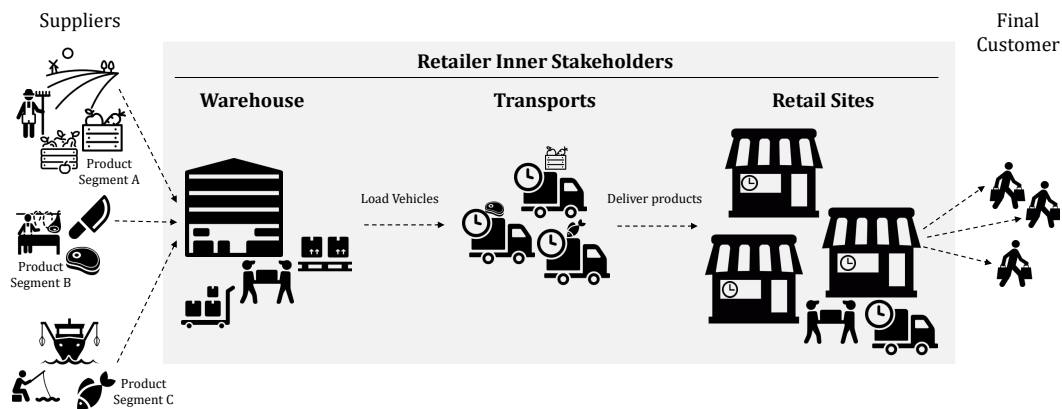


Figure 5.1 – An example of a food retailer including its stakeholders. Products are received from many suppliers and prepared in one warehouse. Afterwards, vehicles deliver products to each retail site, respecting product specific time windows.

In the warehouse, products are received and prepared in shipping units (usually, products are palletized). Each supplier may have a different delivery lead time and time window. Therefore, different product segments are received by the warehouse in different periods of the day, influencing the preparation cycles. For instance, if fresh products are required to be in the retail site before its opening time, fresh product suppliers need to perform night deliveries so that the warehouse is able to prepare them in time.

The transportation activity also adds several constraints to be addressed. In multi-product operations, each product segment requires different logistics equipment and operations. Vehicles are often required to maintain product compatible temperatures. Retail sites may have accessibility constraints enforcing the fleet to comprise vehicles with different dimensions and heterogeneous trailers. Additionally, labour laws do not allow changes in the drivers' shifts for a given period of time. This means that the number of drivers available along a day is not constant. For instance, when one driver ends his shift and another driver starts working using the same vehicle, the vehicle is idle at the depot for some time. These drivers will never perform routes that are incompatible with their shift switching period, which is not easily changed due to law restrictions. Hence, shift switching periods limit the number of available drivers.

Retail site constraints often demand certain products to be delivered during specified peri-

ods of the day. For illustration purposes, it is quite common for some final customers to prefer buying fresh products (such as fruits or fish) early in the morning. This behaviour forces some products to be delivered to the retail site early enough so that the sales areas are ready at the opening hour. Likewise, laws to prevent noise near populated areas during the night time or traffic congestions at rush hours may demand specific delivery time windows at certain retail sites. Moreover, some products may only be received by expert collaborators that are a scarce resource only available during specific periods of the day. The aforementioned reasons demand the definition of product dependent time windows to ensure feasible unloading operations at the retail sites.

Assigning delivery time windows to each retail site is a tactical planning problem that needs to be tackled by retailers. These decisions largely impact the operational decisions taken in transportation planning, namely routing and fleet costs (Deflorio et al. [2012]). Companies operating in similar environments to that described above are obviously interested in optimizing their integrated activities to pursue substantial cost savings. It is definitely worth a methodology to define the set of time windows capable of providing cost reduction opportunities.

The underlying problem is very complex as it involves assigning time windows based on transportation plans. The Time Window Assignment Vehicle Routing Problem (TWAVRP) can be defined as a two-stage stochastic optimization problem which integrates the tactical time window assignment decisions (first-stage) with the operational vehicle routing decisions (second-stage). Given a set of locations to be visited within a regular time period, the first stage decisions are to assign a set of time windows to each location, before demand is known. In the second stage, after the demand is revealed for each day, a delivery schedule respecting the assigned time windows is defined. Since the time windows remain unchanged for a reasonably long period of time, the goal is the way to perform this assignment while allowing for a daily efficient transportation planning across every demand scenario. Given their importance in many applications, time window assignment problems are now receiving an increasingly level of attention.

Since the second stage decisions of the Time Window Assignment Vehicle Routing Problem (TWAVRP) demand the definition of a delivery schedule respecting time windows for each demand scenario, it is closely related to the VRP with Time Windows (VRPTW). In the VRPTW a fleet of vehicles is used to satisfy the demand of a set of customers while respecting vehicle capacity and delivery time window constraints (Toth and Vigo [2001]). Despite its extensive literature (see [Cordeau et al., 2001], [Baldacci et al., 2012], and [Kallehauge, 2008]) the VRPTW is still one of the most challenging problems in combinatorial optimization, which takes the TWAVRP to, at least, the same level of complexity. This latter problem has not been well studied. To the best of our knowledge, there are only four papers addressing it.

Spliet and Gabor [2015] introduce the TWAVRP and propose a branch-price-and-cut algorithm to find the optimal expected travelling cost. The computational results show that the algorithm is capable of solving instances of the problem up to 25 customers and three demand scenarios. The authors state that using time windows defined for an average demand scenario results in 1.85% costs increase compared to TWAVRP solutions.

Spliet and Desaulniers [2015] tackle the same problem with discrete time windows (DT-

WAVRP), which is more applicable to real-world contexts. The authors develop an improved branch-price-and-cut algorithm and a Tabu Search (TS) based on a column generation heuristic to minimize the total transportation cost. The approach is tested on randomly generated instances with up to 30 customers and 5 demand scenarios. The results show that considering 5 scenarios allows for an average cost reduction of 3.64% compared to a single-scenario approach.

Recently, [Split et al. \[2017\]](#) propose an extended mathematical formulation considering time-dependent travel times and apply a branch-price-and-cut to obtain optimal solutions for instances with up to 25 customers.

Despite labelling it a vehicle routing problem with self imposed time windows (VRP-SITW), [Jabali et al. \[2015\]](#) also address the TWAVRP. The proposed approach considers uncertainty in travel times. A TS heuristic assigns customers to vehicles and the timing decisions are subsequently generated by solving a linear subproblem. The objective is to optimize the expected effect of one disruption. The approach is able to solve adapted VRPTW instances with up to 100 customers. However, the instances only consider demands for one period.

The four aforementioned papers are the ones closer to our problem definition however there are additional works tackling similar problems with different approaches. [Agatz et al. \[2011\]](#) use aggregate-level routing models and continuous approximation methods to estimate expected transportation costs while defining customer time windows. The problem is defined as a Time Slot Management Problem (TSMP). The idea is to define sets of time windows to be offered to a set of potential customer services, with uncertain demands, occurring in different zip code areas. Since the routing operation is not explicitly taken into consideration this approach is not applicable to cases where routing constraints invalidate most of the time window assignments.

Other works try to achieve solutions for the TWAVRP based on consistency requirements. The concept of consistent service is intimately related with the ideas explored in the aforementioned papers. The goal is to design consistent routes which satisfy any of the following requirements whenever a customer is served: (i) arrival-time consistency, wherein a customer should be visited roughly at the same time during the day, (ii) person-oriented consistency, in which a customer needs to be visited by the same driver (as in [Braekers and Kovacs \[2016\]](#)), and (iii) delivery consistency, when a customer should receive the same quantity of goods. The Consistent Vehicle Routing Problem (conVRP) with arrival-time consistency requirements is the most similar to the TWAVRP. [Kovacs et al. \[2015\]](#) aim at having the same driver visiting the same customers at roughly the same points in time. The authors experiment different input parameters to trade-off the travel cost against a customer satisfaction measure and conclude that by allowing more than one driver to visit each customer, large cost savings are obtained. [Groër et al. \[2009\]](#) develop an heuristic for serving a set of customers with known demands, using consistent routes. The algorithm is tested over five simulated data sets with up to 1000 customers and a real-world data set with more than 3700 customers from a small package shipping company. The authors state that the consistent routes are able to achieve customer service objectives with low total travel time. [Subramanyam and Gounaris \[2016\]](#) deal with consistency of arrival times in a Travelling Salesman Problem (TSP). They propose a branch-and-cut algorithm which is the first exact

method for the problem. The solution approach is tested on instances with 50 customers and 5 periods, showing that arrival time consistency can be achieved with a small increase in routing costs. A summary of the literature review is provided in Table 5.1 together with the positioning of our challenge.

Table 5.1 – The literature regarding arrival-time consistency is still scarce. Most approaches are found in the conVRP literature where time window amplitude is a decision. We are particularly interested in the TWAVRP with fixed time window amplitude.

Authors (year)	Problem	T.W. Var.	S.D.	Problem Description			C.	Periods	Solution Method	
				Travel Time	Products	Vehicles			Type	Approach
Agatz et al. [2011]	(2011) TSMP	Discrete	-	Deterministic	Single	Limited (Hom)	✓	Single	H	Approximation (no routing)
Spliet and Gabor [2015]	(2015) TWAVRP	Continuous	-	Deterministic	Single	Unlimited (Hom)	✓	Multiple	E	Branch-Price-And-Cut
Spliet and Desaulniers [2015]	(2015) DTWAVRP	Discrete	-	Deterministic	Single	Unlimited (Hom)	✓	Multiple	E	Branch-Price-And-Cut
Spliet et al. [2017]	(2017) TWAVRP	Continuous	-	Uncertain	Single	Unlimited (Hom)	✓	Multiple	E	Branch-Price-And-Cut
Jabali et al. [2015]	(2015) VRP-SITW	Continuous	-	Uncertain	Single	Limited (Hom)	✓	Single	H	Tabu Search
Groër et al. [2009]	(2009) ConVRP	Continuous	-	Deterministic	Single	Limited (Hom)	✓	Multiple	H	Record-To-Record
Kovacs et al. [2015]	(2015) ConVRP	Continuous	-	Deterministic	Single	Limited (Hom)	✓	Multiple	H	Large Neighbourhood Search
Subramanyam and Gounaris [2016]	(2011) ConTSP	Continuous	-	Deterministic	Single	Single	-	Multiple	E	Branch-And-Cut
Our Approach	TWAVRPP	Discrete	✓	Deterministic	Multiple	Unlimited (Het)	✓	Multiple	MH	Fix-And-Optimize

Legend:

T.W. Var. - Time Window Variables | S.D. - Split Deliveries | C. - Capacitated | Hom. - Homogeneous | Het. - Heterogeneous | H - Heuristic | E - Exact | MH - Matheuristic

To the best of our knowledge, no single approach has considered split deliveries and multiple product segments. Therefore, none of the publications explores product dependent time windows and multiple product routes (i.e., different product segments are consolidated). Regarding solution methods, all the approaches proposed for time window assignment are either purely heuristic or exact, hence hybrid methods have not been explored. Furthermore, an important aspect for real-world implementation is the consideration of the number of active vehicles in each point in time. In vehicle routing literature, some works propose set-partitioning formulations which consider the fixed vehicle cost embedded in the cost of each route (as in [Salhi et al. \[2013\]](#)), ignoring the fact that a vehicle can perform multiple routes during the day.

In this paper, we build upon the aforementioned ideas and aim at extending the TWAVRP to deal with product dependent time windows, while complying with specific business constraints. Our approach is inspired by the case of a large European food retailer which owns two warehouses with a fleet for serving around 200 retail sites with time windows defined for different product segments. The state-of-the-art approaches in this field do not provide any solution for such challenge, as there is a significant complexity increment over the current formulations and approaches available in the literature.

Hence, the main contributions of this paper are the following. We propose an extended mathematical formulation for assigning time windows in real-world contexts, capable of dealing with (1) product dependent time windows; (2) multiple product deliveries; (3) split deliveries. A novel objective function, considering both the travelled distance and fleet requirements cost, is also proposed. We develop a matheuristic for dealing with real-world sized instances, with business-related decomposition strategies for accelerating the convergence of the algorithm. Extensive computational experiments are presented on a set of real-world instances. We test three operational models to assess the impact of different time window conditions on the solutions obtained by the proposed approach. A sensitivity analysis is performed over the number of retail sites where time windows changes are introduced. We provide interesting business-related insights as well as real-world consider-

ations that have been revealed during the development and implementation of our solution approach.

The remainder of this paper is organized as follows. In Section 5.2 we introduce the TWAVRP with product dependent deliveries and present the proposed formulation for the problem. Section 5.3 details our matheuristic solution approach. In Section 5.4, the computational experiments are analysed. In Section 5.5, the main conclusions of this work are presented as well as directions for future work.

5.2. Problem Description and Mathematical Formulation

To describe the TWAVRP, consider a complete graph $G = (N, E)$, where the set of vertices $N = \{0, 1, \dots, n\}$ is partitioned into vertex 0, the warehouse (depot), and vertices $\{1, \dots, n\}$, corresponding to n locations to be served. Edges $(i, j) \in E$ are associated with a travelling cost c_{ij} , measured in monetary units (m.u.), and a travelling time t_{ij} , measured in time units (t.u.). Let $P = \{1, \dots, p\}$ be the set of product segments that can be ordered by each location. A set Ω of scenarios is given, where each scenario corresponds to a realization of services to be satisfied on a certain day. Let $S = \{(v, h) : v \in N, h \in P\}$ be the set of services composed by a pair of a location v and a product segment h . The quantity demanded by a service $s \in S$ in scenario $\omega \in \Omega$ is given by d_s^ω . The probability of occurrence of scenario ω is given by p_ω and $\sum_{\omega \in \Omega} p_\omega = 1$. Two types of time windows are considered in a time window assignment problem: exogenous and endogenous. The former indicate a time period where delivery time windows can be defined from. For example, the retail site opening hours is an exogenous time window (fixed). The latter are simply the delivery time windows known from other optimization problems (such as the VRPTW) that are defined inside exogenous time windows (decision). For each service s , let $[a_s, b_s]$ be an exogenous time window within which the endogenous time window must be defined. The service has an expected service time st_{rs} and needs to start and finish within the boundaries of the exogenous time window. To perform the deliveries, an heterogeneous fleet is available. The fleet is composed by a set K of vehicle types, where each vehicle type $k \in K$ has capacity q_k . Each vehicle type can only transport the product segments in the set P_k that are compatible with the temperature of its trailer. The fixed cost of each vehicle type is given by fc_k .

In this paper, we use $\langle E_r, S_r, T_r, k_r, pt_r, st_r, et_r, dur_r, c_r \rangle$ to define a route r belonging to the entire set of routes Θ . $E_r \subseteq E$ are the sequenced edges to be traversed by route r . $S_r \subseteq S$ is the set of services that can be served by route r considering all the operational constraints related to the transportation planning. T_r is an ordered set with the arrival times to the locations visited in the route. Parameter t_{rs} indicates the arrival time of route r to the location of service s , and k_r is the index of the vehicle type used to perform route r . This vehicle type has capacity q_{k_r} and fixed cost fc_{k_r} . Let pt_r be the preparation starting time of the vehicle loading. Parameters st_r , and et_r are the route start time, and the route end time, respectively. The route duration is given by $dur_r = \sum_{(i,j) \in E_r} t_{ij}$ and the travelling cost is given by $c_r = \sum_{(i,j) \in E_r} c_{ij}$. For modelling reasons, we define the subset Θ^{mp} containing all the routes where a multiple product delivery can be performed. Additionally, for multiple product deliveries, we define the parameter \underline{q} which indicates the minimum quantity to

be delivered related to the main product. For instance, if the time window for the fresh segment is used in a multiple product route (containing fresh products), the respective product segment needs to deliver at least a quantity q .

Let W be the set of discrete time slots within one scenario. Each time slot $w = [\underline{w}, \bar{w}] \in W$ has fixed amplitude $\Delta = \bar{w} - \underline{w}$ (for instance, 30-minute slot within the day). Associated with each route r , a parameter e_{rsw} indicates whether it is possible to provide service s in time slot w . Thus, e_{rsw} is equal to 1, if $\langle r, s, w \rangle$ belongs to the set $RS = \{\langle r, s, w \rangle | r \in \Theta, s \in S, w \in W : \underline{w} \leq t_{rs} \leq \bar{w} + st_{rs} \wedge \underline{w} + st_{rs} \leq b_s \wedge \bar{w} + \Delta > a_s\}$, 0 otherwise. Moreover, h_{rw} indicates whether route r is active during time slot w . h_{rw} is equal to 1 if time slot w intersects the interval $[pt_r, et_r]$.

In order to formulate the Time Window Assignment Vehicle Routing Problem with Product Dependent Deliveries (TWAVRPP) as a two-stage stochastic optimization problem, we use decision variables y_{sw} for first stage decisions (i.e., tactical time window assignment) and variables $z_r^\omega, x_{rs}^\omega, w_{rs}^\omega$ for second stage decisions (i.e., operational vehicle routing). Variables y_{sw} are used to assign the beginning of a time window for service s to time slot w . Let z_r^ω be the binary route variables indicating the selection of a route r in scenario ω . Continuous variables x_{rs}^ω define the portion of service s demand served with route r in scenario ω . Binary variables w_{rs}^ω are used in multi-product deliveries to define the main product of the route (i.e., the product segment of the time window to be used). To quantify the maximum number of vehicles used in each scenario ω , auxiliary integer variables v_k^ω are used.

Our proposed set-partitioning formulation for the TWAVRPP reads:

(TWAVRPP):

$$\text{minimize } \sum_{\omega \in \Omega} p_\omega \cdot \left(\sum_{r \in \Theta} c_r \cdot z_r^\omega + \sum_{k \in K} fc_k \cdot v_k^\omega \right) \quad (5.1)$$

s.t.

$$\sum_{w \in W} y_{sw} = 1 \quad \forall s \in S \quad (5.2)$$

$$\sum_{r \in \Theta} x_{rs}^\omega = 1 \quad \forall s \in S, \omega \in \Omega \quad (5.3)$$

$$\sum_{s \in S} d_s^\omega \cdot x_{rs}^\omega \leq z_r^\omega \cdot q_{k_r} \quad \forall r \in \Theta, \omega \in \Omega \quad (5.4)$$

$$x_{rs}^\omega \leq \sum_{w \in W} e_{rsw} \cdot y_{sw} \quad \forall r \in \Theta, s \in S, \omega \in \Omega \quad (5.5)$$

$$\sum_{s \in S} w_{rs}^\omega = z_r^\omega \quad \forall r \in \Theta^{mp}, \omega \in \Omega \quad (5.6)$$

$$\underline{q} \cdot w_{rs}^\omega \leq d_s^\omega \cdot x_{rs}^\omega \quad \forall r \in \Theta^{mp}, s \in S, \omega \in \Omega \quad (5.7)$$

$$w_{rs}^\omega \leq \sum_{w \in W} e_{rsw} \cdot y_{sw} \quad \forall r \in \Theta^{mp}, s \in S, \omega \in \Omega \quad (5.8)$$

$$x_{rs}^\omega \leq \sum_{s' \in S} w_{rs'}^\omega \quad \forall r \in \Theta^{mp}, s \in S, \omega \in \Omega \quad (5.9)$$

$$\sum_{s' \in S \setminus \{s\}} d_{s'}^\omega \cdot x_{rs'}^\omega \geq w_{rs}^\omega + \underline{q} \cdot (z_r^\omega - w_{rs}^\omega) \quad \forall r \in \Theta^{mp}, s \in S, \omega \in \Omega \quad (5.10)$$

$$\sum_{r \in \Theta} h_{rw} \cdot z_r^\omega \leq v_k^\omega \quad \forall \omega \in \Omega, w \in W \quad (5.11)$$

$$x_{rs}^\omega \in \mathbb{R}^+, \quad y_{sw}, z_r^\omega, w_{rs}^\omega \in \{0, 1\}, \quad v_k^\omega \in \mathbb{N}_0. \quad (5.12)$$

The objective function (5.1) minimizes the total fleet requirement and travelling costs across all scenarios. Constraints (5.2) ensure that an endogenous time window is defined for each service. Constraints (5.3) impose that all services are satisfied in each scenario. Constraints (5.4) are vehicle capacity requirements. Constraints (5.5) ensure that a route can only provide a service if its arrival time at the store is comprised within an assigned endogenous time window of the service demand. Constraints (5.6) - (5.10) are dedicated to multi-product routes. Multi-product routes impose the selection of main service to define the service time window that will be used to perform the consolidated delivery. Constraints (5.6) force the selection of a main service when a multi-product route is performed. After selecting the main service, constraints (5.7) guarantee that the delivery quantity of the main service is larger than \underline{q} . Constraints (5.8) require the time window of the main service to be compatible with the arrival time of the route. Constraints (5.9) allow a multi-product route to perform a delivery in case its main service is selected. By definition, multi-product routes deliver more than one product, as stated in constraints (5.10). Finally, expressions (5.11) count the maximum number of active vehicles in each scenario. Additional constraints for adapting the model to common real-world applications are presented in 5.A.

5.3. Solution Method

Due to the complexity of the TWAVRP we chose to solve it with an heuristic approach. Since we are interested in real-world applications of the TWAVRPP, solving large problem instances is just one of the challenges to be addressed by our solution method. In business contexts, planners are often interested in a good trade-off between solution quality and computational time. Furthermore, new business constraints need to be considered from time to time. Considering these requirements, we support that a mathematical programming based matheuristic is a well-balanced choice. Our solution method is based on a three-phase approach to be detailed in the following subsections of the paper. In the

first phase data is pre-processed to create a set of routes to be used by the proposed set-partitioning formulation. The set of routes comprises routes of different types, visiting one or many locations and delivering one or many products. The second phase builds an initial solution for the problem. The problem is solved considering one scenario at a time, fixing the variables of previously optimized scenarios and relaxing variable integrality constraints in the remaining scenarios. In the third phase, the algorithm improves the solution found in phase two by iteratively solving a series of sub-problems which focus in smaller subsets of decisions at a time. An overview of the solution method is depicted in Figure 5.2.

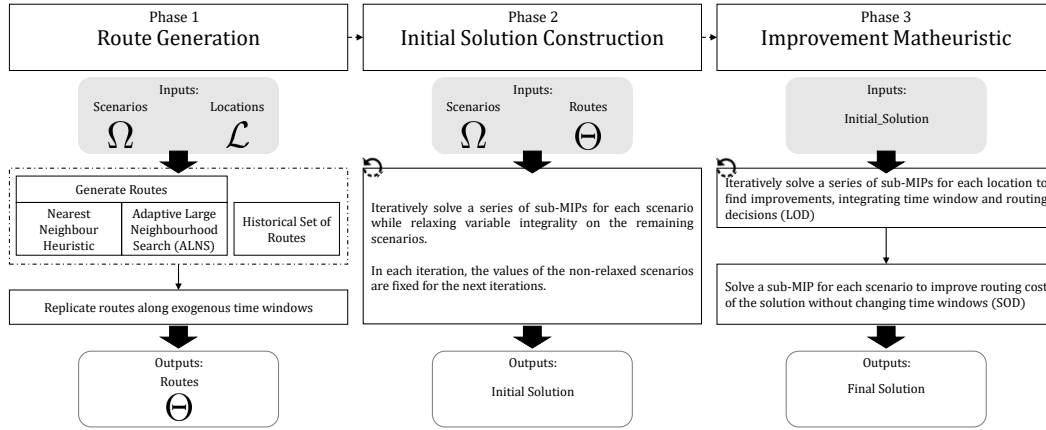


Figure 5.2 – The proposed solution method comprises three phases which aim at decomposing and providing good solutions for large TWAVRP instances. The first phase generates a route set Θ that is used in the second phase to build the initial solution for the problem. Afterwards, in the third phase, location (LOD) and scenario-based (SOD) decompositions are used to define subproblems that are iteratively solved to seek for solution improvements.

5.3.1 Route Generation

The route generation phase aims at preparing all the parameters related to the routes to be used by the two remaining phases. Since we use a set-partitioning formulation, most of the routing constraints are exogenous to the mathematical formulation. Therefore, the procedure used in the route generation phase should ensure that all the generated routes are feasible in the problem to be solved. The inputs to our route generation phase correspond to data related to a set of demand scenarios Ω and the characteristics of the set of locations \mathcal{L} where the deliveries are to be made. In this phase, the developed procedure creates three different types of routes:

- **Direct Routes** deliver only one product to one location, thus they serve only one service.
- **Multi-Location Routes** deliver one product to many locations, serving different services of the same product in different locations.
- **Multi-Product Routes** deliver many products to one location, serving different services in the same location with a consolidated shipment composed of more than one

product.

We do not consider routes delivering multiple products to multiple locations. Although these may be the most flexible routes, reducing the number of drops while minimizing travelled distance, they usually are undesired for adding complexity to the loading operation. Nonetheless, the approach can be adapted to consider this type of routes. Figure 5.3 depicts an example with the allowed routes.

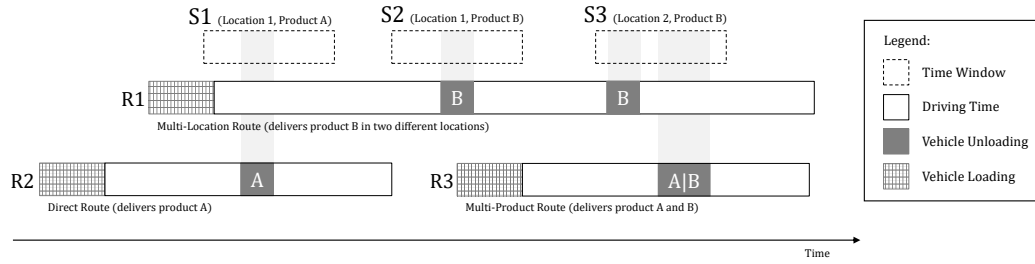


Figure 5.3 – Route type and delivery possibilities example showing three time windows being used to receive the deliveries of three routes. Route R1 visits two locations and delivers product B to both locations. The route uses the time windows of services S2 and S3 which are dedicated to product B. Route R2 is a direct route delivering product A in the time window of service S1, dedicated to product A. Route R3 is a multi-product route where a consolidated delivery is performed. The route delivers products A and B in the time window of service S3. Despite being dedicated to product B, the time window of service S3 allows deliveries of other products (in this case product A), as long as at least some products of type B are delivered.

To create the set of routes, the generator uses mainly two different approaches. The first approach generates routes based on the cheapest insertion heuristic. For each seed location, subsets with 0 to $n_{Nearest}$ nearest locations are defined. For each subset, we open a route with the seed location and insert the other locations in the cheapest position. The selection of the next location to enter each route is performed randomly. The second approach uses demand scenarios as an input to create optimized delivery schedules by means of an Adaptive Large Neighbourhood Search (ALNS) procedure. By doing so, we build a diverse set of routes that is able to cope with the demand patterns embedded in the scenarios. Note that at this point, endogenous time windows have not been defined yet. Therefore, we use the exogenous time windows to ensure the feasibility of the delivery schedules. Since we are dealing with a set-partitioning formulation where path variables z_r^ω represent entire routes, it is also possible to use historical data provided by companies to correctly describe their routing operations and derive less disruptive solutions.

The last step of the route generation process is to create copies of the same route along the exogenous time windows. This procedure clones the routes and modifies the preparation time, the departure time, and the arrival times at each location. Algorithm 1 provides the pseudo-code describing the route generation phase.

Algorithm 1 Route Generation

```

1: procedure ROUTEGENERATION( $\Omega, \mathcal{L}$ )
2:    $\Theta \leftarrow \emptyset$ 
3:   foreach  $l \in \mathcal{L}$  do
4:     foreach  $p \in P$  do
5:        $\Theta \leftarrow$  Build routes by nearest neighbour heuristic using  $l$  as seed and exogenous time windows
6:        $\Theta \leftarrow$  Build a direct route serving all products (for multiple product deliveries)
7:     foreach  $\omega \in \Omega$  do
8:        $\Theta \leftarrow$  Build routes for scenario  $\omega$  using ALNS considering exogenous time windows
9:      $\Theta \leftarrow$  Add external routes from historical data
10:     $\Theta \leftarrow$  Create copies of each route along exogenous time windows
11:     $\Theta \leftarrow$  Check feasibility and exclude infeasible routes
12:   return  $\Theta$ 

```

The generated routes need to satisfy the compatibility between the vehicle, service, and each endogenous time window. This means that several combinations between temperature, vehicle capacity, and arrival times need to be generated for each sequence of retail sites to be visited and added to the considered set of routes. Analysing this set in conjunction with the exogenous time windows of each service results in a set of compatibility matrix of routes and endogenous service time windows. Parameter $e_{r,sw}$ (used in our formulation) indicates whether route r is able to serve service s in an endogenous time window starting in slot w .

Figure 5.4 represents an example of the rational behind the feasibility checker of the route generator (line 11, Algorithm 1).

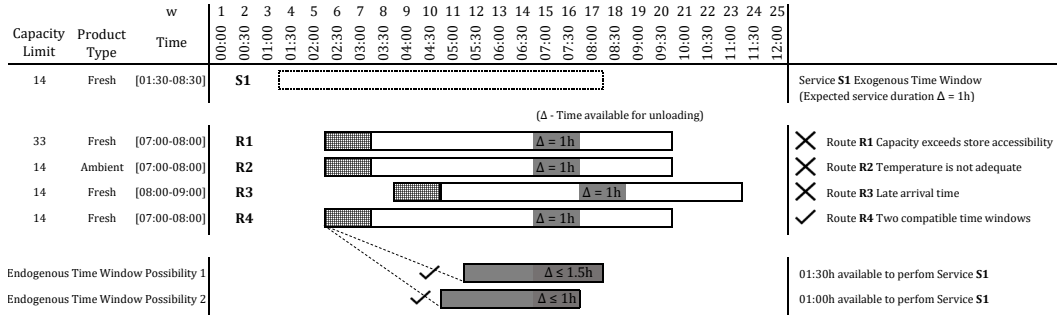


Figure 5.4 – In the example there is one route, R4, compatible with the exogenous time window for fresh products. Two endogenous time windows can be defined to allow route R4 to be performed.

Service S1 has an exogenous time window spanning through the interval [01:30-08:30]. In the example, four routes are analysed. The first three routes are rejected due to capacity, temperature, and late arrival time constraints, respectively. The fourth route allows for two compatible endogenous time windows, one starting at 05:00 (slot $w = 11$) and another starting at 05:30 (slot $w = 12$). These are the only possible endogenous time windows for serving service S1 using route R4 (considering 30-minute granularity). If the endogenous time window possibility 1 would advance in time, it would go out of the exogenous time window of the service. If the endogenous time window possibility 2 would be sooner, there

would not be enough time to serve the service due to its expected duration.

The pre-processing rational to define service, route, and time window compatibility eliminates infeasible routes taking into account the rules that should be followed by each delivery schedule. Therefore, our approach can be adapted to several operational models with distinct constraints, as long as the route set is entirely feasible.

5.3.2 Initial Solution Construction

The initial solution is constructed aiming at defining delivery schedules for each scenario, by solving a series of sub-problems sequentially, using the set of routes Θ to satisfy the demand of the set of scenarios Ω . This procedure is based on the ideas of the Relax-and-Fix heuristic proposed by Pochet and Wolsey [2006]. In each sub-problem, our routing variables z_r^ω are partitioned into three groups: The first group, \mathcal{ST}^{fix} , contains routing variables with fixed values, the second, \mathcal{ST}^{opt} , contains routing to be optimized, and the third, \mathcal{ST}^{rel} , contains routing variables for which the integrality constraints are relaxed. The procedure starts by choosing a scenario $\omega \in \Omega$ to be considered which is the one whose routing variables will be optimized. All the combinations between route and scenario (r, ω) are added to \mathcal{ST}^{opt} . To be able to change the variable bounds of the initial formulation, we define a sub-problem TWAVRPP-INIT as follows:

(TWAVRPP – INIT) : minimize objective function (5.1) subject to constraints (5.2) – (5.11) and the additional constraints to define variable bounds:

$$x_{rs}^\omega \geq 0, \quad y_{sw} \in \{0, 1\}, \quad v_k^\omega \in \mathbb{N}_0, \quad (5.13)$$

$$z_r^\omega \in \{0, 1\} \quad \forall (r, \omega) \in \mathcal{ST}^{opt}, \quad (5.14)$$

$$z_r^\omega \in [0, 1] \quad \forall (r, \omega) \in \mathcal{ST}^{rel}, \quad (5.15)$$

After being optimized, the variables in the set \mathcal{ST}^{opt} are transferred to the set \mathcal{ST}^{fix} . Additionally, to fix their values \bar{z}_r^ω , the following constraints are added to the formulation:

$$z_r^\omega = \bar{z}_r^\omega \quad \forall (r, \omega) \in \mathcal{ST}^{fix}. \quad (5.16)$$

The whole process is repeated until all the routing variables are fixed. Therefore, variables will pass from the set \mathcal{ST}^{opt} , then to the set \mathcal{ST}^{rel} , and finally they end up in the set \mathcal{ST}^{fix} . In each iteration, we define the routing schedule for a new scenario ω , maintaining the values of the previously optimized scenarios fixed. Algorithm 2 shows the procedure of the initial solution construction phase.

Algorithm 2 Initial Solution Construction

```

1: procedure INITIALSOLUTIONCONSTRUCTION( $\Omega, \Theta$ )
2:    $\omega \leftarrow 0$ ;  $fractional\_initial\_solution \leftarrow \emptyset$ ;  $initial\_solution \leftarrow \emptyset$ 
3:    $\mathcal{ST}^{rel} \leftarrow$  Relax variable integrality on all routing variables  $z_r^\omega$ 
4:   while  $\omega < |\Omega|$  do
5:      $\hat{\omega} \leftarrow$  Define the current scenario as  $\omega$ 
6:      $\mathcal{ST}^{opt} \leftarrow$  Enforce integrality on routing variables  $z_r^{\hat{\omega}}$  of current scenario  $\hat{\omega}$ 
7:      $fractional\_initial\_solution \leftarrow$  Solve TWAVRP-INIT considering  $\Omega, \Theta, \mathcal{ST}^{opt}, \mathcal{ST}^{rel}$ , and  $\mathcal{ST}^{fix}$ 
8:      $\mathcal{ST}^{fix} \leftarrow$  Update the set of fixed variables and fix their values in the model
9:      $initial\_solution \leftarrow$  Fix the values of the integer routing variables in  $\mathcal{ST}^{fix}$ 
10:     $\omega \leftarrow \omega + 1$ 
11:  return  $initial\_solution$ 

```

5.3.3 Improvement Matheuristic

To improve the solution of the TWAVRPP, we propose an improvement matheuristic that is inspired by the fix-and-optimize (F&O) version of [Helber and Sahling \[2010\]](#). Since the number of integer variables determines the major portion of the computational burden, we are interested in dealing with these variables by iteratively solving a series of sub-problems derived from the TWAVRPP formulation. The resulting sub-problems may be solved to optimality by a Mixed-Integer Program (MIP) solver. Given that the number of integer variables in each sub-problem is reduced, the time spent in each iteration is also smaller.

5.3.3.1 Definition of Variable Subsets and Subproblems

In order to define a sub-problem TWAVRPP-SUB, we need to select combinations of services s and scenarios ω , $(s, \omega) \in \mathcal{ST}$. Each combination $(s, \omega) \in \mathcal{ST}^{opt} \subseteq \mathcal{ST}$ indicates that all the variables related to the pair (s, ω) are to be optimized in the sub-problem. The remaining variables, related to the pairs $(s, \omega) \in \mathcal{ST}^{fix} = \mathcal{ST} \setminus \mathcal{ST}^{opt}$, are to be fixed with their current values. For instance, if the pair $(1, 1)$ is selected, it means that the time window variables y_{1w} and the routing variables z_r^1 are freed and re-optimized in the sub-problem. Note that modern solvers are able to detect and cut routes that are not useful to serve the services that are able to be optimized in the set \mathcal{ST}^{opt} . A sub-problem TWAVRPP-SUB can be stated as follows:

(TWAVRPP – SUB) : minimize objective function (5.1) subject to constraints (5.2) – (5.12) and the additional constraints:

$$y_{sw} = \bar{y}_{sw} \quad \forall s | (s, \omega) \in \mathcal{ST}^{fix}, w \in W \quad (5.17)$$

$$z_r^\omega = \bar{z}_r^\omega \quad \forall \omega | (s, \omega) \in \mathcal{ST}^{fix}, r \in \Theta \quad (5.18)$$

where \bar{y}_{sw} and \bar{z}_r^ω are solution values that come from the previous solution.

5.3.3.2 Decomposition Strategies

In order to define the subset \mathcal{ST}^{opt} various decomposition strategies can be used, depending on the entities that are selected. Note that the structure of the TWAVRPP offers an additional difficulty compared to non-integrated problems. In fact, in this case, decomposition strategies are not very effective if they do not consider the tactical and operational components at the same time. If the routing variables are not freed, it is unlikely to find improvements solely by changing time window variables because these decisions are very related to each other. Hence, first and second stage decisions need to be jointly addressed. We follow two different decomposition strategies:

- **Scenario-Oriented Decomposition (SOD):** each sub-problem considers variables of a single scenario ω . All the routing and time window variables are freed for a single scenario.
- **Location-Oriented Decomposition (LOD):** each sub-problem considers all the variables related to a single location v . This means that all the service time window variables related to a certain location and all the routing variables visiting that location to serve any service are freed for all scenarios.

5.3.3.3 Iterative Algorithm

The basic idea of our matheuristic approach is to iteratively solve a series of sub-problems based on the aforementioned decomposition strategies. Since our objective is to define a set of service time windows and the delivery schedules to be executed under those time windows, we needed to define a strategy that is able to capture interactions between these entities to explore sub-problems where the potential for finding improvements is high. For instance, if the time windows of every service are fixed, the problem can be decomposed in several Vehicle Routing Problems (VRPs), thus it makes no sense to consider more than one scenario at a time in the optimization process because no further improvement would be found.

The SOD focuses on a single scenario and will most likely work on routing variables, making small adjustments in time window variables. The routes used in fixed scenarios will trap most of the service time windows. Therefore, this type of decomposition is to be used whenever we assume that for the current set of time windows, large improvements can be obtained in the routing variables. On the other hand, LOD works on the time window and routing variables of every scenario, focusing on a certain location (or a set of locations). Time windows can be changed so that new routes become possible for improving the objective function.

Given the characteristics of each type of decomposition, we conclude that the SOD is most useful in the beginning of the algorithm, after a certain number of iterations using other decompositions, and in the end of the algorithm to ensure the best routing schedule for considered set of time windows. LOD should be used in the majority of the running time as it is not so easy to reach local optima by solving the sub-problems based on this decomposition. Algorithm 3 shows the general steps of our F&O approach.

Algorithm 3 Improvement Matheuristic

```

1: procedure FAO(initial_solution, noimpmax, tlimit)
2:   stop  $\leftarrow$  false, noimp  $\leftarrow$  0, solutionbest  $\leftarrow$  0
3:   solutionbest  $\leftarrow$  SOD(initial_solution)
4:   while not stop do
5:     r  $\leftarrow$  select random route from  $\Theta^{ml}$ 
6:      $\mathcal{ST}^{opt}$   $\leftarrow$  select set of services and scenarios based on r
7:     solution  $\leftarrow$  LOD(solutionbest,  $\mathcal{ST}^{opt}$ )
8:     if noimp  $>$   $\frac{noimp_{max}}{2}$  then
9:       solution  $\leftarrow$  SOD(solution)
10:    if solution  $<$  solutionbest then
11:      solutionbest  $\leftarrow$  solution, noimp  $\leftarrow$  0
12:    else
13:      noimp  $\leftarrow$  noimp + 1
14:    if noimp  $>$  noimpmax or time  $>$  tlimit then
15:      stop  $\leftarrow$  true
16:    solutionbest  $\leftarrow$  SOD(solutionbest)
17:  return solutionbest

```

We start by setting the best solution after applying SOD to the initial solution (line 3 of Algorithm 3). Afterwards, the algorithm iteratively focuses on the decisions related to subsets of locations. In order to select a subset of locations, the algorithm randomly selects a route (line 5, Algorithm 3) from the set of multi-location routes $\Theta^{ml} \subseteq \Theta$, which comprises all routes visiting more than one location. Since they are comprised in the same route, these locations are associated for proving good savings in the routing cost and should be considered in the same sub-problem. Therefore, after selecting a route $r \in \Theta^{ml}$, the algorithm selects the subset \mathcal{ST}^{opt} (line 6 of Algorithm 3), containing all the combinations (s, ω) between scenarios and services related to the locations visited by the route r . The LOD procedure solves a sub-problem (line 7 of Algorithm 3) where all the time window and routing variables of the services contained in \mathcal{ST}^{opt} are freed. Improvements can be found both in the routing cost and in the cost of vehicle fleet since sub-problems based on the LOD are able to simultaneously select new routes and slide the beginning of the service time windows. Whenever the number of non-improvements *noimp* reaches half of the maximum number of non-improvements *noimp_{max}*, we apply SOD (line 9, Algorithm 3). Solutions are accepted if their objective value is better than the one of the current best solution *solution_{best}* (line 10 of Algorithm 3). The algorithm stops after *noimp_{max}* iterations without improving or when the time limit *tlimit* is reached. One last SOD procedure is used to optimize the delivery schedules for the incumbent set of service time windows.

5.4. Computational Experiments

This section presents the numerical results of our matheuristic solution approach and provides managerial insights regarding the TWAVRPP. First, we quantify the additional savings obtained by considering the stochastic optimization problem relatively to the case where a set of time windows is computed *a priori*, considering one scenario with average demands. Afterwards, we test three different operational models providing information

regarding the potential savings achieved in each one, compared to a baseline solution (provided by a large European food retailer). Furthermore, we analyse the structure of each solution by providing various routing indicators, such as the average number of routes performed, the average number of visited locations per route, the average number of product segments delivered per route, and time window dispersion.

All numerical tests were conducted on Intel®Xeon®E5-2650 processing units at 2.00GHz and 16GB memory. A single thread was used. The tests were implemented in C++ (Visual Studio 2015) using CPLEX 12.6 solver.

5.4.1 Instance Set

The data to build our instance set was provided by a large food retailer operating in Europe. The data set includes information related to the month of January of 2017 indicating the number of pallets of each type of product that was delivered to each location, as well as the set of all the routes that was used by the company during that scenario. Since the food retailer owns two independent warehouses, that serve two different regions, we create a set of instances related to context **N** and another one related to context **S**. For each context, the data are divided in four instances of one week, totalling eight instances with real-world data. The locations of each retail site were maintained and the distance and time matrices were computed using Google Maps API. The instances consider three product segments which have different exogenous time windows in each context. All the endogenous time windows have an amplitude of 60 minutes. Retail sites have different accessibility constraints and can only be visited by a compatible vehicle. Demand scenarios occurrence is considered to be equal. The description of our instances is summarized in Table 5.2.

Table 5.2 – Instance data set description

Demand	Number of pallets delivered of each service in January 2017
Locations	Real-world coordinates
Travel distances and times	Google Maps API
Product segments	{Fresh, Frozen, Ambient}
Exogenous time windows for Fresh products	$[1200, 480]^N$, $[1200, 480]^S$
Exogenous time windows for Frozen products	$[360, 1440]^N$, $[360, 1440]^S$
Exogenous time windows for Ambient products	$[0, 1440]^N$, $[0, 1440]^S$
Endogenous time window width	60 minutes for every product segment
Vehicle and retail site capacities	{16, 20, 22, 33, 24, 26}
Scenario distribution	Equal probabilities of occurrence
Fixed cost per vehicle per scenario	191.32
Variable cost per Km	0.432

Contexts **N** and **S** are different in terms of retail site coordinates, demand quantities, and average number of visits per retailer. In Table 5.3, we provide a brief description of a set of relevant indicators for each context. The context **S** considers a set of retail sites that are further away from the depot but it has a lower demand average per retail site. One of the most important indicators is the lower bound on the average number of necessary visits to each retail site. The lower bound on the necessary number of visits to satisfy a given retail site l in a given scenario ω is given by $\sum_{s \in S} d_s^\omega / q_l$, where q_l is the capacity of the largest

vehicle that can access retail site l . Note that in the worst case, 6 visits (split deliveries) may be necessary to satisfy the total demand of a certain retail site comprised in context **S**.

Table 5.3 – Data context description

Context	Travel distance from depot (km)			Travel time from depot (min)			Demand (Pallets)		Lower bound on necessary visits	
	Min	Max	Avg	Min	Max	Avg	Avg	AvgKmPerUnit	Avg	Max
N	2.3	265.9	67.1	6.1	156.3	45.8	43.1	3.1	1.9	5.0
S	0.5	320.3	95.6	2.2	176.0	62.4	41.5	2.8	1.9	6.0

5.4.2 Stochastic Optimization versus Average Demand Scenario

One of the main factors driving the relevance of our approach lies in the fact that considering a set of demand scenarios to define time windows is likely to outperform an approach considering only one scenario with average demands. For the single product case, [Spliet and Gabor \[2015\]](#) state that adopting time windows obtained by average demand scenarios results in 1.85% cost increase. We performed similar tests on our multiple product case. The solutions for the Stochastic Optimization (SO) approach are obtained by applying our solution approach to the instances comprising various demand scenarios. The solutions for the Average Demand Scenario (AD) approach are obtained in two steps: (1) an instance with average demands is created and solved (AVG approach), defining a set of time windows; (2) the instances comprising various demand scenarios are solved with the set of time windows defined in the first step (i.e., there is no time window assignment in the second step). For this experiment, we generated 8 instances (4 for each context) by randomly selecting seven demand scenarios amongst the available data set. The results are presented in Table 5.4.

Table 5.4 – The AD approach was outperformed by the SO in every instance. The total average improvement for considering the SO approach is 5.3%.

Context	Instance	AVG	AD	SO	$\frac{AD-SO}{AD}$
N	1_N	13 824	104 395	98 492	5.7%
	2_N	13 135	103 376	98 171	5.0%
	3_N	13 613	107 868	99 037	8.2%
	4_N	13 550	104 548	98 619	5.7%
N Avg					6.1%
S	1_S	17 884	136 973	131 380	4.1%
	2_S	17 849	138 244	133 411	3.5%
	3_S	17 720	136 539	130 534	4.4%
	4_S	18 270	140 847	132 303	6.1%
S Avg					4.5%
Total Avg					5.3%

Legend:

AVG - Average Demand, free windows | AD - 7 Scenarios, AVG windows | SO - 7 Scenarios, free windows

The results show that the SO approach always outperforms the AD approach. In context **N** the SO approach yields an average improvement of 6.1%. In context **S**, the average improvement achieved by the SO approach is 4.5%. The instances of context **S** are more difficult to solve. We suspect that the smaller improvements found for this context are due to a larger deviation from the optimal solution. Additionally, the AVG column shows that obtaining a smaller cost in the first step of the AD approach (solving the average scenario) does not necessarily mean a smaller cost in the second step (solving various VRPTW with fixed time windows).

5.4.3 Operational Models and Change Levels

In real-world applications, introducing changes in the planning process is generally very difficult due to a large change resistance that companies demonstrate. Very disruptive solutions are usually impracticable for real-world implementation. Therefore, we perform a sensitivity analysis on a few time window conditions. We test our algorithm considering three different operational models: baseline, increase window, and product flexibility (model parametrization presented in Table 5.5).

- (1) In the **baseline** operational model, the characteristics of each time window are maintained as in the data set provided by the company. Therefore, in this operational model there is a time window for each service with a 60-minute amplitude and multi-product routes need to deliver at least \underline{q} units of the main product.
- (2) In the **increase window** operational model, the amplitude of each time window is increased by 60 minutes (30 on each side) while maintaining the obligation of delivering at least \underline{q} units of the main product in multi-product routes.
- (3) In the **product flexibility** operational model, it is not necessary to deliver at least \underline{q} units of the main product in multi-product routes. Thus, time windows are only product dependent when a route visits more than one location. This operational model considers the original amplitude of 60 minutes.

Table 5.5 – Parameters to define each operational model in the mathematical formulation

Operational Model	Amplitude Increase (min)	\underline{q}
Baseline	0	$\frac{1}{2} \cdot \min\{q_k k \in K\}^*$
Increase Window	60	$\frac{1}{2} \cdot \min\{q_k k \in K\}^*$
Product Flexibility	0	0

*Percentage of the smallest vehicle in the fleet

(i.e., If the smallest vehicle has a capacity of 16 pallets, at least 8 pallets of the main product need to be delivered)

Furthermore, for each operational model we test four retail site time window change levels: 0%, 25%, 50%, and 100%. The change level indicates the percentage of retail sites in which at least one of the service time windows can be moved back or forth. This means that the beginning of the time window is altered in relation to its original position in the delivery plan.

5.4.3.1 Sensitivity Analysis

The results of our sensitivity analysis are presented in Table 5.6. The first three columns indicate the context, change level, and operational model considered, respectively. Since our algorithm includes a non-deterministic selection of entities to define sub-problems, each instance (considering one week) is run five times. Each row represents the average values obtained for the four weeks of data, totalling 20 runs. Column Avg Runtime shows the average runtime whereas column Avg Objective indicates the average values of the objective function. To describe the deviation between runs, we show the average standard deviation on column Avg StdDev. Column Avg Gap indicates the average gap relatively to the best-known solution of each instance (obtained during our tests). The last four columns represent different views of the savings that are obtained. Column Avg Savings (Change) indicates the savings relatively to the baseline scenario of each change level. Column Avg Savings (Global) indicates the savings relatively to the baseline operational model of each context with a change level of 0%. Columns Avg FC Savings (Change) and Avg FC Savings (Global) indicate the savings coming from the fixed cost term in the objective function, which represents the fleet cost. For instance, in the second row of Table 5.6, 28% of savings are achieved and 68% of these savings come from the fixed cost term.

Table 5.6 – Computational results for every combination of operational model and change level. The best operational model in terms of savings is shown in bold.

Context	Change Level	Operational Model	Avg Runtime (s)	Avg Objective	Avg StdDev	Avg GAP*	Avg Savings (Change)	Avg FC Savings (Change)**	Avg Savings (Global)	Avg FC Savings (Global)**
N	0%	Original	2 491	247 680	681	0.00%	-	-	-	-
		IncreaseWindow	2 872	178 086	1 357	0.01%	28%	68%	28%	68%
		ProductFlexibility	2 990	142 051	774	0.00%	43%	59%	43%	59%
	25%	Original	13 241	151 385	2 125	1.08%	-	-	39%	75%
		IncreaseWindow	13 312	133 795	1 330	1.31%	11%	56%	46%	72%
		ProductFlexibility	4 157	118 072	1 944	1.02%	21%	43%	52%	67%
	50%	Original	9 929	123 238	1 193	0.71%	-	-	50%	76%
		IncreaseWindow	23 395	111 983	1 840	1.78%	8%	47%	55%	73%
		ProductFlexibility	4 317	111 590	1 210	0.15%	9%	-23%	55%	67%
	100%	Original	9 406	115 398	1 289	0.58%	-	-	53%	75%
		IncreaseWindow	21 448	107 320	2 703	3.53%	6%	62%	57%	72%
		ProductFlexibility	4 138	111 387	1 392	0.15%	3%	-206%	55%	67%
S	0%	Original	11 195	234 014	47	0.00%	-	-	-	-
		IncreaseWindow	14 381	195 394	2 643	0.61%	17%	61%	17%	61%
		ProductFlexibility	13 168	163 993	1 504	0.00%	30%	53%	30%	53%
	25%	Original	12 605	182 819	2 458	0.27%	-	-	22%	69%
		IncreaseWindow	28 459	157 550	1 283	1.02%	14%	66%	33%	67%
		ProductFlexibility	13 317	145 251	5 345	3.46%	20%	64%	38%	69%
	50%	Original	27 091	146 180	2 373	1.16%	-	-	38%	75%
		IncreaseWindow	39 096	137 896	827	0.50%	5%	44%	41%	71%
		ProductFlexibility	12 658	138 697	4 600	3.31%	4%	25%	41%	69%
	100%	Original	33 838	137 558	1 207	0.64%	-	-	41%	73%
		IncreaseWindow	50 271	132 461	875	0.42%	3%	32%	43%	71%
		ProductFlexibility	16 876	135 591	2 981	1.25%	1%	-500%	42%	67%

* Relatively to the best-known solution (each instance is run 5 times)

** Percentage of savings obtained in the fixed cost

(Change) Savings related to the best-known solution for the baseline operational model inside the change level group

(Global) Savings related to the best-known solution for the baseline operational model and a change level of 0% inside the context

The results show that it is possible to obtain large savings in both contexts, being slightly larger savings in context **N**. However, there are plenty of combinations between operational model and change level to achieve satisfactory savings across both contexts. Nonetheless, it is important to note that changing time windows requires some implementation effort. Table 5.6 is a useful instrument to trade-off the implementation efforts versus the potential savings that can be achieved. In some cases, it may be easier to maintain the baseline operational model and move some time windows back and forth whereas in other cases it may be preferable to change the operational model without moving time windows.

Regarding the operational models that are tested, we conclude that widening time windows amplitude or increasing the product flexibility of multi-product routes are very promising alternatives. Even when the start of the time windows is maintained, the increase window operational model achieves savings of 28% in context **N** and 17% in context **S**. The product flexibility operational model achieves savings of 43% in **N** and 30% in context **S**. Thus, increasing the product flexibility of multi-product routes may be the best decision when the position of time windows cannot be changed in any retail site.

With respect to the change level parameter, large savings can be obtained even under the baseline operational model. If the starting time of the windows of 25% of the retail sites is allowed to be changed, the context **N** achieves average savings of 39% and the context **S** 22%. For every operational model, no significant differences occur when the number of retail sites changed is larger than 50%. Therefore, there is no need to change more than 50% of the retail sites to achieve the largest savings potential. Additionally, savings seem to converge across every operational model when the change level parameter increases above 50%. However, the increase window and product flexibility operational models may obtain better results in the long run as they offer increased flexibility to cope with more disruptive demand scenarios. Figure 5.5 provides a summary of the savings that can be achieved for each operational model and change level.

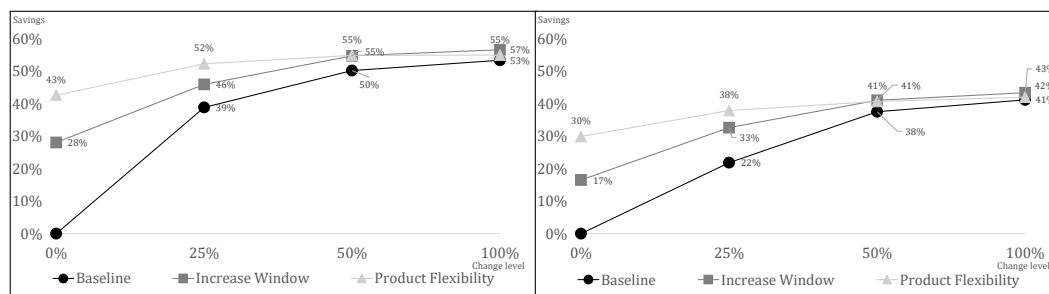


Figure 5.5 – Average **savings** (Y-axis) achieved in context **N** (left) and context **S** (right) for each change level (X-axis) and operational model (compared to the baseline operational model with 0% changes)

Finally, we need to emphasize the portion of the savings that comes from the fixed cost term of the objective function. The last column of Table 5.6 shows that a large portion of the savings is due to the minimization of the number of vehicles needed in each scenario. For a visual representation of the role of our novel objective function, Figure 5.6 shows an example of two different profiles describing the number of necessary vehicles in each time

slot. Note that, vehicle requirement profiles can be very different depending on the delivery schedule. In this case, although the problem is constrained by an exogenous time window for fresh products, it is possible to smooth the peak that existed nearby. In terms of fixed cost, the savings for a scenario with the profiles shown in Figure 5.6 are given by $\Delta V \cdot fc$, where ΔV is the reduction on the number of vehicles and fc is the fixed cost associated with each vehicle.

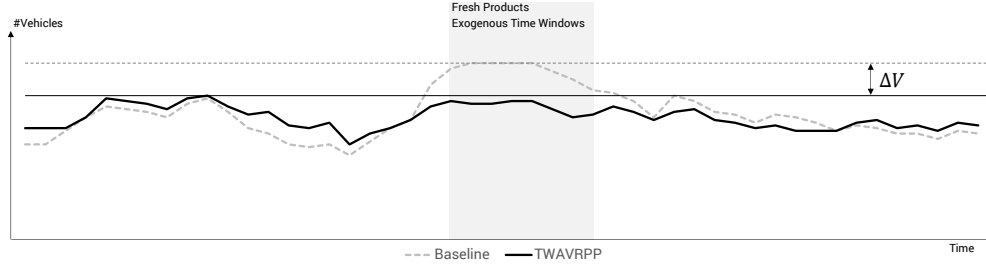


Figure 5.6 – The figure shows the Baseline and the profile obtained after optimizing time windows with our solution approach (TWAVRPP).

The effects caused by imposing exogenous time windows are generally translated into a large number of overlapping time windows in some time slots. The worst case would require one vehicle to be arriving at each store in the same point in time. In the example, the baseline solution shows a peak of vehicle needs during the exogenous time window for fresh products. The idea of the new objective function is to smooth the peak while minimizing the total travelled distance, complying with business constraints. Note that in this case, the number of necessary vehicles decreases by ΔV . This issue has been addressed in some works related to different problems (as in [Bhusiri et al. \[2014\]](#)), yet the time window assignment literature ignores the fixed cost. For these reasons, we consider that this term should not be ignored when approaching the TWAVRP.

5.4.3.2 Solution Structure Analysis

In order to further explore the differences between the structure of the solutions obtained in each combination of operational model and change level, we analyse six dimensions that are of utmost importance for deriving conclusions from problems with a strong routing component: **(1)** number of routes performed, **(2)** type of routes performed, **(3)** number of drops per route, **(4)** vehicle occupation, **(5)** number of product segments delivered per route, **(6)** number of stores with overlapping time window per time slot. We interpret the interaction between each dimension with the savings obtained for each combination of operational model and change level, providing a set of figures with operational indicators

To analyse dimensions **(1)** and **(2)**, Figure 5.7 depicts the average number and type of routes performed for every combination of operational model and change level (for both contexts).

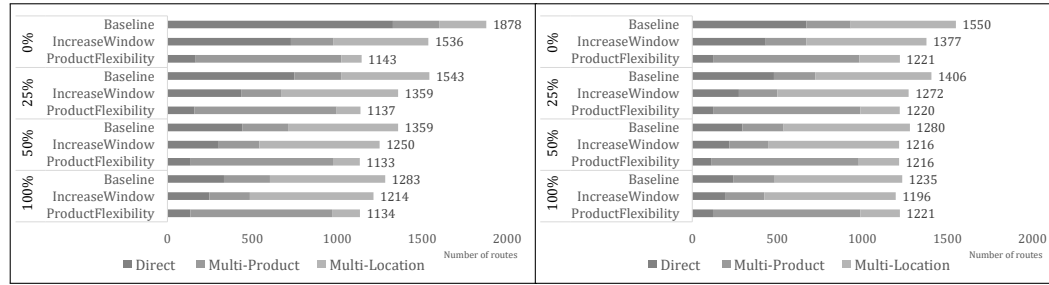


Figure 5.7 – Average **number of routes** (Y-axis) performed in context **N** (left) and context **S** (right) for each change level (X-axis) and operational model

As expected, one of the main reasons for obtaining the substantial savings described in Figure 5.5 is due to a large reduction in the number of routes. Figure 5.7 allows us to draw two different conclusions that apply to both contexts, **N** and **S**. First, by providing additional flexibility to change the beginning of a larger number of time windows, new routes become available in every operational model. Therefore, for larger change levels, a smaller number of routes is generally achieved. Second, for the same change level, increasing time window amplitude or adding product flexibility allows for a reduction in the number of routes, relatively to the baseline operational model. However, in each operational model, we find different reasons to achieve such reduction. When the amplitude of the time windows is increased, the savings come from new multi-location routes that are performed, which are more efficient than direct routes as they increase the number of drops per route (routes can visit more locations). When product flexibility is given, the savings come from new multi-product routes that are performed, which are more efficient than direct routes because they reduce the number of drops to each location (loads are consolidated). These findings are in-line with the research on consolidation presented by [Mesa-Arango and Ukkusuri \[2013\]](#).

Dimensions (3) and (4) are analysed with Figure 5.8, which provides information on the number of drops per route, as well as the average vehicle occupation achieved for each combination of operational model and change level.

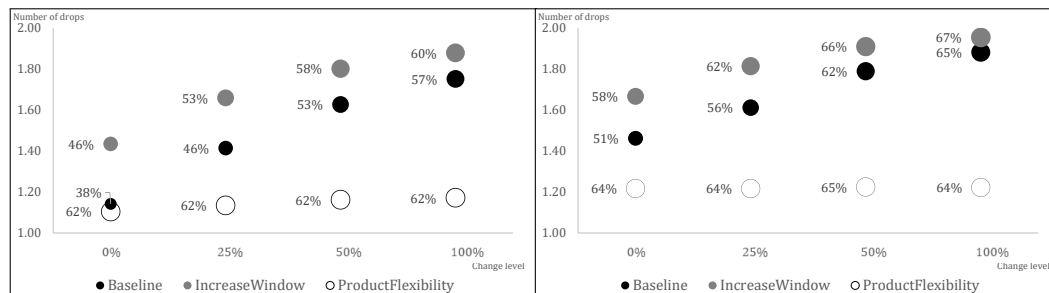


Figure 5.8 – Average **number of drops** (Y-axis) performed in context **N** (left) and context **S** (right) for each change level (X-axis) and operational model (bubble size and label represents average vehicle occupation)

The number of drops per route is intimately connected with the type of routes that is per-

formed. In Figure 5.8, for each operational model and change level, we show the average number drops per route as well as the average vehicle occupation. Savings in the baseline and increase window operational models come from visiting more locations within the same route. Note that as the number of locations increases, the vehicle occupation also increases (even by performing routes delivering only one product segment). In the product flexibility operational model, although the number of drops does not vary substantially, the new routes allow for larger vehicle occupations. Since more product segments are consolidated, the number of drops is reduced. In fact, in most cases, one drop is enough to deliver the total demand of a certain retail site (regardless of the product segment).

Regarding dimension (5), the number of products per route, we present Figure 5.9.

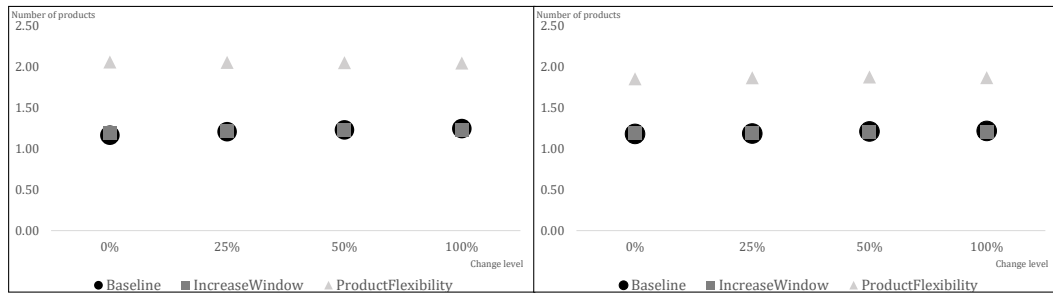


Figure 5.9 – Average **number of product segments** (Y-axis) delivered in each route in context **N** (left) and context **S** (right) for each change level (X-axis) and operational model

Figure 5.9 allows us to confirm that in both contexts, the number of product segments delivered per route is larger in the product flexibility operational model. It seems that delivering more product segments per route is the reason why the product flexibility operational model achieves larger savings, even showing the smallest number of drops per route. Comparing the baseline and increase window operational models, we show that the average number of product segments delivered per route is the same. Therefore, the savings obtained in the increase window operational model are mostly due to an increase in the number of drops per route (as concluded from Figure 5.8). The change level parameter does not influence the number of products per route.

Finally, Figure 5.10 analyses dimension (6) by trying to capture the dispersion of the time windows assigned in the tactical problem.

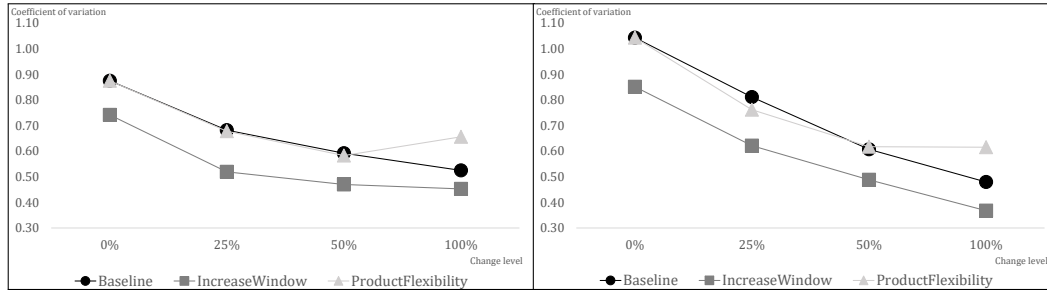


Figure 5.10 – Coefficient of variation of the number of active time windows per time slot (Y-axis) in context N (left) and context S (right) for each change level (X-axis) and operational model

Intuition tells us that having time window assignments with a large number of overlapping time windows may induce delivery schedules with larger fleet requirements for certain time slots. For instance, if the time window assignment defines a solution with all the time windows for fresh products overlapped, this could mean that a vehicle is needed in each store at the same time (number of vehicles \approx number of stores). For that reason, we try to measure the dispersion of the stores with some active time window in each time slot by computing the coefficient of variation of the number of stores with overlapping time windows. Thus, vehicle profiles with large peaks are penalised with large coefficients of variation. Figure 5.10 shows the referred coefficient of variation for every tested operational model and change level. In general terms, we can conclude that for smaller coefficients of variation (when time windows are sparser), larger gains are achieved. This impact gets clearer when analysing the change level parameter. In case more retail sites can be changed, smaller coefficients of variation are obtained, resulting in larger savings. Therefore, time window dispersion is a characteristic that is inherent to good solutions for the TWAVRP.

5.5. Conclusion

This research proposes and explores an extension for the TWAVRP. We model it as a two-stage stochastic optimization problem and propose a novel mathematical formulation where product dependent time windows need to be assigned, while minimizing the total cost incurred in travelled distance and fleet requirements. The latter term is usually ignored in TWAVRP literature, however it is shown that it can assume a huge importance both in terms of cost as well as in the type of routes that is used to satisfy demand. Indeed, it may be beneficial to perform shorter routes outside the time slot in which the fleet requirement is maximum, instead of performing better routes overlapping that time slot. Therefore, smoothing the peak of vehicle needs may be beneficial even if the routing cost needs to increase by a small amount to decrease the number of vehicles. For that reason, we propose the inclusion of a new term in the objective function to capture the maximum number of vehicles that is necessary in each scenario. Furthermore, we add realism to the TWAVRP by proposing a set of constraints to handle routes delivering more than one product. The developed formulation provides higher levels of flexibility as it is able to adapt to various

operational models.

In order to tackle large realistic instances, we develop a three-phase solution approach that relies on a pre-generated set of routes that can be adapted to a large number of routing models, including delivery time windows, heterogeneous fleet, multiple product split deliveries, among other features. After building an initial solution, we improve the incumbent solution by means of a matheuristic, which iteratively solves a series of sub-problems where retail sites with large potential are jointly optimized. The solution approach is effective and can solve real-world instances in reasonable time.

Our computational experiments are performed over a set of instances provided by a large food retailer, considering two different business contexts. First, we assess the value of considering a stochastic optimization approach compared to using an approach based on a single scenario with average demand. In our multiple product context, the results show an average cost decrease of 5.3% when various demand scenarios are used in the stochastic optimization approach. Second, to validate the flexibility of the model and provide valuable managerial insights on the TWAVRP, we extend our computational experiments to cover three different operational models crossed with a sensitivity analysis on the number of retail site time windows allowed to be changed. The results show that widening the time windows or relaxing the product dependency of each time window can result in large savings compared to the baseline operational model. Results indicate that it suffices to arrange time windows on 25% of the retail sites to obtain interesting savings in both contexts. Additionally, arranging time windows on more than 50% of the retail sites does not provide a substantial increase in the potential savings. This is a valuable finding considering that the change level can be seen as an indicator of the implementation effort. We summarize the findings coming from our analysis on five managerial and operational insights:

- i) In order to achieve substantial savings, conditions to reduce the number of routes need to be created (such as some time windows that need to be moved to allow for more efficient routes).
- ii) It is possible to achieve substantial savings with a small number of drops per route if the load is more consolidated (as more product segments are delivered per route).
- iii) It is possible to achieve substantial savings by delivering a small number of product segments per route if routes are longer (as more drops are performed per route).
- iv) In the sensitivity analysis, increasing product-segment consolidation showed to be superior to widening time window amplitude in terms of potential savings.
- v) Time window dispersion (i.e., a measure for the number of overlapping time windows) should be a goal while solving the tactical time window assignment.

Finally, we show the importance of including the fleet requirement cost in the objective function. The majority of the savings obtained is related to the fixed cost incurred on the number of vehicles to perform the routes of each scenario. However, as future work, and based on the experience achieved during the implementation of our solutions, we consider that it is very important to include other types of costs related to every stakeholder that is

involved along the supply chain. For instance, the cost of human resources, infrastructure development, and equipment have not been explicitly considered in this work but can have a large impact on the solution space. Changing time windows impacts every link of the supply chain. Therefore, to correctly assess the trade-offs associated with the TWAVRP and implement solutions in the real-world, it is necessary to further develop and add realism to the formulations that are proposed in the literature.

Acknowledgements

This research was partly supported by the PhD grant SFRH/BD/108251/2015, awarded by the Portuguese Foundation for Science and Technology (FCT), and by the project TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020, financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This support is gratefully acknowledged.

Bibliography

- Agatz, N., Campbell, a., Fleischmann, M., and Savelsbergh, M. (2011). Time slot management in attended home delivery. *Transportation Science*, 45(3):435–449.
- Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.
- Bhusiri, N., Qureshi, A. G., and Taniguchi, E. (2014). The trade-off between fixed vehicle costs and time-dependent arrival penalties in a routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 62(Supplement C):1 – 22.
- Braekers, K. and Kovacs, A. A. (2016). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological*, 94(Supplement C):355 – 377.
- Cordeau, J. F., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52(8):928–936.
- Deflorio, F., Gonzalez-Feliu, J., Perboli, G., and Tadei, R. (2012). The influence of time windows on the costs of urban freight distribution services in city logistics applications. *European Journal of Transport and Infrastructure Research*, 12(3):256–274.
- Groër, C., Golden, B., and Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643.

- Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2):247–256.
- Jabali, O., Leus, R., van Woensel, T., and de Kok, T. (2015). Self-imposed time windows in vehicle routing problems. *OR Spectrum*, 37(2):331–352.
- Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers and Operations Research*, 35(7):2307–2330.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2015). The Generalized Consistent Vehicle Routing Problem. *Transportation Science*, 49(4):796–816.
- Mesa-Arango, R. and Ukkusuri, S. V. (2013). Benefits of in-vehicle consolidation in less than truckload freight transportation operations. *Procedia - Social and Behavioral Sciences*, 80(Supplement C):576 – 590.
- Pochet, Y. and Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York.
- Salhi, S., Wassen, N., and Hajarati, M. (2013). The fleet size and mix vehicle routing problem with backhauls: Formulation and set partitioning-based heuristics. *Transportation Research Part E: Logistics and Transportation Review*, 56(Supplement C):22 – 35.
- Spliet, R., Dabia, S., and van Woensel, T. (2017). The time window assignment vehicle routing problem with time-dependent travel times. *Transportation Science*, Articles in Advance.
- Spliet, R. and Desaulniers, G. (2015). The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, 244(2):379–391.
- Spliet, R. and Gabor, A. F. (2015). The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, 49(4):721–731.
- Sternbeck, M. G. and Kuhn, H. (2014). An integrative approach to determine store delivery patterns in grocery retailing. *Transportation Research Part E: Logistics and Transportation Review*, 70(Supplement C):205 – 224.
- Subramanyam, A. and Gounaris, C. E. (2016). A branch-and-cut framework for the consistent traveling salesman problem. *European Journal of Operational Research*, 248(2):384–395.
- Toth, P. and Vigo, D., editors (2001). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Appendix 5.A Other Operational and Business Constraints

We provide extensions to the proposed formulation so as to include additional operational and business constraints related to the stakeholders involved in the time window assignment process.

5.A.0.1 Docks Availability

Given that vehicles need to be loaded at the warehouse, the number of available docks must be considered in the formulation. The number of docks available for loading vehicles may differ between time slots, as the docks may perform other activities, such as unloading goods from suppliers. In order to consider dock availability, we define two new parameters. l_{rw} is one if route r is loading during time window w , zero otherwise. $maxDocks_w$ refers to the number of available docks during time window w . Constraints (5.19) are added to ensure that the number of used docks in each time slot w is respected in each scenario ω .

$$\sum_{r \in \Theta} l_{rw} \cdot z_r^\omega \leq maxDocks_w \quad \forall w \in W, \omega \in \Omega \quad (5.19)$$

5.A.0.2 Driver Shifts

Logistics providers define certain time periods for switching drivers assigned to vehicles that are available 24 hours a day. This means that a smaller number of routes should be active during time periods allowed for drivers' changes. For that reason, we define H^{change} as the set of time slots w designated for changing drivers. Given the maximum number of available drivers during time slot w ($maxDrivers_w$), constraints (5.20) are added to limit the number of active routes in H^{change} .

$$\sum_{r \in \Theta} h_{rw} \cdot z_r^\omega \leq maxDrivers_w \quad \forall \omega \in \Omega, w \in H^{change} \quad (5.20)$$

5.A.0.3 Refrigerated Products Time Window

Food retailers pay special attention to fresh products. Since fresh products demand delicate reception operations, retail sites are interested in knowing the exact time at which they will be delivered. This means that even in multi-product deliveries, fresh products must always respect their time window. For that end, constraints (5.21) are added for the set of services considering fresh products S^{Fresh} .

$$e_{rsw} \cdot x_{rs}^\omega \leq y_{sw} \quad \forall \omega \in \Omega, r \in \Theta, w \in W, s \in S^{Fresh} \quad (5.21)$$

5.A.0.4 Consolidated Deliveries at Smaller Stores

Some retail sites receive small quantities of goods and demand a single consolidated delivery for certain sets of products. For each store, we must force some time windows to be overlapped in case they are within set S_l , which comprises the services of retail site l that should be served at the same time. Constraints (5.22) ensure that the time windows of services that are to be consolidated begin in the same time slot.

$$y_{s_1 w} = y_{s_2 w} \quad \forall s_1, s_2 \in S_l, l \in L, w \in W \quad (5.22)$$

5.A.0.5 Autonomous Deliveries

The infrastructure of some retail sites allows them to receive goods without the need of a receptionist. Depending on the conditions available on the reception chamber, some products can be received with the store closed (i.e., during the night). Although these deliveries provide a large flexibility while planning distribution, they require an assistant to help in the unloading operation. Routes with autonomous deliveries need to incorporate the cost of the assistant in their cost Θ^{Auton} where at least one autonomous delivery is performed and a parameter a_s indicating whether the route r is able to serve a certain service. Constraints (5.23) forbid autonomous deliveries to retail sites without the necessary conditions.

$$x_{rs}^{\omega} \leq a_s \quad \forall r \in \Theta^{Auton}, s \in S, \omega \in \Omega \quad (5.23)$$

Enforcing driver and time consistency in vehicle routing

Consistent vehicle routing problem with service level agreements: a case study in the pharmaceutical distribution sector

Pedro Campelo^{*†} · Fábio Neves-Moreira^{*} · Pedro Amorim^{*†} · Bernardo Almada-Lobo^{*†}

Submitted to *European Journal of Operational Research*, 2017 (under review)

Abstract In this paper, a mathematical model is developed to tackle a Consistent Vehicle Routing Problem which considers customers with multiple daily deliveries, different service level agreements, such as time windows, and time limits between ordering and delivering. In order to solve this problem, an instance size reduction algorithm and a mathematical programming based decomposition approach are developed. This solution approach is benchmarked against a commercial solver. Results indicate that it allows the usage of the mathematical model to solve instances of large size, enabling its application to real-life scenarios. A case study in a pharmaceutical distribution company is analyzed. Consistent routes are planned for several warehouses, comprising hundreds of orders. A simulation model evaluates the performance of the generated route plans. Significant improvements in terms of the total distance traveled and the total travel times, and ultimately in distribution costs, are obtained when compared to the company's current planning process.

Keywords Consistent Vehicle Routing Problem · Service Level Agreements · Matheuristics · Pharmaceutical Industry

^{*}INESC T0EC and Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, 4200-465, Porto, Portugal

[†]LTP Labs, Rua Doutor Júlio de Matos, 4200-355 Porto, Portugal

6.1. Introduction

Customer satisfaction is becoming one of the main drivers in competitive markets, and therefore new factors need to be considered during the planning and execution of delivery operations. In some markets, customers try to simultaneously decrease their overall stock levels and increase response time in case of a stock-out. This puts pressure on distributors, especially when faced with stiff competition, to increase the frequency of delivery and to be more concerned about good service levels rather than directly reducing operating costs. In this setting, the operational efficiency of the resources involved in this entire process is crucial, as the line separating profit from loss is a fine one. From an Operations Research point of view, these delivery operations often boil down to vehicle routing problems that arise whenever goods need to be distributed to a series of delivery locations by more than one vehicle. While this broad description fits a wide variety of applications, it is common for several specific business characteristics to hamper the decision making process regarding this problem, which in turn leads to the development of increasingly specific models that relate more closely to practical issues faced by companies.

In the distribution business, it is common for service level agreements to include a maximum time span between ordering and delivering or multiple deliveries throughout the day. While the existence of time windows is often the case in vehicle routing problems, the presence of multiple daily customer-specific shipping periods increases planning complexity as it is not possible to aggregate orders from the same customer on the same route if they are placed at different periods of the day. In this research we are concerned with a scenario in which customers have different service level agreements. In this setting, customers have different shipping periods, time-windows of varying length, and also different allowed time spans between ordering and delivering.

These distribution systems are, of course, challenged by uncertainty, which makes planning and execution even more difficult. Uncertainty arises from several sources such as demand, service times and customers' locations. To tackle it there are two main operational models that may be used: flexible routes that are routed dynamically based on the current orders that have to be delivered or consistent routes where the set of clients that are served per route has been defined a priori. The first option could theoretically lead to better sizing of the fleet and more optimized routes, but it ignores the fact that, upstream, the orders that are known a couple of hours beforehand, have to be prepared and consigned to a given delivery vehicle. On the other hand, the use of consistent routes increases driver familiarity with their own routes and territories, which improves driver efficiency. Moreover, the increased focus on customers makes driver recognition an even more important factor in route planning. Consistent routes also provide an easier, more stable way to manage delivery schedules and enable the possibility of delivering to customers at approximately the same time every day. The consistency of customer-route assignment may also be motivated by operational level constraints when customer' order placement deadlines are very close to vehicle departure. In these situations, it may be necessary to start picking operations and loading the vehicles before the full list of orders for the period is known. Finally, the assignment of each order to a vehicle is easier to manage and less error-prone with consistent routes.

Companies that use this type of consistent routes are fairly common in small package distribution industries faced with strong competition and highly demanding customers, with pharmaceutical and automobile spare parts distribution being the best-known examples. Other activities in which consistent routing is relevant include home care services, and also the transportation of children, the elderly or the disabled.

The tactical challenge of these logistics systems is in the design of the best set of consistent routes that result in the lowest operating costs. This problem is known in the literature as the Consistent Vehicle Routing Problem (conVRP). In recent years, the conVRP has been attracting the attention of the academic community as innovative and better solutions are being requested by companies. In problems such as this, customers are assigned to specific routes before any order has been placed during a tactical planning phase. On the operational level, every order placed by a customer will be served by the same vehicle at approximately the same time over multiple days. However, developing stable, consistent route plans that perform well in practice while still being efficient is a difficult task that many companies struggle with. Tools that assist decision-making in these conditions and take business characteristics into consideration are therefore very useful and desired by practitioners.

Several approaches have been proposed to address variants of the problem tackled in this paper. Nevertheless, our contribution is two-fold. Firstly, to the best of our knowledge, there is no literature that tackles the service level agreements often encountered in these logistics systems, in which not only time windows are settled, but also the time between ordering and receiving on the same day is defined as well as multiple shipping periods per day. This drastically limits the number of clients to be paired and the flexibility of the departure times of the vehicles. Secondly, although several formulations have been put forward to solve conVRPs, they were never used to address real-world instances. In this paper, we propose a solution method leveraged by a Fix-and-Optimize approach that fully utilizes the mathematical model developed and aims to be well-suited for application in real-life business situations. In order to test its validity and potential, the solution method is used with historical data to plan the consistent routes of a pharmaceutical distribution company with over 3,000 daily deliveries in an environment with both expected and unexpected customers with uncertain demand. The proposed plans are then simulated and their performance is compared to real-world plans in a case study.

The remainder of this paper is organized as follows. Section 6.2 introduces a literature review of the vehicle routing problem and some of its specific formulations with more relevance to our problem. Section 6.3 presents the formulation of the problem and the corresponding mathematical model, while Section 6.4 comprehensively describes the proposed solution approach. The case study is then introduced in Section 6.5 with a brief description of both the company's operations and all the tools developed and used during the analysis. In Section 6.6, the results of the proposed model in the real case study are shown and analyzed. Finally, conclusions and improvement opportunities are discussed in Section 6.7.

6.2. Literature Review

Since its introduction by [Dantzig and Ramser \[1959\]](#), the research community has been extensively studying different VRP variants and applications. As expected, the complexity of the challenges addressed also underwent a massive increase as the value of the optimization techniques attracted the interest of most competitive logistics operators. The computational power that is available nowadays enables the scientific community to further develop new mathematical models as well as the necessary methods to solve them. This remarkable and logical evolution is described by [Laporte \[2009\]](#).

Our main challenge is inspired in the real case of a pharmaceutical distributor which points various reasons for adopting consistent routes over a reasonably long period of time. In this case, the term consistent accounts both for drivers and delivery times of each route. Thus, our routing plans need to define a set of routes that are driver and time consistent. Since we consider these to be the main aspects of the problem we aim to solve, we position the problem in the conVRP literature. Besides the consistency aspects, the conVRP is intimately related with two other problems: the Period Vehicle Routing Problem (PVRP) and the VRP with Time Windows (VRPTW). In fact, in the conVRP, after defining a set of time windows, we still need to solve a PVRP which incorporates a VRPTW in each period. Although this literature review resorts to the three aforementioned problems, the reader is referred to the Rich Vehicle Routing Problem (RVRP) literature surveys provided by [Doerner and Schmid \[2010\]](#), [Caceres-Cruz et al. \[2014\]](#), and [Lahyani et al. \[2015\]](#). Indeed, to adapt our formulation for the real-world application, various business-related constraints need to be added so as to obtain company-compatible solutions.

6.2.1 Vehicle Routing Problem With Time Windows

Besides the consistency constraints regarding the number of drivers servicing each customer, the problem we aim to tackle considers time-windows for each request. This feature is usually necessary in real-world problems as customers need to be prepared to receive shipments. This type of constraint is responsible for a significant increase in complexity and most successful approaches are obtained with non-exact methods. Additionally, it is worth mentioning that our challenge also considers release dates for each order, which is a concept that was recently introduced by [Cattaruzza et al. \[2016\]](#). Release dates are intimately related with the VRPTW as they roughly change the bounds of the variables controlling the beginning of each time window. For these reasons, we review related VRPTW literature.

Solution approaches based on the Tabu Search (TS) metaheuristic have been proven to be quite efficient. [Taillard et al. \[1997\]](#) propose a TS heuristic for a Vehicle Routing Problem (VRP) with soft time-windows. The vehicles are allowed to arrive late at customer locations, although a penalty is incurred in the objective function. The algorithm uses a stochastic insertion heuristic to construct different solutions. Then, the TS heuristic is applied to each solution and the resulting routes are stored in an adaptive memory, which will be later used to construct other solutions. A post optimization procedure is applied to each individual route. This methodology has produced the best-known solutions for

VRPTW benchmark instances. Extending the work of Shaw [1998], Pisinger and Ropke [2007] present a general heuristic for several VRPs, including the VRPTW. An Adaptive Large Neighborhood Search (ALNS) metaheuristic is applied, improving 183 best-known solutions out of 486 benchmark tests. In this metaheuristic, a number of simple algorithms compete to modify the current solution. In each iteration, one algorithm destroys and another algorithm repairs the solution. The choice of the algorithms to be used in each iteration is made by an adaptive layer, which is biased according to the past performance of each algorithm. The authors describe this methodology as a sequence of fix-and-optimize operations. The fix operation selects a subset of variables that are fixed at their current value whereas the optimize operation seeks to find a near-optimal solution by changing non-fixed variables. This is one of the most successful methods among the VRP community and therefore we consider that exploring different techniques to fix and optimize parts of routing problems is a promising research direction. Cordeau et al. [2001] propose a unified TS heuristic for the VRPTW where an initial solution is constructed, without guaranteeing feasibility. Afterwards, the TS algorithm generates a certain number of solutions and chooses the best, feasible one. This solution is then post-optimized by applying a specialized heuristic for the Traveling Salesman Problem with Time Windows (TSPTW) to each individual route. The computational experiments show that the proposed algorithm may not be the best available for the VRPTW. Nevertheless, this weakness is compensated by the flexibility, the speed of execution, and memory usage of the approach. The program can run on any computer with minimal resources, solving instances with up to 100 customers. Recently, Vidal et al. [2015] uses different approximation methods to explore the contribution of exploring infeasible solutions in heuristic searches for the VRPTW. All tested relaxations introduce positive impacts in terms of solution quality, computational time or scalability.

Purely exact methods are not as common in the literature as they are not able to solve instances with a large number of customers. The survey presented by Kallehauge [2008] reviews four different formulations for the VRPTW and describes two main lines of development concerning exact algorithms. One focused on general decomposition approaches and on the solution of dual problems associated with the VRP. The other is concerned with the analysis of the polyhedral structure of the problem. Baldacci et al. [2012] review recent exact methods for the VRPTW and report a comparison between different approaches. It is interesting to observe that state-of-the-art approaches can only solve instances with less than 100 customers.

6.2.2 Periodic Vehicle Routing Problem

The PVRP demands the definition of vehicle routes for several periods without considering consistency constraints. Unlike the conVRP, the PVRP has been studied extensively for more than forty years and its applications cover numerous contexts. Recently, many applications have been considering the classic PVRP as a basis to which additional constraints or alternative objectives are added [Campbell and Wilson, 2014]. The conVRP is an application of the PVRP considering consistency constraints and the two problems are closely related. Therefore, it is worth reviewing the literature concerning this topic. The period-

icity of VRP may be considered in three different ways: a) a predefined set of allowable delivery schedule alternatives [Christofides and Beasley, 1984]; b) the intervals between deliveries to each customer [Cordeau et al., 2001]; c) minimum and maximum required spacing between deliveries [Gaudioso and Paletta, 1992]. In this paper, we are particularly interested in the first and second cases, as they are closely related to our challenge in the sense that service level agreements may be modeled with the help of these two concepts. The PVRP has been tackled by numerous solution approaches. Cordeau et al. [1997] propose a TS metaheuristic for a PVRP with multiple depots, solving instances with up to 360 customers and 9 time periods. Hemmelmayr et al. [2009] present a VNS for a PVRP with time-windows. The initial solution is constructed by solving a VRP for each day using the Clarke and Wright savings algorithm [Clarke and Wright, 1964]. Afterwards, the algorithm searches for better solutions by applying the ideas presented by Hansen and Mladenović [2001]. The algorithm finds new best solutions for instances containing up to 400 customers and time periods. Vidal et al. [2012] tackle a PVRP by means of a hybrid GA. The computational experiments for the PVRP, considering a set of instances with up to 400 customers, show that the method can identify either the best-known solutions or new best solutions for all benchmark instances. Therefore, this is the state-of-the-art metaheuristic for the problem.

Few exact approaches are available for the PVRP. Baldacci et al. [2011] present a mathematical formulation that is used in an exact approach. The effectiveness of the proposed method is shown on benchmark and new sets of test instances. The paper provides the first evaluation of the best-known solutions for the PVRP instances reported over the last 30 years.

Given that it is quite clear that exact methods are still unable to solve large instances, researchers try to hybridize exact and approximate phases to enhance the algorithms for the PVRP. Pirkwieser and Raidl [2009] propose a column generation where a mathematical formulation selects a set of optimal routes (columns) to solve a PVRP. Recently, Cacchi-ani et al. [2014] propose a hybrid optimization algorithm, which also relies on a column generation approach to constrain the route set.

Although there are not many approaches based on this idea, using exact algorithms to iteratively solve smaller sub-problems may be relevant. Indeed, there is evidence that approaches such as fix-and-Optimize [Helber and Sahling, 2010] can be quite efficient in different optimization problems. The idea is to iteratively find improvements in a given solution by fixing parts and optimizing the smaller sub-problems. This makes it possible to solve larger instances, as one can fix as many variables as the solver needs.

6.2.3 Consistent Vehicle Routing Problem

Although we have particular interest in reviewing VRPTW and PVRP literature, this research focuses mainly on a less studied VRP extension which is the conVRP. This optimization problem demands the definition of vehicle routes for several periods, maintaining a certain level of consistency on pre-selected metrics. The objective is to achieve minimum cost routing plans satisfying the classical routing constraints as well as predefined consistency requirements. Generally, customer-oriented routing considers two types of

consistency for customer satisfaction: driver consistency and time consistency [Kovacs et al., 2014a]. Driver consistency is measured by the number of different drivers that visit a customer, whereas time consistency is related to the maximum difference between the earliest and the latest arrival times at each customer. Time window assignment problems, as presented in Spliet and Gabor [2015] and Spliet and Desaulniers [2015], achieve time consistency in a slightly different manner by setting a single time window with a constant amplitude for each customer, thus delays are not allowed.

The conVRP arises in many industries where customer satisfaction is considered a distinctive factor in competitiveness. Particularly in industries transporting small packages, providing a standard service with a single driver and approximately at the same time of the day enables customers to be prepared for a delivery, strengthening supplier/customer relationships [Kovacs et al., 2014c]. For further insights on VRPs where consistency is important, the reader is referred to the survey provided by [Kovacs et al., 2014b].

Despite the advantages of adopting consistent routes, few papers have addressed the conVRP and most approaches resort to approximate methods. Groër et al. [2009] formulate the conVRP as an MIP and improve the algorithm used by Li et al. [2005] to solve very large VRP. The consistent routes obtained for a real-world data set are less than 10% longer on average, compared to inconsistent routes. Tarantilis et al. [2012] propose a TS algorithm to iteratively generate template routes improving the best reported results over all conVRP benchmark instances. Kovacs et al. [2014c] construct template routes by means of an ALNS. It is shown that solving daily VRPs may lead to inconsistent routes whereas consistent long-term solutions can be generated by using historical template routes. Kovacs et al. [2014a] state that by assigning one and only one driver to each customer and bound the variation in the arrival times over a given planning horizon may be too restrictive in some applications. They propose the generalized conVRP in which a customer is visited by a limited number of drivers and the variation in the arrival times is penalized in the objective function. A Large Neighborhood Search (LNS) metaheuristic generates solutions without using template routes. The computational results on different variants of the conVRP prove the efficiency of the algorithm, as it outperforms all published algorithms. Sungur et al. [2010] consider a real-world courier delivery problem where customers appear probabilistically. Although the authors do not call it a conVRP, their assumptions are in line with problems such as this. The proposed approach generates master plans and daily schedules with the objective of maximizing both the coverage of customers and the similarity between the routes served each day. Braekers and Kovacs [2016] explore driver consistency in a multi-period dial-a-ride problem. The authors propose different formulations and assess their efficiency using a branch-and-cut scheme. Additionally, they propose an ALNS that generates near optimal solutions.

The papers proposing a mathematical formulation to deal with consistency features are still scarce in the literature. These formulations are only able to solve small instances (i.e. Kovacs et al. [2014a] solves instances containing less than 12 customers). To the best of our knowledge, no single approach is able to efficiently make use of a mathematical formulation to solve conVRP instances with realistic size. However, hybrid solution approaches combining exact and approximate methods have been very successful in providing a good trade-off between solution quality and computational times Archetti and Speranza [2014].

6.3. Problem Statement and Mathematical Formulation

In this section, the Consistent Vehicle Routing Problem with Service Level Agreements (conVRPSLA) is formally expressed in terms of a Mixed Integer Programming (MIP) model.

Let \mathcal{D} be the set of days to be carried out, \mathcal{P} the set of distinct shipping periods of the depot, and \mathcal{C} the set of customers to be served. Each customer has a set of ordering windows that, according to their SLA, may or may not have an associated time window $[a_i, b_i]$, a previously assigned shipping period p_i and/or an order release date rd_i (Figure 6.1). The time window refers to the time at which the order should be delivered to the customer, whereas the period specifies whether this order should be delivered via a certain route in a specific shipping period. Finally, the order release date refers to the earliest time at which that specific order can be ready for shipping.

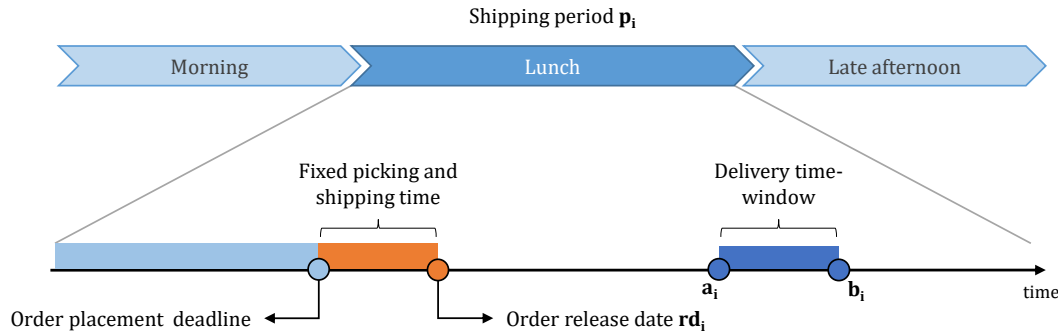


Figure 6.1 – Parameters defining the service level agreements

Let \mathcal{N} be the set of n nodes, each representing a pair customer-shipping period with c_i being the customer of node i . In this way, every different shipping period of each customer is to be treated as a different node (Figure 6.2). Two additional fictitious nodes 0 and $n + 1$ are added to the set to represent the depot as the departure node and the return node, respectively, with \mathcal{N}_C being the subset of all non-depot nodes. Let \mathcal{N}_{TW} be the subset of nodes with a time window different from the depot's time window, \mathcal{N}_P the subset of nodes with a specific shipping period and \mathcal{N}_{OD} the subset of nodes with a relevant order release date. On each day, customers may or may not place an order in any of the shipping periods. As such, let \mathcal{O} be the set of orders (i, t) placed by node i on day t , which are defined by their service time st_{it} and quantity ordered q_{it} .

Furthermore, a set \mathcal{A} of arcs connecting nodes i and j on day t is considered. Note that arc $(0, n + 1, t)$ exists and will be used when a route is not to leave the depot on that day. Each arc has an associated travel time tt_{ij} and a travel distance td_{ij} , which are both independent of the day. These arcs may be pre-processed in order to eliminate combinations that will never be used and are therefore unnecessary to use in the model. For each day t , in order for an arc to exist connecting node i to node j , both need to be nodes with orders placed on day t or one of the depot nodes, 0 or $n + 1$. If i and j are customer nodes, they must belong to different customers ($c_i \neq c_j$) and their specific shipping periods, if any, must be the same.

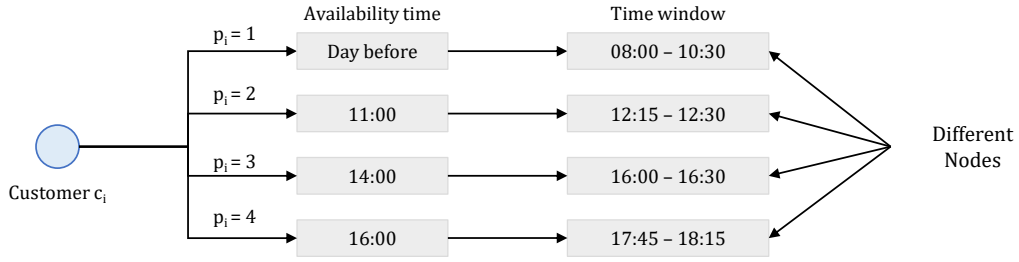


Figure 6.2 – Different nodes belonging to the same customer

A set of routes \mathcal{R} is also considered, with each route r representing the departure of a vehicle on a specific shipping period rp_r . The routes are assumed to be performed by the same driver on each day. This is the reason why driver consistency is needed. The vehicles that will serve the routes are considered to be homogeneous with capacity Q . In order to adjust the model to address labor related legal constraints, a maximum single route duration MD is also considered in the model. A depot time window $[aW, bW]$ is also present, representing the operating hours of the depot and therefore defining the earliest departure time and latest return time for all routes.

In this model, there are three main binary decision variables: X_{ijt}^r , which defines each arc (i, j, t) that is traveled by route r ; Z_i^r , which defines each node's route assignment; Y_r , which defines whether route r is active, i.e. if any customer is assigned to that route. Additionally, three time related decision variables are also considered, with W_{it} representing the arrival time at customer i on day t , dW^r the planned departure time of route r and rW_t^r the time at which route r arrives at the depot on day t . Moreover, since daily ordering patterns are uncertain, service level, capacity and route duration constraints might not be fulfillable at all times. As such, constraint softening decision variables were defined along with their associated objective function weights. Early and late arrivals at node i on day t are represented as ϕ_{it}^- and ϕ_{it}^+ , respectively. Moreover, ω_t^r defines how much route r exceeds maximum route duration L on day t and θ_t^r is the excess capacity on route r and day t . A compact description of all sets, parameters and decision variables may be found in 6.A.

6.3.1 Mathematical formulation

The conVRPSLA mixed integer linear formulation is as follows.

The objective function (6.1) primarily minimizes the total distance traveled since this is usually the main cost driver in transportation operations. A penalty for the number of routes is also included if there is a fixed cost associated with their execution. The remaining parcels represent the different penalties for breaking constraints regarding customer arrival times, route duration and vehicle overloads, respectively.

$$\min \sum_{(i,j,t) \in \mathcal{A}} \sum_{r \in \mathcal{R}} td_{ij} \cdot X_{ijt}^r + \sum_{r \in \mathcal{R}} \alpha_1 \cdot Y_r + \sum_{i \in \mathcal{N}} \sum_{t \in \mathcal{D}} (\alpha_2 \cdot \phi_{it}^- + \alpha_3 \cdot \phi_{it}^+) + \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{D}} (\alpha_4 \cdot \omega_t^r + \alpha_5 \cdot \theta_t^r) \quad (6.1)$$

Firstly, each node must be assigned to only one route (6.2). Also, each node must be visited exactly once, but only when it has placed an order on that particular day and via the route

it was assigned to (6.3). To ensure a correct flow, the same reasoning applies to the number of departures from each node (6.4).

$$\sum_{r \in \mathcal{R}} Z_i^r = 1 \quad \forall i \in \mathcal{N}_C \quad (6.2)$$

$$\sum_{i \in \mathcal{N} \setminus \{(i,j,t) \in \mathcal{A}\}} \sum_{t \in \mathcal{D}} X_{ijt}^r = Z_j^r \quad \forall j \in \mathcal{N}_C, \forall r \in \mathcal{R} \quad (6.3)$$

$$\sum_{j \in \mathcal{N} \setminus \{(i,j,t) \in \mathcal{A}\}} \sum_{t \in \mathcal{D}} X_{ijt}^r = Z_i^r \quad \forall i \in \mathcal{N}_C, \forall r \in \mathcal{R} \quad (6.4)$$

Each route must also depart from the depot exactly once every day (6.5). If no orders are placed by the nodes on route r on a given day t , then the existence of arc $(0, n+1, t)$ allows for the fulfillment of this constraint. The return of the route to the depot is already assured by the basic flow constraints (6.3) and (6.4). When at least one customer is assigned to a route r , then it is an active route (6.6).

$$\sum_{j \in \mathcal{N} \setminus \{(0,j,t) \in \mathcal{A}\}} X_{0jt}^r = 1 \quad \forall t \in \mathcal{D}, r \in \mathcal{R} \quad (6.5)$$

$$Z_i^r \leq Y_r \quad \forall i \in \mathcal{N}_C, r \in \mathcal{R} \quad (6.6)$$

As the several shipping periods for each customer are treated as different nodes, they have to be forced to be shipped on different routes (6.7). Moreover, when an order has a specific shipping period assigned to it, it can only be delivered by routes operating in that period (6.8).

$$Z_i^r + Z_j^r \leq 1 \quad \forall r \in \mathcal{R}, i, j \in \mathcal{N}_C : c_i = c_j \quad (6.7)$$

$$rp_r + M(1 - Z_i^r) \geq p_i \geq rp_r - M(1 - Z_i^r) \quad i \in \mathcal{N}_P, r \in \mathcal{R} \quad (6.8)$$

In order to enable the existence of time windows in the model, the arrival time at all customers must be traced. As such, the starting time of each service at a given node j must be greater than or equal to the time at which the previous node i is served plus its service time and the travel time between the two (6.9). Additionally, these constraints serve as subtour elimination constraints. For each node, the arrival time at each day is compared to its target time window (6.10) to capture the deviation from the window $[a_i, b_i]$. Therefore,

these constraints are responsible for the time consistency of the solution.

$$W_{it} + st_{it} + tt_{ij} \leq W_{jt} + M(1 - X_{ijt}^r) \quad \forall (i, j, t) \in \mathcal{A}, i, j \in \mathcal{N}, \forall r \in \mathcal{R} \quad (6.9)$$

$$a_i - \phi_{it}^- \leq W_{it} \leq b_i + \phi_{it}^+ \quad \forall (i, t) \in \mathcal{O}, i \in \mathcal{N}_{TW} \quad (6.10)$$

Both the depot departure and return times need to be coordinated with the first (6.11) and last (6.12) deliveries, respectively.

$$dW^r + tt_{0j} \leq W_{jt} + M(1 - X_{0jt}^r) \quad \forall (0, j, t) \in \mathcal{A}, r \in \mathcal{R} \quad (6.11)$$

$$W_{it} + st_{it} + tt_{i0} \leq rW_t^r + M(1 - X_{i(n+1)t}^r) \quad \forall (i, 0, t) \in \mathcal{A}, r \in \mathcal{R} \quad (6.12)$$

In addition, vehicles need to leave and return to the warehouse during operating hours (6.13, 6.14). Also, each route must depart only after all of its orders are ready to be shipped, as stated in constraints (6.15).

$$dW^r \geq aW \quad \forall r \in \mathcal{R} \quad (6.13)$$

$$bW \geq rW_t^r \quad \forall t \in \mathcal{D}, r \in \mathcal{R} \quad (6.14)$$

$$dW^r \geq rd_i - M(1 - Z_i^r) \quad \forall i \in \mathcal{N}_{OD}, r \in \mathcal{R} \quad (6.15)$$

Furthermore, it is necessary to keep track of each vehicle's total load (6.16) and daily duration (6.17) in order to define the penalties for excesses on any day.

$$\sum_{i \in \mathcal{N}_C} q_{it} \cdot Z_i^r \leq Q + \theta_t^r \quad \forall r \in \mathcal{R}, t \in \mathcal{D} \quad (6.16)$$

$$rW_t^r - dW^r \leq MD + \omega_t^r \quad \forall r \in \mathcal{R}, t \in \mathcal{D} \quad (6.17)$$

Finally, integrality, binary and non-negativity conditions are set by constraints (6.18).

$$X_{ijt}^r, Z_i^r, Y_r \in \{0, 1\}; \quad W_{it}, dW_t^r, rW_t^r, \phi_{it}^-, \phi_{it}^+, \omega_t^r, \theta_t^r \geq 0. \quad (6.18)$$

6.4. Solution approach

As in most vehicle routing problems, the proposed model proved to be hard to solve using exact methods comprised in most commercial solvers, as the instances considered are quite large. Therefore, a Fix-and-optimize (FO) based approach was designed to tackle this problem by attempting to make use of its structural characteristics. In this section, an overview of the methodology is given along with a detailed explanation of the proposed algorithm.

6.4.1 Methodology overview

In the conVRPSLA, the main driver for complexity is the number of nodes considered as they increase the number of arcs and, consequently, the number of decision variables and constraints, exponentially [Ordóñez et al., 2005]. Hence, a solution method was devised with the goal of (1) reducing the number of nodes, (2) iteratively exploring the solution space, and (3) leveraging the developed mathematical model. This method includes a pre-processing stage in which the dataset is treated in order to simplify the problem followed by the actual problem solving stage.

The FO procedure fixes most of the nodes and arcs and optimizes smaller subproblems at each iteration. This allows for the exploration of a high percentage of the solution space without splitting the problem into different instances and while maintaining efficiency. This method requires a feasible initial solution which can be very difficult to achieve with the mathematical model, given the size of the instances. Therefore, an initial solution construction algorithm was implemented to provide fast feasible solutions to be used in the FO phase.

Since the subsets of variables (related to nodes and arcs) that are fixed and released in every iteration can be built in many different ways and with different cardinalities, a systematic approach is preferable. Indeed, our method combines the principles of FO and of the Variable Neighborhood Decomposition Search (VNDS) – see Seeanner et al. [2013]. VNDS is a variant of VNS to make it possible to tackle large-sized problems. It “decomposes” the problem by only regarding different parts of the solution space (a result of a kind of decomposition). The basic idea explored in this paper is to apply the concept of VNDS in order to methodically adapt the variables sets for the FO.

6.4.2 Node grouping

The node grouping step is based on the algorithm proposed by Dondo and Cerdá [2007] with the intention of reducing the computational effort of subsequent phases. The goal is to aggregate small groups of nodes with compatible time windows that are geographically

very close to each other when compared to the global set. This significantly reduces the number of variables and constraints in the model by assuming that these customers will always be served by the same vehicle, which is most likely the case. In order not to limit the ability of the model to allocate nodes to different periods, this stage is only performed for nodes with the same shipping period.

The grouping algorithm (Algorithm 1) starts with a pool of all nodes from \mathcal{N} ordered first by a_i and then by b_i (Algorithm 1, line 3), while requiring a maximum inter-node distance of $maxD$ and a maximum vehicle waiting time of δ_{wait} . Also, let \mathcal{N}^* be a set of grouped nodes i^* to be populated by the algorithm and \mathcal{N}' a temporary set of unassigned regular nodes. Then, and until the original set \mathcal{N} is empty, groups of nodes i^* will be created, populated and then added to the final set of grouped nodes \mathcal{N}^* . Every time a new grouped node i^* is initialized, it is populated with the first node of \mathcal{N} , which is removed from the original set. The temporary set \mathcal{N}' is then created as a copy of \mathcal{N} and iterated for each of its nodes i . Node i is added to group i^* if they both share the same shipping period, if the distance to its closest node is less than the maximum allowed, if the vehicle capacity is respected, and if the time window is fulfillable (Algorithm 1, line 9). For this last condition to be met, the time to serve all the nodes in the group needs to be traced. Every time a new node joins the group, both its own service time and the travel duration between it and the node of the group it is closest to are added to the previous group service time. The latest time window in the node should then be fulfillable assuming service starts at the beginning of the first node's time window and that a maximum delay in the delivery to the last customer of the grouped node of δ_{wait} is allowed. While δ_{wait} should be defined as a relatively small period of time compared to expected route duration as it represents how late it is acceptable to arrive at a customer in order for it to be grouped with nearby nodes, D_{max} should vary according to the node density of each specific instance. In the end, all distances $td_{i^*j^*}^*$ and travel times $tt_{i^*j^*}^*$ are recalculated respectively as $td_{i_n^*,j_0^*}^*$ and $tt_{i_n^*,j_0^*}^*$, assuming that the service of the new group of nodes starts at the first node and progresses according to their placement order.

6.4.3 Initial solution construction

The initial solution construction stage is straightforward as its only purpose is to allow for a quick start to the improvement phase when there is no initial solution provided. During this step, it is important to evaluate how close one node is to another, but the simple concept of distance is insufficient when dealing with different time windows. For this purpose, an incompatibility score between nodes τ_{ij} is introduced as shown in equation (6.19), attempting to measure the distance between nodes and penalizing them when their time windows are not compatible.

$$\tau_{ij} = td_{ij} + \left[\left(\frac{a_j + b_j}{2} - \frac{a_i + b_i}{2} - st_{it} \right) \cdot \bar{v} - td_{ij} \right]^2, \quad (6.19)$$

with \bar{v} being the average speed of the vehicles to convert time into distance. This speed can be computed from an average of the ratio between each of the travel distances and travel

Algorithm 1 Node grouping stage

```

1: function GROUPNODES( $\mathcal{N}, D_{max}, \delta_{wait}$ )
2:    $\mathcal{N}^* \leftarrow \emptyset$ 
3:   Order  $\mathcal{N}$  by  $a_i$ , then by  $b_i$ 
4:   repeat
5:      $i^* \leftarrow \text{First } i \in \mathcal{N}$ 
6:      $\mathcal{N}' \leftarrow \mathcal{N} \setminus \{i^*[0]\}$ 
7:     repeat
8:        $i \leftarrow \text{First } i \in \mathcal{N}'$ 
9:       if  $p_i = p_{i^*}$ 
10:        and  $\min\{td_{ji} \mid j \in i^*\} \leq D_{max}$ 
11:        and  $q_{i^*} + q_i \leq Q$ 
12:        and  $a_{i^*} + st_{i^*} + \min\{tt_{ji} \mid j \in i^*\} \leq \max\{b_i, b_{i^*}\} + \delta_{wait}$  then
13:          Add  $i$  to  $i^*$  after closest node  $j$ 
14:           $q_{i^*} \leftarrow q_{i^*} + q_i$ 
15:           $st_{i^*} \leftarrow \max\{st_{i^*} + tt_{ji} + st_i, a_i + st_i - a_{i^*}\}$ 
16:           $b_{i^*} \leftarrow \max\{b_{i^*}, b_i\}$ 
17:          Remove  $i$  from  $\mathcal{N}'$  and  $\mathcal{N}$ 
18:        else
19:          Remove  $i$  from  $\mathcal{N}'$ 
20:      until  $\mathcal{N}' = \emptyset$ 
21:      Add  $i^*$  to  $\mathcal{N}^*$ 
22:   until  $\mathcal{N} = \emptyset$ 
23:   return  $\mathcal{N}^*, td^*, tt^*$ 

```

durations in the instance. In the special case in which some nodes are to be placed on specific routes, a subset of nodes \mathcal{N}' and the corresponding list of node-route assignment variables Z' should also be provided. This can be especially useful if the algorithm is run on a previously encountered solution as an improvement mechanism (section 6.4.4).

The constructive algorithm (Algorithm 2) is performed in several steps. Firstly, routes are populated with the provided nodes \mathcal{N}' , if any. Then, each of the remaining nodes i is assigned the fixed route i_{fix} which contains the closest node from all the period compatible routes. With this information, the minimum incompatibility score found so far is traced by $minIncomp$ (Algorithm 2, line 5), while Boolean $foundNode$ tracks whether any node was found (Algorithm 2, line 12). The period-compatible route of the closest node found is then selected as initial delivery route for node i as well. If no compatible node is found, which happens when no previous route assignment is provided for the initial solution, the node is assigned a random route from all the period compatible ones. The value of variables Z'_i are then set accordingly, to 1 or 0, and an iteration through all the days and routes follows. In this step, orders are sorted by their earliest time-window and variables X_{ijt}^r are set assuming the route is performed in this sequence (Algorithm 2, line 20). As time windows are not hard constraints, as long as no shipping period compatibility constraint is violated, every partial solution constructed in this way is feasible.

Algorithm 2 Constructive heuristic.

```

1: function BUILDINITIALSOLUTION(Optional  $N'$ , Optional  $Z'$ )
2:   for all  $i \in N', r \in \mathcal{R} : Z_i^r = 1$  do
3:      $i_{fix} \leftarrow r$ 
4:   for all  $i \in N \setminus N'$  do
5:      $minIncomp \leftarrow 0$ 
6:      $foundNode \leftarrow false$ 
7:     for all  $r \in \mathcal{R} : rp_r = p_i, j \in N : j_{fix} = r$  do
8:       if  $minIncomp = 0$  or  $\tau_{ij} \leq minIncomp$  then
9:          $minIncomp \leftarrow \tau_{ij}$ 
10:         $i^* \leftarrow j$ 
11:         $foundNode \leftarrow true$ 
12:     if  $foundNode$  then
13:        $i_{fix} \leftarrow i^*$ 
14:     else
15:        $i_{fix} \leftarrow random\ r \in \mathcal{R} : p_i = rp_r$ 
16:   for all  $t \in \mathcal{D}, r \in \mathcal{R}$  do
17:      $O_t^r \leftarrow (i, t') \in \mathcal{O} : i_{fix} = r$  and  $t' = t$ 
18:     Order  $O_t^r$  by  $a_i$ , then by  $b_i$ 
19:     for  $k = 0$  to  $size(O_t^r) - 2$  do
20:        $X_{O_t^r[k]O_t^r[k+1]}^r \leftarrow 1$ 

```

6.4.4 Fix-and-Optimize algorithm

The last and main stage of the solution method is a neighborhood search improvement matheuristic based on a FO approach and of the VNDS. The reasoning behind this approach is grounded on the high computational burden of having a very large number of integer variables and on how easy it is to define a neighborhood as a set of nodes that are very close to one another. By defining a subset N_R as the set of all nodes which can change their values from the current solution and subsequently limiting a very large portion of the integer decision variables, the MIP becomes substantially easier to solve. Let a node be labeled as *fixed* if the variables Z_i^r are to maintain their values for every route r , and *released* if these same variables will be able to change values. Furthermore, let a route r be labeled as *fixed* if, for all nodes $i \in N$, Z_i^r is to remain equal to the incumbent solution and *released* if the nodes are allowed to both join and leave this route. In practice, fixing a binary variable means imposing its bounds to equal the value of the incumbent solution.

The FO algorithm (Algorithm 3) starts by setting a counter n_{noimp} to keep track of the number of iterations with no improvement to the incumbent solution and a time tracker ω to account for the time expended by the algorithm. Then, a loop with three main steps runs until the time counter is over the limit ω_{max} or the current neighborhood size n_{nodes} reaches maximum neighborhood size n_{max} . Firstly, the list of nodes to release N_R is computed according to the current neighborhood size. Then, both the node assignment variables Z_i^r and the arc usage variables X_{ijt}^r need to be either fixed to their current values or released.

This is done according to the previously defined list of nodes to release and the relevant arc releasing parameters.

After these pre-processing stages, the model is solved with a time limit of ω_{mod} and the next procedure depends on the solution found. Let sol represent the new solution, with sol_{obj} being its objective value and $sol_{\mathcal{V}}$ the full set of its variable values. If the incumbent solution is not improved, i.e. the objective value is not better than the overall best obj_{min} , the counter of non-improving iterations is updated. Then, when the counter reaches a pre-established limit n_{step} , the neighborhood size n_{nodes} is increased by $step$. This process increases the number of nodes that will be released in the next iteration, which allows for a larger search space that helps escape local optima. This systematic increase of the subsets (and therefore of the neighborhood structures) is inspired on the principles of the Variable Neighborhood Decomposition Search [Hansen and Perez-Britos, 2011]. On the other hand, if the best overall objective value is improved, the new solution is saved as the new overall best by updating obj_{min} and the overall best set of variables \mathcal{V} is set to current variable values in order to be used in the next iterations until a new best solution is found. In addition, if the current neighborhood size is larger than the initial one, n_0 , it is reset to restrict the exploration space and therefore refocus the search on the newly found solution.

Algorithm 3 Fix and Optimize algorithm

```

1:  $n_{noimp}, \omega \leftarrow 0$ 
2:  $n_{nodes} \leftarrow n_0$ 
3: repeat
4:    $n_{noimp} \leftarrow n_{noimp} + 1$ 
5:    $i_0 \leftarrow$  Select a random node from  $\mathcal{N}$ 
6:    $\mathcal{N}_R \leftarrow$  Select the  $n_{nodes}$  most compatible with  $i_0$  from  $\mathcal{N}$ 
7:    $sol \leftarrow$  solve subproblem conVRPSLA fixing the variables according to  $\mathcal{V}, \mathcal{N}_R, \delta$ 
8:   if  $sol_{obj} \leq obj_{min}$  then
9:      $obj_{min} \leftarrow sol_{obj}$ 
10:     $n_{noimp} \leftarrow 0$ 
11:     $\mathcal{V} \leftarrow sol_{\mathcal{V}}$ 
12:     $n_{nodes} \leftarrow n_0$ 
13:   else
14:     if  $n_{noimp} = n_{step}$  then
15:        $n_{noimp} \leftarrow 0$ 
16:        $n_{nodes} \leftarrow n_{nodes} + step$ 
17:   update  $\omega$ 
18: until  $\omega \geq \omega_{max} \vee n_{nodes} \geq n_{max}$ 

```

The first neighborhood definition stage deals with the selection of which nodes to release. The first node to be released, i_0 (Algorithm 3, line 5), is selected randomly from all the nodes in the set. Then, and until n_{nodes} nodes have been selected, the most compatible node i^* with the first random node is also selected for release [Shaw, 1998]. By using the incompatibility score (Equation (6.19)), the selection process considers both distance and time-window compatibility. Nodes are only considered for selection when they share the

shipping period with the previously released nodes or when they do not have a specific period for delivery (Algorithm 3, line 6). In the end, all the routes containing the selected nodes are released as well.

After selecting the routes and nodes to be fixed and released, this information needs to be converted to set the actual binary variables of the model. Note that, as the time the model will take to explore the solution space is extremely dependent on the number of binary variables it considers, this decision is of utmost importance to assure the algorithm performs efficiently. Hence, a new concept of a connection node is introduced representing the nodes that, on any of the days of the incumbent solution, are at a maximum of δ positions apart from any of the released nodes on their route. All the arcs connecting each connection node to released nodes or other connection nodes on its route will be released as well. By doing this, released nodes will be able to move freely between routes, while the connection nodes will remain on their current route. Therefore, this position buffer δ controls the trade-off between the number of released variables and the size of the explored neighborhood. In Figure 6.3, a representation of fixed (1), released (2) and connection nodes (3) shows how the latter are defined based on the released nodes around them. Note that, on day $t + 1$ in the example, non-ordering customers might not belong in that particular day's route, which creates additional connection nodes according to δ .

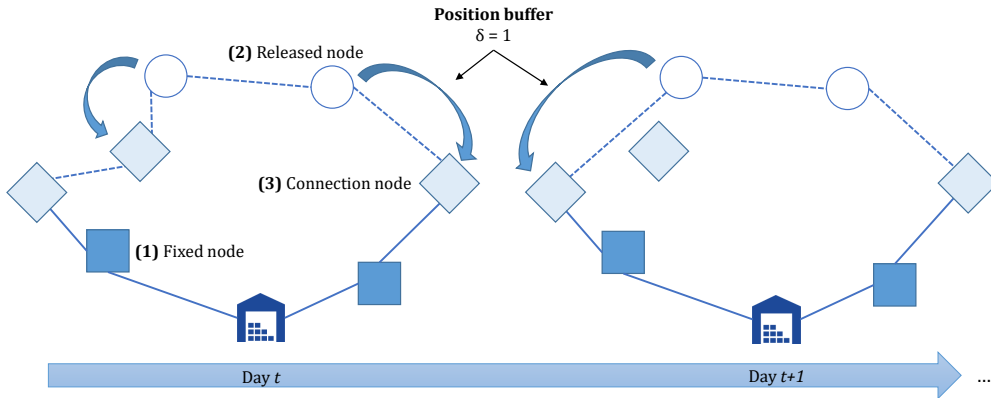


Figure 6.3 – Different types of nodes in a route

Let \mathcal{N}_F be the subset of fixed nodes, \mathcal{N}_C the subset of connection nodes and \mathcal{N}_R the subset of nodes to be released. Also, let similar subsets be created for routes, with \mathcal{R}_F being the subset of fixed routes and \mathcal{R}_R the subset of released routes. The first step is to define which nodes are connection nodes. In order to achieve this, each order (i, t) is given an attribute, pos_{it} , defining the sequential position on which it is fulfilled by its route during day t . Then, for each route r on day t , if a node i is within δ positions of a released node, it is transferred from the subset of fixed nodes \mathcal{N}_F to the connection node subset \mathcal{N}_C .

The next step is to set the binary variables of the model (Algorithm 3, line 7). Regarding node-route assignment variables, Z_i^r is fixed to its current value if $i \in \mathcal{N}_F \cup \mathcal{N}_C$ and it is released if $i \in \mathcal{N}_R$. As seen in Figure 6.4, an arc X_{ijt}^r is released in one of three situations. Firstly, if both $i \in \mathcal{N}_R \wedge j \in \mathcal{N}_R$, the arc is released for all $t \in \mathcal{D}, r \in \mathcal{R}_R$ (1), therefore allowing these nodes to freely join any of the released routes in between other released nodes. In

this stage, released routes may be from many different periods, which allows for nodes that do not have a specific one to freely change the shipping period. However, when just releasing these arcs, a node can only join another route if there are released nodes ordering on every day of the horizon in which it has also placed orders. This makes it hard for nodes to change routes unless the number of released nodes is very large, hence the connection nodes were introduced.

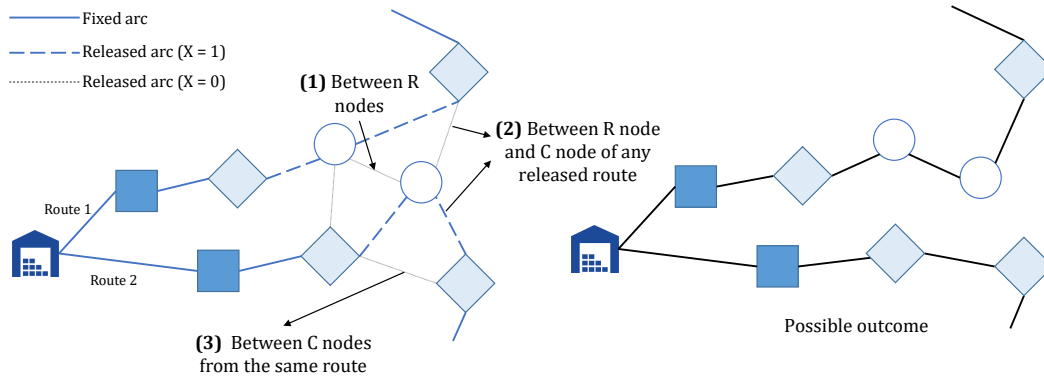


Figure 6.4 – Different types of released arcs to allow for changes in customer assignment

In order to allow for any released node to join routes in between connection nodes, all arcs X_{ijt}^r are released if either $i \in \mathcal{N}_R \wedge j \in \mathcal{N}_C$ or $j \in \mathcal{N}_R \wedge i \in \mathcal{N}_C$ for the connection node's current route for all $t \in \mathcal{D}$ (2). Finally, the arcs between connection nodes are also released for every $t \in \mathcal{D}$ if $i \in \mathcal{N}_C \wedge j \in \mathcal{N}_C$ (3) to enable breaking their direct connection. This last release is only done within route $r : Z_i^r = Z_j^r = 1$ because the purpose of connection nodes is not for them to change routes but instead to make room for the addition of new nodes to the route.

All the remaining arcs are fixed at their current values, effectively reducing the size of the problem by decreasing not only the number of binary variables, but also the number of constraints. This makes the MIP much faster to solve and suitable to apply in an iterative solution method in which many significant subproblems are solved.

6.4.5 Validation of the solution approach

In order to validate the solution approach both a comparison to a commercial solver and a sensitivity analysis on the key parameters of the algorithm are performed. The proposed solution approach aims to solve instances of large size. Instances of different sizes (in terms of number of customers) have been solved to prove the methodology is able to obtain close-to-optimal solutions in reasonable time when optimal values are obtainable by commercial solvers. The mathematical models and the matheuristic were solved using commercial solver CPLEX 12.6 and a two 12-core Intel Xeon with 60GB RAM machine with a maximum time limit of one hour. Table 6.1 shows average results on achieved objective values and time taken until reaching the best solution for randomly generated instances. Complete results for the instances used in this comparison are provided in 6.B.

Table 6.1 – Average results by number of customers

#Customers	Solver			Matheuristic		Relative Obj. Difference (%)
	Objective Value	Optimality Gap (%)	Runtime (s)	Objective Value	Runtime (s)	
10	1845.90	0.00	5.45	1852.20	0.46	+0.34
15	2265.05	0.28	199.90	2265.05	1.75	0.00
20	2567.80	1.70	807.25	2567.80	46.62	0.00
35	4075.00	19.18	3616.53	4050.07	355.69	-0.61
50	6501.53	43.97	3544.80	5393.87	785.87	-17.04

Results indicate that the matheuristic is able to achieve very similar results to the commercial solver for small size instances. For instances with more than 30 customers, the approximate solution method clearly outperforms the commercial solver in terms of solution quality and time. Note that for instances with 50 customers the optimality gap of the solver after one hour is close to 50%.

6.5. Case study in a pharmaceutical distribution company

6.5.1 Case study overview

The company targeted by the case study is a distributor that is part of a pharmaceutical wholesaler operating mainly in Portugal. The pharmaceutical industry business is characterized by its low margins for the distribution players, for being a highly regulated market, and for being part of a well-established competitive environment. The company mainly provides delivery services to a limited number of merchandise suppliers by specializing in the transportation of pharmaceutical items. These suppliers consist of other businesses belonging to the parent company, which need deliveries to pharmacies, supermarkets, perfume shops, and other large wholesalers.

The parent company's final customers (referred to as internal) are well defined in the information system, place frequent orders, and their packages are sent in uniform containers retrieved automatically from the company's warehouse. They may or may not have several deliveries in the different shipping periods and they may place orders until a deadline, which is normally around 30 minutes before the assigned vehicle departure. On the other hand, the deliveries made for other external companies (referred to as external) have fairly uncertain customers, non-uniform containers and arrive at the warehouse at the beginning of the day. As there is no unique identifier for the recipient of these orders, each package comes with its own delivery address and is assigned to a route according to its zip code prefix. Moreover, the types of service level agreements in place are different from customer to customer, with some of them having a strict time windows, a specific period to be shipped in and an order deadline just in time for the route departure, while others have no restrictions at all (Table 6.2). These different delivery characteristics with distinct service level agreements complicate the route planning process significantly, especially when developing consistent routes.

Table 6.2 – Main characteristics of the company's different customers

Customer origin	Container	Frequency	Time window	Order deadline	Route allocation
Internal	Uniform	Several daily deliveries	Strict	Before route departure	Consistent routes plan
External	Unpredictable	Single delivery	Site-dependent	Beginning of the day	Postal code prefix

The quantity demanded by each customer is also uncertain, as is the possibility of not ordering at all. There is a very subtle seasonality effect over the course of the year, peaking in the cold season, which is not very relevant to distribution. However, the main concern during the planning stage is the beginning of each month, when the customers tend to increase the amount ordered up to 10%, mostly due to increased budget availability and commercial target resets, which increases upstream pressure on the warehouse and challenges vehicle capacity.

Currently, the company operates from 5 different warehouses, to which customers have been previously assigned based mainly on geographical factors. One of these warehouses acts as the main distribution center by receiving from suppliers and stocking most of the merchandise, transferring stock to the remaining warehouses on a daily basis. By using existing customer assignments, each warehouse ends up having routes which are independent of the others, thus enabling the separation of the overall problem into warehouse-dependent instances.

In order to find consistent routes for this case study the proposed solution method with real data from a specific week was used. Afterwards, in order to validate the proposed routes and to compare them with current operations, a simulation model was developed to evaluate how the new plan would perform with real requests. The simulation used historical data from all the weekdays of that month so that the results could be compared with the actual company performance.

6.5.2 Route planning

The route planning stage is implemented by directly applying the conVRPSLA solution method with real data from the time span of a week. Since seasonality only occurs within each month and not so much over the year, with the peak happening in the first week, the first five weekdays of the month were used as the planning horizon. The company decision makers considered this period to be sufficient, as almost every internal customer is bound to order at least once during a full week period. Also, by using the peak period, the resulting routes should be more resilient and allow for better service levels, which is also one of the main goals of the company. In case relevant seasonality patterns exist throughout the year, additional weeks from different months or the peak month should be considered during this planning stage. If any large customer changes occur meanwhile, the company should adjust to them by making a new plan.

In order to speed up the solution procedure, the current customer routes, when a specific assignment existed, were used during the initial solution building stage. Also service time

in minutes (Equation (6.20)) was approximated according to a statistical study recently performed by the planning team in which actual service times at many different destinations were measured.

$$st_{it} \approx 2.5 + 1.5 \cdot \left\lfloor \frac{q_{it}}{6} \right\rfloor \quad (6.20)$$

There is a fixed component present at all deliveries which accounts for parking, picking, walking time and paperwork and also a variable component which accounts for the need of additional trips back and forth to the vehicle when the driver is not able to carry the whole load at once. The quantity considered is the measure approximated from the number of boards and boxes actually delivered. Finally, both the travel distance and travel duration matrices were obtained using the Google Maps API.

6.5.3 Simulation model

The consistent routes plan and the historical data, along with some other parameters, are the main inputs of the simulation model. The plan consists of simply allocating the main customers considered during the route optimization phase to the routes. The data that will be used during this stage consist of the demand in a month which was not used during the previous route planning step. By using different days for each stage, the results of the simulation are not influenced by having routes planned specifically for them and they are therefore actually tested for their practical application (without over-fitting). The simulation logic is presented in Algorithm 4, whose detailed description follows.

Algorithm 4 Simulation algorithm.

- 1: **for all** $i \in \mathcal{N} : i \notin \mathcal{N}_{Plan}$ **do**
 - 2: Look for closest node $j \in \mathcal{N}_{Plan}$ in a compatible route r
 - 3: Assign the route r to i
 - 4: **for all** $r \in \mathcal{R}$ **do**
 - 5: Define departure as the planned start time
 - 6: Define the return depot node's location
 - 7: **for all** $r \in \mathcal{R}, t \in \mathcal{D}$ **do**
 - 8: Run TSPTW model for day t and route r
-

Since customers are uncertain, new destinations which have never ordered before are very likely to appear. Hence, the first step of the simulation is to choose which route is going to serve requests which do not have a specific allocation. A simple allocation was performed by choosing, from all the routes that are able to fulfill its service level agreements, the one with the node which it is closest to the new request. After the requests that each route will have to serve on each day are known precisely, the next step is to actually calculate all the parameters throughout the whole simulation horizon. The main decision to be made in this stage concerns the order in which each request will be served and vehicle departure time, as all the other relevant parameters are easily computed with this information. The

distances and durations used during this stage will be the deterministic ones retrieved from the Google Maps API. These parameters are optimized through a mathematical model that solves the basic TSPTW problem [Gutin and Punnen, 2006] with the addition of the two-commodity flow subtour elimination constraints, as described by Oncan et al. [2009]. Since routes have been planned by taking capacity constraints into account during peak demand scenarios, overcapacity scenarios are fairly rare. Additionally, the company had contingency plans in place in case a large order could not be fulfilled in which an additional route was performed by a vacant vehicle or the order was delayed. As such, capacity constraints are generally not relevant on a daily basis in the simulated scenarios.

6.6. Results

In this section, the results obtained from solving the case study are provided. Firstly, the main characteristics of the solved instances are described, with the relevant parameters for the intermediate steps also provided. The results of the application of the FO approach to plan the routes of the company are then presented. Finally, the simulated results, which evaluate the impact of the planned routes on the actual operational KPIs are shown and discussed for each warehouse/depot and under a general scope as well.

6.6.1 Instance characteristics

As the model and solution method were tested as a tool to plan the routes of the company in the case study, and since the warehouses operate independently from one another regarding distribution, instances were created for each one of them. These instances were generated from historical data from the company's operations and the most relevant parameters are shown in Table 6.3. The customers from the larger warehouses were split into geographical clusters in order to improve the performance of the FO. The centroids of the routes in the current company were used in a *k-means* clustering algorithm in order to determine the clusters of customers to be routed separately.

The number of clusters was kept to a minimum value which allowed for the FO algorithm to load the model in a matter of seconds during each iteration. The node grouping parameters were then set for each depot, with the maximum inter-group node distance D_{max} being defined according to the geographical density of the customers, whereas the maximum deviation from the time window δ_{wait} was set to the delay value considered to be acceptable by the company. The depots situated in less dense areas have higher distances as, generally, there are no different routes serving areas within a few kilometers of each other. The different objective function weights were set based on discussions with the decision makers. The weights of the delays are higher in areas where customers are more likely to complain about late deliveries and the weight for the number of routes only exists in depots whose fleet is partially owned by the company and where the drivers are company employees.

The instances were then generated for each cluster for every warehouse. The number of nodes considered is effectively lower than the real number of customers due to not taking non-recurring destinations into account in the planning stage and by applying the node

Table 6.3 – Parameters used to define the instances

Warehouse	Number of clusters	$D_{max}[km]$	$\delta_{wait}[min]$	α_1	α_2	α_3	α_4	α_5
A	1	5	15	0	0.5	1	2	1
B	1	3	15	0	0.5	1	2	1
C	2	1.5	15	0	0.5	2	2	1
D	5	0.5	15	100	0.5	2	2	1
E	2	1	15	250	0.5	1	2	1

grouping algorithm. All the characteristics that indicate the size of each of the solved instances, specifying the results of each node reduction step, are shown in Table 6.4.

Table 6.4 – Main characteristics of the solved instances

Instance	Total customers	Recurring customers	Grouped Nodes	Grouping % reduction	Days	Routes	Orders
A	197	152	109	-28.3%	5	12	439
B	441	232	166	-28.4%	5	18	623
C1	319	268	181	-32.5%	5	16	762
C2	445	321	253	-21.2%	5	19	905
D1	367	338	304	-10.1%	3	29	817
D2	257	220	166	-24.5%	3	12	446
D3	457	384	283	-26.3%	3	21	774
D4	453	376	274	-27.1%	3	23	707
D5	397	320	220	-31.3%	3	16	578
E1	609	435	340	-21.8%	3	21	683
E2	445	295	259	-12.2%	3	14	521

The reduction algorithm is able to significantly reduce the size of the instances, achieving a reduction of over 20% in most cases. The largest warehouse whose routes were planned during this case study has over 1900 different customers, makes more than 5000 weekly deliveries and operates with 101 daily routes.

6.6.2 conVRPSLA results

The FO was developed in C# using CPLEX 12.6 for solving sub-problems. The tests were run on a personal computer with 12GB of random access memory and a i7-4710HQ 64-bit processor with 8 threads and maximum frequency of 3.5GHz.

In order to run the solution approach, the FO parameters had to be defined in such a way that the algorithm would be able to release as many nodes as possible while still having time to perform enough iterations to explore the whole instance. The position buffer δ was kept with a value of 2 since nodes that are more positions apart are already quite far from each other in most situations. The initial number of nodes in the released neighborhood, n_0 , was set to 15 to allow for the selection of a few nodes from the whole daily horizon in a small geographical area. The maximum number of iterations without improvement before increasing the neighborhood size, n_{step} , was set to 20, with the neighborhood size increasing by 10 nodes at a time. Finally, the time limit for each instance ω_{max} was set to 3

hours.

The total number of iterations, the objective value of the initial solution, that was defined using the routes currently made by the company, and the best objective value found are shown in Table 6.5. Note that both objective values represent the sum of all clusters for a given depot. The total variation (Var) between these two objective values represents how much the model would be able to improve the results if no uncertain element were present in daily operations.

Table 6.5 – Results of the conVRPSLA applied with the developed matheuristic

Warehouse	n_{iter}	Initial objective value	Best objective value	Objective value Var	Initial distance [km]	Best distance [km]	Distance Var
A	398	10306.75	10038.62	-2.6%	10143	9838	-3.0%
B	303	16209.25	15258.5	-5.9%	14741	14020	-4.9%
C	695	33555.25	31654.25	-5.7%	29382	28110	-4.3%
D	935	53712.0	51112.95	-4.8%	35999	34452	-4.3%
E	699	27469.0	23006.75	-16.2%	14818	12882	-13.1%
Total (avg)	3030	141252.3	131071.1	-7.2%	105083	99302	-5.5%

Overall, the model is able to improve the objective value by 7.2%, but with significant differences between the warehouses. In warehouse A, customers have a very low geographical density, hence the improvement is the lowest both in overall objective value and in distance traveled. On the other hand, warehouse E has the biggest share of uncertain customers being assigned to routes by their zip code prefix, which explains the much larger margin for improvement by the model.

As this optimization stage served the purpose of defining the new route plan for the company, the final results come from the simulation of the proposed plans using historical data.

6.6.3 Simulation results

The routes defined by the conVRPSLA were tested under real operating conditions. Note that, as the delivery locations of external customers are not known in advance by the company, it is operationally impossible to include them in the tactically defined route plans. Hence, route plans originally consist of internal customers while external customers are dynamically added to existing routes when their orders are actually placed as described on the simulation process.

Results are simulated for three different configurations in each warehouse. Firstly, the routes in the current plan are simulated in the same delivery order that was actually used by the company. Then, the same plan is used but, for each route and day, a TSPTW instance is created and solved to assure the optimum path is used. Finally, the routes proposed by the conVRPSLA model are simulated using the TSPTW model as well. In this way, it is possible to report two distinct results (Table 6.6), namely those obtained by optimizing the sequence in which routes are served ($\Delta_{sequenced}$) and those that result from changing the assignment of customers to routes while also optimizing their sequencing ($\Delta_{assignment}$).

Table 6.6 – Simulation results for the different company's warehouses

Depot	Simulation type	Assignment changes	Routes	Total duration [h]	Total distance [km]	Estimated cost [€]
A	Δ sequenced	-	-	-9.4%	-1.2%	-1.1%
	Δ assignment	11	0	-2.3%	-4.0%	-5.8%
B	Δ sequenced	-	-	-8.0%	-9.4%	-11.2%
	Δ assignment	9	0	-4.8%	-17.1%	-22.9%
C	Δ sequenced	-	-	-8.8%	-7.9%	-8.0%
	Δ assignment	100	-1	-5.9%	-17.5%	-14.5%
D	Δ sequenced	-	-	-4.4%	-9.5%	-7.9%
	Δ assignment	100	-2	-9.1%	-16.0%	-11.8%
E	Δ sequenced	-	-	-16.7%	-19.1%	-9.9%
	Δ assignment	48	0	-13.2%	-25.6%	-12.7%
Total / Average	Δ sequenced	-	-	-8.5%	-10.5%	-9.9%
	Δ assignment	268	-3	-8.4%	-17.4%	-12.7%

The simulation showed improvements in all warehouses. As expected, these results come from different sources and the different configurations help to identify the major factors explaining them. Warehouses A and B have very few nodes being reassigned, but they lead to large estimated cost reductions due to having many external customers being delivered to by better suited routes. Both warehouses C and D had many nodes changing routes and this is the main cause for the improvements, along with the reduction in the total number of routes. Warehouse E is very different from the others due to its large share of external customers. As stated in the route optimization stage, there is a large margin for improvement in reassigning the customers that are not present in the consistent routes plan, and the simulation stage further reiterates this statement. This warehouse also has by far the biggest improvements due to having better route sequencing, which was expectable since many customers are uncertain, making the drivers' decision regarding which sequence to follow much harder.

Overall, all the proposed changes have an estimated impact on the influenceable costs of operations of 12.7%, with benefits originating from both better route sequencing and an optimized assignment of external customers. These improvements consist of an overall expected reduction of 8.4% of total route durations and a decrease in total traveled distance of 17.4%.

6.7. Conclusions and future work

This paper focused on further exploring the conVRP, which is an increasingly popular problem both in literature and in practice as many businesses are shifting to a more service level focused planning. An extension to this problem, the conVRPSLA, was proposed, which considers different service level agreements that are very common in the pharmaceutical or spare parts distribution industry. Furthermore, in order to apply the model to

larger instances a matheuristic based on a Fix-and-Optimize decomposition approach was developed.

The solution approach was validated in a real case study of a Portuguese pharmaceutical distribution company that serves pharmacies several times a day as well as other external customers with different service level agreements. The mathematical model and matheuristic developed were used to plan the consistent routes of the company by using historical data. In order to test the proposed routes and to analyze how different scenarios might impact the operations of the company, a simulation model was developed.

The route optimization stage, which used the proposed matheuristic, found a new set of routes that improved the overall objective value by 7.2% and reduced the total distance traveled by 5.5% by assigning both types of customers to new routes. However, the operating conditions require a consistent pre-defined schedule, so, in order to validate the proposed consistent routes plan, the simulation model was used to test its performance using different historical data. In addition, simulations were run with the current routes so as to optimize the sequencing and thus evaluate whether the drivers are serving them efficiently.

The simulation stage showed that better sequencing of the routes could reduce the total duration of the routes by 8.5% and the distance traveled by 10.5%, which had an estimated impact of 9.9% on the costs considered in the company studied. The routes proposed by the optimization stage, which consisted of several assignment changes, and the elimination of 3 routes, had a small impact on the total duration after the optimized sequencing, but further improved the reduction in the distance traveled to a total of 17.4%. Overall, all the proposed changes had an estimated impact of 12.7% on the cost considered to be influenceable by implementing better planning processes.

In order to further develop the proposed methodology, a better initial solution construction procedure could be developed so as to obtain consistent route plans from scratch in a shorter period of time. Furthermore, an interesting additional analysis would be to compare the performance of the proposed matheuristic to different solution methods, namely metaheuristics which forego the use of the mathematical model and hence the flexibility to easily integrate additional operational constraints. Finally, it would be interesting to use a combination of lexicographical objectives and constraints instead of a weighted sum in the objective function to improve the control of the decision maker.

Acknowledgements

This research was partly supported by the PhD grant SFRH/BD/108251/2015, awarded by the Portuguese Foundation for Science and Technology (FCT) and by the project TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020, financed by the North Portugal Regional Operational Program (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This support is gratefully acknowledged.

Bibliography

- Archetti, C. and Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246.
- Baldacci, R., Bartolini, E., Mingozzi, A., and Valletta, A. (2011). An exact algorithm for the period routing problem. *Operations research*, 59(1):228–241.
- Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.
- Braekers, K. and Kovacs, A. A. (2016). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological*, 94:355–377.
- Cacchiani, V., Hemmelmayr, V. C., and Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, 163(PART 1):53–64.
- Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. A. (2014). Rich Vehicle Routing Problem: A Survey. *ACM Computing Surveys*, 47(2):1–28.
- Campbell, A. M. and Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1):2–15.
- Cattaruzza, D., Absi, N., and Feillet, D. (2016). The Multi-Trip Vehicle Routing Problem with Time Windows and Release Dates. *Transportation Science*, 50(2):676–693.
- Christofides, N. and Beasley, J. E. (1984). The period routing problem. *Networks*, 14(2):237–256.
- 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.
- Cordeau, J. F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119.
- Cordeau, J. F., Laporte, G., and Mercier, a. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52(8):928–936.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- Doerner, K. F. and Schmid, V. (2010). *Survey: Matheuristics for Rich Vehicle Routing Problems*, pages 206–221. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Dondo, R. and Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176:1478–1507.

- Gaudioso, M. and Paletta, G. (1992). A Heuristic for the Periodic Vehicle Routing Problem. *Transportation Science*, 26(2):86–92.
- Groër, C., Golden, B., and Wasil, E. (2009). The Consistent Vehicle Routing Problem. *Manufacturing & Service Operations Management*, 11(4):630–643.
- Gutin, G. and Punnen, A. P. (2006). *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media.
- Hansen, Pierre, M. N. and Perez-Britos, D. (2011). Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350.
- Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467.
- Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, pages 1–22.
- Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802.
- Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers and Operations Research*, 35(7):2307–2330.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014a). The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816.
- Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014b). Vehicle routing problems in which consistency considerations are important: A survey. *Netw.*, 64(3):192–213.
- Kovacs, A. A., Parragh, S. N., and Hartl, R. F. (2014c). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, 63(1):60–81.
- Lahyani, R., Khemakhem, M., and Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1):1–14.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.
- 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.
- Oncan, T., Altinel, I. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.
- Ordóñez, F., Sungur, I., and Dessouky, M. (2005). Factors that impact solution run times of arc-based formulations of the vehicle routing problem. *USC-ISE Working paper*, 3.

- Pirkwieser, S. and Raidl, G. R. (2009). A column generation approach for the periodic vehicle routing problem with time windows. *Proceedings of the International Network Optimization Conference*, 2009:1–6.
- Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.
- Seeanner, F., Almada-Lobo, B., and Meyr, H. (2013). Combining the principles of variable neighborhood decomposition search and the fix&optimize heuristic to solve multi-level lot-sizing and scheduling problems. *Computers & Operations Research*, 40(1):303–317.
- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 417–431. Springer.
- Spliet, R. and Desaulniers, G. (2015). The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, 244(2):379–391.
- Spliet, R. and Gabor, A. F. (2015). The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, 49(4):721–731.
- Sungur, I., Ren, Y., Ordonez, F., Dessouky, M., and Zhong, H. (2010). A Model and Algorithm for the Courier Delivery Problem with Uncertainty. *Transportation Science*, 44(2):193–205.
- Taillard, É., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186.
- Tarantilis, C. D., Stavropoulou, F., and Repoussis, P. P. (2012). A template-based Tabu Search algorithm for the Consistent Vehicle Routing Problem. *Expert Systems with Applications*, 39(4):4233–4239.
- 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. (2015). Time-window relaxations in vehicle routing heuristics. *Journal of Heuristics*, 21(3):329–358.

Appendix 6.A Nomenclature

Sets and subsets

\mathcal{A}	set of arcs
\mathcal{C}	set of customers
\mathcal{D}	set of days
\mathcal{N}	set of nodes
\mathcal{N}_C	set of non-depot nodes
\mathcal{N}_{TW}	subset of nodes with time windows
\mathcal{N}_P	subset of nodes with shipping period
\mathcal{N}_{OD}	subset of nodes with order deadlines
\mathcal{O}	set of orders
\mathcal{P}	set of shipping periods
\mathcal{R}	set of routes

Parameters

a_i	earliest limit of node i 's time window
b_i	latest limit of node i 's time window
p_i	mandatory shipping period of node i
rd_i	time at which orders from node i are available for shipping
n	number of nodes representing customers
st_{it}	service time of node i on day t
q_{it}	quantity ordered by node i on day t
td_{ij}	travel distance between node i and node j
tt_{ij}	travel time between node i and node j
rp_r	shipping period of route r
Q	vehicle capacity
L	maximum allowed route duration
aW	time at which the depot allows vehicles to leave
bW	time until which vehicles may return to the depot
M	big number
α_w	weights of objective function components with $w \in \{1, \dots, 5\}$

Decision variables

X_{ijt}^r	$= \begin{cases} 1 & \text{if arc } (i, j, t) \text{ is traveled by route } r, \\ 0 & \text{otherwise.} \end{cases}$
Z_i^r	$= \begin{cases} 1 & \text{if node } i \text{ is part of route } r, \\ 0 & \text{otherwise.} \end{cases}$
Y_r	$= \begin{cases} 1 & \text{if route } r \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$
W_{it}	time at which node i is served on day t
dW^r	time at which route r is set to departure from the depot
rW_t^r	time at which route r arrives at the depot on day t
ϕ_{it}^-	earliness at node i on day t
ϕ_{it}^+	lateness at node i on day t
ω_t^r	overduration of route r on day t
θ_t^r	overcapacity of route r on day t

Appendix 6.B Comparison between the solver and the matheuristic

Table 6.7 – Detailed results presenting the relative difference between the objective values of each approach ($\frac{\text{Matheuristic}-\text{Solver}}{\text{Solver}}$)

#Customers	Solver			Matheuristic		Relative Obj. Difference (%)
	Objective Value	Relative Gap (%)	Runtime (s)	Objective Value	Runtime (s)	
10	2089	0.00	3	2089	0.00	0.0
10	1854	0.00	3	1854	0.00	0.0
10	2023	0.00	4	2023	0.00	0.0
10	1964	0.00	3	1964	0.20	0.0
10	1071	0.00	4	1197	1.90	11.8
10	2155	0.00	5	2155	0.00	0.0
10	1820	0.00	5	1820	0.30	0.0
10	2301	0.01	5	2301	0.30	0.0
10	1521	0.00	4	1521	0.40	0.0
10	1581	0.00	9	1581	0.00	0.0
10	1848	0.00	9	1848	0.00	0.0
10	2407	0.00	2	2407	0.00	0.0
10	1447	0.00	7	1447	0.40	0.0
10	1162	0.00	11	1162	0.60	0.0
10	1662	0.00	7	1662	0.50	0.0
10	2397	0.00	6	2397	0.00	0.0
10	2362	0.00	3	2362	0.00	0.0
10	1562	0.00	1	1562	0.50	0.0
10	2176	0.00	13	2176	0.00	0.0
10	1516	0.00	5	1516	4.10	0.0
15	2444	0.01	84	2444	3.00	0.0
15	2610	5.50	3604	2610	2.90	0.0
15	3246	0.01	60	3246	7.00	0.0
15	2203	0.00	18	2203	1.60	0.0
15	2214	0.00	10	2214	0.90	0.0
15	2272	0.00	12	2272	1.00	0.0
15	2291	0.00	11	2291	0.80	0.0
15	2455	0.00	19	2455	0.00	0.0
15	1864	0.00	10	1864	0.00	0.0
15	1803	0.00	3	1803	0.70	0.0
15	2298	0.01	23	2298	1.20	0.0
15	2059	0.01	14	2059	0.70	0.0
15	1982	0.00	28	1982	1.10	0.0
15	1417	0.01	35	1417	3.80	0.0
15	2327	0.00	9	2327	0.60	0.0
15	2980	0.00	35	2980	5.50	0.0
15	1790	0.00	3	1790	0.60	0.0
15	2188	0.01	4	2188	0.70	0.0
15	2120	0.01	12	2120	2.20	0.0
15	2738	0.00	4	2738	0.60	0.0
20	2680	0.01	741	2680	166.80	0.0
20	3140	0.01	161	3140	43.90	0.0
20	2264	0.01	66	2264	14.20	0.0

#Customers	Solver			Matheuristic		Relative Obj. Difference (%)
	Objective Value	Relative Gap (%)	Runtime (s)	Objective Value	Runtime (s)	
20	2524	9.52	3608	2524	41.40	0.0
20	1908	0.00	47	1908	1.20	0.0
20	2462	0.00	53	2462	2.00	0.0
20	3337	6.53	3609	3337	246.50	0.0
20	2689	0.00	93	2689	1.70	0.0
20	2468	0.01	76	2468	1.20	0.0
20	2135	0.01	53	2135	74.50	0.0
20	2549	0.01	56	2549	2.40	0.0
20	2450	2.20	3601	2450	17.10	0.0
20	2492	0.01	32	2492	3.20	0.0
20	2442	0.00	70	2442	2.00	0.0
20	1943	0.01	92	1943	3.40	0.0
20	1892	0.00	44	1892	4.30	0.0
20	2448	0.01	13	2448	7.20	0.0
20	2310	15.74	3608	2310	128.80	0.0
20	2725	0.01	85	2725	154.20	0.0
20	4498	0.01	37	4498	16.40	0.0
35	3587	20.81	3614	3984	536.60	11.1
35	4025	20.73	3602	4008	391.20	-0.4
35	3308	13.98	3600	3240	451.70	-2.1
35	2694	25.83	3602	2623	533.00	-2.6
35	4513	19.30	3760	4708	268.80	4.3
35	4903	13.66	3602	4766	151.00	-2.8
35	4687	33.29	3602	4310	616.80	-8.0
35	3998	19.03	3626	3956	307.40	-1.1
35	4856	21.48	3603	4767	207.20	-1.8
35	4810	30.53	3604	4038	314.10	-16.0
35	3950	32.05	3602	3646	318.70	-7.7
35	4528	10.78	3610	4850	451.00	7.1
35	3702	9.10	3608	3702	240.80	0.0
35	4760	13.26	3609	4764	283.40	0.1
35	2804	3.93	3604	3389	263.70	20.9
50	5804	46.63	3615	4784	1118.00	-17.6
50	4326	37.72	3613	3733	603.60	-13.7
50	6242	45.01	3603	5141	1135.60	-17.6
50	9133	62.24	3603	6262	1149.50	-31.4
50	9875	64.50	3612	5709	858.20	-42.2
50	6287	45.81	3602	4902	866.60	-22.0
50	5997	30.17	3601	5929	757.30	-1.1
50	7080	38.30	3680	6244	693.10	-11.8
50	6150	48.48	3606	4622	757.20	-24.8
50	7171	42.45	3604	6416	725.80	-10.5
50	4456	40.85	3614	4456	973.10	0.0
50	6725	27.80	3620	6889	428.10	2.4
50	6938	47.74	1888	5611	595.70	-19.1
50	5214	37.81	3615	4715	422.20	-9.6
50	6125	43.97	4296	5495	704.00	-10.3

