UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

The flexible periodic vehicle routing problem: modeling alternatives and solution techniques

by

Diana Lucia Huerta Muñoz

Universitat Politècnica de Catalunya

Department of Statistics and Operations Research

The Flexible Periodic Vehicle Routing Problem: Modeling alternatives and solution techniques

by

Diana Lucia Huerta Muñoz

Dissertation for the degree of

Philosophy Doctor (PhD)

in Statistics and Operations Research

Barcelona, Spain

January 2018

To my greatest motivation in life, my family.

To my beloved homeland, México.

Contents

List of Figures

List of Tables

Acknowledgments

This research has been partially supported by the Spanish Ministry of Economy and Competitiveness and EDRF funds through grant MTM2015-63779-R (MINECO/FEDER), the Mexican National Council for Science and Technology (CONACyT), The Mexican Secretariat of Public Education (SEP), and *La Caixa* Foundation. This support is gratefully acknowledged.

I would like to express my sincere gratitude to my supervisors, Dr. Elena Fernández and Dr. Claudia Archetti, for guiding me on the right path during my PhD studies. Thank you for always supporting me in all I needed and for celebrating with me every achievement obtained. You are a great role model for me and it will always be a pleasure to work with you. Eskerrik asko, Elena!. Grazie mille, Claudia!.

My gratitude is extended to Dr. Roger Z. Ríos who has always believed in me and has supported me during my postgraduate studies. Thank you for being an excellent mentor and a great inspiration. You are an important piece of the beginning of my PhD.

Thank you to all the members from the Departament d'Estadística i Investigació Operativa for collaborating and sharing their valuable time to carry out all the activities we developed for the department, working with this amazing group has been a great experience for me. Special thanks to Jordi Castro, head of the PhD program, for supporting us in the initiatives proposed by our PhD students' group. Also I want to thank to Sonia Navarro, Carme Macias, Celia Martin, and Toni Font for always having a positive and helpful attitude every time I needed your help for any administrative and technical issue during my stay in the department.

I want to thank to all my friends and partners in Barcelona for being an essencial part of my life giving me the opportunity to live new experiences abroad. I want to thank especially to my very dear friends Daría Hernández and Claudia Teutli for their unconditional friendship that made the difference every day. Thank you for always listening to me and giving me a good advice when I need it the most.

I would like to thank to my friends and colleagues from the Department of Economics and Management of the Università di Brescia for every warm welcome you gave me during each research stay. My special thanks to my dear friends Valentina Morandi, Dario Fontana, and Michele Vindigni for all the invited coffees and the new Italian words learnt, but especially for all the great moments that you have shared with me which always make me feel like home. I am so lucky to met you all.

"Love is patient, love is kind. It does not envy, it does not boast, it is not proud. It does not dishonor others, it is not self-seeking, it is not easily angered, it keeps no record of wrongs." *1 Corinthians 13:4-5.* Thank you, Iván, my favorite travel companion, for giving to these words a special and stronger meaning with your unconditional love and support throughout all these years. You are a blessing in my life.

I allow myself to thank my family using my native language. Mamá, mi ángel en vida, gracias por tu apoyo incondicional en todo momento para el logro de mis metas y sueños, porque sé lo difícil que ha sido para ti cada despedida y aún así me comprendes y celebras conmigo cada logro. Gracias por todas las oraciones que siempre haces por mi. Gracias a mis hermanos por su apoyo en cada momento y a mis sobrinos por todos esos abrazos y muestras de cariño tanto a distancia como en persona que me motivan a dar lo mejor de mi para ser un buen ejemplo a sus ojos. Jamás dejaré de agradecer a mi ángel en el Cielo, mi proveedor de fuerza y mi primer y mejor profesor de matemáticas, mi papá, porque nunca dejó de creer en mi y siempre estuvo a mi lado apoyándome y brindándome sabios consejos que perdurarán por siempre en mi vida.

And last but not least, thank to all my friends and people, from Mexico and other countries who always had a kind word to encourage me to do my best during this phase of my life.

"...Con gente como esa, me comprometo para lo que sea por el resto de mi vida, ya que por tenerlos junto a mí, me doy por bien retribuido".

M. Benedetti

Thank you God for never leaving me alone in this walk.

Summary

Diana Lucia Huerta Muñoz.

Candidate for the degree of Philosophy Doctor in Statistics and Operations Research.

Universitat Politècnica de Catalunya.

Department of Statistics and Operations Research.

Thesis title:

The Flexible Periodic Vehicle Routing Problem:
Modeling alternatives and solution techniques

Abstract: In this thesis the Flexible Periodic Vehicle Routing Problem is introduced and studied. In this problem a carrier must establish a distribution plan to serve a given set of customers over a planning horizon using a fleet of homogeneous capacitated vehicles. The total demand of each customer is known for the time horizon and it can be satisfied by visiting the customer in several time periods. There is, however, a limit on the maximum quantity that can be delivered at each visit. The aim is to minimize the total routing cost. This problem can be seen as a generalization of the Periodic Vehicle Routing Problem which, instead, has fixed service schedules and fixed delivered quantities per visit. On the other hand, the Flexible Periodic Routing Problem shares some characteristics with the Inventory Routing Problem in which inventory levels are considered at each time period, the delivery of product is a decision of the problem and, typically, an inventory cost is involved in the objective function. The relation among these periodic routing problems is discussed and a worst-case analysis, which shows the advantages of the studied problem with respect to the problems with periodicity mentioned above, is presented. Furthermore, alternative mixed-integer programming formulations are described and computationally tested.

Given the difficulty to optimally solve the studied problem for small size instances, a matheuristic is developed, which is able to solve large size instances efficiently. Extensive computational experiments illustrate the characteristics of the solutions of the problem and show that, also in practice, allowing flexible policies may produce substantial savings in the routing costs in comparison with both the Periodic Vehicle Routing Problem and the Inventory Routing Problem.

Motivation:

- **The importance in real–world applications:** Vehicle Routing Problems are considered one of the most important class of problems in combinatorial optimization due to their variety of real–world applications. Focusing on versions where customers have periodic demand throughout a given time horizon, the study of Periodic Vehicle Routing Problems has increased, mostly in the last years, as many real–world applications related to recycling, periodic deliveries of products to customers, and periodic visits for providing specific services have a substantial impact nowadays.

- **The benefit of incorporating flexible service policies:** According to Campbell and Wilson (2014), one of the future directions of growth in the study of Periodic Vehicle Routing Problems is the increase of operational flexibility in their definitions. This is largely motivated by the increment of the real–world applications with periodic demand, which are usually addressed by limiting service visits to a predefined set of schedules, which, if it is not well defined, it can severely affect the quality of the final solution. Since one of the most important criteria in this type of problems is the minimization of transportation costs, incorporating flexible service policies may produce significant savings. In this thesis the term *flexible service policy* refers to service policies where the frequency of the visits to each customer as well as the delivered quantities are not determined a priori. Thus the time periods when each customer will be served and the quantity to be delivered in each visit have to be decided.

Scope:

- A new generalization of vehicle routing problems with periodic demands where flexible service policies are allowed is introduced and studied.

- Flexible service policies involve two types of decisions concerning when and how to satisfy the customers demands: those related to the frequency of visits to each customer and those related to the quantities to be delivered at each visit.

- This research includes an analysis of the mathematical properties and suitable mixed–integer programming formulations for the studied problem, as well as the development of exact and approximate solution methods.

Objectives:

- **General:**
  - To introduce and study a new vehicle routing problem with periodicity criteria and thus generate knowledge related to this area of research.
  - To analyze the effects of flexible service policies in vehicle routing problems with periodic demand, i.e., when visits and deliveries to customers can be determined by the decision maker instead of fixing them a priori (as they are usually managed in the literature).

- **Specific:**
  - To offer alternatives for addressing periodic vehicle routing problems by means of new mathematical models in which this type of flexibility is allowed.
  - To design and implement exact and approximate solution algorithms for solving the new problem.
  - To provide a significant analysis based on extensive computational experiments to show the advantages of the proposed approaches.

Methodology to achieve objectives:

1. Analysis of the state of the art on the related field to show the advantages and disadvantages of the existing models and solution methods.

2. Worst–case analysis for determining the potential savings obtained when flexible service policies are included.

3. Proposal of mathematical programming formulations for modeling the new vehicle routing problem with periodic demand.

4. Design and implementation of exact and approximate solution methods for the proposed problem.

5. Evaluation and analysis of the performance and efficiency of the solution algorithms, particularly for the case of large-size instances.

SCIENTIFIC DISCLOSURE: The research developed in this thesis has produced several publications and participations in conferences, workshops, PhD schools and research stays. All of them are briefly listed below.

- **Publications**

  – Archetti, C., Fernández, E., and Huerta-Muñoz, D.L. (2017a). The Flexible Periodic Vehicle Routing Problem. *Computers & Operations Research*, 85:58–70.

  – Archetti, C., Fernández, E., and Huerta-Muñoz, D.L. (2017b). A two-phase solution algorithm for the Flexible Periodic Vehicle Routing Problem. Submitted.

- **Presentations**

2018 **A two-phase algorithm for the Flexible Periodic Vehicle Routing Problem.** D.L. Huerta–Muñoz, C. Archetti, E. Fernández. ELAVIO XXII, Marbella, Chile. March 5th–9th, 2018.

2017 **A matheuristic for the Flexible Periodic Routing Problem.** D.L. Huerta–Muñoz, C. Archetti, E. Fernández. 18th EU/ME Workshop on Metaheuristics for a better world, Rome, Italy. April 3rd–4th, 2017.

2015 **Periodic Vehicle Routing Problems: Modeling Alternatives and Solution Techniques.** D.L. Huerta–Muñoz, C. Archetti, E. Fernández. Workshop on Combinatorial Optimization, Routing and Location (CORAL 2015), Salamanca, Spain. Sep 30th - Oct 2nd, 2015.

2015 **Problemas Periódicos de Rutas de Vehículos: Alternativas de Modelización y Métodos de Solución.** D.L. Huerta–Muñoz, C. Archetti, E. Fernández, XXXV National Congress of Statistics and Operational Research (SEIO 2015), Pamplona, Spain. May 26th–29th, 2015.

- **PhD schools**

2016 EURO PhD School in Matheuristics. Lorient, France. April 20th-28th, 2016.

2015 EURO PhD School on Routing and Logistics. Brescia, Italy. June 24th–July 3rd, 2015.

2015 Winter School on Network Optimization. Estoril, Portugal. January 12th–16th, 2015.

- **Research stays**

2017 2-month stay at Università degli Studi di Brescia. Brescia, Italy.

2015 3-month stay at Università degli Studi di Brescia. Brescia, Italy.

Chapter 1

Introduction

Finding better vehicle routes in the management of freight transportation and distribution is an aspect of crucial relevance. A mismanagement can generate high increments in operational costs, affecting competitiveness of companies and the perception of the service quality in customers. Some of the most significant costs that a company incurs include the transportation costs (mainly generated by the price of fuel) and the associated wage costs. The main aim is to minimize them. Some applications of this type of problems arise in product distribution companies, transportation services, periodic deliveries, waste collection, and many others (Golden et al., 2002).

The *Vehicle Routing Problem* (Toth and Vigo, 2014, VRP), is the name given to a class of problems that incorporate optimization tools in order to reduce routing costs. Early studies focus on the *Traveling Salesman Problem* (Shmoys et al., 1985, TSP) that assumes one single vehicle without any type of capacity limitation as well as on the most basic VRP models that integrate capacity constraints on an available fleet of vehicles. Recent works progressively incorporate more complex elements, trying to increase the similarity between the theoretical VRP models and the real–world applications. Some added elements are time windows, priorities, service frequencies, and inventory costs (Goel and Gruhn, 2006; Hashimoto et al., 2006; Lee, 2013).

Several methods proposed in the literature to address VRPs define a mathematical formulation to represent the problem and apply an exact solution algorithm for solving instances to optimality. Due to the limitation on the size of the instances that can be solved to optimality, an alternative for dealing with larger instances is to implement an approximate solution algorithm. Some works in the literature propose a combination of exact and approximate methods for addressing different parts of the original problem. These methods are known as *matheuristics* and have been successfully applied in VRPs (Archetti and Speranza, 2014).

This thesis introduces and studies a generalization of the VRP, called the *Flexible Periodic Vehicle Routing Problem* (FPVRP), which extends the *Periodic Vehicle Routing Problem* (Campbell and Wilson, 2014, PVRP) and focuses on *flexible service policies.* The term *flexible service policy* refers to a service policy where the frequency of the visits to each customer, as well as the quantities to be delivered in each visit, are not fixed in advanced and are part of the decision making process. The main objective of this thesis is to develop new modeling alternatives and appropriate solution methods that outperform, from a cost minimization perspective, the existing models where the frequency of visits and the amount to be delivered to customers are fixed. We define, model and develop formulation alternatives for the FPVRP. A first formulation uses a vehicle–index representation of the decision variables. An alternative formulation represents vehicle routes through their loads using a set of continuous variables. In addition, we propose two types of solution methods for solving the FPVRP. The first method is based on exact techniques applied to the proposed formulations. It includes several inequalities and optimality cuts to strengthen the formulations, as well as separation procedures for the families of constraints of exponential size. The second solution method, which is a matheuristic, is a two–phase algorithm integrating a mixed–integer linear programming (MILP) formulation and a Tabu Search (Glover and Laguna, 1997, TS) heuristic to obtain efficiently good quality solutions of large size FPVRP instances. Extensive computational experiments have been run in order to evaluate and compare the performance of the proposed solution algorithms.

This thesis is organized as follows. In Chapter 2, an extensive review of the literature related to the studied problem is carried out. Chapter 3 gives the formal definition of the FPVRP and studies its relation to other VRPs with periodic demand. Two illustrative examples and the theoretical worst-case analysis of the FPVRP with respect to other related problems are also presented. Several MILP formulations for the FPVRP as well as for the PVRP and IRP are proposed in Chapter 4. In Chapter 5 the exact solution algorithms developed for solving the FPVRP to optimality, the benchmark instances used in the computational experience and a summary of the analysis of the extensive tests performed to evaluate the proposed solution methods, are presented. Chapter 6 shows the description of a two–phase solution algorithm developed to solve medium and large size instances of the problem efficiently. Also, the analysis of the results of the corresponding computational experience is provided. Finally, in Chapter 7, the conclusions and future directions of the work done are presented. Some appendices are included to provide complementary results obtained during the development of this thesis.

Chapter 2

Literature review

2.1 The Vehicle Routing Problem

VRP (Toth and Vigo, 2014) is the term used to identify a class of problems focused on designing optimal routes for a fleet of capacitated vehicles that depart from a given depot in order to satisfy the demand of a set of customers. The minimization of the total transportation cost is one of the objectives most often considered in these problems. VRP constraints include those that model the assumption that vehicles have a limited capacity, so that the overall demand satisfied by a vehicle route does not exceed the capacity of the vehicle. These problems were introduced by Dantzig and Ramser (1959) in the work *The Truck Dispatching Problem* in which a set of routes was sought for a fleet of fuel dispatching vehicles, which should travel from a depot to several service stations to satisfy their demand.

The VRP is an NP–Hard problem (Garey and Johnson, 1979) because it is a generalization of two well-known combinatorial problems: the *Traveling Salesman Problem* (Shmoys et al., 1985, TSP) and the *Bin Packing Problem* (Coffman et al., 1984, BPP). In practice VRPs are much more difficult to solve than TSPs, due to the difficulty of the additional constraints. According to Laporte (2009) and Uchoa et al. (2017), the best–known exact algorithms can solve to optimally instances with up to 100 nodes. Because of the difficulty of finding good quality solutions in small computing times for real–world applications, a considerable amount of current work on VRPs deals with approximate methods to handle large size instances.

Different variants and extensions have been proposed for modeling several real–world applications. For example, the *Multi–Depot* VRP (Salhi et al., 2014, MDVRP), which considers two or more depots that must attend customer demands through their vehicles. Another VRP extension is the one that considers a heterogeneous fleet (Baldacci et al., 2008, HVRP). The *Split Delivery* VRP (Archetti et al., 2008a, SDVRP), allows that the demand of each customer is met by two or more vehicles. The VRP *with Time Windows* (Lau et al., 2003, VRPTW) assumes that deliveries to each customer must be attended within a certain time interval, which varies among customers. In *Dynamic* VRPs (Pillac et al., 2013; Albareda-Sambola et al., 2014, DVRP), some delivery orders are known but during the day new orders arrive, which must be incorporated into the current plan. Additionally, there are other interesting research areas that study VRPs with multiple decision criteria called *multiobjective* VRPs (Alabaş-Uslu, 2007; García Calvillo, 2010), which involve more than one objective to be optimized simultaneously. In practice, several elements of the input data of VRPs may be uncertain (Zhang et al., 2013). On the other hand, VRPs *with Pickup and Delivery* (Côté et al., 2012; Hernández-Pérez et al., 2016, VRPPD) allow routes with two different types of customers: customers who require loading product and customers who require unloading product. The book by Toth and Vigo (2014) overviews a good number of VRPs that have been studied by different authors.

This thesis focuses on a VRP variant that considers periodic customers demands throughout a specific time horizon. Some of the problems related to the one that we study are overviewed below.

2.2 Vehicle Routing Problems with periodic demand

Some real–world VRP applications require to serve the customers demand throughout a given time horizon. The vehicle capacity constraints as well as inventory limitations at the customers locations usually suggest to address this type of problems resorting to some periodicity in the visits to customers. Periodic routing problems have been studied for over forty-three years since the idea of introducing periodicity was first proposed by Beltrami and Bodin (1974).

According to Campbell and Wilson (2014), the most common applications of routing problems with periodicity can be classified in terms of how customers demands are satisfied, i.e., by *picking up a product* (garbage, recyclable, wastes, autoparts, oil, factory goods, etc.), by *delivering it* (groceries, blood, vending machines, hospitals, etc.), or by *giving on-site service* (maintenance of equipment, home health care, quality inspectors, etc.). Two comprehensive surveys that show such applications are Francis et al. (2008) and Campbell and Wilson (2014).

Three important groups of VRPs with periodic demand, which are related to the problem studied in this thesis are the classical *Periodic* VRP (PVRP), the PVRP *with Service Choice* (PVRP–SC) and the *Inventory Routing Problem* (IRP).

2.2.1 The Periodic Vehicle Routing Problem

Five years after the seminal work of Beltrami and Bodin (1974), Russell and Igo (1979) used the name "Assignment Routing Problems" to refer to this family of problems. Christofides and Beasley (1984) used the name "Period Routing Problem", and provided the first mathematical formulation, while the current name "Periodic Vehicle Routing Problem" was coined by Gaudioso and Paletta (1992).

The PVRP (Campbell and Wilson, 2014) is a generalization of the classical VRP in which vehicle routes must be constructed over a given time horizon using predefined schedules that indicate the time periods when customers should be visited. Feasible schedules for a given customer reflect the frequency with which the customer should be visited according to its service demand. Each day of the time horizon vehicles travel along routes starting and ending at a specific depot depending on the selected schedule. According to Christofides and Beasley (1984), three important considerations must be taken into account in PVRPs: define a schedule for the set of customers, assign customers to vehicles and find the best route that each vehicle must take in order to serve their demands. Depending on the frequency of visits to a given customer, a fraction of its total demand will be delivered at each time visit. The aim is to select a feasible schedule for each customer, and to find a set of routes that minimize the total travel cost satisfying vehicle capacity and customer visit requirements.

Several extensions of the original PVRP have been addressed over the last forty-three years: *Periodic* TSPs (PTSP), PVRPs *with Time Windows* (PVRPTW), *Multi-depot* PVRPs (MDPVRP), PVRPs *with Intermediate Facilities* (PVRP-IF), and many others.

2.2.2 The Periodic Vehicle Routing Problem with Service Choice

The PVRP–SC is a variation of the PVRP in which the visit frequency is a decision variable and it is allowed to visit customers more often than their predefined frequencies. This problem was proposed and formulated as an integer programming (IP) formulation by Francis et al. (2006) and a continuous approximation model was presented by Francis and Smilowitz (2006). In Francis et al. (2006) the objective function combines routing and service decisions. They propose a solution method based on a Lagrangian relaxation of the formulation, which allows to obtain tight lower and upper bounds. If the bounds do not coincide, a branch and bound method is applied to close the gap between them. A variation of this algorithm is used as a heuristic in order to obtain high quality solutions to large instances. In Francis and Smilowitz (2006) the authors show that the proposed continuous approximation is useful using a set of benchmark instances.

2.2.3 The Inventory Routing Problem

The IRP (Coelho et al., 2013) is a VRP with periodic demand that includes inventory management and delivering–scheduling decisions, which depend on the inventory levels at each time period. This family of problems was introduced by Bell et al. (1983) as the *Vendor–Managed Inventory* (VMI). Three decisions must be made: when to serve customers, the amount of product to deliver at each visit, and the design of the service routes at each day of the time horizon. The aim is to minimize the total inventory holding cost plus the total routing cost.

According to Coelho et al. (2013), variants of the IRP allow alternative replenishment or inventory handling policies, or different criteria related to inventory and routing decisions. Two common replenishment policies are: the Maximum–Level policy and the Order–Up–to level policy. In the first one the quantity to replenish can be any amount that does not exceed the capacity available per customer, while in the second one the total inventory capacity level of each visited customer is filled at each visit. The inventory handling policies include alternative options like allowing (or not) stock–out, avoiding negative inventory (back–orders), or undelivered demand. Archetti et al. (2014) analyzed and evaluated different formulations and valid inequalities for a *Multi–Vehicle* IRP. An extensive review of the IRP literature is given in Coelho et al. (2013).

Different matheuristics have been used for solving large size IRP instances. According to Bertazzi and Speranza (2012b), some of the most common matheuristics for IRPs correspond to the following classification:

- *Routing–based*: The aim is to minimize only the routing costs. However, considering only routing costs may result in solutions of low quality when inventory costs are added to the final solution.

- *Inventory–first routing–second*: First, a subproblem that focuses on the inventory criterion is solved. Then routing is obtained by solving a TSP for each time period.
- *Cluster–first inventory–routing second*: First, customers are grouped into segments, then a small inventory–routing model is solved to optimality for each segment.
- *Intensified TS*: Combines a TS scheme with MILP formulations to intensify the search of the solution space.

2.3 State of the art for Periodic Vehicle Routing

Several solution methods have been proposed for solving different classes of PVRPs. Given the difficulty of exact methods for finding optimal solutions for large size instances, most of them are approximate approaches. The most relevant methods use sophisticated techniques to obtain high quality solutions. Those considered state of the art are shown in Figure 2.1 and are explained below according to their year of publication.

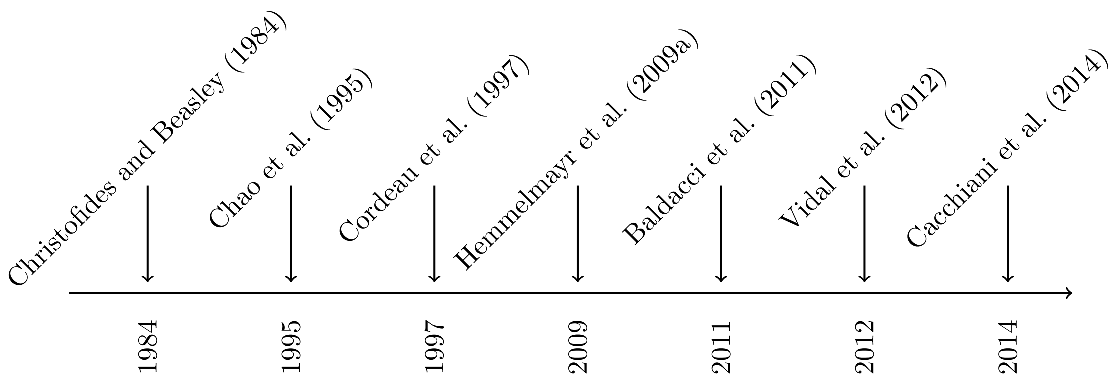


Figure 2.1: State-of-the-art approaches for PVRPs.

Christofides and Beasley (1984) provided the first IP formulation for the PVRP. However, this formulation was not used for solving the problem. Instead, a heuristic was used to initially assign visit days to customers and then the VRP resulting for each time period was solved.

Chao et al. (1995) proposed a heuristic consisting of an initialization, an improvement phase and a feasibility recovery step. In the initialization phase, an IP formulation is solved in order to obtain the assignment of customers to schedules. Then, in the second phase, a record to record improvement heuristic is applied to obtain better solutions. If the final solution is infeasible a procedure is applied to recover feasibility. In order to analyze its effectiveness, the heuristic was applied to existing benchmark instances and to new instances generated by the authors. In particular, the heuristic improved the best–known solutions for most of the benchmark instances tested.

Cordeau et al. (1997) proposed a TS algorithm for solving three generalizations of PVRPs. This algorithm generates an initial solution in which feasibility is not required. The initial solution is then modified by applying different moves and by avoiding recent ones. The heuristic was tested with the PVRP instances used by Chao et al. (1995). The results showed that in 24 out of the 32 considered instances, the proposed TS produced better solutions than the best–known results of the literature. In Cordeau and Maischberger (2012) this TS heuristic was improved by adding a local search as well as diversification and parallelization tools.

Hemmelmayr et al. (2009a) developed a hybrid Variable Neighborhood Search (VNS) heuristic for solving the PVRP. This VNS generates a new solution from an initial random solution by applying different neighborhoods sequentially. Then, a 3–opt local search is applied in order to further improve it. A worse solution can be accepted with a certain probability using a Simulated Annealing (SA) criterion. This algorithm produced 24 new best–known solutions for the considered PVRP benchmark instances.

Concerning exact solution methods, the algorithm of Baldacci et al. (2011) is considered as state of the art for the classical PVRP and two of its generalizations. In this algorithm, three relaxations of a set-partition formulation are solved in order to obtain tight lower bounds of the problem. Moreover, five bounding procedures are developed, outperforming some of the best–known upper bounds of Hemmelmayr et al. (2009a) and producing very high quality solutions, which, on average, have a 1% deviation with respect to the obtained lower bounds.

Vidal et al. (2012) proposed a metaheuristic based on a Genetic Algorithm (GA) and a local search. This GA operates with both feasible and infeasible solutions. This heuristic was tested using benchmark PVRP instances used in Baldacci et al. (2011) improving 20 of them within 0.02% of optimality precision.

Cacchiani et al. (2014) developed a hybrid optimization algorithm for solving the

PVRP. This algorithm integrates the solution of a MILP within a heuristic framework. A relaxation of a set-covering formulation of the PVRP is solved by column generation and then a local search is applied considering a fix–and–relax procedure and a TS heuristic. Results on benchmark instances showed that good quality solutions can be obtained by the proposed algorithm. Some best–known solutions of the literature were improved.

2.4 Flexible service policies in Periodic Vehicle Routing

According to Francis et al. (2008) flexible service policies may produce great savings in the routing cost of a PVRP. Several works have studied these policies in PVRPs in order to improve the solutions in comparison with those obtained with standard models in the literature.

For example, Rusdiansyah and Tsao (2005) integrated IRP and PVRPTW features to solve a problem for the delivery of products in vending–machine supply chains. They developed a mathematical formulation that combined both inventory and periodic routing with the difference that the objective function attempts to minimize routing, inventory holding and visit frequency costs. However, due to the complexity of the problem, they developed four variants of heuristics to solve it. These heuristics were evaluated and the obtained results were compared with the best–known PVRPTW solutions in the literature. They were able to obtain relevant savings for most of the instances through the incorporation of inventory and vehicle routing decisions in their model.

Francis et al. (2006) introduced the PVRP–SC. In the PVRP–SC flexibility in service frequency is considered as a decision of the model. The authors proposed a mathematical formulation and an exact solution algorithm. Computational results showed that adding service choice (flexibility in visit frequency) can improve the system efficiency. In order to compare the quality of the final solution, Francis and Smilowitz (2006) evaluated their formulation considering different service levels. They also compared the advantages and disadvantages of considering the service choice term in the objective function. They noticed that savings are greater when customers with high frequency of visits are closer to the depot. In general, their results showed that adding service choice can help to provide better designs of service options.

Francis et al. (2007) developed a TS for a PVRP that incorporates different operational flexibility options: flexible service choice, crew flexibility, greater number of schedule options, and delivery strategies. They analyzed the trade-off between operational flexibility and operational complexity through the incorporation of a set of quantitative measures. They concluded that adding operational flexibility increases the operational complexity to find the solutions, that the location of customers affects the savings obtained, and that the reduction of the crew flexibility reduces the operational complexity. Their proposed TS produced solutions within 3% of optimality for instances from the literature.

Hemmelmayr et al. (2009b) developed several solution approaches based on an IP formulation and VNS to evaluate delivery strategies for blood products supplies. For their IP formulation they combined IRP and PVRP features and considered two alternative delivery strategies: *regionalization* (creation of regions with fixed routes) and *delivery regularity* (repeating delivery patterns for each hospital). The aim was to minimize the traveling costs. The results of the computational experiments showed that allowing more flexible strategies it is possible to obtain about 30% of savings.

Pacheco et al. (2012) proposed a MILP formulation and a two–phase method based on a Greedy Randomized Adaptive Search Procedure (Feo and Resende, 1995, GRASP) and Path Relinking (Glover et al., 2000, PR) for a real–world problem of a bakery company. They addressed the problem as a generalization of the Capacitated Vehicle Routing Problem (Ralph et al., 2003, CVRP) because preliminary experiments showed that modeling the problem as a PVRP was more complex to handle. They introduced flexibility on the delivery dates in their approximate method and solutions were compared with those produced by state–of–the–art metaheuristics. Their results showed that adding flexibility to their model made it possible to obtain high quality solutions (about 20% of reduction of the total traveling costs in real–world instances) in much less time than other solution methods proposed in the literature.

Aksen et al. (2012) proposed two different flow commodity formulations that combined PVRP and IRP features for a waste vegetable oil collection problem. The aim was to minimize the total collection, inventory and purchasing costs. Neither of the proposed formulations assumed fixed visit frequencies or predetermined schedules. Some valid inequalities were proposed to strengthen them obtaining about 3.28% optimality gaps on average on small size instances.

Archetti et al. (2015) studied a multi–period vehicle routing problem in city logistics where customers have to be served considering due dates. In this work, alternative flow-based and load-based formulations that combine PVRP and IRP features are proposed and the benefit of incorporating flexibility in the due dates and in the number of vehicles is analyzed. Results showed that the load-based formulation outperforms the flow-based formulation and more savings can be obtained when due dates are extended.

Table 2.1 shows the relation among works that include flexibility in the models that are addressed. Other works found in literature where flexibility criteria are applied, can be found in Hashimoto et al. (2006), and Há et al. (2014).

Reference	Application	Type of flexibility	Model and solution method	Objective (Min)
Rusdiansyah and Tsao (2005)	Vending-machine supply chains	Visit frequency is a decision variable	IRP model based on PVRPTW and five heuristics	Sum of the average inventory holding and travel costs
Francis et al. (2006)	Library deliveries	Visit frequency is a decision variable	PVRP-SC: exact algorithm and heuristic variation of the exact method	Total travel cost plus service benefit
Francis and Smilowitz (2006)	Periodic distribution with Service Choice	Visit frequency is a decision variable	Continuous PVRP-SC reduced by geographic decomposition and variable substitution	Total travel cost plus service benefit of each subregion
Francis et al. (2007)	Periodic distribution	Visit frequency, crew flexibility, schedule options, delivery strategy	PVRP embedded in a Tabu Search method	Total travel cost plus service benefit
Hemmelmayr et al. (2009b)	Blood product supplies	Routing decisions: regions/fixed routes and delivery regularity	IP formulation based on IRP, a basic heuristic and a Variable Neighborhood Search	Total traveling cost
Pacheco et al. (2012)	Bakery company	Dates of delivery	CVRP: metaheuristic (GRASP & Path Relinking)	Total distance traveled
Aksen et al. (2012)	Waste vegetable - oil collection	Visit frequency is not fixed nor a limited number of predetermined schedules is assumed	Two MILPs based on IRP and PVRP and partial linear relaxations to generate lower bounds	Total transportation costs, vehicle operation costs, holding costs, and purchasing costs
Archetti et al. (2015)	City logistics	Due date, crewsize, vehicle capacity	Three formulations reinforced with valid inequalities: Flow based formulation (FF), FF with assignment variables and load–based formulation	Transportation costs, inventory costs and penalty costs for postponed service

Table 2.1: Literature review of flexible service policies in periodic delivery operations.

Chapter 3

The Flexible Periodic Routing Problem

The *Flexible Periodic Vehicle Routing Problem* can be seen as a new generalization of the PVRP that allows service policies that are flexible with respect to the frequency of visits and the amount delivered at each visit. The motivation for this research stems from the variety of real–world PVRP applications that have been constrained to deliver for each customer a fixed amount of product with a pre-established frequency. As it will be seen, allowing flexible service policies for these two criteria may produce considerable operational savings, both theoretically and empirically. The general aim of this research is to provide a suitable mathematical and algorithmic framework for PVRPs in which flexible distribution plans are allowed to improve the quality of the final solutions.

This chapter is organized as follows. The FPVRP, the PVRP and the IRP are formally defined in Section 3.1 and their relation is described in Section 3.2. Two illustrative examples are given in Section 3.3 to highlight the main advantages of FPVRPs over PVRPs and IRPs. These potential advantages are formalized in Section 3.4 with a theoretical worst-case analysis for the savings that can be obtained with the FPVRP with respect to these related problems.

3.1 Formal definition and related problems

3.1.1 The Flexible Periodic Vehicle Routing Problem

Consider a complete and directed network $G = (N, A)$ with set of nodes $N = \{0\} \cup C$ and set of arcs A. Node $\{0\}$ denotes the depot and $C = \{1, \ldots, n\}$ the set of customers. Let $T = \{1, \ldots, H\}$ be a discrete set of time periods. Each customer $i \in C$ has a total demand W_i over T and a storage capacity w_i. A homogeneous fleet of vehicles $K = \{1, \ldots, m\}$, with capacity Q is available to perform the service. In order to satisfy the customers demand a distribution plan must be defined, indicating the quantity of product to be delivered to each customer at each time period. The quantity delivered to customer $i \in C$ at each visit cannot be greater than w_i and the sum of the quantities delivered over T must be equal to W_i. A cost $c_{ij} \geq 0$ is associated with each arc $(i, j) \in A$ and is paid every time a vehicle traverses the arc.

The FPVRP is the problem of finding the quantity to be delivered to each customer at each time period that, together with the set of routes that satisfy customer demands at the end of the time horizon, minimize the total routing cost.

3.1.2 The Periodic Vehicle Routing Problem

In the PVRP, as defined in Christofides and Beasley (1984), a set of schedules, S, is given. Each schedule consists of a set of days in which customers receive service. This implies that each customer will be visited, receiving the same amount of product w_i at each visit, in every day of the schedule, i.e., $S_i = \{s \in S : \sum_{t \in T} a_{st} = f_i\}$ where S_i is the schedule chosen for customer i, f_i is the visit frequency for customer i, and

$$a_{st} = \begin{cases} 1 \text{ if day } t \in T \text{ belongs to schedule } s \in S, \\ 0 \text{ otherwise.} \end{cases}$$

Three sets of decisions have to be made: select a schedule from all the predefined options for each customer, assign customers to vehicles, and generate the routes to be performed in each period of the time horizon.

The PVRP is the problem of selecting a schedule for each customer and finding a set of routes consistent with the selected schedules of minimum total routing cost.

The relation between the FPVRP and the PVRP can be established by defining the storage capacity w_i as the ratio between the total demand W_i and the *expected frequency* of visits f_i as defined in the PVRP. Thus, $w_i = \frac{W_i}{f_i}$. This way, in the FPVRP customer i has to be visited at least f_i times (the frequency defined in the PVRP). The customer may be visited more frequently if this leads to cost savings.

3.1.3 The Periodic Vehicle Routing Problem with Service Choice

The PVRP–SC is a generalization of the PVRP in which service frequency is a decision of the model. Similarly to the FPVRP, the PVRP–SC is defined on the same network G and the service frequency f_i is a lower bound on the number of days that a customer $i \in C$ must be visited. However, the amount of product to deliver to each customer at each visit is established by the selected schedule, which comes from the set S of possible options known beforehand. The number of days for each schedule $s \in S$ is denoted by γ^s. A stopping cost τ_i^s is defined for each customer $i \in C$ and each schedule $s \in S$. Furthermore, a service benefit α^s and a demand accumulation factor β^s, $s \in S$, are defined.

The aim of the PVRP–SC is to find a set of routes, for each vehicle and each time period, that minimizes the total travel cost plus the service benefit, satisfying the vehicle capacities and the minimum service requirements.

3.1.4 The Inventory Routing Problem

The IRP is defined on the same network G as the FPVRP. The difference with respect to the FPVRP setting is that customers are no longer associated with a total demand W_i. Instead, a demand d_i^t is defined for each customer $i \in C$ and each time period $t \in T$. Moreover, a starting inventory level I_i^0 is associated with each customer, together with a capacity w_i. The distribution plan has to be such that each customer is able, at each time period, to satisfy the demand d_i^t, thus the customer must have a sufficient quantity I_i^t in inventory. Moreover, the quantity delivered at each visit plus the inventory available when the visit is performed should not exceed the capacity w_i. Similarly to Archetti et al. (2014) it is assumed that, at each customer, the inventory level at time $t \in T$ is the inventory level at time $t - 1$ plus the amount delivered at time t minus the amount consumed at time t. No shortages are allowed.

The aim of the IRP is to determine the quantity of product to deliver to each customer and the corresponding service routes, guaranteeing that there is no shortage at each customer in each time period, of minimum total routing cost.

Note that the IRP, as defined in Bertazzi and Speranza (2012a, 2013) and Coelho et al. (2013), includes inventory holding costs in the objective function and inventory constraints at the supplier. In order to have a fair comparison with the FPVRP, none of these elements will be considered from now on. Such a version of the IRP will be referred to as the FPVRP with Inventory Constraints (FPVRP-IC).

3.2 Relationship among periodic routing problems

PVRPs, PVRPs–SC and IRPs share some characteristics: they perform periodic visits to customers in order to deliver a certain quantity of product along a time horizon. These visits incur some costs mostly related to the routing of the vehicles. There are however important differences among these classes of problems as well.

In the PVRP, every customer must be visited with a known periodicity on a specific time horizon. This periodicity (or frequency) must be chosen from a set of schedule plans which are known initially. At each visit the quantity delivered is exactly the same, according to the selected schedule. In the PVRP–SC, the frequency of visits is modeled as a decision variable. In particular, it is allowed to visit customers more often than a minimum predefined frequency. However, the PVRP–SC still depends on a previously known reference schedule for each customer, which determines the amount of product to deliver to each customer at each time visit. A service benefit, which mainly depends on the demand of each customer, is considered to determine the solution cost. On the other hand, the IRP incorporates inventory management and a distribution route design decisions simultaneously. It is not based on a predefined schedule, and customers are visited according to their replenishment policy. In this type of problems, the frequency of visits is implicit and there is no minimum service requirement as in the PVRP–SC. The amount of product to deliver to each customer depends on the customer inventory at that time and it is modeled as a decision variable. Table 3.1 summarizes the main differences among these three types of problems.

Problem	Periodicity	Delivered quantity	Objective
PVRP	Predefined set of schedules	Same quantity at each visit	Minimize routing cost
PVRP–SC	Predefined schedule, visit frequency modeled as a decision variable	Depends on the selected schedule and the delivery strategy at each customer	Maximize net profit: balance between service benefit and routing cost
IRP	No predefined schedule and unconstrained number of visits	Modeled as a decision variable. Depends on the replenishment policy	Minimize holding plus routing costs

Table 3.1: Comparison among PVRP, PVRP–SC and IRP.

3.3 Illustrative examples

In this section we present two examples that motivate the study of the FPVRP and show the potential savings that can be obtained with respect to both the PVRP and the IRP. The first example refers to the comparison between the FPVRP and the PVRP while the second compares the FPVRP and the FPVRP-IC.

Example 1. *Consider a PVRP instance with a number of customers* $|C| = 4$*, a time horizon* $|T| = 6$*, a vehicle capacity* $Q = 8$ *and a fleet of* $|K| = 2$ *vehicles. Suppose that distances are as indicated in Figure 3.1 where* $\epsilon \ll M$ *and* $\alpha \ll M$*. In addition, for each* $i \in C$*, a total demand* $W_i = 12$ *and a number of visits* $f_i = 2$ *are assigned.*

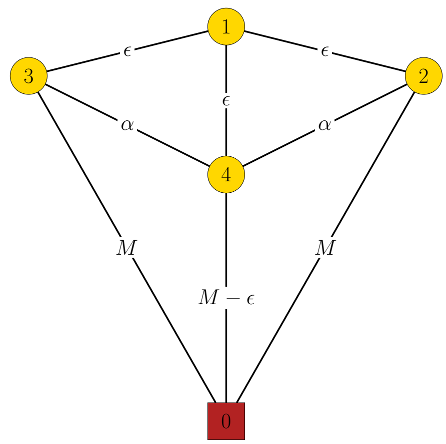


Figure 3.1: Input network for Example 1.

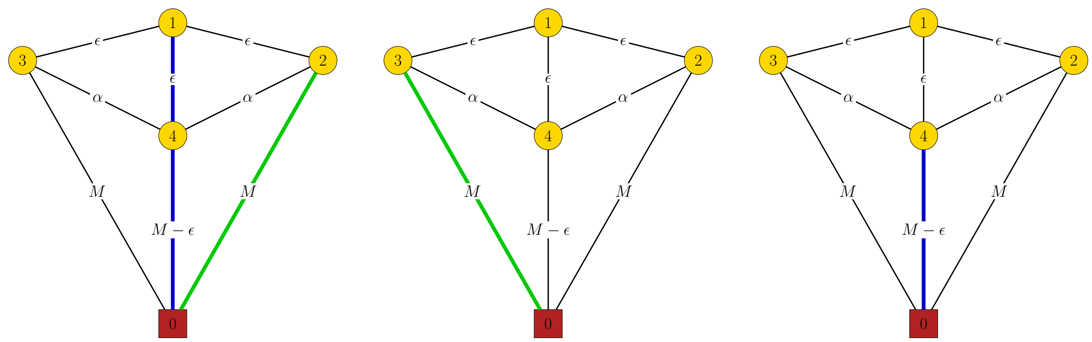
In the classical PVRP each customer has to be visited every third day and the quantity to be delivered at each visit is $w_i = \frac{12}{2} = 6$*. The set of feasible schedules for each customer is* $S_i = \{(1,4), (2,5), (3,6)\}$*,* $i \in C$*. Since the capacity of each vehicle allows to serve one customer per route, any solution minimizing the number of vehicles is optimal for this PVRP instance. The minimum number of routes is* $(4 \times 1) + (2 \times 2) = 8$ *(4 days with one route and 2 days with two routes). An optimal PVRP solution is the one that is shown in Figure 3.2 and is described below:*

3.2a ***Periods 1 and 4:*** *One vehicle delivers 6 units to Customer 1 and another vehicle delivers 6 units to Customer 2. The traveled distance is* $8M$ *(*$2M$ *per vehicle at each time period).*

3.2b ***Periods 2 and 5:*** *One vehicle delivers 6 units to Customer 3. The traveled distance is* $4M$ *(*$2M$ *for each time period).*

3.2c ***Periods 3 and 6:*** *One vehicle delivers 6 units to Customer 4. The traveled distance is* $4M - 4\epsilon$

The overall distance traveled by the 8 vehicles is $16M - 4\varepsilon$*.*



(a) $S_1 = S_2 = \{(1,4)\}$, $q_1^{1,4} = q_2^{1,4} = 6$, $Dist = 2 \times (2 \times 2M.)$ (b) $S_3 = \{(2,5)\}$, $q_3^{2,5} = 6$, $Dist = 2 \times 2M$ (c) $S_4 = \{(3,6)\}$, $q_4^{3,6} = 6$, $Dist = 2 \times (2M - 2\epsilon)$

Figure 3.2: Optimal PVRP solution. Total distance traveled: $16M - 4\epsilon$.

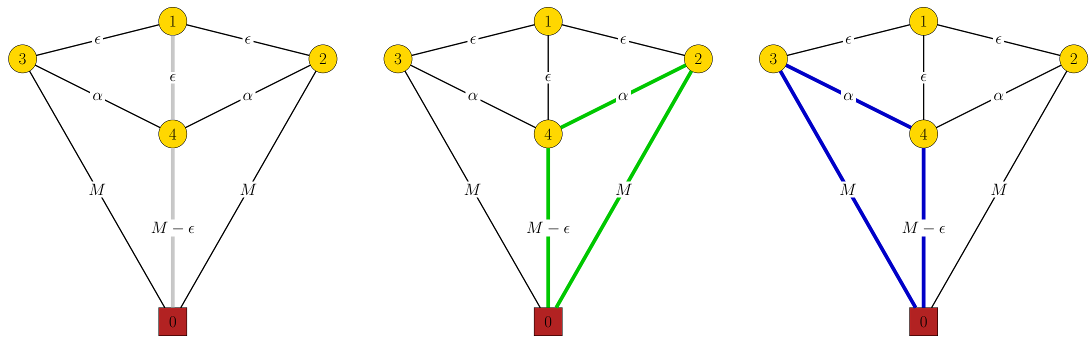
For the FPVRP, the following schedule is optimal:

3.3a ***Periods 1 and 4:*** *One vehicle delivers 6 units to Customer 1 and 2 units to Customer 4. The traveled distance is* $4M$ *(*$2M$ *at each time period).*

3.3b ***Periods 2 and 5:*** *One vehicle delivers 6 units to Customer 2 and 2 units to Customer 4. The traveled distance is* $4M+2\alpha-2\varepsilon$ *(*$2M+\alpha-\varepsilon$ *at each time period).*

3.3c ***Periods 3 and 6:*** *One vehicle delivers 6 units to Customer 3 and 2 units to Customer 4. The traveled distance is* $4M+2\alpha-2\varepsilon$ *(*$2M+\alpha-\varepsilon$ *at each time period).*

The total traveled distance is $12M + 4(\alpha - \varepsilon)$*, which is much smaller than* $16M - 4\varepsilon$ *when* $\epsilon \ll M$ *and* $\alpha \ll M$*. Figure 3.3 shows the FPVRP solution.*



(a) $S_1 = S_4 = \{(1,4)\}$, $q_1^{1,4} = 6, q_4^{1,4} = 2$, $Dist = 2(2M)$

(b) $S_2 = S_4 = \{(2,5)\}$, $q_2^{2,5} = 6, q_4^{2,5} = 2$, $Dist = 2(2M + \alpha - \epsilon)$

(c) $S_3 = S_4 = \{(3,6)\}$, $q_3^{3,6} = 6, q_4^{3,6} = 2$, $Dist = 2(2M + \alpha - \epsilon)$

Figure 3.3: Optimal FPVRP solution. Total distance traveled: $12M + 4\alpha - 4\epsilon$.

Example 2. *Consider now a FPVRP-IC instance with $|C| = 5$ customers, time horizon $|T| = 3$, a vehicle capacity $Q = 228$ and a fleet of $|K| = 2$ vehicles. Distances between each pair of nodes $i, j \in N$ are shown in Figure 3.4. It is assumed that each customer $i \in C$ has a fixed demand at all time periods, i.e., $d_i^t = d_i$ for all $t \in T$. Information about the demand per period (d_i), the initial inventory levels (I_i^0) and the maximum inventory levels at each customer (w_i) is presented in Table 3.5. Note that for the FPVRP $W_i = d_i^t \times |T| - I_i^0$.*

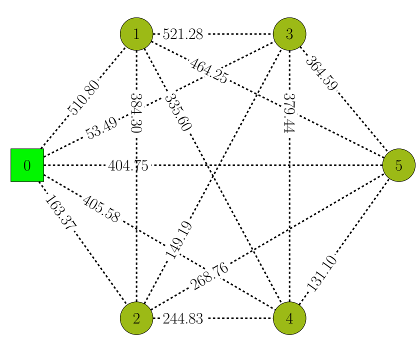


Figure 3.4: Input network for Example 2.

i	d_i	I_i^0	w_i	W_i
1	87	87	174	174
2	86	86	172	172
3	65	65	130	130
4	53	106	159	53
5	13	26	39	13

Table 3.5: Initial data for Example 2.

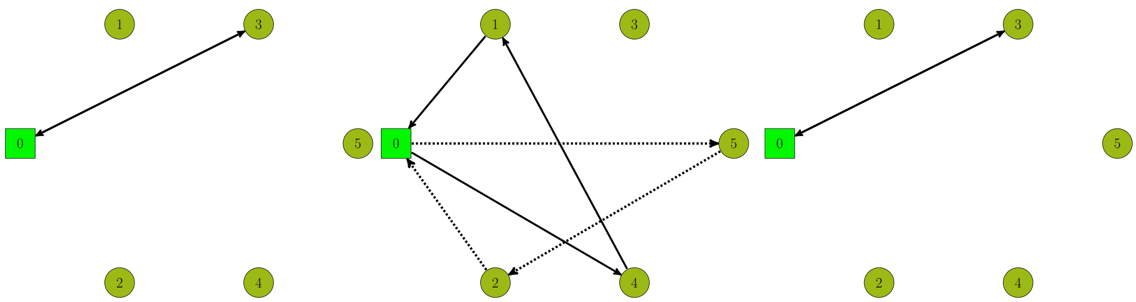
An optimal FPVRP-IC solution for this example is:

3.6a ***Period 1:*** *One vehicle delivers 65 units to Customer 3. The traveled distance is* 106.98*. The customers inventory levels are* $I_i^1 = \{0, 0, 65, 53, 13\}$.

3.6b ***Period 2:*** *One vehicle delivers 13 units to Customer 5 and 172 units to Customer 2. Another vehicle delivers 53 and 174 units to Customers 4 and 1, re-*

spectively. The traveled distance is 2088.86*. The customers inventory levels are* $I_i^2 = \{87, 86, 0, 53, 13\}$.

3.6c ***Period 3:*** *One vehicle delivers 65 units to Customer 3. The traveled distance is* 106.98*. The final inventory levels are all zero.*



(a) Period 1: Dist 106.98 (b) Period 2: Dist 2088.86 (c) Period 3: Dist 106.98

Figure 3.6: Optimal FPVRP-IC Solution. Total distance traveled: 2302.82.

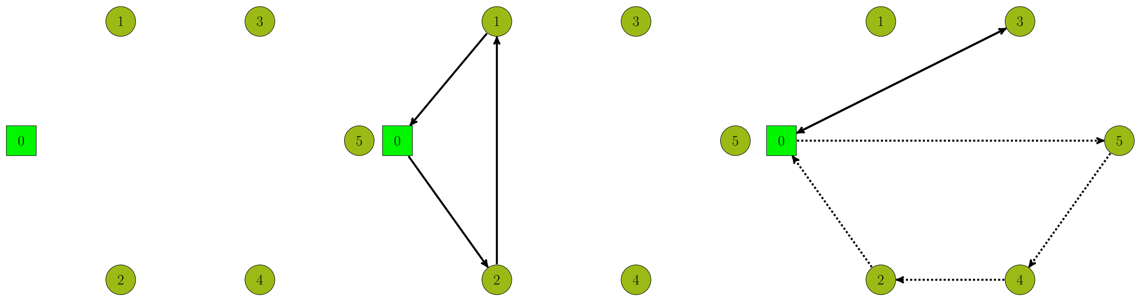
The total distance of the optimal FPVRP–IC solution is 2302.82.

If the FPVRP associated with this instance is considered, then $W_i = d_i^t \times |T| - I_i^0$ *as inventory levels are no longer considered at the customers. Then, an optimal solution for the FPVRP is as follows:*

3.7a ***Period 1:*** *No deliveries are performed. The traveled distance is 0.*

3.7b ***Period 2:*** *One vehicle is used to deliver 10 units to Customer 2 and 174 units to Customer 1. The traveled distance is* 1058.47*.*

3.7c ***Period 3:*** *One vehicle delivers 130 units to Customer 3 and another vehicle delivers 13, 53 and 162 units to Customers 5, 4 and 2, respectively. The distances traveled by the vehicles are* 106.97 *and* 944.05*, respectively.*



(a) Period 1: Dist 0 (b) Period 2: Dist 1058.47 (c) Period 3: Dist 1051.03

Figure 3.7: Optimal FPVRP Solution. Total distance traveled 2109.51.

The total distance of the optimal FPVRP solution is 2109.51. Therefore, in this example, the FPVRP produces an improvement of more than 8% *with respect to the FPVRP-IC (Cost = 2302.82).*

3.4 Worst-case analysis

In the examples of Section 3.3 it was shown that substantial savings can be obtained by the FPVRP with respect to both the PVRP and the FPVRP-IC. This section quantifies and formalizes the maximum potential savings in each case. Let $z(P)$ denote the optimal value to a given instance of problem P.

Theorem 1. *There exists no finite bound for the ratio* $\frac{z(PVRP)}{z(FPVRP)}$.

Proof. Consider the following instance of the PVRP in which $|T| = 2$, $|K| = Q$ and $f_i = 1$ for each customer i. There are three sets of customers. The first set is composed by Q customers with $w_i = W_i = Q$. In the second set there are Q customers with $w_i = W_i = Q - 1$ and the third set have Q customers with $w_i = W_i = 1$. Moreover, all customers can be visited either in $t = 1$ or in $t = 2$. Customers and depot locations are as follows. Each customer in the first set is co-located with a customer of the second set and they are spread around a circle centered at the depot with radius $\delta < 1$ and a distance ϵ apart. Customers of the third set are spread around a circle centered at the depot with a radius 1, perfectly aligned along the radius with the customers of the first and second set, with a distance $\frac{\epsilon}{\delta}$ apart (Figure 3.8a). Note that, as the fleet is composed by Q vehicles and $|T| = 2$, the maximum number of routes during T is $2Q$. Moreover, as the total demand of customers is $2Q^2$, all vehicles must be used and fully loaded in both periods.

The optimal and only solution of the PVRP is the one where Q routes are used to serve the customers in the first set with direct trips to the depot and the other Q routes serve one customer in the second set and one customer in the third set each. The cost of this solution is $2Q\delta + 2Q$ (Figure 3.8b).

The optimal solution of the FPVRP is the following (Figure 3.8c). In $t = 1$ one route is used to serve all customers in the third set. There remains $Q-1$ routes in $t = 1$ and Q routes in $t = 2$. They are constructed as follows. Without loss of generality, choose one customer of the second set as the first customer. Number all customers in a clockwise direction as $i_1, ..., i_{|Q|}$ if they belong to the second set and $j_1, ..., j_{|Q|}$ if they belong to the first set. The first route delivers $Q-1$ units to customer i_1 and 1 unit to customer j_1. The second route delivers $Q-1$ units to customer j_1 and 1 unit to customer i_2. The third route delivers delivers $Q-2$ units to customer i_2 and 2 units to customer j_2, and so on. We obtain $Q+Q-1$ routes where the last route delivers Q units to customer $j_{|Q|}$. The odd routes are performed in $t = 2$ while the even routes are performed in $t = 1$ (as no customer can be served more than once in the same period). The cost of this solution is $2 + (Q-1)(\frac{\epsilon}{\delta}) + 2Q\delta + (Q-1)(2\delta + \epsilon)$.

The ratio between $z(PVRP)$ and $z(FPVRP)$ is therefore $\frac{2Q\delta+2Q}{2+(Q-1)(\frac{\epsilon}{\delta})+2Q\delta+(Q-1)(2\delta+\epsilon)}$. When Q goes to infinity and ϵ, δ and $\frac{\epsilon}{\delta}$ go to 0 this ratio tends to infinity.

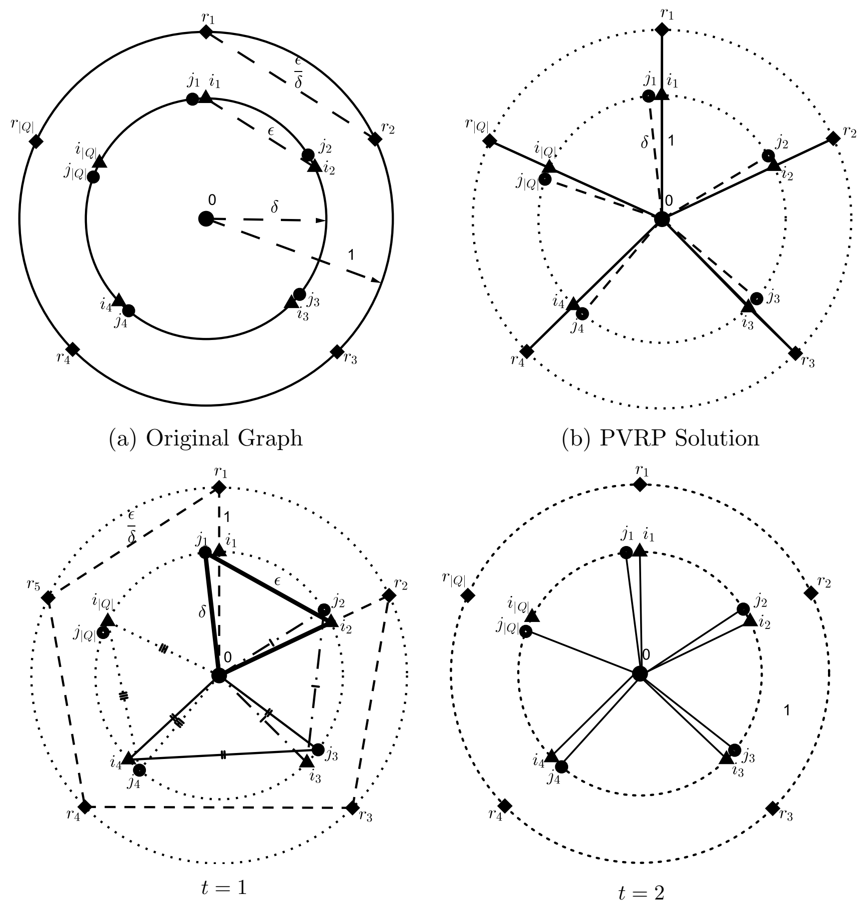


Figure 3.8: Worst–case analysis

■

The proof presented above is quite similar to the one proposed by Gueguen (1999) for the analysis of the benefits of the SDVRP with respect to the VRP.

We now compare the FPVRP with the FPVRP-IC as defined in Section 3.1.4. The following result holds.

Theorem 2. *There exists no finite bound for the ratio* $\frac{z(FPVRP-IC)}{z(FPVRP)}$.

Proof. Let us consider the instance introduced in the proof of Theorem 1. In the FPVRP-IC, initial inventory levels at customers must be defined. We define them as follows: the initial inventory level is equal to 0 for the customers in the first set, to $Q-1$ for the customers in the second set and to 1 for the customers in the third set. The customers demands are equal to $Q-1$ and 1, at time $t=1$ and $t=2$, for customers of the second and third set, respectively. For the customers in the first set, demand at time $t = 1$ is equal to Q and demand at $t = 2$ is equal to 0. Note that the customers of the second and third set cannot be served at time $t = 1$ as their initial inventory level is equal to the storage capacity. On the other side, customers of the first set have to be served at time $t = 1$ as their initial inventory level is 0 and the demand at $t = 1$ is positive. Thus, the only feasible solution for the FPVRP-IC corresponds to the solution of the PVRP shown in the proof of Theorem 1, i.e., Q routes are used to serve the customers in the first set with direct trips to the depot at time $t = 1$ and Q routes serve one customer in the second set and one customer in the third set each at time $t = 2$. The cost of this solution is $2Q\delta + 2Q$. The solution of the FPVRP does not change. Thus the ratio is the same and tends to infinity.

■

Chapter 4

Mixed Integer Linear Programming formulations

Traditional VRP formulations with multiple vehicles use decision variables with a vehicle index to indicate the arcs traversed by each vehicle. This involves a high number of decision variables, particularly in problems where decisions must be made at different periods of a given time horizon, like those studied in this thesis. In order to mitigate this difficulty, recent works on several VRP variants have proposed the use of the so-called load-based formulations, in which decision variables identify the arcs used in the solutions without making explicit the vehicles that traverse them (Letchford and Salazar-González, 2015; Archetti et al., 2014). For this, an additional set of continuous commodity flow variables is needed to guarantee that routes are properly defined. Such formulations tend to be quite effective in practice, although their linear programming (LP) relaxations are usually weaker than their *traditional* counterpart. On the one hand they have a smaller number of variables. On the other hand their implementation does not require the use of sophisticated techniques, like branch-and-cut or column generation.

In this chapter, a vehicle-index and a load-based MILP formulations are proposed for the FPVRP. Load-based formulations are also presented for the FPVR-IC and the PVRP, since they have been used in the computational experiments, for comparative purposes. Alternative vehicle-index formulations for the FPVRP–IC and the PVRP, are given in the Appendices.

4.1 Vehicle-index formulation for the FPVRP

For the vehicle–index FPVRP formulation, we define two sets of binary variables to represent the routes and the visits to customers, and one set of continuous variables for the quantities delivered by the vehicles to customers at each time period.

Decision Variables:

- $y_{ij}^{kt} = \begin{cases} 1 \text{ if arc } (i,j) \in A \text{ is traversed by vehicle } k \in K \text{ at time period } t \in T, \\ 0 \text{ otherwise.} \end{cases}$.
- $z_i^{kt} = \begin{cases} 1 \text{ if node } i \in N \text{ is visited by vehicle } k \in K \text{ at time period } t \in T, \\ 0 \text{ otherwise.} \end{cases}$
- q_i^{kt}: Quantity delivered to customer $i \in C$ by vehicle $k \in K$ at time period $t \in T$.

The vehicle-index MILP formulation for the FPVRP is the following:

$$\min \quad \sum_{t \in T} \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} y_{ij}^{kt} \tag{4.1}$$

$$s.t. \quad q_i^{kt} \leq w_i z_i^{kt} \qquad i \in C, k \in K, t \in T \tag{4.2}$$

$$\sum_{i \in C} q_i^{kt} \leq Q z_0^{kt} \qquad k \in K, t \in T \tag{4.3}$$

$$\sum_{k \in K} z_i^{kt} \leq 1 \qquad i \in C, t \in T \tag{4.4}$$

$$\sum_{j|(i,j) \in A} y_{ij}^{kt} = z_i^{kt} \qquad i \in N, k \in K, t \in T \tag{4.5}$$

$$\sum_{j|(i,j) \in A} y_{ij}^{kt} = \sum_{j|(j,i) \in A} y_{ji}^{kt} \qquad i \in N, k \in K, t \in T \tag{4.6}$$

$$\sum_{\substack{(i,j) \in A \\ i,j \in S}} y_{ij}^{kt} \leq \sum_{i \in S} z_i^{kt} - z_s^{kt} \qquad S \subseteq C, s \in S, k \in K, t \in T \tag{4.7}$$

$$\sum_{t \in T} \sum_{k \in K} q_i^{kt} = W_i \qquad i \in C \tag{4.8}$$

$$z_i^{kt} \in \{0,1\} \qquad i \in N, k \in K, t \in T \tag{4.9}$$

$$q_i^{kt} \geq 0 \qquad i \in C, k \in K, t \in T \tag{4.10}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad (i,j) \in A, k \in K, t \in T. \tag{4.11}$$

The objective function (4.1) minimizes the routing costs. Constraints (4.2) impose that, at each time period, no vehicle delivers any customer $i \in C$ a quantity that exceeds w_i. Constraints (4.3) establish that the total quantity delivered by each vehicle k at time period t does not exceed the vehicle capacity. Constraints (4.4) ensure that, at each time period, at most one vehicle serves the demand of customer i. Constraints (4.5) state that, for every vehicle and time period, one arc has to exit from the node of every visited node. Constraints (4.6) are flow conservation constraints on the entering and leaving arcs at each customer, vehicle and time period. Constraints (4.7) are a reinforcement of the classical subtour elimination constraints (SECs) $\sum_{\substack{(i,j)\in A\\ i,j\in S}} y_{ij}^{kt} \leq |S|-1$. These enhanced SECs are better than the classical SECs due to their stronger linear programming relaxation, i.e., when z variables are fractional, the solution space is better delimited by using Constraints (4.7). Constraints (4.8) impose that the total quantity delivered to each customer at the end of the time horizon is equal to W_i. Finally, Constraints (4.9)–(4.11) define the domain of the variables.

This formulation has $|K||T|\left(|A|+|N|\right)$ binary variables and $|K||T||C|$ continuous variables. The size of the family of constraints (4.7) is exponential in the number of customers. All other families of constraints are of polynomial size.

4.1.1 Valid inequalities and optimality cuts

In order to strengthen formulation (4.1)–(4.11), several families of inequalities have been proposed and tested. Note that classical inequalities that can be used when the amount of product delivered to each customer at each time period is fixed, like the ones described in Letchford and Salazar-González (2015), cannot be applied to these formulations because this amount is a decision variable. Then, the following inequalities have been considered for the proposed formulation. They are similar to the ones used by Archetti et al. (2014) for the IRP.

Valid Inequalities $\hat{I}1$: Consistency constraints. They strengthen the routing part. Constraints (4.12) avoid to use vehicles if they do not depart from the depot.

$$z_i^{kt} \leq z_0^{kt} \qquad i \in C, t \in T, k \in K \tag{4.12}$$

Valid Inequalities $\hat{I}2$: Symmetry-breaking for vehicles. They are used to avoid replicating a given solution by just interchanging the indices of some of the vehicles. Constraints (4.13) establish that vehicle $k+1$ cannot be used unless vehicle k is also used. Constraints (4.14) are called the *lexicographic ordering constraints* in which a number is given to each customer and vehicles are assigned according to the order of these.

$$z_0^{kt} \geq z_0^{k+1,t} \qquad 1 \leq k \leq m-1, t \in T \tag{4.13}$$

$$\sum_{i=1}^{j} 2^{(j-i)} z_i^{kt} \geq \sum_{i=1}^{j} 2^{(j-i)} z_i^{k+1,t} \qquad j \in C, 1 \leq k \leq m-1, t \in T. \tag{4.14}$$

Valid Inequalities Î3: Fractional Capacity–Cut Constraints (FCCCs) are similar to those used in Archetti et al. (2014). These constraints differ from the classical FCCCs (Letchford and Salazar-González, 2015) in that the capacity cannot be rounded since the delivered quantities are modeled as decision variables.

$$Q \sum_{\substack{i \in S \\ j \in C \setminus S}} y_{ij}^{kt} \geq \sum_{i \in S} q_{it}^{k} \qquad S \subseteq C, k \in K, t \in T. \tag{4.15}$$

4.2 Load-based formulation for the FPVRP

For the load-based formulation of the FPVRP two sets of binary decision variables are introduced, which identify the arcs that are traversed and the customers that are visited at each time period. In addition, a set of integer decision variables that indicate the number of vehicles that are used at each time period is defined. This can be easily computed by counting the number of arcs leaving the depot at each time period. Finally, two additional sets of continuous variables are used. The first one indicates the load of the vehicles when they traverse the arcs while the second one, shows the amount of product delivered to each visited customer.

The definition of the decision variables is the following:

- $z_i^t = \begin{cases} 1 & \text{if customer } i \text{ is visited at time period } t, \\ 0 & \text{otherwise.} \end{cases} \quad (i \in C, t \in T)$

- $y_{ij}^t = \begin{cases} 1 & \text{if arc } (i,j) \text{ is traversed at time period } t, \\ 0 & \text{otherwise.} \end{cases} \quad ((i,j) \in A, t \in T)$
- z^t: Number of vehicles used at time period $t \in T$.
- l_{ij}^t: Load of the vehicle traversing arc $(i,j) \in A$, at time period $t \in T$.
- q_i^t: Quantity delivered to customer $i \in C$ at time period $t \in T$.

The formulation of the FPVRP is as follows:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{(i,j) \in A} c_{ij} y_{ij}^t && && (4.16) \\
s.t. \quad & q_i^t \le w_i z_i^t && i \in C, t \in T && (4.17) \\
& \sum_{i \in C} q_i^t \le Q z^t && t \in T && (4.18) \\
& \sum_{j|(i,j) \in A} y_{ij}^t = z_i^t && i \in C, t \in T && (4.19) \\
& \sum_{j|(i,j) \in A} y_{ij}^t = \sum_{j|(j,i) \in A} y_{ji}^t && i \in N, t \in T && (4.20) \\
& \sum_{j|(i,j) \in A} l_{ij}^t - \sum_{j|(j,i) \in A} l_{ji}^t = \begin{cases} -q_i^t, i \in C \\ \sum_{i' \in C} q_{i'}^t, i = 0 \end{cases} && i \in N, t \in T && (4.21) \\
& l_{ij}^t \le Q y_{ij}^t && (i,j) \in A, t \in T && (4.22) \\
& \sum_{j|(0,j) \in A} y_{0j}^t \le m && t \in T && (4.23) \\
& z^t = \sum_{i \in C} y_{0i}^t && t \in T && (4.24) \\
& \sum_{t \in T} q_i^t = W_i && i \in C && (4.25) \\
& q_i^t \ge 0 && i \in C, t \in T && (4.26) \\
& z^t \in \mathbb{Z} && t \in T && (4.27) \\
& z_i^t \in \{0,1\} && i \in C, t \in T && (4.28) \\
& y_{ij}^t \in \{0,1\}, l_{ij}^t \ge 0 && (i,j) \in A, t \in T. && (4.29)
\end{aligned}
$$

The objective function (4.16) minimizes the routing costs. Constraints (4.17) impose that none of the quantities delivered to each customer exceeds w_i. Constraints

(4.18) establish that the total quantity delivered at time t does not exceed the total capacity of the vehicles used at time t. Constraints (4.19) state that, at each time period, one arc has to exit from the node of every visited customer. Constraints (4.20) are flow conservation constraints on the entering and leaving arcs at each customer and time period. Constraints (4.21) are the load conservation constraints, which are imposed for each customer and time period. Constraints (4.22) impose that the vehicles loads do not exceed their capacity; they also link the y and the l variables. Constraints (4.23) ensure that the number of vehicles used is at most m. Constraints (4.24) guarantee that the value of variables z^t coincides with the number of vehicles used at each time period. Constraints (4.25) impose that the total quantity delivered to each customer at the end of the time horizon is equal to W_i. Finally, Constraints (4.26)–(4.29) define the domain of the variables.
The above formulation has $|T|\,(|C|+|A|)$ binary variables, $|T|$ general integer variables, and $|T|\,(|C|+|A|)$ continuous variables. The number of constraints is $|T|\,(6|C|+2|A|+6)+|C|$.

4.2.1 Valid inequalities and optimality cuts

The families of inequalities proposed to strengthen formulation (4.16)–(4.29) are listed below.

Inequalities $I1$. Sum of final loads: All vehicles return to the depot with an empty load. Indeed, these inequalities are not valid, since there are feasible FPVRP solutions that do not satisfy them. Instead, they are optimality cuts, since there is at least an optimal solution that satisfies them. Therefore, they can be used to reduce the domain of the solutions that are explored.

$$\sum_{t\in T}\sum_{j\in C} l^t_{j0} = 0 \tag{4.30}$$

Inequalities $I2$. Symmetry-breaking of routes: The following inequalities partially break the symmetry of the routes by exploiting the fact that arc costs are symmetric. According to them only the routes with a certain orientation will be considered (as the same route in the opposite orientation will have the same cost). Among the two possible orientations for a route, the one that starts with the lowest-index customer is chosen. For this, it is imposed that

$$y_{i0}^t \leq \sum_{r \leq i} y_{0r}^t, \qquad \forall i \in C, \forall t \in T \tag{4.31}$$

That is, if arc $(i, 0)$ enters the depot at time period t, then there must be a lowest index arc $(0, r)$ with $r \leq i$ which exits the depot at this time period. The symmetry-breaking inequalities are also optimality cuts, but not valid inequalities.

Valid Inequalities $I3$: The relation between variables y and z can be imposed in several ways. In the FPVRP formulation this is done in a *two-step* fashion. On the one hand, the flow balance constraints (4.21) relate the load variables l to the q, which, in turn, activate the z. On the other hand, constraints (4.22) relate the load variables l to the arc variables y. Nevertheless, this relation can be stated in a more direct way, similarly to the constraints imposed in Christofides and Beasley (1984) for the PVRP. In particular, no arc y_{ij} can be used at time period t unless customers i and j are visited at the same time period. Therefore, the following inequalities are valid:

$$y_{ij}^t \leq \frac{z_i^t + z_j^t}{2}, \qquad \forall i, j \in C, \forall t \in T \tag{4.32}$$

Taking into account that no arc will be traversed in both directions in the same time period the above inequalities can be reinforced to:

$$y_{ij}^t + y_{ji}^t \leq \frac{z_i^t + z_j^t}{2}, \qquad \forall i < j \in C, \forall t \in T \tag{4.33}$$

4.3 Load-based formulation for the FPVRP-IC

Below a load-based formulation for the FPVRP-IC is presented, which is largely based on the formulation for the IRP proposed by Archetti et al. (2014). The inventory levels are evaluated after the delivery of q_i^t and the consumption of d_i^t.

Note that inventory levels cannot exceed $w_i - q_i^t$. In addition to the l, q, y, and z variables used in the FPVRP formulation (4.16)–(4.29), the formulation for the FPVRP-IC uses another set of continuous variables that represent the customers inventory levels at each time period.

- I_i^t: Inventory level at customer $i \in C$ at the end of time period $t \in T$.

The FPVRP-IC formulation is the following:

$$\min \quad \sum_{t \in T} \sum_{(i,j) \in A} c_{ij} y_{ij}^t \tag{4.34}$$

$$s.t. \quad I_i^t = I_i^{t-1} - d_i^t + q_i^t \qquad i \in C, t \in T \tag{4.35}$$

$$q_i^t \leq w_i - I_i^{t-1} \qquad i \in C, t \in T \tag{4.36}$$

$$q_i^t \leq w_i z_i^t \qquad i \in C, t \in T \tag{4.37}$$

$$\sum_{i \in C} q_i^t \leq Q z^t \qquad t \in T \tag{4.38}$$

$$\sum_{j|(i,j) \in A} y_{ij}^t = z_i^t \qquad i \in C, t \in T \tag{4.39}$$

$$\sum_{j|(i,j) \in A} l_{ij}^t - \sum_{j|(j,i) \in A} l_{ji}^t = \begin{cases} -q_i^t, & i \in C \\ \sum_{i \in C} q_i^t, & i = 0 \end{cases} \qquad i \in N, t \in T \tag{4.40}$$

$$l_{ij}^t \leq Q y_{ij}^t \qquad (i,j) \in A, t \in T \tag{4.41}$$

$$\sum_{j|(i,j) \in A} y_{ij}^t = \sum_{j|(j,i) \in A} y_{ji}^t \qquad i \in N, t \in T \tag{4.42}$$

$$\sum_{j|(0,j) \in A} y_{0j}^t \leq m \qquad t \in T \tag{4.43}$$

$$z^t = \sum_{i \in C} y_{0i}^t \qquad t \in T \tag{4.44}$$

$$I_i^t \geq 0 \qquad i \in N, t \in T \tag{4.45}$$

$$q_i^t \geq 0 \qquad i \in C, t \in T \tag{4.46}$$

$$z_i^t \in \{0,1\} \qquad i \in C, t \in T \tag{4.47}$$

$$z^t \in \mathbb{Z} \qquad t \in T \tag{4.48}$$

$$y_{ij}^t \in \{0,1\}, l_{ij}^t \geq 0 \qquad (i,j) \in A, t \in T. \tag{4.49}$$

The objective function (4.34) is the minimization of the total routing cost. Constraints (4.35) and (4.45) determine the inventory levels over time and avoid stock–out situations. Since $T = \{1, \dots, H\}$, when $t = 1$, $I_i^{t-1} = I_i^0$ corresponds to the initial inventory at customer i. Constraints (4.36) ensure that delivered quantities do not exceed the maximum quantity needed by each customer at each time period. The remaining constraints have the same meaning as in the FPVRP formulation.

4.4 Load-based formulation for the PVRP

A new formulation for the PVRP, which has been used in the computational experiments is presented. To the best of our knowledge this is the first load-based formulation in the literature for the classical PVRP. It uses the same y, z and l variables as the FPVRP formulation. In addition, the following set of binary decision variables is defined to determine the schedule that is chosen for each customer:

- $v_i^s = \begin{cases} 1 \text{ If customer } i \in C \text{ is visited according to schedule } s \in S_i, \\ 0 \text{ otherwise.} \end{cases}$

Then, the load-based formulation for the PVRP is:

$$\min \quad \sum_{t \in T} \sum_{(i,j) \in A} c_{ij} y_{ij}^t \tag{4.50}$$

$$s.t. \quad \sum_{s \in S_i} v_i^s = 1 \qquad i \in C \tag{4.51}$$

$$z_i^t = \sum_{s \in S_i} v_i^s a_{st} \qquad t \in T, i \in C \tag{4.52}$$

$$y_{ij}^t \leq \frac{z_i^t + z_j^t}{2} \qquad t \in T; i \neq j \in C \tag{4.53}$$

$$\sum_{j|(i,j) \in A} y_{ij}^t = \sum_{j|(j,i) \in A} y_{ji}^t \qquad i \in N, t \in T \tag{4.54}$$

$$\sum_{j|(i,j) \in A} y_{ij}^t = z_i^t \qquad i \in C, t \in T \tag{4.55}$$

$$\sum_{j|(i,j) \in A} l_{ij}^t - \sum_{j|(j,i) \in A} l_{ji}^t = \begin{cases} -w_i z_i^t, & i \in C \\ \sum_{j \in C} w_j z_j^t, & i = 0 \end{cases} \qquad i \in N, t \in T \tag{4.56}$$

$$l_{ij}^t \leq Q y_{ij}^t \qquad (i,j) \in A, t \in T \quad (4.57)$$

$$\sum_{j \in C} y_{0j}^t \leq m \qquad t \in T \quad (4.58)$$

$$v_i^s \in \{0,1\} \qquad i \in C, s \in S_i \quad (4.59)$$

$$z_i^t \in \{0,1\} \qquad i \in C, t \in T \quad (4.60)$$

$$y_{ij}^t \in \{0,1\} \qquad (i,j) \in A, t \in T \quad (4.61)$$

$$l_{ij}^t \geq 0 \qquad (i,j) \in A, t \in T. \quad (4.62)$$

The objective function (4.50) minimizes the total routing costs. Constraints (4.51) ensure that a schedule is assigned to each customer. Constraints (4.52) relate the selected schedules with customer visits. Constraints (4.53) allow to connect two customers only if both are served in the same period. Constraints (4.54) are the flow conservation constraints. Constraints (4.55) ensure that exactly one arc leaves each visited customer at each time period. Constraints (4.56) are load conservation constraints. Constraints (4.57) ensure that the load of vehicles does not exceed their capacity. Constraints (4.58) indicate that the number of vehicles used must be at most the maximum number of vehicles available at each time period. Finally, Constraints (4.59)-(4.62) determine the domain of variables.

Chapter 5

Exact solution algorithms for the FPVRP

This chapter describes the exact methods developed to solve the MILP formulations for the FPVRP presented in Chapter 4, as well as some valid inequalities and optimality cuts added for their reinforcement. Furthermore, the benchmark instances and the computational experiments performed for the proposed exact solution algorithms are described.

5.1 Description of the algorithms

Figure 5.1 shows the general scheme of the exact algorithms that are proposed for solving the FPVRP formulations.

The vehicle-index FPVRP formulation requires the set of constraints (4.7) to avoid the creation of subtours. Adding SECs directly to the formulation becomes impracticable, since their number grows exponentially with the number of customers. Therefore, only the SECs that are needed are incorporated to the formulation. For this we apply an iterative algorithm that solves at each iteration the LP relaxation of formulation (4.1)–(4.6), (4.8)–(4.11) reinforced with only a subset of inequalities (4.7), and applies a separation procedure that identifies whether or not any not yet considered inequality (4.7) is violated by the current solution. If so, the violated inequality is added to the current formulation and the process is repeated. Furthermore, in order to strengthen the formulation, other families of inequalities are also included.

On the contrary, SECs are not needed for the load-based formulation, since its feasible solutions already satisfy them. Hence the load-based formulation is only reinforced with valid inequalities and optimality cuts.

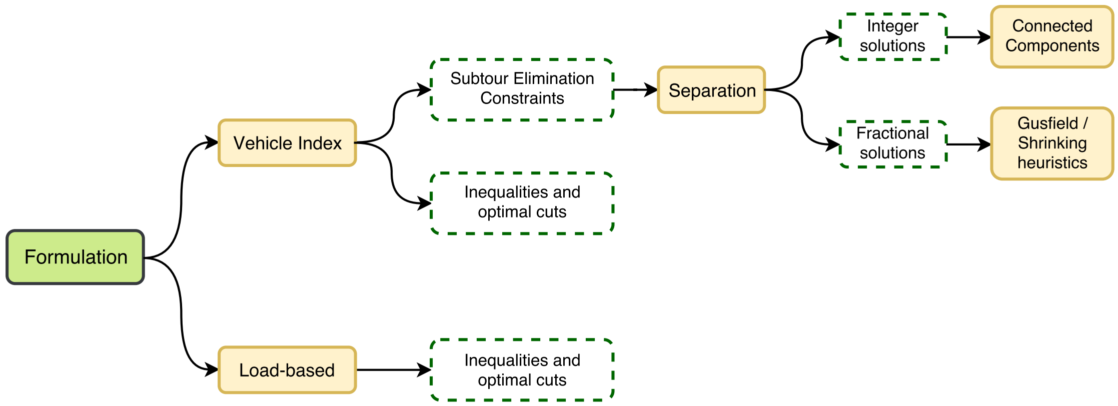


Figure 5.1: Exact solution method for the FPVRP.

Below we describe the separation of SECs and the valid inequalities and optimality cuts used to reinforce the formulations.

5.1.1 Separation of SECs

The separation procedure applied to identify violated SECs differs depending on whether or not the current LP solution is integer or fractional. For each each vehicle $k \in K$ and time period $t \in T$, let $G^{kt} = (S^{kt}, A^{kt})$ denotes the subgraph induced by the solution of the LP relaxation of the current formulation, where $S^{kt} \subseteq N$ denote the set of nodes visited by vehicle k (i.e. $z_i^{kt} > 0$), and A^{kt} the set of arcs used by vehicle k (i.e. $y_{ij}^{kt} > 0$). Figure 5.2 shows an example of an induced graph.

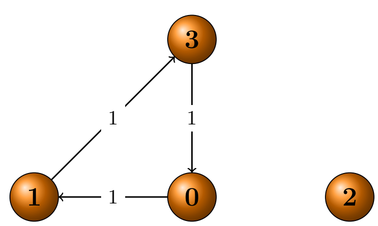


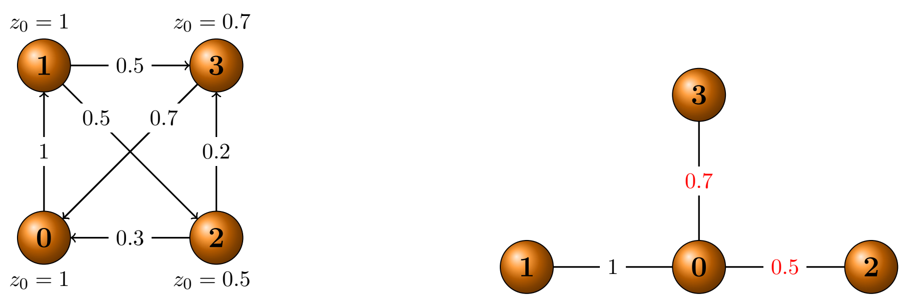
Figure 5.2: Induced graph G^{kt} (visited {0,1,3}, non visited {2}).

- **Integer solution:**

  When the current LP solution is integer, violated SECs can be obtained by identifying the *connected components* of each graph G^{kt}. If there are two or more components, a violated SEC is associated with each of them. To find these *connected components*, the *disjoint–set data structures* (Galler and Fisher, 1964) are used.

- **Fractional solution:**

  When the current LP solution is fractional, violated SECs can be found, if they exist, from the tree of min–cuts of each subgraph G^{kt}. In particular, each cut–set of value strictly smaller than 1, defines a violated SEC (Drexl, 2013). It is well–known that the tree of min–cuts of a given graph can be found by solving a series of *Max–Flow* problems (Ford and Fulkerson, 1956). In our case we apply the *Gusfield algorithm* (Gusfield, 1990). Figure 5.3 shows an example of the graph G^{kt} and the min-cuts tree $\hat{T}$ obtained after applying the Gusfield algorithm.



(a) Induced graph G^{kt}. (b) Cut tree $\hat{T}$.

Figure 5.3: Generation of a Gusfield tree of min–cuts.

The set of constraints (4.15) is of exponential size in the number of customers, so it must be handled in a similar way as SECs. We have implemented three alternative algorithms to find violated cuts from this family: a Greedy Shrinking heuristic (Augerat et al., 1998), an Extended Shrinking heuristic (Nagamochi and Ibaraki, 1992), and the Gusfield algorithm (Gusfield, 1990) as a last option.

5.2 Benchmark instances

In the development of this thesis several series of computational experiments have been run in order to test the proposed formulations and the performance of the solution algorithms. Before presenting the numerical results we describe the sets of benchmark and new instances used in the experiments.

5.2.1 Description of the set of instances

We have classified the instances into five different sets:

- **Set 1 (S1)**: The IRP instances proposed by Archetti et al. (2014). This set consists of 40 benchmark instances with $|C| \in \{5, 10, 15, 20\}$ and a 3–period time horizon, i.e. $|T| = 3$.

- **Set 2 (S2)**: The PVRP instances proposed by Francis et al. (2006). This set consists of 24 instances with $|C| \in \{7, 9, 11, 15, 49\}$ and time horizon $|T| = 5$. Instances were generated in a *similar* way to those used by Francis et al. (2006) and according to the database provided by one of the authors. Schedules for each customer are assigned according to their frequency of visits as explained in the following.

- **Set 3 (S3)**: A newly generated set of 35 PVRP instances of medium size with *clustered* customers. The time horizon used in the whole set is $|T| = 5$. Other parameters are: the number of customers $|C|$, the vehicle capacity Q, the number of clusters p, a radius r, which determines the *coverage area* of each cluster, and a parameter β, which together with r determines the minimum distance $\beta \times r$ among the *centers* of the clusters. Five instances were generated for each combination of $|C| \in \{10, 15, 20\}$ and $r \in \{0.15, 0.30\}$, plus five more instances with $|C| = 20$ and $r = 0.50$. Vehicles capacities, Q, have been set to 200, 250 and 300 for 10, 15 and 20 customers, respectively. For $|C| = 10$, the number of clusters is set to $p = 2$, whereas for instances with $|C| \in \{15, 20\}$ is set to $p = 3$. When $r \in \{0.15, 0.30\}$, the value of β has been set to 2, which avoids clusters overlapping. Instead, for $r = 0.50$ is fixed to $\beta = 1$, allowing customers to belong to more than one cluster.

- **Set 4 (S4)**: A set of 10 larger instances, generated in a similar way to the ones in S3, with $|C| \in \{50, 100\}$, $r \in \{0.15\}$ and vehicle capacity of $Q = 500$.
- **Set 5 (S5)**: A set of 5 PVRP benchmark instances from the literature (Chao et al., 1995; Baldacci et al., 2011) and available in `http://neumann.hec.ca/chairedistributique/data/pvrp/old/`.

Section 5.2.2 shows the procedure of generating instances of sets S3 and S4.

5.2.2 Generation of FPVRP instances

The generation of instances of the sets S3 and S4 is done according to the following steps (see Figure 5.4 for a graphical example):

- The depot is located at the center of the unit square.
- p *centers* (one for each cluster) are randomly generated from a uniform distribution in the unit square. Each generated center must satisfy the minimum distance condition $\beta \times r$ with respect to the others.
- Once all the centers have been fixed, the customers are generated in such a way that clusters are balanced, i.e. each cluster contains up to $\left\lceil \frac{|C|}{p} \right\rceil$ customers. Clusters are progressively filled: first, one customer is generated for each cluster; then, a second one; and so on, until $|C|$ customers have been randomly generated using an uniform distribution in the circles around the clusters' centers of radius r.
- A storage capacity w_i, $i \in C$, is randomly generated from an integer uniform distribution in $[1, Q]$.
- A number of visits is randomly associated with each customer according to the following options: $f_i = 5$, 3 or 2.

- A set of schedules S_i is assigned to each customer according to its frequency f_i, which is randomly selected. The number of visits associated with each schedule is $f_i = 5$, 3 or 2. This step is needed because the comparisons with the PVRP require a predefined schedule. For FPVRP tests, this is not necessary as the FPVRP does not depend on predefined schedule.
  - **Schedules for S2, S3 and S4**: Different schedule options are considered for the tests.
    * $f_i = 2 : S_i = \{(0,0,1,0,1),(0,1,0,0,1),(0,1,0,1,0),(1,0,0,1,0),(1,0,1,0,0)\}$,
    * $f_i = 3 : S_i = \{(0,1,0,1,1),(0,1,1,0,1),(1,0,1,0,1),(1,0,1,1,0),(1,1,0,1,0)\}$,
    * $f_i = 5 : S_i = \{(1,1,1,1,1)\}$.
- Once all the customer demands are generated, the number of vehicles is set to $m = \left\lceil \frac{\sum_{i \in C} w_i}{Q} \right\rceil$ for S3 and to $m = 1 + \frac{\sum_{i \in C} w_i f_i}{QH}$ for S4.

Figure 5.4 shows an example of an instance with $|C| = 10$, $Q = 10$ and $p = 4$. As it can be seen, two clusters have three customers each and the remaining ones have two customers each. Nodes with the same color represent customers with the same assigned schedule. The number inside each node represents the assigned storage capacity w_i.

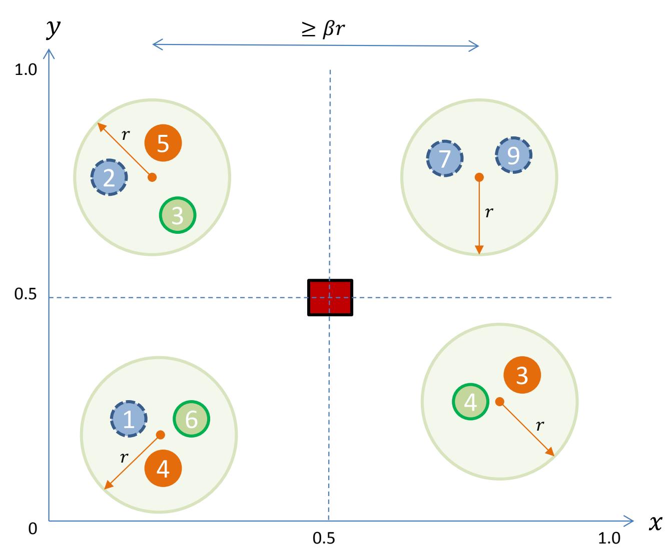


Figure 5.4: Clustered instance with $|C| = 10$, $Q = 10$, $p = 4$, and $|S| = 3$.

5.3 Computational experience

In this chapter we present and discuss the numerical results obtained with the exact algorithms. The aim of the experiments that were run was twofold: on the one hand to analyze the computational difficulty of the FPVRP and the effectiveness of the proposed formulations. On the other hand, to highlight the benefits derived from allowing flexibility in the PVRP and the FPVRP-IC, by comparing the solutions produced by the proposed models.

For the computational experiments, all formulations were implemented in C++ with ILOG Concert Technology API and CPLEX 12.5.0.0, running on a HP Intel(R)-Xeon(R) 2.4GHz Workstation with 32GB RAM (Win Server 2012, 64 bits). Default parameters were used. All computing times were limited to 14400 seconds. The sets of instances used for these tests were S1, S2, and S3.

5.3.1 Vehicle–index vs load–based formulations

A preliminary test was performed in order to compare the effectiveness of the proposed vehicle–index and load–based FPVRP formulations. The one with the best performance was considered for later experiments. Table 5.1 shows the results obtained for both formulations. They were applied to a small subset of instances of S3 ($r = 0.15$). The vehicle-index formulation includes all the valid inequalities proposed in Chapter 4.1.1.

It is clear that the load–based formulation is much better than the vehicle–index version as it was able to obtain feasible solutions for all instances (*Status* column) and the optimality gaps were much better (*Gap%* column). Thus, the vehicle–index formulation was excluded from any further consideration in our experiments.

Instance	Vehicle–index				Load–based			
	Status	*BestSol*	*Gap%*	*Time*	*Status*	*BestSol*	*Gap%*	*Time*
n10k5t5_1	F	20.79	2.03	14399	F	20.78	2.31	14400
n10k4t5_2	F	12.44	7.64	14399	F	12.44	2.58	14400
n10k5t5_3	F	13.45	4.66	14400	F	13.23	2.71	14400
n10k4t5_4	F	13.53	5.75	14399	F	13.53	3.55	14400
n10k8t5_5	F	26.64	2.69	14399	F	25.91	1.28	14400
n15k10t5_1	F	35.26	13.23	14400	F	34.26	2.53	14403
n15k6t5_2	F	18.44	12.54	14401	F	17.42	5.23	14400
n15k10t5_3	I	—	—	14401	F	25.25	3.17	14400
n15k8t5_4	F	32.72	11.15	14402	F	32.11	2.66	14400
n15k7t5_5	F	24.21	9.12	14400	F	23.89	4.39	14400
n20k10t5_1	I	—	—	14400	F	24.58	4.06	14402
n20k12t5_2	I	—	—	14400	F	36.10	1.82	14400
n20k11t5_3	I	—	—	14400	F	23.70	3.82	14404
n20k10t5_4	I	—	—	14400	F	35.35	2.32	14401
n20k10t5_5	I	—	—	14400	F	29.45	1.91	14399

Table 5.1: Comparison between vehicle–index and load-based FPVRP formulations.

5.3.2 Evaluation of inequalities and optimality cuts for the FPVRP

The first set of experiments was focused on the evaluation of the effectiveness of the different inequalities for the load–based FPVRP formulation.

Initially, several variants of the formulation presented in Chapter 4.2 were tested and the obtained results indicated that the best performance was attained with formulation (4.16)-(4.29). Then, a comparison of this base formulation against alternative reinforcements resulting from the addition of different combinations of inequalities presented in Chapter 4.2.1 was made. For this test, a subset of benchmark instances whose optimality was particularly difficult to prove with the base formulation was selected. This subset consists of 16 S2 instances (with $|K| \in \{3, 4\}$) plus the 35 new S3 instances.

Average results (over all instances in each group) of the percentage optimality gaps at termination are summarized in Table 5.2. These optimality gaps have been computed as $100 \times \frac{BestSol-LB}{BestSol}$, where $BestSol$ is the best-known feasible solution value for each instance (among all tested versions), and LB is the lower bound at termination of the tested version. Each row corresponds to a group of instances. The first column indicates the set to which the group of instances belongs. The next three columns give some characteristics of the instances of each group: r (for instances in $S3$), $|C|$ and Q (an average value is reported for $S2$). The number of instances in each group is given in column *Inst.* Average results over all the combinations tested are given in the following six columns: entries in column *Base* correspond to the base formulation (4.16)-(4.29), whereas under *I1*, *I2*, *I3*, the results of the base formulation reinforced respectively with (i) the empty load return inequalities (4.30); (ii) the symmetry breaking inequalities (4.31); and, (iii) the valid inequalities (4.33) are shown. The last two columns correspond to the combinations of (i) with (ii) and of (i) with (iii). In each case, the number of instances optimally solved, within the allowed computing time, is given in parenthesis next to the percentage optimality gap. These values are only given for the groups of instances of S2 because it was never possible to prove the optimality for any of the instances in S3.

Other combinations of inequalities were also tested but did not give significantly different results (see Appendix C.1).

Set	r	$\|C\|$	Q	Inst.	*Base*	*I1*	*I2*	*I3*	*I1+I2*	*I1+ I3*
S2		15	999.38	8	0.43 (3)	0.30 (3)	0.38 (5)	0.45 (3)	0.26 (5)	0.36 (3)
		15		8	0.32 (3)	0.29 (4)	0.31 (5)	0.52 (3)	0.26 (5)	0.35 (3)
S3	0.15	10	200	5	2.58	2.78	2.65	3.16	2.49	2.93
		15	250	5	3.83	3.82	3.76	4.15	3.55	3.90
		20	300	5	5.71	5.71	5.67	5.68	5.65	5.57
	0.30	10	200	5	1.51	1.56	1.54	1.92	1.51	1.71
		15	250	5	1.97	1.97	1.94	2.15	1.89	2.05
		20	300	5	2.09	2.15	2.16	2.35	2.15	2.26
	0.5	20	300	5	2.59	2.50	2.55	2.97	2.49	2.85

Table 5.2: Summary of results of FPVRP for different combinations of inequalities.

Results show that strengthened formulations reduced the percentage gap in nearly all cases. Broadly speaking, both the empty load return inequalities (4.30) and the symmetry breaking inequalities (4.31) are effective. The effectiveness of inequalities (4.33) that relate y and z variables is not so clear and in some cases the base formulation alone gives better results than if they are used. This behavior may be due to the high number of inequalities of this family ($\mathcal{O}(|C|^2|T|)$). As can be seen, the best results are those of column *I1+I2*, which corresponds to the base formulation reinforced with (4.30) and (4.31). Therefore, all subsequent tests used the corresponding strengthened FPVRP formulation.

The results obtained with the strengthened FPVRP formulation for the full set of benchmark instances are summarized in Table 5.3. Again, rows correspond to groups of instances with similar characteristics. The first five columns describe the characteristics of the instances: r (for instances in $S3$), $|C|$, $|K|$, Q, and D_t (the total demand that must be distributed at each time period). When not all the instances in the group have the same parameters, minimum and maximum values are displayed. In the following columns, *Inst.* gives the number of instances in each group, and *O/F* gives the number of instances of the group that terminated with a solution with proven optimality (O) and with only a feasible solution (F). The average percentage optimality gaps at termination are given in column *%Gap*. Finally, the last two columns refer to the computing times: *avrg.* for the average times and *range* for the minimum and maximum of such values.

Set	r	$\|C\|$	$\|K\|$	Q	D_t	Inst.	O/F	$\%Gap$	Time *avrg.*	Time *range*
S1		5	2 - 3	79 - 175	193 - 304	10	10/0	0.00	1.40	0 - 7
		10	2 - 3	272 - 480	458 - 640	10	10/0	0.01	202.90	7 - 942
		15	2 - 3	340 - 619	681 - 845	10	9/1	0.34	3086.80	101 - 14400
		20	2 - 3	512 - 867	999- 1156	10	6/4	2.32	7214.50	147 - 14400
S2		7	3	496 - 871	699	4	4/0	0.00	0.25	0 - 1
		9	3	1037 - 1208	1206	4	4/0	0.00	27.25	9 - 57
		11	3	546 - 1521	851	8	7/1	0.26	1965.50	22 - 14400
		15	3	757 - 1240	1360	8	5/3	0.24	6313.88	17 - 14400
		49	4	1111	4443	1	0/1	49.76	14400	14400
S3	0.15	10	4 - 8	200	664 - 1401	5	0/5	2.49	14400	14400
		15	6 - 10	250	1495 - 2484	5	0/5	3.55	14400	14400
		20	10 - 12	300	2813 - 3408	5	0/5	5.65	14400	14400
	0.30	10	5 - 8	200	820 - 1436	5	0/5	1.51	14400	14400
		15	6 - 9	250	1488 - 2225	5	0/5	1.89	14400	14400
		20	10 - 13	300	2824 - 3691	5	0/5	2.15	14400	14400
	0.50	20	7 - 14	300	2059 - 4076	5	0/5	2.49	14400	14400

Table 5.3: Summary of results of the FPVRP for the complete set of instances.

The obtained results highlight the difficulty of the FPVRP. This will become more evident when the results of the FPVRP are compared with the results of the FPVRP-IC and the PVRP. Still, it was possible to optimally solve 35 out of the 40 S1 instances and 20 out of the 24 S2 instances. For the set S1, the percentage optimality gap at termination was always below 5%, except for one 20 customer instance (abs1n20_2). Optimality gaps of unsolved S2 instances were always below 1%, except for a 11 customer instance (Instn12t5k3_1), with a 2.11% gap, and the largest 49 customer instance for which the relative percentage deviation between the upper and lower bounds at termination was of nearly 25%. The results of the individual instances in S1 and S2 (see Tables 5.4, 5.5, and 5.6 below) show that, when all other parameters are similar, a clear indicator of the difficulty of an instance is the fleet size.

As can be seen, the new instances of set S3 are considerably harder to solve than those of sets S1 and S2 of similar sizes. This is possibly due to two factors. The first one is the fleet size which is much larger in S3 than in S1 and S2. The second one is the *proximity* of the clustered customers (particularly for the small radius 0.15), which makes it particularly difficult to *discriminate* among solutions that permute the order in which neighboring customers are visited.

5.3.3 Comparison of the FPVRP with other VRPs with periodic demand

The following series of experiments was oriented to analyze the potential advantage of the FPVRP relative to the FPVRP-IC and the PVRP. Since all three models focus on the overall routing costs throughout the time horizon, potential advantages can be quantified in terms of the percentage relative reduction in the objective function value of the compared models. In particular, this value will be computed as follows:

$$\%Imp = \max\left\{0, \frac{Z_{\text{Mod}} - Z_{\text{FPVRP}}}{Z_{\text{Mod}}}\right\} \times 100,$$

where Z_{FPVRP} and Z_{Mod} are the best-known values for the FPVRP and the compared model, respectively.

Comparison between the FPVRP and the FPVRP-IC

For the comparison between the FPVRP and the FPVRP-IC, the set of benchmark instances S1 was used. Each instance was run with both the strengthened FPVRP formulation and the FPVRP-IC formulation with a time limit of 14400 seconds. The results are shown in Table 5.4 where the first five columns show the name of the instances and their characteristics. The next two blocks of four columns each correspond to the FPVRP and to the FPVRP-IC results, respectively. Column *Status*, indicates whether the instance was solved to proven optimality (O), or a feasible solution was found but its optimality was not proven (F); *BestSol* gives the value of the best solution obtained in the run, *BestLB* is the lower bound at termination, and *Time* is the computing time consumed (in seconds). The last column of the table, *%Imp*, gives the percentage relative improvement obtained with the FPVRP with respect to the FPVRP-IC.

					FPVRP				FPVRP-IC				
Instance	$\|C\|$	$\|K\|$	Q	D_t	*Status*	*BestSol*	*BestLB*	*Time*	*Status*	*BestSol*	*BestLB*	*Time*	$\%Imp$
abs1n5_1	5	2	144	193	O	1301.85	1301.85	0	O	1301.85	1301.85	0	0.00
abs1n5_2		3	96	193	O	1335.88	1335.86	0	O	1335.88	1335.88	0	0.00
abs2n5_1		2	118	158	O	1088.72	1088.72	0	O	1088.72	1088.72	0	0.00
abs2n5_2		3	79	158	O	1494.37	1494.37	2	O	1494.37	1494.23	4	0.00
abs3n5_1		2	228	304	O	2109.51	2109.51	1	O	2302.82	2302.82	2	8.39
abs3n5_2		3	152	304	O	2864.95	2864.87	1	O	2864.95	2864.95	0	0.00
abs4n5_1		2	134	179	O	1504.27	1504.27	7	O	1650.73	1650.59	0	8.87
abs4n5_2		3	89	179	O	2224.13	2223.94	3	O	2224.13	2224.13	1	0.00
abs5n5_1		2	175	234	O	1091.97	1091.97	0	O	1091.97	1091.97	0	0.00
abs5n5_2		3	117	234	O	1386.18	1386.18	0	O	1386.18	1386.18	4	0.00
abs1n10_1	10	2	476	635	O	1936.15	1936.01	96	O	1960.99	1960.82	18	1.27
abs1n10_2		3	317	635	O	2369.40	2369.16	737	O	2429.55	2429.55	31	2.48
abs2n10_1		2	408	545	O	2491.71	2491.52	7	O	2554.79	2554.79	14	2.47
abs2n10_2		3	272	545	O	3194.02	3193.71	185	O	3214.05	3213.88	19	0.62
abs3n10_1		2	343	458	O	1980.71	1980.71	17	O	1980.71	1980.71	6	0.00
abs3n10_2		3	229	458	O	2372.91	2372.73	13	O	2410.50	2410.50	39	1.56
abs4n10_1		2	411	548	O	2115.97	2115.78	9	O	2240.93	2240.73	35	5.58
abs4n10_2		3	274	548	O	2756.47	2756.20	942	O	2943.14	2942.87	279	6.34
abs5n10_1		2	480	640	O	1746.02	1746.02	11	O	1848.20	1848.20	14	5.53
abs5n10_2		3	320	640	O	2014.42	2014.42	12	O	2151.45	2151.45	19	6.37
abs1n15_1	15	2	619	826	O	1915.91	1915.77	175	O	1915.91	1915.89	21	0.00
abs1n15_2		3	413	826	O	2349.28	2349.05	5437	O	2402.36	2402.14	295	2.21
abs2n15_1		2	592	790	O	2161.09	2160.89	898	O	2185.68	2185.46	194	1.13
abs2n15_2		3	395	790	O	2388.97	2388.73	719	O	2388.97	2388.97	39	0.00
abs3n15_1		2	633	845	O	2373.10	2372.90	101	O	2373.10	2373.10	11	0.00
abs3n15_2		3	422	845	O	2646.11	2645.86	116	O	2646.11	2646.11	20	0.00
abs4n15_1		2	538	718	O	2064.15	2063.95	455	O	2199.78	2199.57	188	6.17
abs4n15_2		3	359	718	O	2403.11	2402.87	7297	O	2572.55	2572.30	705	6.59
abs5n15_1		2	510	681	O	2192.45	2192.23	1271	O	2309.75	2309.53	88	5.08
abs5n15_2		3	340	681	F	2678.85	2634.11	14399	O	2959.31	2959.02	5829	9.48
abs1n20_1	20	2	799	1066	O	2345.27	2345.04	6768	O	2410.91	2410.70	6343	2.72
abs1n20_2		3	533	1066	F	3004.62	2741.40	14399	F	3103.40	2867.75	14399	3.18
abs2n20_1		2	782	1043	O	2148.82	2148.63	156	O	2148.82	2148.82	18	0.00
abs2n20_2		3	521	1043	O	2365.64	2365.40	4511	O	2393.13	2392.90	658	1.15
abs3n20_1		2	768	1024	O	2283.53	2283.31	615	O	2283.53	2283.53	23	0.00
abs3n20_2		3	512	1024	O	2529.42	2529.19	147	O	2529.42	2529.28	16	0.00
abs4n20_1		2	749	999	F	2888.29	2858.71	14399	O	3136.22	3135.91	2782	7.91
abs4n20_2		2	499	999	F	3408.32	3287.19	14399	F	3664.52	3602.53	14399	6.99
abs5n20_1		3	867	1156	O	2854.24	2853.96	2352	O	2859.60	2859.35	379	0.19
abs5n20_2		3	578	1156	F	3562.11	3403.85	14399	F	3567.47	3507.53	14399	0.15
										Average Improvement			2.56

Table 5.4: Comparison between FPVRP and FPVRP-IC with S1 instances.

The results of Table 5.4 show that, except for the small instances with only five customers, in general, the computing times required by the FPVRP are substantially greater than those of the FPVRP-IC. This becomes more evident as the size of the instances increases and is reflected by the fact that five instances were not optimally solved for the FPVRP within the maximum computing time, whereas this number reduces to three for the FPVRP-IC. The entries in column *%Imp* show that there are instances where both models have the same optimal value, although in most cases the FPVRP produces an improvement (up to 9.48%) with respect to the FPVRP-IC, which increases with the size of the instances. It is worth noting that, even for the two instances that were not optimally solved with the FPVRP but were optimally solved with the FPVRP-IC, the best FPVRP solution at termination was considerably better than the optimal FPVRP-IC solution. These results show that, if possible, it is worthwhile to increase flexibility and not consider inventory levels, as done in the FPVRP, as this may lead to remarkable savings.

Comparison between the FPVRP and the PVRP

For the comparison between the FPVRP and the PVRP, instances of both S2 and S3 were used. Each instance in these two sets was run with the strengthened FPVRP formulation (4.16)-(4.29) and with the PVRP formulation (4.50)-(4.59). Again, the time limit for each run was 14400 seconds. The results for the instances of set S2 are presented in Table 5.5, where instances with the same number of customers are ordered by decreasing capacity of the vehicles. The columns labeled as *%Gap* show the relative percentage optimality gap at termination of each optimization.

Most of the instances with $|C| = 7$, $|C| = 9$ and $|C| = 11$ were solved to optimality with both the FPVRP and the PVRP formulations. Once more, it can be observed that, in terms of the computing times, the FPVRP formulation is more demanding than that for the PVRP. This could be expected as the FPVRP incorporates additional decisions to the ones of the PVRP. While the computing times to optimally solve the PVRP instances remain negligible, they become significant for the FPVRP as the size of the instances increases. In particular, for five of the instances the optimality of the best solution found could not be proven within the allowed computing time. Concerning the improvement of the FPVRP with respect to PVRP, it is noticed that no improvement can be perceived related to the flexibility in the delivered quantities since the total vehicles capacities are enough to cover the demand of all customers in the same day. Nevertheless, the FPVRP still shows slight improvements related to the non-dependency of schedules. Moreover, although the FPVRP produced no improvement in the large 49 customers instance, it was able to find a feasible solution within the time limit. On the contrary, the PVRP produced no solution for that instance.

					FPVRP					PVRP			%Imp
Instance	$\|C\|$	$\|K\|$	D_t	Q	*Status*	*BestSol*	*BestLB*	*%Gap*	*Time*	*Status*	*%Gap*	*Time*	
Instn8t5k3_1	7	3	699	871	O	5.82	5.82	0	0	O	0	0	0.00
Instn8t5k3_3				765	O	5.82	5.82	0	1	O	0	0	0.00
Instn8t5k3_2				705	O	5.82	5.82	0	0	O	0	0	0.00
Instn8t5k3_4				496	O	6.30	6.30	0	0	O	0	0	0.00
Instn10t5k3_3	9	3	1206	1208	O	6.76	6.76	0	9	O	0	6	0.05
Instn10t5k3_1				1058	O	6.86	6.86	0.01	17	O	0.01	7	0.57
Instn10t5k3_2				1045	O	6.90	6.90	0.01	26	O	0	6	0.00
Instn10t5k3_4				1037	O	6.90	6.90	0.01	57	O	0	7	0.00
Instn12t5k3_6	11	3	851	1521	O	4.47	4.47	0.01	22	O	0	6	0.00
Instn12t5k3_5				1491	O	4.47	4.47	0.01	31	O	0	7	0.00
Instn12t5k3_7				1399	O	4.47	4.47	0.01	37	O	0	9	0.00
Instn12t5k3_8				1146	O	4.47	4.47	0.01	26	O	0	9	0.00
Instn12t5k3_4				925	O	4.47	4.47	0.01	44	O	0	8	0.00
Instn12t5k3_2				802	O	4.50	4.50	0.01	102	O	0.01	30	0.67
Instn12t5k3_3				748	O	4.54	4.54	0.01	1062	O	0.01	29	0.47
Instn12t5k3_1				546	F	4.83	4.73	2.01	14400	O	0.01	1404	0.59
Instn16t5k3_3	15	3	1360	1240	O	5.62	5.62	0.01	52	O	0	22	0.00
Instn16t5k3_4				1232	O	5.62	5.62	0	17	O	0	26	0.00
Instn16t5k3_1				1056	O	5.62	5.62	0	31	O	0	33	0.00
Instn16t5k3_2				1030	O	5.63	5.63	0.01	2100	O	0.01	1743	0.40
Instn16t5k3_8				1027	O	5.63	5.63	0.01	5113	O	0.01	1356	0.40
Instn16t5k3_7				851	F	5.74	5.70	0.73	14400	O	0.01	8054	0.38
Instn16t5k3_5				802	F	5.76	5.72	0.76	14400	O	0	37	0.07
Instn16t5k3_6				757	F	5.78	5.75	0.45	14399	O	0	26	0.00
Instn50t5k4	49	4	4443	1111	F	18.69	12.48	33.25	14400	I	—	—	0.00
										Average Improvement			0.14

Table 5.5: Comparison between FPVRP and PVRP formulations with S2 instances.

On the other hand, Table 5.6 shows the results of the comparison between the FPVRP and the PVRP with the S3 instances. For the PVRP instances, computing times increased significantly in some of them. The exception was instance n10k5t5_1, for which no feasible PVRP solution could be found in the 14400 seconds allowed. The strengthened FPVRP formulation was not able to prove the optimality of the best solution found for any of the instances in the set, although it was always able to find a feasible solution. In general, the percentage optimality gaps at termination are small, with higher values (up to 5.51%) for the instances with the smallest radius $r = 0.15$. Despite not knowing whether or not the best-known FPVRP solutions are optimal, in all cases they produce substantial improvements in the routing costs with respect to the optimal PVRP solutions. In these instances the benefit of using FPVRP is more noticeable for instances of S2 because of their structure. In fact, the improvement is significant since it is allowed to partition customers demands into several periods. The range of such improvements is 3.03%- 12.27%, with an average improvement of 7.43% in comparison to the PVRP.

						FPVRP					PVRP			%*Imp*
Instance	r	$\|C\|$	$\|K\|$	Q	D_t	*Status*	*BestSol*	*BestLB*	*% Gap*	*Time*	*Status*	*% Gap*	*Time*	
n10k5t5_1	0.15	10	5	200	938	F	20.78	20.30	2.36	14400	I	—	14399	—
n10k4t5_2			4		664	F	12.44	12.12	2.64	14400	O	0.01	145	4.37
n10k5t5_3			5		874	F	13.23	12.87	2.80	14400	O	0	11	4.10
n10k4t5_4			4		738	F	13.53	13.05	3.68	14400	O	0.01	35	10.81
n10k8t5_5			8		1401	F	25.91	25.58	1.29	14400	O	0	1	6.60
n15k10t5_1		15	10	250	2317	F	34.26	33.39	2.61	14403	O	0.01	2445	5.55
n15k6t5_2			6		1495	F	17.42	16.51	5.51	14400	F	1.69	14399	12.27
n15k10t5_3			10		2484	F	25.25	24.45	3.27	14400	O	0.01	1213	10.12
n15k8t5_4			8		1767	F	32.11	31.25	2.75	14400	O	0.01	4587	5.81
n15k7t5_5			7		1606	F	23.89	22.84	4.60	14400	F	1.54	14399	7.28
n20k10t5_1		20	10	300	2813	F	24.58	23.58	4.24	14402	F	2.33	14399	5.25
n20k12t5_2			12		3408	F	36.10	35.45	1.83	14400	F	0.48	14399	7.71
n20k11t5_3			11		3138	F	23.70	22.79	3.99	14404	F	2.47	14399	11.03
n20k10t5_4			10		2957	F	35.35	34.53	2.37	14401	F	4.81	14399	10.34
n20k10t5_5			10		2874	F	29.45	28.89	1.94	14399	F	4.31	14399	6.36
n10k6t5_1	0.3	10	6	200	1028	F	19.04	18.97	0.37	14400	O	0.01	79	5.58
n10k6t5_2			6		1176	F	13.89	13.59	2.21	14399	O	0.01	122	11.05
n10k5t5_3			5		922	F	14.50	14.16	2.40	14400	O	0.01	269	7.03
n10k5t5_4			5		820	F	14.38	14.04	2.42	14400	O	0	19	4.39
n10k8t5_5			8		1436	F	19.40	19.34	0.31	14400	O	0.01	8	10.08
n15k9t5_1		15	9	250	2116	F	27.16	26.64	1.95	14400	O	0.01	105	3.03
n15k9t5_2			9		2225	F	29.72	29.19	1.82	14401	O	0.01	161	9.60
n15k7t5_3			7		1692	F	27.84	27.38	1.68	14400	F	1.68	14399	4.57
n15k7t5_4			7		1530	F	17.43	17.06	2.17	14401	O	0.01	8992	4.58
n15k6t5_5			6		1488	F	20.59	20.18	2.03	14400	F	0.88	14399	7.70
n20k10t5_1		20	10	300	2824	F	29.57	28.97	2.07	14400	F	3.44	14399	9.41
n20k12t5_2			12		3537	F	31.55	30.78	2.50	14400	F	0.58	14399	5.94
n20k10t5_3			10		2849	F	26.07	25.18	3.53	14400	F	4.27	14399	8.89
n20k13t5_4			13		3691	F	42.61	41.62	2.38	14407	F	1.47	14399	8.07
n20k12t5_5			12		3308	F	34.05	33.76	0.86	14400	F	2.30	14403	8.41
n20k14t5_1	0.5	20	14	300	4076	F	32.30	31.57	2.31	14400	F	0.69	14400	6.32
n20k10t5_2			10		2786	F	29.33	28.77	1.95	14400	F	5.46	14399	9.55
n20k7t5_3			7		2059	F	23.25	22.68	2.51	14400	F	1.61	14399	5.12
n20k10t5_4			10		2825	F	24.80	24.02	3.25	14400	F	4.09	14399	11.45
n20k11t5_5			11		3043	F	36.45	35.50	2.68	14400	F	2.38	14399	4.22
													Average Improvement	7.43

Table 5.6: Comparison between FPVRP and PVRP formulations with S3 instances.

5.3.4 Analysis of the FPVRP benefits

This section focuses on the analysis of the benefits of the FPVRP with respect to the FPVRP-IC and to the PVRP that emerged in the computational experiments described in the previous section. To this aim, the improvements of the FPVRP objective function relative to the compared models are summarized for different instance parameters. First we focus on the comparison between the FPVRP and the FPVRP-IC on the instances of set S1. Results are shown in Table 5.7 where instances are clustered on the basis of the number of customers, first, and fleet size, second. For each cluster of instances, the average and maximum improvements achieved by the FPVRP are reported.

$\|C\|$	Avrg. Improvement	Max. Improvement
5	1.73%	8.87%
10	3.22%	6.37%
15	3.07%	9.48%
20	2.23%	7.91%
$\|K\|$	Avrg. Improvement	Max. Improvement
2	3.11%	8.87%
3	2.02%	9.48%

Table 5.7: Improvement of FPVRP versus FPVRP-IC on S1 instances.

Table 5.7 shows that the improvements increase with the number of customers except for the case with $|C| = 20$, while they slightly decrease when the number of vehicles increases. The maximum improvement is 9.48%.

Now, the benefits of the FPVRP with respect to the PVRP are analyzed. Table 5.8 summarizes the results for the instances of set S2. Instances are clustered by number of customers only as all instances have the same number of vehicles. Instance Instn50t5k4 is not considered in this analysis.

$\|C\|$	Avrg. Improvement	Max. Improvement
7	0.00%	0.00%
9	0.16%	0.57%
11	0.43%	0.67%
15	0.21%	0.40%

Table 5.8: Improvement of FPVRP versus PVRP on S2 instances.

The results show that the FPVRP benefits are quite fluctuating with respect to the number of customers. Nevertheless, the FPVRP still shows slight improvements with respect to the PVRP.

Finally, Table 5.9 focuses on the comparison between the FPVRP and the PVRP on the instances of Set S3. Instances are clustered by the number of customers and radius. The number of vehicles for these instances is not considered as it varies with the number of customers.

$\|C\|$	Avrg. Improvement	Max. Improvement
10	7.11%	11.05%
15	7.05%	12.27%
20	7.87%	11.45%
r	Avrg. Improvement	Max. Improvement
0.15	7.69%	12.27%
0.30	7.22%	11.05%
0.50	7.33%	11.45%

Table 5.9: Improvement of FPVRP versus PVRP on S3 instances.

For the instances of set S2 it can be noticed that the benefits are quite fluctuating with respect to the number of customers and the radius. The maximum savings are 12.27%.

When comparing the benefits of the FPVRP versus the FPVRP–IC and PVRP, respectively, it can be noticed that much larger improvements are achieved by the FPVRP with respect to the PVRP than with respect to FPVRP–IC. This is clearly due to the way the instances are constructed and also to the fact that the PVRP suffers from the rigidity derived from fixed schedules and fixed delivery quantities. Part of this rigidity is overcome with the FPVRP–IC. However, the improvements of the FPVRP versus the FPVRP–IC show that inventory constraints may remarkably affect solution costs.

Finally, additional computational experiments were run in order to investigate the impact of customer demands on the benefits produced by the PVRP. The results confirm that the highest savings can be attained when the capacity is tight and demands are rather homogeneous, whereas the smallest savings are obtained with larger capacities and high variations in the demands. A similar observation was made in Archetti et al. (2008b) concerning the benefits of the SDVRP with respect to the VRP.

Chapter 6

A two-phase solution algorithm

As it was seen in the previous chapter, the exact solution algorithms based on the mathematical formulations can be highly demanding in terms of the computing time they may require, particularly as the size of the instances increases. In this chapter, an approximate solution method that can be used to solve medium to large size FPVRP instances is proposed.

6.1 Description of the algorithm

We developed a two–phase solution algorithm, classified as a *matheuristic*, that operates according to an iterative scheme. At each iteration a MILP is solved to determine a plan for the periods to visit the customers and their corresponding quantities. Then suitable routes consistent with the distribution plan are designed with a TS heuristic. Figure 6.1 shows the main components of the proposed approach.

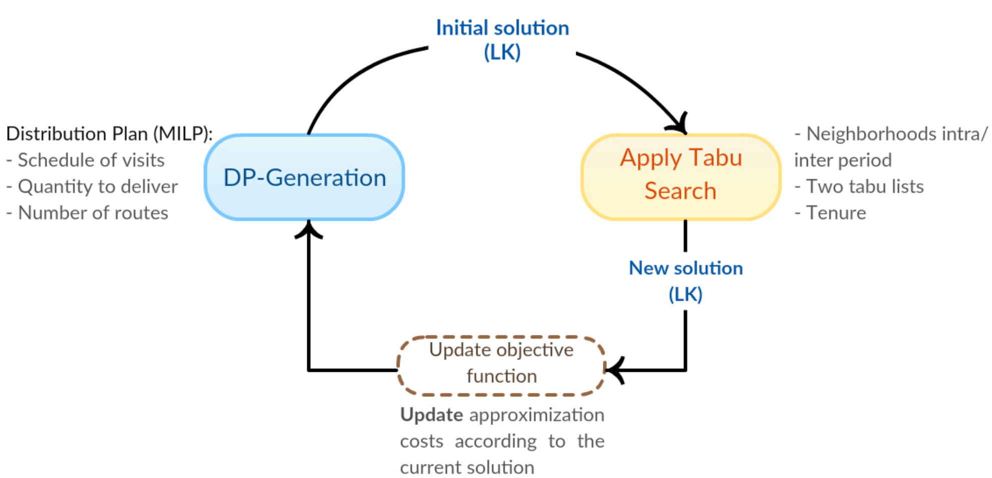


Figure 6.1: A two-phase heuristic scheme to solve the FPVRP.

These two main phases are described below:

1. *Phase 1 – Distribution Plan (DP) Generation*: Builds an initial feasible solution by solving independent subproblems limited to only a subset of the FPVRP decisions.
2. *Phase 2 – Tabu Search*: Improving phase, which applies a TS algorithm taking as input the solution produced by Phase 1.

Algorithm 1 shows the general framework of the proposed approach.

Algorithm 1 Two-phase matheuristic

1: $\hat{C}_i^t = 1 \quad i \in C, t \in T$
2: $\lambda_t = 1 \quad t \in T$
3: $s^* \leftarrow \emptyset$
4: `BestSolCost = BigNumber`
5: **while** *a stopping condition is not true* **do**
6:     $s \leftarrow$ `DP-Generation(`$\hat{C}, \lambda$`)`
7:     $\bar{s} \leftarrow$ `TS(`s`)`
8:     **if** $f(\bar{s}) <$ `BestSolCost` **then**
9:         `BestSolCost` $= f(\bar{s})$
10:         $s^* \leftarrow \bar{s}$
11:     **end if**
12:     Update $\hat{C}$
13: **end while**
14: return (s^*)

The two main phases (lines 6 and 7) are applied iteratively until a stopping criterion is met. At each iteration, in the first phase, an initial solution s is obtained by applying the `DP-Generation`. Then, in the second phase, a TS procedure is applied to s to obtain a new solution $\bar{s}$. If $\bar{s}$ improves the solution cost of s^*, then s^* is updated. Values of $\hat{C}$ and λ, where

- $\hat{C}_i^t$ is the approximated routing cost for visiting customer i in period t, and
- λ_t is a parameter greater than or equal to 1 used to penalize infeasibility at time t,

are set as follows. At the beginning of the matheuristic they are set to 1. At the following iterations (line 12), if the solution cost $f(\bar{s})$ is different from the solution cost obtained in the previous iteration, then the removal savings are computed, i.e., if i is visited in period t then $\hat{C}_i^t = c_{\rho s} - (c_{\rho i} + c_{i\varsigma})$, where ρ and ς denote the predecessor and successor of customer i in its route in period t. If i is not visited at t, $\hat{C}_i^t$ is equal to the cheapest insertion cost of i at time t. On the other hand, if $f(\bar{s})$ remains the same, the values of $\hat{C}$ are generated randomly between [-1,-100]. The rationale for this is to provide different initial solutions to the TS performed in Phase 2. The value of coefficients λ remains the same as the ones computed in the last call to the `DP-Generation` phase. At the end of the matheuristic, the best solution found is reported as the final solution. Below, each phase is described in detail.

6.1.1 Phase 1: DP-Generation

The first phase aims at building an initial feasible solution to the problem. Note that three main decisions have to be taken when dealing with the FPVRP:

1. *Visiting periods*: The periods at which each customer is visited.
2. *Delivered quantities*: The quantity to deliver to each customer at each visit.
3. *Routing*: Vehicle routes at each time period, i.e., determining the assignment of customers to vehicles and, for each vehicle, the sequence in which customers must be visited.

The `DP-Generation` phase works in two steps. In the first step, it builds a distribution plan handling the first two decisions, i.e., it determines the visiting periods for each customers and the delivered quantities. The DP is then taken as input to the second step which builds vehicle routes.

In particular, the first step consists in solving a MILP called, from now on, `DP-MILP`, which determines the visiting periods (*calendar*) and the quantities to be delivered to all customers.

The formulation of the `DP-MILP` is as follows:

$$\min \quad \sum_{t\in T}\sum_{i\in C}\hat{C}_i^t z_i^t + z^t Q\lambda_t \tag{6.1}$$

$$s.t. \quad q_i^t \le w_i z_i^t \qquad i\in C, t\in T \tag{6.2}$$

$$\sum_{i\in C} q_i^t \le \left\lfloor \frac{Q}{\lambda_t} \right\rfloor z^t \qquad t\in T \tag{6.3}$$

$$\sum_{t \in T} q_i^t = W_i \qquad i \in C \tag{6.4}$$

$$z^t \leq m \qquad t \in T \tag{6.5}$$

$$z_i^t \leq 1 \qquad i \in C, t \in T \tag{6.6}$$

$$q_i^t \geq 0 \qquad i \in C, t \in T \tag{6.7}$$

$$z_i^t \in \{0, 1\} \qquad i \in C, t \in T \tag{6.8}$$

$$z^t \in \mathbb{Z} \qquad t \in T. \tag{6.9}$$

Variables z and q have the same meaning as described in Chapter 4.2. The objective function (6.1) aims at minimizing the sum of the approximated routing costs and the number of vehicles used. Constraints (6.2) establish the maximum deliverable quantity to each customer while (6.3) are vehicle capacity constraints. In particular, constraints (6.3) are aggregated vehicle capacity constraints that fix the maximum quantity that can be delivered in each time period. This maximum amount corresponds to $\left\lfloor \frac{Q}{\lambda_t} \right\rfloor$ multiplied by the number of vehicles used, with $\lambda_t \geq 1$. Note that a solution satisfying constraints (6.3) may not produce a feasible FPVRP solution because it may not exist a feasible way to pack quantities q_i^t into z^t vehicles. The total demand of each customer is satisfied through constraints (6.4). Constraints (6.5) fix the maximum number of vehicles used to m while split deliveries are forbidden through (6.6) since at most one vehicle can serve each customer demand at each time period. Constraints (6.7)–(6.9) define the variable domain.

The solution of the `DP-MILP` determines, for each time period, the subset of visited customers and the amount delivered to each of them. This information provides the distribution plan. The DP is then taken as input to the second step which aims at building vehicle routes. In particular, the second step consists in solving a CVRP for each time period on the basis of the information provided by the DP. Each CVRP is solved by applying the Clarke and Wright (1964) heuristic implemented with the VRPH package of the Coin OR library (Groër et al., 2010). Given that this algorithm works for the case where there is no limit for the fleet size, it may obtain a solution where the number of vehicles used in a given period is higher than m. In this case, another iteration is made by updating the values of $\hat{C}$ and λ and solving the `DP-MILP` again. Finally, if after a certain number of iterations the solution is still infeasible, a procedure to recover feasibility in which customers of the surplus routes are reallocated in different periods, is applied. The scheme of the `DP-Generation` phase is sketched in Algorithm 2 where

- `DP-MILP`$(\hat{C}, \lambda)$ returns the optimal solution of the `DP-MILP` when the values of the approximated routing costs and the infeasibility penalties are specified by $\hat{C}$ and λ, respectively.
- `Routing`(DP) returns the solution of the CVRP for each time period on the basis of the distribution plan DP obtained through the VRPH package (Groër et al., 2010).
- `RecoverFeasibility(`s`)` transforms an infeasible solution s into a feasible one. This procedure works as follows.

  - Select one of the periods in which the number of routes is more than m.
  - Select the route with fewer customers and sequentially remove customers by redistributing in other periods the quantity they received.
  - If all customers of the selected route are reallocated, remove the empty route and follow the same procedure until the number of vehicles used is at most m for all periods.

- `LK`(s) returns an improved solution by applying the Lin-Kernighan algorithm (Lin and Kernighan, 1973, LK) to each route of the solution s. The implementation code for this routine is provided in `http://www.akira.ruc.dk/~keld/research/LKH/` (Helsgaun, 2000).
- s is the solution obtained at the end of the second step.

As shown in Algorithm 2, if an infeasible solution is obtained, the values of $\hat{C}$ and λ are updated (line 7) as follows:

a) $\hat{C}$: if customer i is visited at time t, then $\hat{C}_i^t$ is equal to the removal savings $\hat{C}_i^t = (c_{\rho i} + c_{i\varsigma}) - c_{\rho\varsigma}$, where ρ and ς denote the predecessor and successor of customer i in its route at period t. Instead, if customer i is not visited at time t, then $\hat{C}_i^t$ is equal to the cheapest insertion cost of i at time t.

Algorithm 2 `DP-Generation`($\hat{C}, \lambda$)

1: **while** *a stopping condition is not true* **do**
2: $DP \leftarrow$`DP-MILP`($\hat{C}, \lambda$)
3: $s \leftarrow$ `Routing`(DP)
4: **if** s is a feasible FPVRP solution **then**
5: Stop and Go to line 13
6: **else**
7: Update the values of $\hat{C}$ and λ
8: **end if**
9: **end while**
10: **if** s is not feasible **then**
11: `RecoverFeasibility`(s)
12: **end if**
13: Return `LK`(s)

b) λ: if at t the number of vehicles is not greater than m, then the value of λ_t remains unchanged, otherwise it is increased by ϵ, i.e., $\lambda_t = \lambda_t + \epsilon$. If the `DP-MILP` becomes infeasible (because of a very large value of λ_t), then, $\lambda_t = \max\{1, \lambda_t - \epsilon\}$.

Note that, once a feasible solution s is obtained, the LK algorithm is applied to each route in s in an attempt to reduce the routing cost (line 13).

6.1.2 Phase 2: Tabu Search heuristic

The aim of the Phase 2 is to improve the solution obtained at the end of the first phase. The idea is to define different neighborhoods and embed them in a TS scheme where each selected move is recorded and considered *tabu* for a certain number of iterations (tenure) to avoid cycling. Each neighborhood is explored exhaustively (best improvement), unless a selected move improves the incumbent (aspiration criterion). In that case the exploration stops and the improved solution is chosen as the next solution. Otherwise, the best non–tabu move found among all neighborhoods will be chosen. The selected move is considered *tabu* for a certain number of iterations.

Let $\mathcal{N} = \{N_1, ..., N_l\}$ be the set of neighborhoods with $|\mathcal{N}| = l$. The proposed TS scheme is shown in Algorithm 3.

Algorithm 3 TS(s)

Require: Initial Solution s

Ensure: Best solution found s^*

1: `BestSolCost` $= f(s)$, Iter $= 1$
2: $s \leftarrow$ `SplitOperator(s)`
3: **while** *a stopping condition is not true* **do**
4: `BestLocalCost` $=$ `BigNumber`
5: `Tenure` $=$ `computeTenure()`
6: $\tilde{l} = 1$
7: **repeat**
8: $s' \leftarrow$ Explore Neighborhood $N_{\tilde{l}}(s)$
9: **if** $f(s') <$ `BestLocalCost` **then**
10: `BestLocalCost`$= f(s')$
11: $\tilde{s} \leftarrow s'$
12: **if** $f(\tilde{s}) <$ `BestSolCost` **then**
13: Go to line 18
14: **end if**
15: **end if**
16: $\tilde{l} = \tilde{l} + 1$
17: **until** $\tilde{l} \leq l_{\max}$
18: Update tabu list `TL(`$\tilde{s}$`)` $=$ Iter $+$ `Tenure`
19: $\tilde{s} \leftarrow$ `LK`($\tilde{s}$), Iter $=$ Iter $+ 1$
20: **if** $f(\tilde{s}) <$ `BestSolCost` **then**
21: `BestSolCost` $= f(\tilde{s})$
22: $s^* \leftarrow \tilde{s}$
23: **end if**
24: **end while**
25: Return s^*

The TS begins with an initial feasible solution obtained in Phase 1. If possible, the routes of this solution are split using the `SplitOperator` in order to increase the possibility of applying moves which may improve the solution during the search. The resulting solution is taken as the initial solution to be applied for all neighborhoods in $\mathcal{N}$. The neighborhoods are explored independently and the best solution among all of them is selected. The corresponding move is considered *tabu* for a certain number of iterations (line 18). Then, the LK algorithm is applied to the best solution found (line 19). This procedure stops when a stopping criterion is reached (line 3). The three main ingredients of the proposed TS are: the `SplitOperator`, the set of neighborhoods $\mathcal{N}$ and the tabu lists. They are explained below.

A) The `SplitOperator`: This operator splits one route into two without increasing the solution cost. This situation happens when a route travels through an edge (i, j) whose cost c_{ij} is equal to $c_{i0} + c_{0j}$. In this case, the route is split in two smaller routes, traversing the edge $(i, 0)$ (first route) and the edge $(0, j)$ (second route). This is done only if the total number of routes used is lower than mH. The idea behind this operator is to create routes with a larger residual capacity to allow a wide range of modifications when the neighborhoods described in the following are applied. When a route at time t is split, two situations may happen:

- **The number of routes used at time period t is lower than m.** In this case, the two new routes are both performed at time period t and no further change is made.
- **The number of routes used at time period t is m.** In this case, we cannot assign both new routes to time period t. Thus, at least one route must be performed in a different day. Let us define:
  - r_1 and r_2: the two routes obtained from the splitting.
  - S_r: subset of customers served in route r.
  - $S(t)$: subset of customers served at time period t.
  - $\tilde{q}_r$: residual capacity of route r.
  - r_i^t: route serving customer i at time period t.

  Then, a route r can be moved from period t to period $t' \neq t$ if $S_r \cap S(t') = \emptyset$ or the following holds for each customer $i \in S_r \cup S(t')$:

  1. $q_i^t + q_i^{t'} \leq w_i$ and

2. either $q_i^t \leq \tilde{q_{r_i^{t'}}}$ or $q_i^{t'} \leq \tilde{q}_r$.

Thus, if either r_1 or r_2 satisfies the above conditions the split is performed, otherwise it is discarded.

B) Neighborhoods $\mathcal{N}$: They are listed in the order they are applied according to Algorithm 3.

1. *Intra-period moves*: The following moves are applied to each period independently.
   (a) *1-move* (N_1): Consider a customer visited in period t. Remove the customer from the route that currently serves it and insert it in another route, using the cheapest insertion rule. The best route (in terms of insertion cost) that can feasibly accommodate the quantity delivered to the customer is chosen.
   (b) *1-swap* (N_2): Consider two customers i and j served in two different routes, r and r', in period t and swap the two customers. The swap is made as follows. First, remove both customers from their current route and then insert them in the new route through the cheapest insertion method, as done in the *1-move*.
2. *Inter-period moves*: These moves are applied to pairs of periods t and t' in order to change the visit plan of customers. Similarly to the intra-period moves, there are two neighborhoods considered.
   (a) *1-move* (N_3): Consider a customer visited in period t. Remove the customer from the route that currently serves it and insert it in a route in period t', using the cheapest insertion rule as done in the *1-move* intra-period.
   (b) *1-swap* (N_4): Consider customer i served in t and customer j served in t' and swap them. The swap is made as follows. First, remove both customers from their current route. The insertion is made by applying the cheapest insertion criterion to all routes performed in the period where the customer has to be inserted choosing the best one.

In any of the above mentioned neighborhoods, every time a customer is removed from a route and inserted into another one, the same quantity delivered in the original route is moved to the newly assigned route, if this is feasible. If this is not feasible (either because the vehicle capacity or the customer capacity are exceeded), then the excess quantity is assigned to the other customer visits if feasible, i.e., if neither vehicle capacity nor customer capacity are exceeded and the move made in the corresponding period is not tabu. The assignment is done in chronological order, i.e., from the first to the last visit. If the excess quantity cannot be reassigned, then the move is infeasible and, thus, discarded.

For each neighborhood, all feasible non-tabu moves are evaluated and the best one is chosen, unless there is a move which improves the incumbent (aspiration criterion). In that case, the evaluation stops (for all neighborhoods) and the solution obtained with that move is chosen as the best one. That is, the algorithm exits from the loop in lines 7–17 in Algorithm 3 and goes to line 19. If this is not the case, once all the neighborhoods are explored, the best non-tabu solution is chosen and it becomes tabu for a certain number of iterations as detailed in Algorithm 3.

C) Tabu lists: There are two different tabu lists, one for intra–period moves and another one for inter–period moves:

- Intra–period list $\texttt{TL}(i, r, t)$: If customer i is removed from route r in period t, then it is tabu to reinsert i in r in period t for a certain number of iterations.
- Inter–period list $\texttt{TL}(i, t)$: If customer i is removed from period t, then it is tabu to reinsert i in period t for a certain number of iterations.

The number of iterations for which a move remains tabu is determined by the *tabu tenure*.

6.2 Computational experience

In this chapter, we present and analyze the results obtained with the computational experiments that were performed in order to test the behavior of the matheuristic with different sets of instances. The computational environment and the benchmark instances used for the tests are first described in Section 6.2.1. Then, Section 6.2.2 describes the tests run in order to calibrate the parameters of the matheuristic. Finally, Section 6.2.3 presents and analyzes the results of several tests that were run to show the efficiency of the proposed solution approach.

6.2.1 Initial data

The matheuristic was implemented in C++ and for the `DP-MILP` the ILOG Concert Technology API (CPLEX 12.5.0.0) was used. All tests were carried out on a HP Intel(R)-Xeon(R) 2.4GHz Workstation with 32GB RAM (Win Server 2012, 64 bits). The set of instances used for the tests of this chapter are S3, S4, and S5. Note that for S3, all the instances have been slightly modified, in particular, they differ in the number of available vehicles which is now computed as $m = 1 + \frac{\sum_{i \in C} w_i f_i}{QH}$ (the same as S4) while in Chapter 5.3 it was set to $m = \frac{\sum_{i \in C} w_i}{Q}$. The reason for this change is that the number of vehicles used in the optimal solutions was much smaller than the number of available vehicles. Table 6.1 shows the detailed information about instances used in this chapter including the modifications made for S3.

Set	r	**Instance**	$\|N\|$	$\|T\|$	f_i	Q	$\|K\|$	$\sum_{i \in C} W_i$
S3	0.15	**n10k5t5_1**	11	5	2,3,5	200	5	3760
		n10k4t5_2					4	2640
		n10k5t5_3					4	2869
		n10k4t5_4					4	2363
		n10k8t5_5					7	5107
		n15k10t5_1	16	5	2,3,5	250	9	9015
		n15k6t5_2					5	4684
		n15k10t5_3					7	7202
		n15k8t5_4					7	7175
		n15k7t5_5					6	5526
		n20k10t5_1	21	5	2,3,5	300	7	8842
		n20k12t5_2					9	11334
		n20k11t5_3					9	10957
		n20k10t5_4					9	11102
		n20k10t5_5					7	8976
	0.3	**n10k6t5_1**	11	5	2,3,5	200	5	3562
		n10k6t5_2					5	3922
		n10k5t5_3					5	3114
		n10k5t5_4					5	3038
		n10k8t5_5					7	5139
		n15k9t5_1	16	5	2,3,5	250	7	6877
		n15k9t5_2					8	7642
		n15k7t5_3					6	5973
		n15k7t5_4					5	4621
		n15k6t5_5					6	4947
		n20k10t5_1	21	5	2,3,5	300	8	9551
		n20k12t5_2					9	11675
		n20k10t5_3					7	8047
		n20k13t5_4					10	12723
		n20k12t5_5					9	11996
	0.5	**n20k14t5_1**	21	5	2,3,5	300	10	13225
		n20k10t5_2					7	8862
		n20k7t5_3					6	7490
		n20k10t5_4					7	8373
		n20k11t5_5					9	11437
S4	0.15	**n50k10t5_1**	51	5	2,3,5	500	10	22226
		n50k8t5_2					8	15690
		n50k9t5_3					9	19145
		n50k9t5_4					9	19737
		n50k11t5_5					11	23282
		n100k17t5_1	101	5	2,3,5	500	17	38700
		n100k18t5_2					18	40870
		n100k20t5_3					20	46018
		n100k21t5_4					21	48959
		n100k18t5_5					18	40782
S5		**p01**	51	2	1	160	2	937
		p14	21	4	1,2,4	20	2	120
		p15	39	4	1,2,4	30	2	200
		p16	57	4	1,2,4	40	2	280
		p32	154	6	2,3,5	20	9	1134

Table 6.1: Instance information.

6.2.2 Calibration of parameters

Several preliminary tests were run in order to establish the best setting for the matheuristic. The following parameter values were used:

- **Overall parameters:**
  - The overall solution algorithm is repeated for five global iterations, provided that it does not exceed a maximum computing time of 14400 seconds. That is, the algorithm stops when one of the two conditions is verified.
- **DP-Generation:**
  - The maximum number of `DP-MILP` solved in Algorithm 2 is set to 5.
  - The maximum number of iterations of the initialization is set to 5.
  - $\epsilon = 0.1$.
- **TS:**
  - The maximum number iterations without improvement is `IterNImp = 15n`.
  - `Tenure`=10.

The maximum number of `DP-MILP` solved in Algorithm 2 indicates the maximum number of times the loop in lines 1–9 of Algorithm 2 is repeated. The limit will not be reached if the current `DP-MILP` produces a feasible solution. Parameter ϵ is used when updating the value of λ_t as specified in Section 6.1.1. Concerning the TS, the stopping criterion is reached when the maximum number of iterations without improvement is equal to $15n$ or when the time limit is reached.

Test 1: Neighborhood performance

A first test was carried out to analyze the performance of each of the neighborhoods used in the TS. Only a small subset of the benchmark instances was used. The results are summarized in Table 6.2. Columns in blocks (N_1 - N_4) show the frequency of use (*Freq*), computed as the number of iterations when a move from that neighborhood was chosen as the best one, and the computing time (*Time*) required by each neighborhood in the overall computation. According to the results, N_4 is the most resource-consuming neighborhood for the TS, with a small average frequency of use of 77.43 but a high computation time requirement of 2143.21 seconds on average. This is more than 8 times the computing time of using N_3, more than 7 times of N_2 and more than 64 times of N_1.

		N_1		N_2		N_3		N_4		*Total*	
Instance	$\|N\|$	*Freq*	*Time*	*Freq*	*Time*	*Freq*	*Time*	*Freq*	*Time*	*Freq*	*Time*
p01	51	665	25.42	279	431.12	4901	51.31	79	3888.8	5924	4396.64
p14	21	0	1.17	0	22.70	1714	10.79	21	291.25	1735	325.91
p15	39	0	12.29	0	198.70	3053	65.33	32	2731.50	3085	3007.81
p16	57	0	36.09	0	854.75	4575	193.47	26	10775.00	4601	11859.30
p32	154	212	88.11	95	1563.71	1451	2282.47	36	6952.50	1794	10886.79
n10k5t5_1	11	197	3.42	347	7.76	651	10.24	46	46.18	1241	67.60
n15k10t5_1	16	764	29.07	255	43.09	642	67.57	41	138.76	1702	278.50
n20k10t5_1	21	292	52.71	96	153.56	2269	162.98	47	1178.06	2704	1547.31
n10k6t5_1	11	182	3.73	50	4.55	780	7.21	37	51.44	1049	66.93
n15k9t5_1	16	422	16.11	352	39.82	913	53.61	116	281.14	1803	390.67
n20k10t5_1	21	61	54.00	17	120.97	1742	123.96	11	673.29	1831	972.21
n20k14t5_1	21	89	55.99	26	113.64	1540	164.27	425	1108.97	2080	1442.86
n20k10t5_2	21	264	28.58	141	117.68	1738	88.87	111	852.90	2254	1088.02
n20k7t5_3	21	417	60.41	19	177.30	2235	137.02	56	1035.09	2727	1409.82
Average		254.64	33.36	119.79	274.95	2014.57	244.22	77.43	2143.21	2466.43	2695.74
Total		3565	467.08	1677	3849.33	28204	3419.08	1084	30005.37	34530	37740

Table 6.2: Individual neighborhood performance.

Given the above results, another set of tests was run, excluding neighborhood N_4. The results, which are summarized in Table 6.3, allow to compare the performance of the two-phase algorithm with and without the use of N_4. Column labeled *Best–Known* gives the value of the best-known solution for each instance. These values were obtained with the FPVRP formulation of Archetti et al. (2017a), except for instance p32, for which it was taken from Baldacci et al. (2011). The next two columns give the best solution values (*BestSol*) and the best lower bounds (*BestLB*) produced by the FPVRP formulation of Archetti et al. (2017a). Then, there are two blocks of three columns each, one labeled $\mathcal{N}$ and another one $\mathcal{N} \setminus N_4$, corresponding to the results of TS with and without the use of the interperiod swap neighborhood (N_4). Each block gives the values of the best solution produced by the corresponding version of TS, the computing times and the percentage gaps between the values of the best solutions obtained and the best-known values of column *Best–Known*. The latter are computed as follows.

$$Gap = \frac{Z_{\text{Heur}} - Z_{\text{BEST}}}{Z_{\text{Heur}}} \times 100\%, \tag{6.10}$$

where Z_{BEST} and Z_{Heur} are the best–known solution value and the one produced by the heuristic, respectively.

The results of Table 6.3 clearly indicate that removing N_4 from the TS reduces significantly the computing time (almost 44%) and the average solution quality is not affected; on the contrary, it improves. Therefore, all subsequent tests were carried out with TS without N_4.

			FPVRP		$\mathcal{N}$			$\mathcal{N} \setminus N_4$		
Instance	$\lvert N\rvert$	***Best–Known***	*BestSol*	*BestLB*	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***
p01	51	**524.61**	524.93	510.46	533	6389.90	**1.57%**	538.12	2699.76	**2.51%**
p14	21	**954.81**	954.81	954.71	954.80	525.41	**0.00%**	954.80	587.96	**0.00%**
p15	39	**1862.63**	1862.63	1825.04	1916.90	3797.25	**2.83%**	1890.70	1920.70	**1.48%**
p16	57	**2875.24**	2875.24	2814.29	2939.29	13705.90	**2.18%**	2981.45	3764.68	**3.56%**
p32	154	**78072.88**	—	—	86371.20	14454	**9.61%**	84763.20	14451.70	**7.89%**
n10k5t5_1	11	**20.79**	20.79	20.27	21.90	98.85	**5.10%**	21.48	54.26	**3.24%**
n15k10t5_1	16	**34.28**	34.28	33.48	35.08	324.41	**2.29%**	35.05	168.31	**2.21%**
n20k10t5_1	21	**24.58**	24.58	23.57	25.62	1653.20	**4.03%**	25.58	471.83	**3.89%**
n10k6t5_1	11	**19.04**	19.04	18.99	19.63	92.18	**2.99%**	19.57	54.00	**2.72%**
n15k9t5_1	16	**27.16**	27.16	26.65	27.17	425.34	**0.07%**	27.53	142.75	**1.37%**
n20k10t5_1	21	**29.59**	29.59	29.06	31.77	1021.75	**6.86%**	31.05	391.41	**4.71%**
n20k14t5_1	21	**32.30**	32.30	31.61	33.05	1484.67	**2.27%**	32.98	423.16	**2.05%**
n20k10t5_2	21	**29.37**	29.37	28.78	30.46	1419.93	**3.58%**	30.47	615.46	**3.61%**
n20k7t5_3	21	**23.25**	23.25	22.65	24.09	1826.29	**3.49%**	24.14	705.16	**3.68%**
			Average			3372.79	**3.35%**		1889.37	**3.07%**

Table 6.3: TS performance with/without N_4.

Test 2: Tenure selection

Another preliminary test was run to analyze the impact of the `Tenure` value in the performance of the TS. The following options of `Tenure` were considered:

- `Tenure1` = 10.

- `Tenure2` $= 5 + \lfloor \mathrm{randN}(1, \alpha \cdot \sqrt{|C|R(s')}) \rfloor$, similar to the proposed by Archetti et al. (2012).
- `Tenure3` $= \lfloor \alpha \cdot (|N| + \mathrm{randN}(1, \sqrt{|C|R(s')})) \rfloor$.
- `Tenure4` $= \lfloor \mathrm{randN}(1, \alpha \cdot \texttt{IterNImp}) \rfloor$.

In all cases $\alpha = 0.5$. Table 6.4 shows that the best average results are obtained using `Tenure3` with 2.44% in 1984.87 seconds on average. `Tenure4` is the second best option with a percentage gap of 2.55% in 1740.23 seconds. Even though the results of instance p32 are not considered (the atypical values), `Tenure3` is still the best option among all candidates. It is clear that `Tenure1` and `Tenure2` produced worse results in terms of average percentage gaps. Moreover, unlike the remaining tenure values considered, `Tenure3` only needs the calibration of the value of α, and does not depends on any constant apart from those related to the instance, which reduces the complexity of its calibration process. Hence, the formula for the `Tenure3` is the one considered for further tests.

			FPVRP		`Tenure1`			`Tenure2`			`Tenure3`			`Tenure4`		
Instance	$\|N\|$	***Best Known***	*BestSol*	*BestLB*	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***
p01	51	**524.61**	524.93	510.46	538.12	2699.76	**2.51%**	530.32	3835.95	**1.08%**	532.60	2528.45	**1.50%**	555.37	855.19	**5.54%**
p14	21	**954.81**	954.81	954.71	954.80	587.96	**0.00%**	956.69	711.65	**0.20%**	954.80	943.83	**0.00%**	958.58	264.81	**0.39%**
p15	39	**1862.63**	1862.63	1825.04	1890.70	1920.70	**1.48%**	1915.01	1624.55	**2.74%**	1888.82	2125.45	**1.39%**	1898.05	1990.75	**1.87%**
p16	57	**2875.24**	2875.24	2814.29	2981.45	3764.68	**3.56%**	2961.99	4075.00	**2.93%**	2875.24	3856.35	**0.00%**	2902.29	3796.90	**0.93%**
p32	154	**78072.88**	—	—	84763.20	14451.70	**7.89%**	89303.70	14453.30	**12.58%**	88813.70	14439.80	**12.09%**	86712.20	14444.20	**9.96%**
n10k5t5_1	11	**20.79**	20.79	20.27	21.48	54.26	**3.24%**	21.48	53.44	**3.24%**	21.55	61.72	**3.53%**	21.15	60.48	**1.70%**
n15k10t5_1	16	**34.28**	34.28	33.48	35.05	168.31	**2.21%**	34.99	195.79	**2.02%**	34.83	191.03	**1.58%**	34.94	207.46	**1.89%**
n20k10t5_1	21	**24.58**	24.58	23.57	25.58	471.83	**3.89%**	24.86	508.36	**1.13%**	24.87	597.14	**1.15%**	24.94	536.79	**1.43%**
n10k6t5_1	11	**19.04**	19.04	18.99	19.57	54.00	**2.72%**	19.54	77.88	**2.58%**	19.58	57.48	**2.76%**	19.43	59.07	**2.03%**
n15k9t5_1	16	**27.16**	27.16	26.65	27.53	142.75	**1.37%**	27.24	192.75	**0.32%**	27.27	209.55	**0.43%**	27.18	162.94	**0.07%**
n20k10t5_1	21	**29.59**	29.59	29.06	31.05	391.41	**4.71%**	31.09	514.27	**4.84%**	30.26	719.13	**2.23%**	29.97	465.81	**1.26%**
n20k14t5_1	21	**32.30**	32.30	31.61	32.98	423.16	**2.05%**	32.98	634.8	**2.05%**	32.60	665.33	**0.90%**	32.93	448.35	**1.90%**
n20k10t5_2	21	**29.37**	29.37	28.78	30.47	615.46	**3.61%**	30.69	658.06	**4.30%**	30.44	763.94	**3.52%**	29.91	496.46	**1.79%**
n20k7t5_3	21	**23.25**	23.25	22.65	24.14	705.16	**3.68%**	24.61	1057.32	**5.51%**	24.00	628.99	**3.11%**	24.47	574.00	**4.98%**
			Average			1889.37	**3.07%**		2042.37	**3.25%**		1984.87	**2.44%**		1740.23	**2.55%**
			Average[1]			923.03	**2.69%**		1087.68	**2.53%**		1026.80	**1.70%**		763.00	**1.98%**

[1] Average values without considered the atypical results obtained for the p32 instance.

Table 6.4: Comparison among different `Tenure` options.

TEST 3: CALIBRATION OF α

Finally, an evaluation of different values of α was considered to calibrate `Tenure3`. Five different values were compared, $\alpha \in \{0.10, 0.30, 0.50, 0.70, 1.0\}$. The results presented in Table 6.5 show that $\alpha = 0.7$ outperforms the remaining options, with an average percentage gap of 1.92%. Thus, $\alpha = 0.7$ was the value selected for the following tests.

			FPVRP		$\alpha = 0.10$			$\alpha = 0.30$			$\alpha = 0.50$			$\alpha = 0.70$			$\alpha = 1.0$		
Instance	$\|N\|$	***Best–Known***	*BestSol*	*BestLB*	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***	*BestSol*	*Time*	***Gap***
p01	51	**524.61**	524.93	510.46	549.31	1963.55	**4.50%**	531.14	3810.42	**1.23%**	532.60	2528.45	**1.50%**	535.90	2964.23	**2.11%**	550.36	1454.59	**4.68%**
p14	21	**954.81**	954.81	954.71	958.58	269.11	**0.39%**	956.69	443.93	**0.20%**	954.80	943.83	**0.00%**	954.80	487.96	**0.00%**	954.80	532.91	**0.00%**
p15	39	**1862.63**	1862.63	1825.04	1915.01	1599.29	**2.74%**	1924.24	1668.07	**3.20%**	1888.82	2125.45	**1.39%**	1862.62	2101.34	**0.00%**	1862.62	2607.29	**0.00%**
p16	57	**2875.24**	2875.24	2814.29	2960.87	3040.32	**2.89%**	2902.29	3351.63	**0.93%**	2875.24	3856.35	**0.00%**	2875.24	3485.94	**0.00%**	2875.24	7210.40	**0.00%**
p32	154	**78072.88**	—	—	86458.30	14450.40	**9.70%**	88181.50	14436.30	**11.46%**	88813.70	14439.80	**12.09%**	86389.80	14452.30	**9.63%**	86604.70	14451.90	**9.85%**
n10k5t5_1	11	**20.79**	20.79	20.27	22.01	49.66	**5.55%**	22.01	51.12	**5.55%**	21.55	61.72	**3.53%**	21.34	56.89	**2.61%**	21.34	55.00	**2.61%**
n15k10t5_1	16	**34.28**	34.28	33.48	35.28	151.59	**2.84%**	34.82	241.21	**1.55%**	34.83	191.03	**1.58%**	34.97	188.84	**1.98%**	35.14	242.49	**2.46%**
n20k10t5_1	21	**24.58**	24.58	23.57	26.23	446.75	**6.29%**	24.95	498.77	**1.47%**	24.87	597.14	**1.15%**	24.83	767.49	**1.01%**	24.76	918.08	**0.70%**
n10k6t5_1	11	**19.04**	19.04	18.99	19.97	41.15	**4.68%**	19.39	79.28	**1.83%**	19.58	57.48	**2.76%**	19.42	64.62	**1.95%**	19.35	57.33	**1.60%**
n15k9t5_1	16	**27.16**	27.16	26.65	27.95	126.93	**2.83%**	27.24	161.15	**0.32%**	27.27	209.55	**0.43%**	27.24	177.41	**0.31%**	27.18	154.50	**0.08%**
n20k10t5_1	21	**29.59**	29.59	29.06	32.07	391.28	**7.74%**	31.13	561.54	**4.94%**	30.26	719.13	**2.23%**	30.65	623.25	**3.48%**	30.61	512.78	**3.35%**
n20k14t5_1	21	**32.30**	32.30	31.61	34.05	406.41	**5.14%**	32.60	505.80	**0.90%**	32.60	665.33	**0.90%**	32.59	583.72	**0.89%**	32.72	463.25	**1.27%**
n20k10t5_2	21	**29.37**	29.37	28.78	30.65	460.35	**4.19%**	30.09	473.00	**2.40%**	30.44	763.94	**3.52%**	29.78	562.79	**1.37%**	30.19	537.76	**2.71%**
n20k7t5_3	21	**23.25**	23.25	22.65	24.81	538.86	**6.28%**	23.91	555.56	**2.76%**	24.00	628.99	**3.11%**	23.61	997.25	**1.52%**	23.86	669.87	**2.57%**
Average						1709.69	**4.70%**		1916.98	**2.77%**		1984.87	**2.44%**		1965.29	**1.92%**		2133.44	**2.28%**

Table 6.5: Calibration of α value (`Tenure3`).

6.2.3 Matheuristic performace

The final computational test was carried out with the calibrated two-phase algorithm, using the complete set of benchmark instances. The main objective was to assess the effectiveness of the heuristic and to compare the results it produces with the results of the FPVRP formulation. Table 6.6 summarizes this comparison.

Again, column *Best–Known* shows the value of the best-known solution for each instance, columns in block *FPVRP* reproduce the results obtained with the FPVRP formulation, columns in *Heur-FPVRP* give information related to the performance of the proposed algorithm, and the two columns of block *Gap*, give the percentage deviation gaps of the best solution produced by the heuristic with respect to best-known solutions (column *BK*), and to the lower bound (*LB*) produced by the FPVRP formulation of Archetti et al. (2017a) after four hours of computing time. The entries of column *BK* have been computed with the expression (6.10). Note that negative entries in this column indicate that the heuristic results improve the solutions obtained with the formulation. Percentage deviations in column *LB* have been computed with the expression $\frac{Z_{\text{Heur}} - Z_{\text{BestLB}}}{Z_{\text{BestLB}}} \times 100\%$. Average *BK* gaps range between -7.96% and 2.75%, with a total average gap of 0.69%. No *BK* gaps are reported for the largest instances since CPLEX is not able to obtain any feasible solution within the allowed computing time. Percentage deviations relative to lower

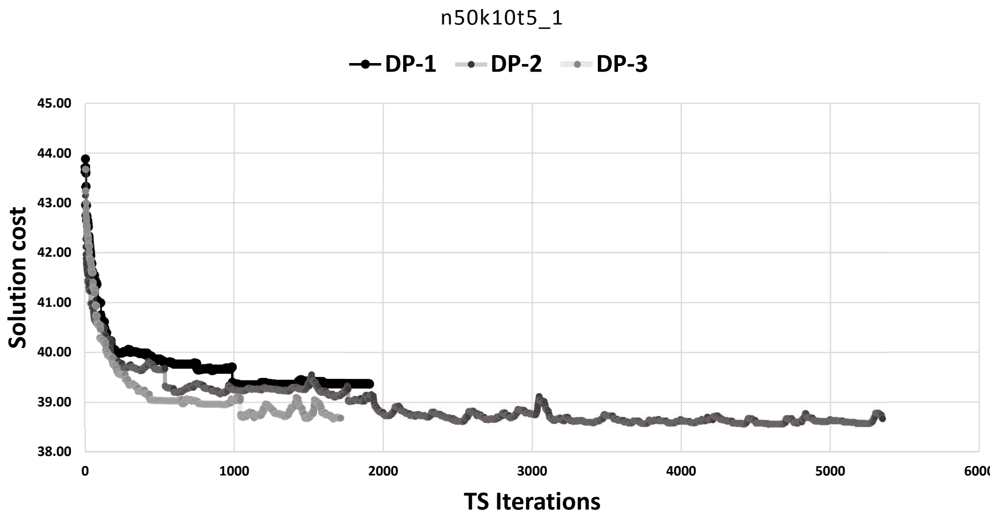
				FPVRP			**Heur-FPVRP**		**Gap**	
Instance	*r*	$\|N\|$	*Best Known*	*BestSol*	*BestLB*	*Time*	*BestSol*	*Time*	*BK*	*LB*
p01		51	524.61	524.93	510.46	14399	535.90	2964.23	2.11%	4.98%
p14		20	954.81	954.81	954.71	11525	954.80	545.83	0.00%	0.01%
p15		28	1862.63	1862.63	1825.04	14399	1862.62	2137.65	0.00%	2.06%
p16		56	2875.24	2875.24	2814.29	14399	2875.24	3080.64	0.00%	2.17%
p32		154	78072.88	—	40990.75	14400	86389.80	14448.70	9.63%	110.75%
								Avg.	**2.35%**	**2.30%**
n10k5t5_1		11	20.79	20.79	20.27	14400	21.34	56.29	2.61%	5.29%
n10k4t5_2			12.44	12.44	12.35	14400	12.44	49.85	0.00%	0.74%
n10k5t5_3			13.23	13.23	12.91	14400	13.45	205.81	1.65%	4.18%
n10k4t5_4			13.53	13.53	13.05	14400	13.93	78.57	2.91%	6.71%
n10k8t5_5			25.91	25.91	25.58	14400	27.74	58.44	6.59%	8.44%
								Avg.	**2.75%**	**5.07%**
n15k10t5_1	0.15	16	34.28	34.28	33.48	14400	34.97	202.90	1.98%	4.45%
n15k6t5_2			17.41	17.41	16.54	14400	17.43	483.24	0.10%	5.42%
n15k10t5_3			25.24	25.24	24.42	14400	25.33	392.46	0.34%	3.71%
n15k8t5_4			32.12	32.12	30.97	14400	32.73	414.56	1.88%	5.68%
n15k7t5_5			23.92	23.92	22.87	14400	24.44	218.01	2.12%	6.86%
								Avg.	**1.29%**	**5.22%**
n20k10t5_1		21	24.58	24.58	23.57	14400	24.83	836.93	1.01%	5.36%
n20k12t5_2			36.08	36.08	35.49	14400	36.27	986.62	0.51%	2.19%
n20k11t5_3			23.69	23.69	22.76	14400	25.36	720.79	6.57%	11.41%
n20k10t5_4			35.36	35.36	34.60	14400	36.23	557.28	2.41%	4.73%
n20k10t5_5			29.44	29.44	28.88	14400	30.27	538.46	2.76%	4.81%
								Avg.	**2.65%**	**5.70%**
n10k6t5_1		11	19.04	19.04	18.99	14400	19.42	67.07	1.95%	2.28%
n10k6t5_2			13.89	13.89	13.63	14400	13.89	74.90	0.02%	1.90%
n10k5t5_3			14.50	14.50	14.10	14400	14.67	69.08	1.19%	4.06%
n10k5t5_4			14.39	14.39	14.11	14400	14.42	86.32	0.27%	2.19%
n10k8t5_5	0.30		19.39	19.39	19.33	14400	19.83	64.39	2.18%	2.56%
								Avg.	**1.12%**	**2.60%**
n15k9t5_1		16	27.16	27.16	26.65	14400	27.24	193.37	0.31%	2.19%
n15k9t5_2			29.72	29.72	29.20	14400	30.28	309.82	1.87%	3.70%
n15k7t5_3			27.84	27.84	27.38	14400	28.12	325.31	1.02%	2.72%
n15k7t5_4			17.42	17.42	17.07	14400	17.42	302.75	0.01%	2.07%
n15k6t5_5			20.58	20.58	20.22	14400	20.63	206.45	0.24%	2.03%
								Avg.	**0.69%**	**2.54%**
n20k10t5_1		21	29.59	29.59	29.06	14400	30.65	674.94	3.48%	5.48%
n20k12t5_2			31.50	31.50	30.82	14400	32.38	591.02	2.72%	5.06%
n20k10t5_3			26.09	26.09	25.17	14400	26.34	549.93	0.95%	4.67%
n20k13t5_4			42.62	42.62	41.69	14400	43.31	676.45	1.60%	3.88%
n20k12t5_5			34.07	34.07	33.80	14400	34.60	480.37	1.52%	2.36%
								Avg.	**2.05%**	**4.29%**
n20k14t5_1		21	32.30	32.30	31.61	14400	32.59	635.37	0.89%	3.10%
n20k10t5_2			29.37	29.37	28.78	14400	29.78	624.37	1.37%	3.47%
n20k7t5_3	0.50		23.25	23.25	22.65	14400	23.61	1211.97	1.52%	4.22%
n20k10t5_4			24.81	24.81	24.12	14400	25.29	921.68	1.91%	4.87%
n20k11t5_5			36.45	36.45	35.60	14400	36.74	791.52	0.79%	3.19%
								Avg.	**1.30%**	**3.77%**
n50k10t5_1		51	44.65	44.65	36.04	14400	38.56	14440.30	-15.78%	7.00%
n50k8t5_2			32.91	32.91	30.36	14400	32.26	14430.50	-2.01%	6.26%
n50k9t5_3			33.13	33.13	28.84	14400	31.55	14438.80	-5.01%	9.39%
n50k9t5_4			29.61	29.61	27.04	14400	29.42	14445.10	-0.65%	8.77%
n50k11t5_5	0.15		39.29	39.29	31.04	14400	33.77	14433.80	-16.36%	8.79%
								Avg.	**-7.96%**	**8.04%**
n100k17t5_1		101	—	—	61.78	14400	67.22	14457.70	—	8.80%
n100k18t5_2			—	—	72.28	14400	78.04	14465.70	—	7.97%
n100k20t5_3			—	—	75.39	14400	82.83	14457.70	—	9.87%
n100k21t5_4			—	—	68.57	14400	74.00	14460.60	—	7.91%
n100k18t5_5			—	—	84.44	14400	91.10	14492.60	—	7.89%
								Avg.		**8.49%**
								Total Average Gap	**0.69%**	**4.85%**

Table 6.6: Performance of the two-phase algorithm.

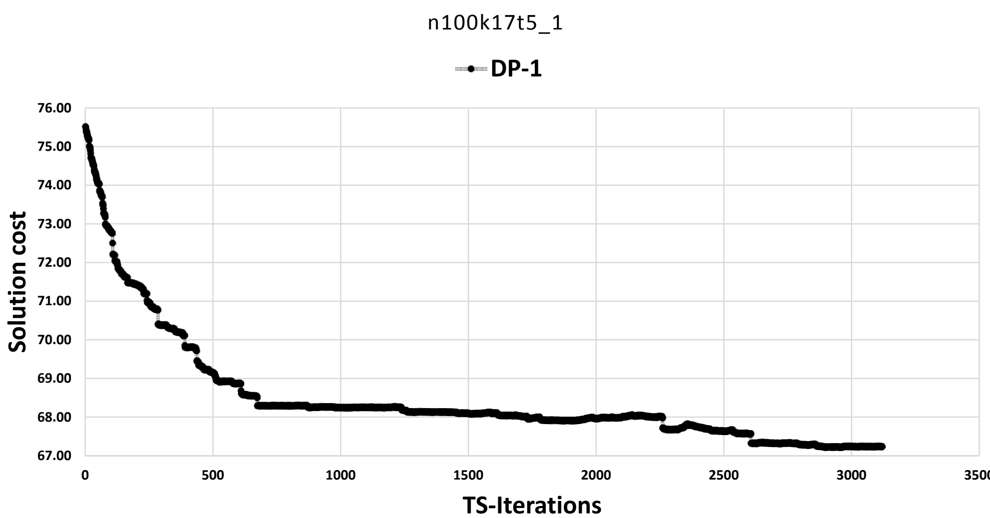
bounds, given in column *LB*, range between 2.30% and 8.49%, with a total average gap of 4.85%. These values indicate that the two-phase algorithm produces high quality solutions. In addition, the lower bounds produced by the FPVRP formulation of Archetti et al. (2017a) are, in general, pretty tight. The only exception is the lower bound for the largest instance, p32, which is an atypical case that has been left out of the calculation of the average.

In general, the computing times required by the heuristic are moderate, taking into account the difficulty and dimensions of the considered instances. Still, it was able to find good quality feasible solutions for all the considered benchmark instances, whereas the FPVRP formulation was not.

To illustrate the evolution of the optimization process, two large instances, one with 50 customers and one with 100 customers, are used. Figure 6.2 shows the reduction in the value of the objective function of the incumbent solution found by the TS during the computation for the two considered instances. Figure 6.2a refers to the instance with 50 customers, whereas Figure 6.2b to the instance with 100 customers. On the vertical axis, the objective function value of the best solution found by the TS is reported, while on the horizontal axis the number of iterations performed by the TS is shown. Also, the value of the objective function of the best solution found by the TS for each DP generation is provided. In particular, for the instance with 50 customers, 3 DP generations (DP1, DP2 and DP3 in Figure 6.2a) were performed before the maximum computing time was reached. For the instance with 100 customers, the maximum computing time was reached during the first run of the TS. It can be observed that large improvements occur before iteration 1000, which is reached after 4108 seconds for the 100 customer instance. For the 50 customer instance, 1000 iterations are reached after 861 seconds for DP1, 841 seconds for DP2 and 969 seconds for DP3. Note that, for the 50 customer instance, the incumbent solution after 1000 iterations is better than the solution produced by the FPVRP formulation after 4 hours of optimization. For the 100 instance no comparison can be done since the formulation does not produce any feasible solution within 4 hours of computation.



(a) 50 customers



(b) 100 customers

Figure 6.2: Value of the incumbent solution for a 50/100 customers instance.

Chapter 7

Conclusions and future directions

Many real-world situations give rise to vehicle routing problems in which it is necessary to deliver products to customers in different days over a specific time horizon. Such problems are usually addressed with classical PVRP models. Multiple elements may give rise to various extensions, which, broadly speaking, usually differ on how strictly delivery schedules are defined. As could be seen in Chapter 2.4, several works in the literature have considered models that allow some flexibility as for when to provide service to customers. In general, most of these works resort to inventory levels or service choice to incorporate flexible service policies.

In this thesis we have defined the *Flexible Periodic Vehicle Routing Problem* in which flexible service policies are considered to decide the frequency of the visits to each customer as well as the quantities delivered at each time visit. Unlike previous work in the literature, our problem addresses these service policies without considering inventory levels or service choice, and it does not depend on predefined schedules and fixed delivery quantities at each visit. In the FPVRP, three important decisions are simultaneously addressed: when to visit customers (schedule), what amount of product to deliver at each visit (delivery), and what are the routes that vehicles must perform in order to visit those customers (routing). The aim of the FPVRP is to minimize the total routing cost over the time horizon.

In Chapter 3.4, a worst-case analysis that shows the theoretical advantages of allowing flexible service policies in periodic routing problems, has been presented. According to this analysis, the savings provided by the FPVRP are significant with respect to both the PVRP and the IRP. In the remainder of the thesis we have presented formulations and developed algorithmic proposals for dealing with this type of models and analyzed their respective performance.

In Chapter 4, we have proposed two different MILP formulations for the FPVRP and families of inequalities to reinforce them. The main difference between both formulations is the way in which vehicle information is represented. The first formulation uses decision variables with a vehicle index and includes a family of SECs of exponential size, which need to be separated. In the second one, instead, decision variables identify the arcs used in the solutions without making explicit the vehicles that traverse them, at the expenses of using an additional set of continuous variables to indicate the load of the vehicles when traversing the arcs. Such load-based formulations tend to be quite effective in practice. On the one hand, they have a smaller number of variables. On the other hand, the number of constraints is polynomially bounded. An adapted version of the IRP called FPVRP-IC has been proposed in order to compare the FPVRP with IRP instances in which an inventory level is defined. Furthermore, a new formulation for the classical PVRP has been proposed in order to compare the FPVRP with this classical problem in which schedules and delivery quantities are known.

The results of the computational experiments that have been carried out show that, also in practice, the FPVRP may produce substantial improvements in the routing costs in comparison to both the PVRP and the IRP. Even if, in practice, the LP relaxation of load-based formulations is in general weaker than its classical counterpart, the computational results show that the lower bounds obtained for the load-based FPVRP are very good.

Given the difficulty of solving to optimality FPVRP instances of relatively small sizes, an efficient matheuristic, able of producing high quality solutions in small computing times, has been proposed to handle medium and large size instances.

The proposed matheuristic consists of two main phases: the `DP-Generation` and the `Improvement` phase. In the `DP-Generation`, the original problem is divided in two subproblems. The solution of the first subproblem produces the schedule of visits or *calendar* and the quantity to be delivered to each customer at each visit. The second subproblem takes the solution of the first subproblem to obtain a consistent set of routes to be performed by the vehicles. It applies a Tabu Search which explores different neighborhoods. The results obtained with the matheuristic showed that it can produce good quality solutions efficiently. The total average percentage gap with respect to the best–known solutions is 0.69% and the average gaps for the groups of instances of different sizes range within -7.96% and 2.75%, where negative values indicate that the solutions produced by the two–phase algorithm are better than the ones obtained with the formulations. The greater savings are obtained with the larger instances, for which the FPVRP formulation is not able to find a better solution after 4 hours of optimization 50 customers instances, and is not even able to find at least one feasible solution for instances with 100 customers.

Future research directions aim at considering similar types of formulations for other periodic VRPs like, for instance, the multi–depot PVRP, the PVRP with heterogeneous fleet of vehicles, PVRPs with Time Windows, and PVRPs with Intermediate Facilities, to mention just a few. Given that most real–world applications of the VRPs with periodic demands are highly related to the collection of garbage, recyclables, wastes, oil, etc., another avenue for future research is to extend the FPVRP to a closely related problem, called the *Green* VRP (Lin et al., 2014), which extends the classical VRP taking into account environmental and social impacts rather than just transportation costs.

Appendix A

Alternative formulation for the PVRP

Christofides and Beasley (1984) developed the first mathematical programming formulation for the PVRP, defined as the problem of designing a set of routes for each day of a given $|T|$-day planning period to meet the required customer visit frequency. Recall from Chapter 3.1.2 that it is assumed that the set of possible schedules for each customer $i \in C$, S_i is known. Furthermore, each schedule $s \in S_i$, $i \in C$ consists of the set of days when customer i is visited according to schedule s, and is represented by a set of binary coefficients a_{st}, indicating whether or not day $t \in T$ belongs to schedule s. The formulation that we propose uses three sets of decision variables: one for the selection of the schedule, one for the visits to customers, and the other one for the routing criterion. Given that the PVRP formulation proposed by Christofides and Beasley (1984) uses several aggregated variables and constraints, we propose a disaggregate formulation, which includes, in addition, the stronger version of the SECs (4.7).

A.1 Vehicle–index formulation for the PVRP

The following sets of variables are considered:

- $p_{ik}^{s} = \begin{cases} 1 \text{ If vehicle } k \in K \text{ visits customer } i \in C \text{ on schedule } s \in S_i, \\ 0 \text{ otherwise.} \end{cases}$
- $y_{ij}^{kt} = \begin{cases} 1 \text{ If vehicle } k \in K \text{ traverses arc } (i,j) \in A \text{ on day } t \in T, \\ 0 \text{ otherwise.} \end{cases}$
- $z_{i}^{kt} = \begin{cases} 1 \text{ If customer } i \in C \text{ is visited on day } t \in T \text{ per vehicle } k \in K, \\ 0 \text{ otherwise.} \end{cases}$

The disaggregated PVRP formulation (PVRP–D) is as follows:

$$\min \quad \sum_{t\in T}\sum_{(i,j)\in A}\sum_{k\in K} c_{ij} y_{ij}^{kt} \tag{A.1}$$

$$s.t. \quad \sum_{s\in S_i}\sum_{k\in K} p_{ik}^{s} = 1 \qquad i\in C \tag{A.2}$$

$$z_i^{kt} = \sum_{s\in S_i} p_{ik}^{s} a_{st} \qquad k\in K, t\in T, i\in C \tag{A.3}$$

$$y_{ij}^{kt} \le \frac{z_i^{kt} + z_j^{kt}}{2} \qquad k\in K; t\in T; i,j\in C (i\neq j) \tag{A.4}$$

$$\sum_{j\in N} y_{ij}^{kt} = \sum_{j\in V} y_{ji}^{kt} \qquad i\in N; k\in K; t\in T \tag{A.5}$$

$$\sum_{i\in N} y_{ij}^{kt} = \begin{cases} z_j^{kt}, & j\in C \\ 1, & j=0 \end{cases} \qquad k\in K, t\in T \tag{A.6}$$

$$\sum_{i,j\in P} y_{ij}^{kt} \le \sum_{i\in P} z_i^{kt} - z_{i'}^{kt} \qquad P\subseteq C, i'\in P, k\in K, t\in T \tag{A.7}$$

$$\sum_{i\in C} w_i \sum_{j\in N} y_{ij}^{kt} \le Q \qquad k\in K, t\in T \tag{A.8}$$

$$z_i^{kt} \in \{0,1\} \qquad i\in C, k\in K, t\in T \tag{A.9}$$

$$p_{ik}^{s} \in \{0,1\} \qquad s\in S_i, i\in C, k\in K \tag{A.10}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad (i,j)\in A, k\in K, t\in T. \tag{A.11}$$

The objective function (A.1) minimizes the total travel costs. Constraints (A.2) ensure that there is a feasible schedule for each customer. Constraints (A.3) define the days within the assigned schedule in which each customer will be visited by vehicle k. Constraints (A.4) restrict arc traversals to those where both end-nodes are customers visited by vehicle $k \in K$ at day $t \in T$. Constraints (A.5) are the flow conservation constraints. Constraints (A.6) relate the y and z variables. Constraints (A.7) are the SECs. Constraints (A.8) are the capacity constraints. Finally, Constraints (A.9)–(A.11) define the domain of the variables.

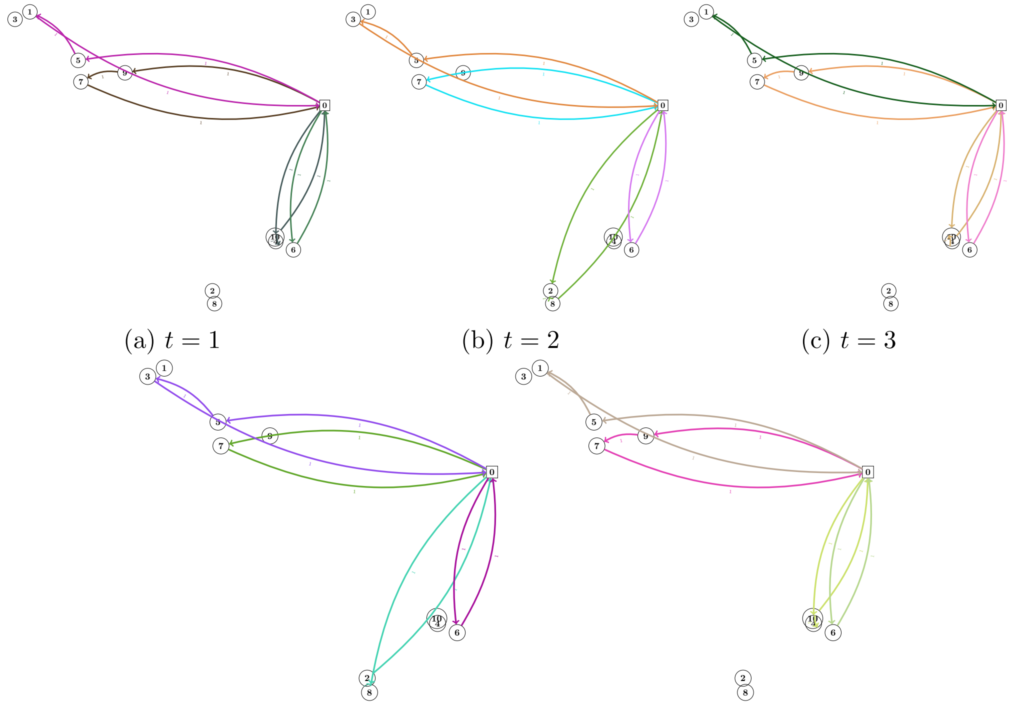
A.2 Comparison among PVRP formulations

For this test, we compare the results obtained with the aggregated formulation of Christofides and Beasley (1984) (PVRP–A) and the proposed disaggregated (PVRP–D) and load–based formulations (PVRP) using a small subset of the FPVRP instances ($r = 0.15$). The time limit was set to 4 hours (14400 sec).

According to results shown in Table A.1, formulations PVRP–D and PVRP produced better quality solutions than those produced by formulation PVRP–A. Moreover, formulation PVRP also outperformed PVRP–D. One of the reasons is that PVRP does not impose that all vehicles must be used at each time period (Figures A.1-A.2), while in PVRP–D it is constrained by (A.6) when $j = 0$. This set of constraints represents the disaggregated form of those used in PVRP–A.

Instance	PVRP–A				PVRP–D				PVRP			
	Status	BestSol	Gap%	Time	**Status**	BestSol	Gap%	Time	**Status**	BestSol	Gap%	Time
n10k5t5_1	**I**	—	—	0.8	**I**	—	—	44.98	**I**	—	—	14400
n10k4t5_2	**F**	15.46	5.49	14400	**O**	15.63	0.01	1080	**O**	13.01	0.01	145
n10k5t5_3	**F**	16.36	3.87	14400	**O**	16.82	0.01	1301	**O**	13.80	0.00	11
n10k4t5_4	**F**	17.80	2.84	14400	**O**	17.95	0.01	869	**O**	15.17	0.01	35
n10k8t5_5	**I**	—	—	0.08	**I**	—	—	0.14	**O**	27.74	0.00	1
n15k10t5_1	**F**	36.77	7.93	14400	**F**	37.04	8.52	14400	**O**	36.27	0.01	2445
n15k6t5_2	**I**	—	—	14400	**F**	22.21	15.91	14400	**F**	19.86	1.69	14400
n15k10t5_3	**I**	—	—	0.17	**I**	—	—	0.53	**O**	28.09	0.01	1213
n15k8t5_4	**F**	37.00	1.12	14400	**O**	37.00	0.01	206	**O**	34.09	0.01	4587
n15k7t5_5	**F**	31.18	9.41	14400	**F**	30.89	6.43	14400	**F**	25.77	1.54	14400
n20k10t5_1	**F**	31.32	1.32	14400	**F**	31.40	1.04	14400	**F**	25.94	2.33	14400
n20k12t5_2	**F**	47.02	2.75	14400	**F**	47.49	3.52	14400	**F**	39.12	0.48	14400
n20k11t5_3	**F**	29.56	3.59	14400	**F**	31.53	9.39	14400	**F**	26.64	2.47	14400
n20k10t5_4	**F**	42.61	12.26	14400	**F**	43.69	14.34	14400	**F**	39.43	4.81	14400
n20k10t5_5	**F**	40.72	4.96	14400	**F**	42.28	8.42	14400	**F**	31.45	4.31	14400

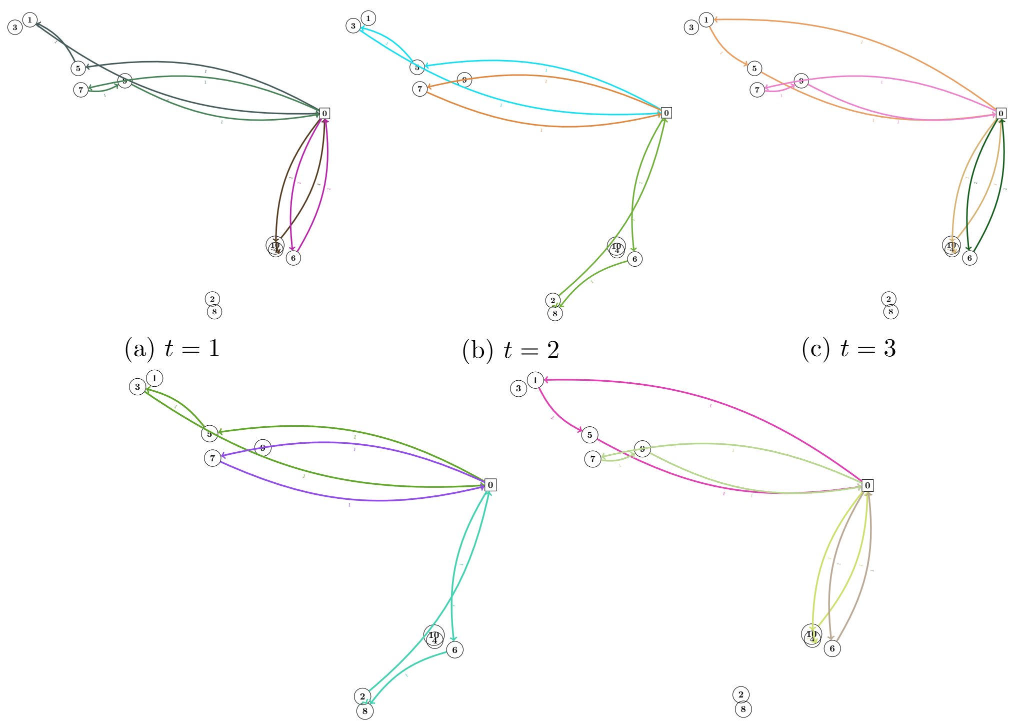
Table A.1: Comparison among PVRP formulations.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$

(d) $t = 4$ (e) $t = 5$

Figure A.1: Optimal solution obtained using the PVRP-D formulation.



(a) $t = 1$ (b) $t = 2$ (c) $t = 3$

(d) $t = 4$ (e) $t = 5$

Figure A.2: Optimal solution obtained using the load-based PVRP formulation.

Appendix B

Alternative formulation for the FPVRP-IC

B.1 Vehicle–index formulation for the FPVRP-IC

$$\min \quad \sum_{t\in T}\sum_{k\in K}\sum_{(i,j)\in A} c_{ij} y_{ij}^{kt} \tag{B.1}$$

$$s.t.\ I_{it} = I_i^{t-1} - d_{it} + \sum_{k\in K} q_i^{kt} \qquad i \in C, t \in T \tag{B.2}$$

$$\sum_{k\in K} q_i^{kt} \le w_i - I_i^{t-1} \qquad i \in C, t \in T \tag{B.3}$$

$$q_i^{kt} \le w_i z_i^{kt} \qquad i \in C, k \in K, t \in T \tag{B.4}$$

$$\sum_{i\in C} q_i^{kt} \le Q z_0^{kt} \qquad k \in K, t \in T \tag{B.5}$$

$$\sum_{k\in K} z_i^{kt} \le 1 \qquad i \in C, t \in T \tag{B.6}$$

$$\sum_{j|(i,j)\in A} y_{ij}^{kt} = z_i^{kt} \qquad i \in N, k \in K, t \in T \tag{B.7}$$

$$\sum_{j|(i,j)\in A} y_{ij}^{kt} = \sum_{j|(j,i)\in A} y_{ji}^{kt} \qquad i \in N, k \in K, t \in T \tag{B.8}$$

$$\sum_{\substack{(i,j)\in A\\ i,j\in S}} y_{ij}^{kt} \le \sum_{i\in S} z_i^{kt} - z_s^{kt} \qquad s \in S, S \subseteq C, k \in K, t \in T \tag{B.9}$$

$$I_{it} \ge 0 \qquad i \in C, t \in T \tag{B.10}$$

$$z_i^{kt} \in \{0,1\} \qquad i \in N, k \in K, t \in T \tag{B.11}$$

$$q_i^{kt} \ge 0 \qquad i \in C, k \in K, t \in T \tag{B.12}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad (i,j) \in A, k \in K, t \in T \tag{B.13}$$

$$y_{0j}^{kt} \in \{0,1,2\} \qquad j \in C, k \in K, t \in T. \tag{B.14}$$

The objective function (B.1) minimizes the total routing costs. Constraints (B.2) and (B.10) determine the inventory level over time and avoid stock–out. Constraints (B.3) impose that, at each time period, no vehicle delivers customer $i \in C$ a quantity that exceeds $w_i - I_i^{t-1}$. Constraints (B.5) impose that the vehicle capacity is not violated. Constraints (B.6), (B.7), (B.8), and (B.9) are the number of vehicles used for each visited customer, the node degree constraints, the flow conservation constraints, and the SECs for each vehicle route and each time period, respectively.

B.2 Comparison between FPVRP-IC formulations

Table B.1 shows the comparison between both vehicle–index and load–based FPVRP-IC formulations. The set S1 of instances is used and a time limit was set to 2 hours (7200 sec). It can be observed that in most of the instances the load–based formulation outperformed its vehicle–index version. The optimality gap of feasible solutions is much smaller than the vehicle–index counterpart (except for instance abs5n15_2).

Instance	Vehicle–index FPVRP-IC						Load–based FPVRP-IC					
	Status	Time	TotCuts*	BestSol	BestLB	Gap %	**Status**	Time	TotCuts	BestSol	BestLB	Gap %
abs1n5_1	**O**	2	13	1301.85	1301.85	0.00	**O**	0	0	1301.85	1301.85	0.00
abs1n5_2	**O**	1	10	1335.88	1335.88	0.00	**O**	0	0	1335.88	1335.88	0.00
abs2n5_1	**O**	1	9	1088.72	1088.72	0.00	**O**	0	0	1088.72	1088.72	0.00
abs2n5_2	**O**	9	24	1494.37	1494.37	0.00	**O**	4	0	1494.37	1494.23	0.01
abs3n5_1	**O**	4	25	2302.82	2302.82	0.00	**O**	2	0	2302.82	2302.82	0.00
abs3n5_2	**O**	2	20	2864.95	2864.95	0.00	**O**	0	0	2864.95	2864.95	0.00
abs4n5_1	**O**	1	6	1650.73	1650.73	0.00	**O**	0	0	1650.73	1650.59	0.01
abs4n5_2	**O**	4	9	2224.13	2224.13	0.00	**O**	1	0	2224.13	2224.13	0.00
abs5n5_1	**O**	1	12	1091.97	1091.95	0.00	**O**	0	0	1091.97	1091.97	0.00
abs5n5_2	**O**	5	19	1386.18	1386.18	0.00	**O**	4	0	1386.18	1386.18	0.00
abs1n10_1	**O**	56	462	1960.99	1960.99	0.00	**O**	18	0	1960.99	1960.82	0.01
abs1n10_2	**O**	184	844	2429.55	2429.37	0.01	**O**	31	0	2429.55	2429.55	0.00
abs2n10_1	**O**	19	117	2554.79	2554.79	0.00	**O**	14	0	2554.79	2554.79	0.00
abs2n10_2	**O**	100	277	3214.05	3213.77	0.01	**O**	19	0	3214.05	3213.88	0.01
abs3n10_1	**O**	10	59	1980.71	1980.71	0.00	**O**	6	0	1980.71	1980.71	0.00
abs3n10_2	**O**	81	387	2410.50	2410.26	0.01	**O**	39	0	2410.50	2410.50	0.00
abs4n10_1	**O**	60	384	2240.93	2240.93	0.00	**O**	35	0	2240.93	2240.73	0.01
abs4n10_2	**O**	338	844	2943.14	2942.90	0.01	**O**	279	0	2943.14	2942.87	0.01
abs5n10_1	**O**	7	23	1848.20	1848.20	0.00	**O**	14	0	1848.20	1848.20	0.00
abs5n10_2	**O**	17	38	2151.45	2151.45	0.00	**O**	19	0	2151.45	2151.45	0.00
abs1n15_1	**O**	19	86	1915.91	1915.91	0.00	**O**	21	0	1915.91	1915.89	0.00
abs1n15_2	**O**	279	678	2402.36	2402.18	0.01	**O**	295	0	2402.36	2402.14	0.01
abs2n15_1	**O**	115	229	2185.68	2185.51	0.01	**O**	194	0	2185.68	2185.46	0.01
abs2n15_2	**O**	229	905	2388.97	2388.97	0.00	**O**	39	0	2388.97	2388.97	0.00
abs3n15_1	**O**	93	410	2373.10	2373.10	0.00	**O**	11	0	2373.10	2373.10	0.00
abs3n15_2	**O**	129	436	2646.11	2645.91	0.01	**O**	20	0	2646.11	2646.11	0.00
abs4n15_1	**O**	87	255	2199.78	2199.78	0.00	**O**	188	0	2199.78	2199.57	0.01
abs4n15_2	**O**	296	349	2572.55	2572.55	0.00	**O**	705	0	2572.55	2572.30	0.01
abs5n15_1	**O**	164	520	2309.75	2309.75	0.00	**O**	88	0	2309.75	2309.53	0.01
abs5n15_2	**O**	903	1003	2959.31	2959.04	0.01	**F**	7199	0	2959.31	2846.19	3.82
abs1n20_1	**O**	1388	3413	2410.91	2410.91	0.00	**O**	6343	0	2410.91	2410.70	0.01
abs1n20_2	**F**	7200	10538	3118.53	2562.24	17.84	**F**	7199	0	3138.27	2885.98	8.04
abs2n20_1	**O**	57	206	2148.82	2148.82	0.00	**O**	18	0	2148.82	2148.82	0.00
abs2n20_2	**O**	852	1199	2393.13	2392.92	0.01	**O**	658	0	2393.13	2392.90	0.01
abs3n20_1	**O**	125	209	2283.53	2283.53	0.00	**O**	23	0	2283.53	2283.53	0.00
abs3n20_2	**O**	127	309	2529.42	2529.42	0.00	**O**	16	0	2529.42	2529.28	0.01
abs4n20_1	**O**	5680	10894	3136.22	3135.91	0.01	**O**	2782	0	3136.22	3135.91	0.01
abs4n20_2	**F**	7200	13530	3874.55	3083.57	20.41	**F**	7200	0	3664.52	3566.13	2.68
abs5n20_1	**O**	1370	4638	2859.60	2859.46	0.00	**O**	379	0	2859.60	2859.35	0.01
abs5n20_2	**F**	7200	11252	3738.12	3171.80	15.15	**F**	7200	0	3567.47	3473.76	2.63

Gap: Cplex MIP gap %. ***SECs and FCC.**

Table B.1: Comparison between vehicle-index and load-based FPVRP-IC formulations.

Appendix C

Additional computational experience for the FPVRP

In this Appendix, results from additional computational experiments are provided to complement the results shown in Chapter 5.3.

C.1 Comparison of inequalities for the load-based FPVRP formulation

We present detailed results of the tests carried out to evaluate the effect of the inequalities and optimality cuts described in Chapter 4.2.1 and discussed in Chapter 5.3.2.

Column description:

- FPVRP: load–based FPVRP formulation without valid inequalities.
- (1): FPVRP + inequalities (4.30)
- (2): FPVRP + ($y^t_{0j} + y^t_{i0} \leq 1 + \sum_{r \leq i} y^t_{0r}, i \in C, t \in T$).
- (3): FPVRP + inequalities (4.32).
- (2S): FPVRP + inequalities (4.31).
- (3S): FPVRP + inequalities (4.33).
- (1&2): Combination of (1) and (2).
- (1&3): Combination of (1) and (3).
- (1&2S): Combination of (1) and (2S).
- (1&3S): Combination of (1) and (3S).

- **Relative percentage optimality gap (RGap)**
  - RGap% $= \frac{|\text{LB} - \text{Best\_UB}|}{(1e{-}10 + |\text{Best\_UB}|)} \times 100.$

Instance	Best Solution									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**524.61**	560.24	524.93	524.61	530.84	534.92	527.68	524.61	524.93	524.61
p14	**954.81**	954.81	954.81	954.81	954.81	954.81	954.81	954.81	954.81	954.81
p15	**1862.63**	1862.63	1862.63	1862.63	1862.63	1862.63	1862.63	1862.63	1862.63	1862.63
p16	**2875.24**	2875.24	2877.13	2875.24	2875.24	2875.24	2875.24	2875.24	2875.24	2875.24
p17	**1609.09**	1612.21	1620.41	1611.45	1620.2	1607.64	1618.87	1607.51	1613	1617.11
Instn16t5k3_1	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_2	**5.63**	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63
Instn16t5k3_3	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_4	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_5	**5.76**	5.76	5.76	5.76	5.76	5.76	5.76	5.76	5.76	5.76
Instn16t5k3_6	**5.78**	5.78	5.78	5.78	5.78	5.78	5.78	5.78	5.78	5.78
Instn16t5k3_7	**5.74**	5.74	5.74	5.74	5.74	5.74	5.74	5.74	5.74	5.74
Instn16t5k3_8	**5.63**	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63
Instn16t5k4_1	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_2	**5.63**	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63
Instn16t5k4_3	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_4	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_5	**5.76**	5.76	5.76	5.76	5.76	5.76	5.76	5.76	5.76	5.76
Instn16t5k4_6	**5.78**	5.78	5.78	5.78	5.78	5.78	5.78	5.78	5.78	5.78
Instn16t5k4_7	**5.74**	5.74	5.74	5.74	5.74	5.74	5.74	5.74	5.74	5.74
Instn16t5k4_8	**5.63**	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63	5.63

Instance	Lower Bound									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**506.13**	506.71	510.63	510.06	509.49	510.27	510	513.95	510.4	511.47
p14	**942.83**	943.33	954.71	941.01	952.51	945.42	954.71	936.63	954.71	935.44
p15	**1810.09**	1809.23	1814.24	1801.69	1823.81	1804.29	1831.92	1803.8	1824.28	1794.62
p16	**2808.86**	2809.9	2812.79	2808.26	2818.81	2810.42	2818.05	2807.69	2814.09	2808.85
p17	**1551.38**	1549.33	1552.13	1552.13	1551.63	1552.13	1551.98	1551.88	1553.1	1550.59
Instn16t5k3_1	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_2	**5.62**	5.62	5.63	5.62	5.63	5.62	5.63	5.62	5.63	5.62
Instn16t5k3_3	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_4	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k3_5	**5.69**	5.72	5.73	5.7	5.72	5.72	5.7	5.71	5.72	5.71
Instn16t5k3_6	**5.72**	5.74	5.73	5.72	5.74	5.74	5.72	5.71	5.75	5.73
Instn16t5k3_7	**5.68**	5.7	5.69	5.68	5.7	5.7	5.68	5.67	5.7	5.69
Instn16t5k3_8	**5.62**	5.62	5.63	5.63	5.63	5.62	5.63	5.62	5.63	5.62
Instn16t5k4_1	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_2	**5.62**	5.63	5.63	5.62	5.63	5.62	5.63	5.62	5.63	5.62
Instn16t5k4_3	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_4	**5.62**	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62	5.62
Instn16t5k4_5	**5.73**	5.72	5.7	5.7	5.73	5.72	5.72	5.7	5.72	5.72
Instn16t5k4_6	**5.73**	5.74	5.72	5.72	5.74	5.73	5.74	5.68	5.74	5.73
Instn16t5k4_7	**5.68**	5.69	5.68	5.67	5.69	5.69	5.68	5.68	5.7	5.69
Instn16t5k4_8	**5.62**	5.63	5.62	5.63	5.63	5.63	5.63	5.62	5.63	5.62

Instance	RGAP%									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**3.52%**	3.41%	2.67%	2.77%	2.88%	2.73%	2.79%	2.03%	2.71%	2.50%
p14	**1.25%**	1.20%	0.01%	1.45%	0.24%	0.98%	0.01%	1.90%	0.01%	2.03%
p15	**2.82%**	2.87%	2.60%	3.27%	2.08%	3.13%	1.65%	3.16%	2.06%	3.65%
p16	**2.31%**	2.27%	2.17%	2.33%	1.96%	2.25%	1.99%	2.35%	2.13%	2.31%
p17	**3.49%**	3.62%	3.45%	3.45%	3.48%	3.45%	3.45%	3.46%	3.38%	3.54%
Instn16t5k3_1	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Instn16t5k3_2	**0.08%**	0.10%	0.01%	0.04%	0.01%	0.06%	0.01%	0.14%	0.01%	0.15%
Instn16t5k3_3	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%
Instn16t5k3_4	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Instn16t5k3_5	**1.29%**	0.74%	0.66%	1.18%	0.70%	0.83%	1.04%	0.99%	0.80%	0.89%
Instn16t5k3_6	**0.97%**	0.65%	0.81%	0.91%	0.65%	0.62%	0.89%	1.09%	0.47%	0.79%
Instn16t5k3_7	**1.04%**	0.77%	0.94%	1.00%	0.69%	0.77%	1.05%	1.20%	0.75%	0.83%
Instn16t5k3_8	**0.08%**	0.13%	0.01%	0.06%	0.01%	0.10%	0.01%	0.17%	0.01%	0.19%
Instn16t5k4_1	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Instn16t5k4_2	**0.05%**	0.02%	0.01%	0.15%	0.01%	0.13%	0.01%	0.10%	0.01%	0.15%
Instn16t5k4_3	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	0.00%
Instn16t5k4_4	**0.00%**	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
Instn16t5k4_5	**0.65%**	0.77%	1.10%	1.10%	0.59%	0.83%	0.77%	1.11%	0.76%	0.84%
Instn16t5k4_6	**0.78%**	0.70%	0.89%	0.89%	0.69%	0.80%	0.64%	1.71%	0.56%	0.79%
Instn16t5k4_7	**0.98%**	0.83%	1.02%	1.17%	0.84%	0.86%	1.00%	1.07%	0.77%	0.82%
Instn16t5k4_8	**0.12%**	0.01%	0.17%	0.01%	0.01%	0.05%	0.01%	0.18%	0.01%	0.16%
Average RGAP	**0.93%**	0.86%	0.79%	0.94%	0.71%	0.84%	0.73%	0.98%	0.69%	0.94%

Instance	CPLEX GAP %									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**3.52**	9.56	2.72	2.77	4.02	4.61	3.35	2.03	2.77	2.5
p14	**1.25**	1.2	0.01	1.45	0.24	0.98	0.01	1.9	0.01	2.03
p15	**2.82**	2.87	2.6	3.27	2.08	3.13	1.65	3.16	2.06	3.65
p16	**2.31**	2.27	2.24	2.33	1.96	2.25	1.99	2.35	2.13	2.31
p17	**3.59**	3.9	4.21	3.68	4.23	3.45	4.13	3.46	3.71	4.11
Instn16t5k3_1	**0**	0	0	0	0	0	0	0	0	0
Instn16t5k3_2	**0.08**	0.1	0.01	0.04	0.01	0.06	0.01	0.14	0.01	0.15
Instn16t5k3_3	**0**	0	0	0	0	0	0	0	0.01	0
Instn16t5k3_4	**0**	0	0	0	0	0	0	0	0	0
Instn16t5k3_5	**1.29**	0.74	0.66	1.18	0.7	0.83	1.04	0.99	0.8	0.89
Instn16t5k3_6	**0.97**	0.65	0.81	0.91	0.65	0.62	0.89	1.09	0.47	0.79
Instn16t5k3_7	**1.04**	0.77	0.94	1	0.69	0.77	1.05	1.2	0.75	0.83
Instn16t5k3_8	**0.08**	0.13	0.01	0.06	0.01	0.1	0.01	0.17	0.01	0.19
Instn16t5k4_1	**0**	0	0	0	0	0	0	0	0	0
Instn16t5k4_2	**0.05**	0.02	0.01	0.15	0.01	0.13	0.01	0.1	0.01	0.15
Instn16t5k4_3	**0**	0	0	0	0	0	0.01	0	0	0
Instn16t5k4_4	**0**	0	0	0	0	0	0	0	0	0
Instn16t5k4_5	**0.65**	0.77	1.1	1.1	0.59	0.83	0.77	1.11	0.76	0.84
Instn16t5k4_6	**0.78**	0.7	0.89	0.89	0.69	0.8	0.64	1.71	0.56	0.79
Instn16t5k4_7	**0.98**	0.83	1.02	1.17	0.84	0.86	1	1.07	0.77	0.82
Instn16t5k4_8	**0.12**	0.01	0.17	0.01	0.01	0.05	0.01	0.18	0.01	0.16

Instance	Status									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**F**	F	F	F	F	F	F	F	F	F
p14	**F**	F	O	F	F	F	O	F	O	F
p15	**F**	F	F	F	F	F	F	F	F	F
p16	**F**	F	F	F	F	F	F	F	F	F
p17	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k3_1	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k3_2	**F**	F	O	F	O	F	O	F	O	F
Instn16t5k3_3	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k3_4	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k3_5	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k3_6	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k3_7	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k3_8	**F**	F	O	F	O	F	O	F	O	F
Instn16t5k4_1	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k4_2	**F**	F	O	F	O	F	O	F	O	F
Instn16t5k4_3	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k4_4	**O**	O	O	O	O	O	O	O	O	O
Instn16t5k4_5	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k4_6	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k4_7	**F**	F	F	F	F	F	F	F	F	F
Instn16t5k4_8	**F**	O	F	O	O	F	O	F	O	F

Instance	Time									
	FPVRP	(1)	(2)	(3)	(1&2)	(1&3)	(2S)	(3S)	(1&2S)	(1&3S)
p01	**14400**	14400	14400	14400	14400	14400	14400	14400	14399	14400
p14	**14400**	14400	11039	14400	14400	14400	12837	14400	11525	14400
p15	**14400**	14400	14400	14400	14400	14400	14400	14399	14399	14399
p16	**14400**	14400	14400	14399	14400	14399	14400	14400	14399	14399
p17	**14400**	14400	14400	14400	14400	14400	14400	14400	14400	14400
Instn16t5k3_1	**21**	40	31	25	27	30	31	27	37	31
Instn16t5k3_2	**14400**	14399	573	14400	839	14399	3678	14400	3203	14399
Instn16t5k3_3	**21**	19	22	20	24	16	19	21	79	30
Instn16t5k3_4	**21**	13	20	18	22	18	22	22	22	16
Instn16t5k3_5	**14400**	14401	14400	14400	14400	14401	14400	14400	14399	14400
Instn16t5k3_6	**14400**	14400	14400	14402	14400	14400	14400	14400	14399	14401
Instn16t5k3_7	**14400**	14402	14400	14400	14400	14400	14400	14400	14399	14400
Instn16t5k3_8	**14400**	14401	5477	14400	3526	14399	7754	14400	6890	14399
Instn16t5k4_1	**28**	31	26	23	26	21	21	19	25	36
Instn16t5k4_2	**14400**	14400	1763	14400	11752	14400	2619	14399	3638	14399
Instn16t5k4_3	**17**	17	18	16	20	14	78	21	23	35
Instn16t5k4_4	**18**	14	19	22	21	19	18	14	25	18
Instn16t5k4_5	**14400**	14400	14400	14402	14400	14400	14399	14400	14399	14399
Instn16t5k4_6	**14400**	14400	14400	14399	14400	14400	14400	14400	14399	14400
Instn16t5k4_7	**14400**	14400	14400	14400	14400	14400	14401	14400	14400	14399
Instn16t5k4_8	**14400**	6572	14400	11580	9625	14399	7517	14400	4579	14401

Table C.1: Comparison among inequalities for the load-based FPVRP formulation - Instances S2 and S5.

Instance	Best Solution									Lower Bound									RGAP%								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k5t5_1	**20.78**	20.78	20.78	20.78	20.78	20.78	20.78	20.78	20.78	**20.15**	20.21	20.21	20.2	20.23	20.02	20.2	20.3	20.06	**3.04%**	2.77%	2.77%	2.81%	2.67%	3.65%	2.80%	2.31%	3.47%
n10k4t5_2	**12.44**	12.44	12.44	12.44	12.44	12.44	12.44	12.44	12.44	**12.19**	12.05	12.03	12	12.06	11.94	12	12.12	12.04	**2.04%**	3.14%	3.30%	3.57%	3.03%	4.00%	3.54%	2.58%	3.24%
n10k5t5_3	**13.23**	13.23	13.23	13.23	13.23	13.23	13.23	13.23	13.23	**12.85**	12.85	12.86	12.86	12.87	12.89	12.86	12.87	12.87	**2.84%**	2.85%	2.76%	2.74%	2.70%	2.55%	2.77%	2.71%	2.72%
n10k4t5_4	**13.53**	13.53	13.53	13.53	13.53	13.53	13.53	13.53	13.53	**13.03**	12.99	13.01	12.98	13.06	12.93	13.02	13.05	13	**3.70%**	4.02%	3.85%	4.09%	3.49%	4.44%	3.80%	3.55%	3.90%
n10k8t5_5	**25.91**	25.91	25.91	25.91	25.91	25.91	25.91	25.91	25.91	**25.57**	25.62	25.57	25.61	25.56	25.61	25.57	25.58	25.57	**1.30%**	1.12%	1.32%	1.15%	1.34%	1.14%	1.31%	1.28%	1.31%
n15k10t5_1	**34.27**	34.26	34.26	34.27	34.26	34.26	34.27	34.26	34.27	**33.46**	33.32	33.46	33.47	33.47	33.25	33.37	33.39	33.34	**2.35%**	2.74%	2.32%	2.32%	2.31%	2.95%	2.60%	2.53%	2.69%
n15k6t5_2	**17.38**	17.4	17.42	17.42	17.38	17.42	17.4	17.42	17.42	**16.51**	16.5	16.52	16.48	16.5	16.49	16.49	16.51	16.49	**5.00%**	5.04%	4.96%	5.16%	5.03%	5.10%	5.10%	4.98%	5.10%
n15k10t5_3	**25.25**	25.25	25.25	25.25	25.25	25.25	25.25	25.25	25.25	**24.39**	24.41	24.38	24.41	24.41	24.4	24.41	24.45	24.4	**3.43%**	3.31%	3.44%	3.31%	3.33%	3.37%	3.32%	3.17%	3.36%
n15k8t5_4	**32.11**	32.11	32.11	32.11	32.11	32.11	32.11	32.11	32.11	**31.21**	31.06	31.15	31.22	31.21	31	31.11	31.25	31.01	**2.79%**	3.26%	3.00%	2.76%	2.78%	3.46%	3.12%	2.67%	3.41%
n15k7t5_5	**23.94**	23.94	23.93	24	23.93	23.93	23.93	23.89	23.93	**22.56**	22.75	22.58	22.77	22.6	22.48	22.76	22.84	22.71	**5.57%**	4.76%	5.46%	4.67%	5.37%	5.87%	4.72%	4.39%	4.94%
n20k10t5_1	**24.61**	24.61	24.64	24.66	24.58	24.61	24.64	24.58	24.61	**23.56**	23.55	23.57	23.53	23.57	23.55	23.59	23.58	23.47	**4.13%**	4.21%	4.11%	4.26%	4.10%	4.20%	4.03%	4.06%	4.51%
n20k12t5_2	**36.1**	36.08	36.08	36.08	36.08	36.08	36.09	36.1	36.08	**35.44**	35.45	35.44	35.43	35.47	35.42	35.47	35.45	35.43	**1.77%**	1.76%	1.78%	1.79%	1.70%	1.84%	1.70%	1.76%	1.80%
n20k11t5_3	**23.7**	23.7	23.7	23.77	23.7	23.68	23.72	23.7	23.71	**22.73**	22.73	22.75	22.72	22.75	22.67	22.74	22.79	22.7	**4.05%**	4.02%	3.95%	4.05%	3.94%	4.29%	3.97%	3.76%	4.17%
n20k10t5_4	**35.36**	35.35	29.5	35.36	35.36	35.35	35.39	35.35	35.39	**34.5**	34.51	28.89	34.56	34.52	34.34	34.18	34.53	34.15	**16.97%**	17.00%	2.07%	17.17%	17.04%	16.41%	15.88%	17.06%	15.78%
n20k10t5_5	**29.4**	29.35	35.37	29.41	29.5	29.35	29.45	29.45	29.41	**28.88**	28.9	34.52	28.88	28.89	28.86	28.9	28.89	28.89	**1.62%**	1.56%	17.61%	1.60%	1.58%	1.68%	1.53%	1.59%	1.58%
Average RGAP																			**4.04%**	4.10%	4.18%	4.10%	4.03%	4.33%	4.01%	3.89%	4.13%

Instance	CPLEX Gap %									Status									Time								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k5t5_1	**3.03**	2.77	2.77	2.81	2.67	3.65	2.8	2.31	3.466	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14400	14400	14400	14401	14400	14400
n10k4t5_2	**2.04**	3.14	3.3	3.57	3.03	4	3.54	2.58	3.238	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14399	14400	14400	14400	14400	14400
n10k5t5_3	**2.85**	2.85	2.76	2.74	2.7	2.55	2.77	2.71	2.722	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14400	14400	14400	14401	14400	14401
n10k4t5_4	**3.7**	4.02	3.85	4.09	3.49	4.44	3.8	3.55	3.895	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14400	14400	14400	14400	14400	14400
n10k8t5_5	**1.3**	1.12	1.32	1.15	1.34	1.14	1.31	1.28	1.306	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14400	14400	14400	14400	14400	14400
n15k10t5_1	**2.38**	2.74	2.32	2.36	2.31	2.95	2.63	2.53	2.721	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14400	14400	14398	14400	14403	14400
n15k6t5_2	**5**	5.15	5.19	5.39	5.03	5.35	5.22	5.23	5.325	**F**	F	F	F	F	F	F	F	F	**14400**	14401	14400	14402	14401	14402	14401	14400	14400
n15k10t5_3	**3.43**	3.31	3.44	3.31	3.33	3.37	3.32	3.17	3.361	**F**	F	F	F	F	F	F	F	F	**14402**	14401	14407	14403	14402	14402	14400	14400	14400
n15k8t5_4	**2.79**	3.26	3	2.76	2.78	3.46	3.12	2.66	3.415	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14399	14400	14400	14400	14400	14400
n15k7t5_5	**5.77**	4.97	5.62	5.11	5.53	6.03	4.88	4.39	5.106	**F**	F	F	F	F	F	F	F	F	**14401**	14400	14401	14400	14401	14400	14401	14400	14401
n20k10t5_1	**4.25**	4.34	4.35	4.6	4.1	4.33	4.27	4.06	4.632	**F**	F	F	F	F	F	F	F	F	**14403**	14400	14401	14400	14400	14400	14401	14402	14401
n20k12t5_2	**1.83**	1.76	1.78	1.79	1.7	1.84	1.73	1.82	1.801	**F**	F	F	F	F	F	F	F	F	**14401**	14400	14400	14402	14401	14404	14400	14400	14400
n20k11t5_3	**4.11**	4.09	4.02	4.41	3.99	4.29	4.1	3.82	4.269	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14400	14404	14400	14400	14400	14404	14400
n20k10t5_4	**2.42**	2.37	2.07	2.26	2.37	2.86	3.42	2.32	3.501	**F**	F	F	F	F	F	F	F	F	**14405**	14403	14401	14400	14402	14400	14400	14401	14401
n20k10t5_5	**1.77**	1.56	2.39	1.79	2.05	1.68	1.85	1.91	1.75	**F**	F	F	F	F	F	F	F	F	**14400**	14400	14403	14399	14401	14400	14401	14399	14400

Table C.2: Comparison among inequalities for the load-based FPVRP formulation - Instances S3 ($r = 0.15$).

Instance	Best Solution									Lower Bound								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k6t5_1	**19.04**	19.04	19.04	19.04	19.04	19.04	19.04	19.04	19.04	**18.9**	18.93	18.95	18.94	18.98	18.89	18.92	18.97	18.89
n10k6t5_2	**13.89**	13.89	13.89	13.89	13.89	13.89	13.89	13.89	13.89	**13.52**	13.61	13.48	13.49	13.5	13.44	13.54	13.59	13.53
n10k5t5_3	**14.5**	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	**14.17**	14.14	14.16	14.17	14.15	14.02	14.17	14.16	14.07
n10k5t5_4	**14.38**	14.38	14.38	14.38	14.38	14.38	14.38	14.38	14.38	**14.18**	14.06	14.12	14.17	14.12	14.12	14.28	14.04	14.15
n10k8t5_5	**19.4**	19.4	19.4	19.4	19.4	19.4	19.4	19.4	19.4	**19.32**	19.31	19.28	19.35	19.34	19.32	19.34	19.34	19.3
n15k9t5_1	**27.16**	27.16	27.16	27.16	27.16	27.16	27.16	27.16	27.16	**26.63**	26.62	26.63	26.62	26.64	26.62	26.62	26.64	26.62
n15k9t5_2	**29.72**	29.72	29.72	29.72	29.72	29.72	29.72	29.72	29.72	**29.18**	29.18	29.19	29.07	29.19	29.06	29.18	29.19	29.13
n15k7t5_3	**27.84**	27.84	27.91	27.84	27.84	27.84	27.84	27.84	27.84	**27.36**	27.38	27.38	27.32	27.38	27.35	27.35	27.38	27.35
n15k7t5_4	**17.43**	17.43	17.43	17.43	17.43	17.43	17.43	17.43	17.43	**17.02**	17.01	16.99	16.99	17.03	16.99	17.05	17.06	17.01
n15k6t5_5	**20.59**	20.59	20.59	20.59	20.59	20.59	20.59	20.59	20.59	**20.16**	20.16	20.17	20.15	20.16	20.12	20.17	20.18	20.15
n20k10t5_1	**29.57**	29.57	29.57	29.57	29.57	29.57	29.57	29.57	29.57	**28.98**	28.96	28.91	28.93	28.93	28.78	28.95	28.97	28.82
n20k12t5_2	**31.56**	31.49	31.49	31.56	31.49	31.56	31.56	31.55	31.56	**30.84**	30.78	30.85	30.85	30.82	30.76	30.86	30.78	30.79
n20k10t5_3	**26.07**	26.06	26.06	26.06	26.07	26.06	26.07	26.07	26.07	**25.19**	25.19	25.18	25.2	25.19	25.16	25.18	25.18	25.18
n20k13t5_4	**42.63**	42.56	42.74	42.61	42.77	42.77	42.65	42.61	42.56	**41.66**	41.62	41.63	41.6	41.59	41.56	41.62	41.62	41.55
n20k12t5_5	**34.05**	34.05	34.05	34.05	34.05	34.05	34.05	34.05	34.05	**33.75**	33.75	33.75	33.75	33.75	33.74	33.76	33.76	33.77

Instance	RGAP%								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k6t5_1	**0.77%**	0.59%	0.52%	0.55%	0.35%	0.83%	0.63%	0.38%	0.83%
n10k6t5_2	**2.67%**	2.01%	2.97%	2.88%	2.80%	3.25%	2.53%	2.11%	2.60%
n10k5t5_3	**2.29%**	2.50%	2.32%	2.28%	2.41%	3.30%	2.29%	2.36%	2.96%
n10k5t5_4	**1.37%**	2.22%	1.80%	1.47%	1.80%	1.80%	0.72%	2.39%	1.61%
n10k8t5_5	**0.44%**	0.47%	0.62%	0.29%	0.32%	0.42%	0.31%	0.32%	0.55%
n15k9t5_1	**1.96%**	1.98%	1.95%	1.98%	1.92%	1.98%	1.97%	1.91%	2.00%
n15k9t5_2	**1.80%**	1.80%	1.78%	2.18%	1.78%	2.20%	1.80%	1.78%	1.98%
n15k7t5_3	**1.70%**	1.64%	1.65%	1.87%	1.65%	1.75%	1.75%	1.65%	1.76%
n15k7t5_4	**2.33%**	2.37%	2.50%	2.50%	2.28%	2.53%	2.14%	2.09%	2.40%
n15k6t5_5	**2.07%**	2.07%	2.04%	2.11%	2.08%	2.27%	2.01%	2.00%	2.11%
n20k10t5_1	**1.99%**	2.04%	2.22%	2.17%	2.14%	2.68%	2.08%	2.03%	2.53%
n20k12t5_2	**2.09%**	2.27%	2.05%	2.03%	2.14%	2.34%	2.03%	2.26%	2.22%
n20k10t5_3	**3.35%**	3.36%	3.38%	3.31%	3.33%	3.48%	3.38%	3.39%	3.37%
n20k13t5_4	**2.13%**	2.21%	2.18%	2.25%	2.28%	2.36%	2.21%	2.20%	2.37%
n20k12t5_5	**0.89%**	0.89%	0.89%	0.89%	0.90%	0.91%	0.87%	0.85%	0.82%
Average RGAP	**1.86%**	1.89%	1.92%	1.92%	1.88%	2.14%	1.78%	1.85%	2.01%

Instance	CPLEX Gap %									Status								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)	**FPVRP**	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k6t5_1	**0.77**	0.59	0.52	0.55	0.35	0.83	0.629	0.38	0.83	**F**	F	F	F	F	F	F	F	F
n10k6t5_2	**2.67**	2.01	2.97	2.88	2.8	3.25	2.535	2.11	2.6	**F**	F	F	F	F	F	F	F	F
n10k5t5_3	**2.29**	2.5	2.32	2.28	2.41	3.3	2.293	2.36	2.96	**F**	F	F	F	F	F	F	F	F
n10k5t5_4	**1.37**	2.22	1.8	1.47	1.8	1.8	0.72	2.39	1.61	**F**	F	F	F	F	F	F	F	F
n10k8t5_5	**0.44**	0.47	0.62	0.29	0.32	0.42	0.311	0.32	0.55	**F**	F	F	F	F	F	F	F	F
n15k9t5_1	**1.96**	1.98	1.95	1.98	1.92	1.98	1.973	1.91	2	**F**	F	F	F	F	F	F	F	F
n15k9t5_2	**1.8**	1.8	1.77	2.18	1.78	2.2	1.805	1.78	1.98	**F**	F	F	F	F	F	F	F	F
n15k7t5_3	**1.7**	1.64	1.9	1.87	1.65	1.75	1.752	1.65	1.76	**F**	F	F	F	F	F	F	F	F
n15k7t5_4	**2.33**	2.37	2.5	2.5	2.28	2.53	2.138	2.09	2.4	**F**	F	F	F	F	F	F	F	F
n15k6t5_5	**2.07**	2.07	2.04	2.11	2.08	2.27	2.006	2	2.11	**F**	F	F	F	F	F	F	F	F
n20k10t5_1	**1.99**	2.05	2.22	2.17	2.14	2.68	2.08	2.03	2.53	**F**	F	F	F	F	F	F	F	F
n20k12t5_2	**2.29**	2.27	2.05	2.24	2.14	2.54	2.23	2.45	2.42	**F**	F	F	F	F	F	F	F	F
n20k10t5_3	**3.4**	3.36	3.38	3.31	3.38	3.48	3.427	3.44	3.42	**F**	F	F	F	F	F	F	F	F
n20k13t5_4	**2.28**	2.21	2.6	2.36	2.76	2.84	2.418	2.32	2.38	**F**	F	F	F	F	F	F	F	F
n20k12t5_5	**0.89**	0.89	0.89	0.89	0.9	0.91	0.868	0.85	0.82	**F**	F	F	F	F	F	F	F	F

Instance	Time								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n10k6t5_1	**14400**	14400	14400	14401	14400	14400	14400	14400	14400
n10k6t5_2	**14400**	14400	14400	14399	14400	14400	14399	14399	14400
n10k5t5_3	**14400**	14400	14400	14400	14400	14400	14400	14400	14399
n10k5t5_4	**14399**	14399	14400	14400	14399	14400	14399	14400	14400
n10k8t5_5	**14399**	14400	14400	14400	14400	14399	14400	14400	14400
n15k9t5_1	**14400**	14400	14400	14400	14400	14403	14401	14400	14401
n15k9t5_2	**14408**	14402	14403	14400	14400	14400	14400	14401	14400
n15k7t5_3	**14400**	14401	14401	14400	14402	14400	14400	14400	14400
n15k7t5_4	**14400**	14401	14403	14400	14401	14401	14401	14401	14400
n15k6t5_5	**14400**	14400	14400	14400	14400	14400	14399	14400	14401
n20k10t5_1	**14401**	14400	14400	14401	14393	14393	14400	14400	14399
n20k12t5_2	**14401**	14401	14400	14401	14401	14403	14401	14400	14400
n20k10t5_3	**14403**	14400	14404	14402	14402	14401	14400	14400	14400
n20k13t5_4	**14402**	14401	14403	14404	14400	14402	14404	14407	14400
n20k12t5_5	**14400**	14400	14401	14400	14404	14401	14400	14400	14402

Table C.3: Comparison among inequalities for the load-based FPVRP formulation - Instances S3 ($r = 0.30$).

Instance	Best Solution								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**32.3**	32.3	32.3	32.3	32.3	32.3	32.3	32.3	32.3
n20k10t5_2	**29.36**	29.36	29.34	29.36	29.33	29.36	29.33	29.33	29.36
n20k7t5_3	**23.25**	23.25	23.28	23.25	23.25	23.25	23.25	23.25	23.25
n20k10t5_4	**24.86**	24.8	24.8	24.86	24.86	24.8	24.86	24.8	24.86
n20k11t5_5	**36.48**	36.45	36.45	36.45	36.45	36.45	36.45	36.45	36.45

Instance	Lower Bound								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**31.55**	31.55	31.54	31.48	31.54	31.43	31.52	31.57	31.41
n20k10t5_2	**28.79**	28.82	28.81	28.79	28.75	28.57	28.78	28.77	28.68
n20k7t5_3	**22.61**	22.7	22.65	22.62	22.66	22.57	22.65	22.68	22.59
n20k10t5_4	**23.99**	23.97	23.99	23.94	24.01	23.92	23.98	24.02	23.92
n20k11t5_5	**35.44**	35.47	35.5	35.46	35.48	35.33	35.47	35.5	35.41

Instance	RGAP%								
	FPVRP	(1)	(2)	(3)	(2S)	(3S)	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**2.32%**	2.35%	2.37%	2.55%	2.36%	2.71%	2.42%	2.28%	2.77%
n20k10t5_2	**1.85%**	1.76%	1.79%	1.85%	1.98%	2.59%	1.87%	1.93%	2.22%
n20k7t5_3	**2.73%**	2.35%	2.56%	2.68%	2.51%	2.91%	2.56%	2.44%	2.82%
n20k10t5_4	**3.27%**	3.34%	3.28%	3.46%	3.21%	3.56%	3.32%	3.16%	3.55%
n20k11t5_5	**2.79%**	2.71%	2.61%	2.72%	2.68%	3.07%	2.71%	2.63%	2.87%
Average RGAP	**2.59%**	2.50%	2.52%	2.65%	2.55%	2.97%	2.58%	2.49%	2.85%

Instance	CPLEX Gap %								
	FPVRP	(1)	(2)	(3)	2S	3S	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**2.32**	2.35	2.37	2.55	2.36	2.71	2.42	2.28	2.77
n20k10t5_2	**1.94**	1.86	1.8	1.95	1.98	2.68	1.88	1.93	2.32
n20k7t5_3	**2.73**	2.35	2.7	2.68	2.51	2.91	2.56	2.44	2.82
n20k10t5_4	**3.5**	3.34	3.28	3.68	3.44	3.56	3.54	3.16	3.78
n20k11t5_5	**2.86**	2.71	2.61	2.72	2.68	3.07	2.71	2.63	2.87

Instance	Status								
	FPVRP	(1)	(2)	(3)	2S	3S	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**F**	F	F	F	F	F	F	F	F
n20k10t5_2	**F**	F	F	F	F	F	F	F	F
n20k7t5_3	**F**	F	F	F	F	F	F	F	F
n20k10t5_4	**F**	F	F	F	F	F	F	F	F
n20k11t5_5	**F**	F	F	F	F	F	F	F	F

Instance	Time								
	FPVRP	(1)	(2)	(3)	2S	3S	(1&3)	(1&2S)	(1&3S)
n20k14t5_1	**14402**	14402	14403	14400	14400	14401	14400	14400	14400
n20k10t5_2	**14402**	14399	14399	14405	14399	14399	14399	14400	14399
n20k7t5_3	**14400**	14399	14400	14400	14399	14400	14399	14400	14403
n20k10t5_4	**14400**	14400	14399	14399	14401	14400	14399	14400	14400
n20k11t5_5	**14400**	14402	14400	14400	14401	14402	14400	14400	14400

Table C.4: Comparison among inequalities for the load-based FPVRP formulation - Instances S3 ($r = 0.50$).

Bibliography

Aksen, D., Kaya, O., Salman, F.S., and Akça, Y. (2012). Selective and periodic inventory routing problem for waste vegetable oil collection. *Optimization Letters*, 6(6):1063–1080.

Alabaş-Uslu, C. (2007). A self-tuning heuristic for a multi-objective vehicle routing problem. *Journal of Operational Research Society*, 59(7):988–996.

Albareda-Sambola, M., Fernández, E., and Laporte, G. (2014). The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, 48:31–39.

Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M.G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.

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.

Archetti, C., Fernández, E., and Huerta-Muñoz, D.L. (2017a). The Flexible Periodic Vehicle Routing Problem. *Computers & Operations Research*, 85:58–70.

Archetti, C., Fernández, E., and Huerta-Muñoz, D.L. (2017b). A two-phase solution algorithm for the Flexible Periodic Vehicle Routing Problem. Submitted.

Archetti, C., Jabali, O., and Speranza, M.G. (2015). Multi–Period Vehicle Routing Problem with Due Dates. *Computers & Operations Research*, 61:122–134.

Archetti, C., Savelsbergh, M.W.P., and Speranza, M.G. (2008a). An Optimization-Based Heuristic for the Split Delivery Vehicle Routing Problem. *Transportation Science*, 42(1):22–31.

Archetti, C., Savelsbergh, M.W.P., and Speranza, M.G. (2008b). To split or not to split: That is the question. *Transportation Research E*, 44:114–123.

Archetti, C. and Speranza, M.G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, 2(4):223–246.

Augerat, P., Belenguer, J.M., Benavent, E., Corberán, A., Naddef, D., and Rinaldi, G. (1998). Computational results with a branch–and–cut code for the capacited vehicle routing problem. Technical Report R–495, Department of Computer Science, Michigan State University.

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., Battarra, M., and Vigo, D. (2008). Routing a Heterogeneous Fleet of Vehicles. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43, pages 3–27, Boston, MA. Springer US.

Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., Mack, R.G., and Prutzman, P.J. (1983). Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer. *Interfaces*, 13(6):4–23.

Beltrami, E.J. and Bodin, L.D. (1974). Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94.

Bertazzi, L. and Speranza, M.G. (2012a). Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, 1(4):307–326.

Bertazzi, L. and Speranza, M.G. (2012b). Matheuristics for Inventory Routing Problems. In Montoya-Torres, J.R., Juan, A.A., Huatuco, L.H., Faulín, J., and Rodríguez-Verjan, G.L., editors, *Hybrid algorithms for service, computing and* *manufacturing systems: routing, scheduling and availability solutions*, pages 1–14. IGI Global.

Bertazzi, L. and Speranza, M.G. (2013). Inventory routing problems with multiple customers. *EURO Journal on Transportation and Logistics*, 2(3):255–275.

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.

Campbell, A.M. and Wilson, J.H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1):2–15.

Chao, I-M., Golden, B.L., and Wasil, E. (1995). An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):25–44.

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.

Coelho, L.C., Cordeau, J.-F., and Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.

Coffman, E.G., Garey, M.R., and Johnson, D.S. (1984). Approximation Algorithms for Bin-Packing – An Updated Survey. In Ausiello, G., Lucertini, M., and Serafini, P., editors, *Algorithm Design for Computer System Design*, volume 284, pages 49–106. Springer Vienna.

Cordeau, J.F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.

Cordeau, J.-F and Maischberger, M. (2012). A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033–2050.

Côté, J.-F., Archetti, C., Speranza, M.G., Gendreau, M., and Potvin, J.-Y. (2012). A branch-and-cut algorithm for the pickup and delivery traveling salesman problem with multiple stacks. *Networks*, 60(4):212–226.

Dantzig, G.B. and Ramser, J.H. (1959). The truck dispatching problem. *Managment Science*, 6(1):80–91.

Drexl, M. (2013). A note on the separation of subtour elimination constraints in elementary shortest path problems. *European Journal of Operational Research*, 229(3):595–598.

Feo, T.A. and Resende, M.G.C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6(2):109–133.

Ford, L.R. and Fulkerson, D.R. (1956). Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404.

Francis, M.P., Smilowitz, K., and Tzur, M. (2008). The Period Vehicle Routing Problem and its Extensions. In Golden, B., Raghavan, S., and Wasil, E., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43, pages 73–102. Springer, New York.

Francis, P. and Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B: Methodological*, 40(10):872–884.

Francis, P., Smilowitz, K., and Tzur, M. (2006). The Period Vehicle Routing Problem with Service Choice. *Transportation Science*, 40(4):439–454.

Francis, P., Smilowitz, K., and Tzur, M. (2007). Flexibility and complexity in periodic distribution problems. *Naval Research Logistics*, 54(2):136–150.

Galler, Bernard A. and Fisher, Michael J. (1964). An improved equivalence algorithm. *Communications of the ACM*, 7(5):301–303.

García Calvillo, I.D. (2010). *Un Enfoque Metaheurístico para un Problema de Ruteo con Flexibilidad en las Fechas de Entrega*. PhD thesis, Universidad Autónoma de Nuevo León, México.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., New York, NY, USA.

Gaudioso, M. and Paletta, G. (1992). A Heuristic for the Periodic Vehicle Routing Problem. *Transportation Science*, 26(2):86–92.

Glover, F.W. and Laguna, M. (1997). *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, USA.

Glover, F.W., Laguna, M., and Martí, R. (2000). Fundamentals of Scatter Search and Path Relinking. *Control and Cybernetics*, 29(3):653–684.

Goel, A. and Gruhn, V. (2006). Solving a Dynamic Real-life Vehicle Routing Problem. In Haasis, H.-D., Kopfer, H., and Schönberger, J., editors, *Operations Research Proceedings 2005*, pages 367–372. Springer.

Golden, B.L., Assad, A.A., and Wasil, E.A. (2002). Routing Vehicles in the Real World: Applications in the Solid Waste, Beverage, Food, Dairy, and Newspaper Industries. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, pages 245–286. Society for Industrial and Applied Mathematics.

Groër, C., Golden, B., and Wasil, E. (2010). A library of local search heuristics for the vehicle routing problem. *Mathematical Programming Computation*, 2(2):79–101.

Gueguen, C. (1999). *'Méthodes de résolution exacte pour les problèmes de tournées de véhicules*. PhD thesis, École Centrale Paris.

Gusfield, D. (1990). Very Simple Methods for All Pairs Network Flow Analysis. *Society for Industrial and Applied Mathematics*, 19(1):143–155.

Há, M.H., Bostel, N., Langevin, A., and Rousseau, L.-M. (2014). An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. *Computers & Operations Research*, 43:9–19.

Hashimoto, H., Ibaraki, T., Imahori, S., and Yagiura, M. (2006). The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics*, 154(16):2271–2290.

Helsgaun, K. (2000). An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130.

Hemmelmayr, V.C., Doerner, K.F., and Hartl, R.F. (2009a). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802.

Hemmelmayr, V., Doerner, K., Hartl, R., and Savelsbergh, M.W.P. (2009b). Delivery strategies for blood products supplies. *OR Spectrum*, 31(4):707–725.

Hernández-Pérez, H., Rodríguez Martín, I., and Salazar González, J.J. (2016). A hybrid heuristic approach for the multi-commodity pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 251(1):44–52.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Lau, H.C., Sim, M., and Teo, K.M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research*, 148:559–569.

Lee, W.L. (2013). Real-life vehicle routing with non-standard constrains. In *Proceedings of the World Congress on Engineering*, volume 1, London, UK.

Letchford, A.N. and Salazar-González, J.J. (2015). Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730 – 738.

Lin, C., Choy, K.L., Ho, G.T.S., Chung, S.H., and Lam, H.Y. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4, Part 1):1118 – 1138.

Lin, S. and Kernighan, B.W. (1973). An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516.

Nagamochi, H. and Ibaraki, T. (1992). Computing edge–connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5(1):54–66.

Pacheco, J., Álvarez, A., García, I., and Ángel Bello, F. (2012). Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates. *Journal of the Operational Research Society*, 63(5):569–581.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A.L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Ralph, T.K., Kopman, L., Pulleyblank, W.R., and Trotter, L.E. (2003). On the capacitated vehicle routing problem. *Mathematical Programming*, 94(2–3):343–359.

Rusdiansyah, A. and Tsao, D. (2005). An integrated model of the periodic delivery problems for vending–machine supply chains. *Journal of Food Engineering*, 70(3):421–434.

Russell, R. and Igo, W. (1979). An assignment routing problem. *Networks*, 9(1):1–17.

Salhi, S., Imran, A., and Wassan, N.A. (2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulations and a variable neighborhood search implementation. *Computers & Operations Research*, 52(Part B):315–325.

Shmoys, D.B., Rinnooy Kan, A.H.G., Lenstra, J.K., and Lawler, E.L., editors (1985). *The Traveling Salesman Problem: A Guide Tour of Combinatorial Optimization*. John Wiley & Sons Ltd., New York, USA.

Toth, P. and Vigo, D., editors (2014). *Vehicle Routing: Problems, Methods, and Applications*. Society for Industrial and Applied Mathematics, Philadelphia, 2 edition.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*, 257(3):845–858.

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.

Zhang, J., Lam, W.H.K., and Chen, B.Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: Trade-off between cost and customer service. *Networks and Spatial Economics*, 13(4):471–496.