THE PERIODIC VEHICLE ROUTING PROBLEM WITH VISUAL ATTRACTIVENESS AND DRIVER CONSISTENCY

by
SAEEDEH AHMADI BASIR

Submitted to the Graduate School of Faculty of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Doctor of Philosophy

Sabancı University
July 2024

THE PERIODIC VEHICLE ROUTING PROBLEM WITH VISUAL ATTRACTIVENESS AND DRIVER CONSISTENCY

Approved by:

Asst. Prof. Amine Gizem Tiniç ....
(Dissertation Supervisor)

Asst. Prof. Esra Koca ....

Asst. Prof. Duygu Taş Küten ....

Assoc. Prof. Çağrı Koç ....

Asst. Prof. Sina Rastani ....

Date of Approval: July 24, 2024

ABSTRACT

THE PERIODIC VEHICLE ROUTING PROBLEM WITH VISUAL ATTRACTIVENESS AND DRIVER CONSISTENCY

SAEEDEH AHMADI BASIR

Industrial Engineering Ph.D DISSERTATION, July 2024

Dissertation Supervisor: Asst. Prof. Amine Gizem Tiniç

Keywords: Vehicle Routing, Logic-Based Benders Decomposition, Column Generation, Visual Attractiveness, Driver Consistency, Parallel Tempering, Adaptive Large Neighborhood Search

This thesis explores advanced methodologies and innovative approaches to the Periodic Vehicle Routing Problem (PVRP) and its variants. Initially, we propose a new vehicle flow formulation for the PVRP and strengthen it with valid inequalities. We also investigate two prominent formulations for the PVRP available in the literature: a commodity flow formulation, referred to as the load-based formulation, and a cut-based formulation which is adapted from a formulation originally developed for a variant of the PVRP. We also extend these formulations to model the PVRP with time windows (PVRPTW) and employ valid inequalities to tighten the resulting formulations. A comprehensive computational study is then carried out to compare the performances of alternative PVRP and PVRPTW formulations on various sets of benchmark instances with different characteristics. The results attest to the robustness and the consistency of the proposed formulation and its strengthened versions in producing good quality solutions, especially for large instances although the load-based formulations tend to perform well in small instances. Subsequently, we address the PVRP using a Logic-Based Benders Decomposition approach and a Column Generation-based heuristic. Our findings reveal that the Column Generation algorithm achieves near-optimal solutions, deviating by an average of only 0.21% from the best-known solutions in the literature. Further, we incorporate visual attractiveness and driver consistency constraints into the PVRPTW, developing a Mixed-Integer Linear Programming formulation for this extended problem

(PVRPTWVADC). To solve the PVRPTWVADC, we propose an Adaptive Large Neighborhood Search (ALNS) algorithm and a Parallel Tempering-based ALNS (PTALNS). Comprehensive computational studies highlight the robustness and superior performance of the PTALNS algorithm.

ÖZET

GÖRSEL ELVERIŞLILIK VE SÜRÜCÜ TUTARLILIĞI KISITLARI ILE PERIYODIK ARAÇ ROTALAMA PROBLEMI

SAEEDEH AHMADI BASIR

Endüstri Mühendisliği DOKTORA TEZİ, Temmuz 2024

Tez Danışmanı: Dr. Öğr. Üyesi Amine Gizem Tiniç

Anahtar Kelimeler: periyodik araç rotalama, mantık-temelli Benders ayrıştırması, sütun türetme, görsel elverişlilik, sürücü tutarlılığı, paralel temperleme, adaptif geniş komşuluk arama

Bu tezde, periyodik araç rotalama problemi (PARP) ve varyantlarına yönelik yenilikçi ve etkin çözüm yaklaşımları geliştirilmesine odaklanılmıştır. İlk olarak, PARP için yeni bir araç akış formülasyonu önerilmiş ve geçerli eşitsizliklerle güçlendirilmiştir. Ayrıca, literatürde PARP için mevcut olan iki önemli formülasyon incelenmiştir: yük-temelli formülasyon ve başlangıçta PARP'nin bir varyantı için geliştirilmiş bir formülasyondan uyarlanan kesi-temelli formülasyon. Bu formülasyonlar, zaman pencereli PARP'yi (PARP-ZP) modelleyecek biçimde genişletilmiş ve elde edilen formülasyonları güçlendirmek için yine geçerli eşitsizliklerden ve optimalite kesilerinden faydalanılmıştır. Farklı özelliklere sahip çeşitli problem örnekleri üzerinde, alternatif PARP ve PARP-ZP formülasyonlarının performanslarını karşılaştırmak için kapsamlı bir hesaplama çalışması gerçekleştirilmiştir. Sonuçlar, özellikle büyük örnekler için, önerilen formülasyonun ve güçlendirilmiş versiyonlarının, iyi kalitede çözümler üretme konusunda gürbüzlüğünü ve tutarlılığını doğrulamakta, ancak yük-temelli formülasyonların küçük ve orta örneklerde üstün performans gösterme eğiliminde olduğuna işaret etmektedir. Ardından, PARP'yi çözmek için bir mantık temelli Benders ayrıştırma (MTBA) yaklaşımı ve sütun türetmeye dayalı bir sezgisel yöntem geliştirilmiştir. Elde edilen bulgular, sütun türetme algoritmasının, literatürde bilinen en iyi çözümlere kıyasla ortalamada yalnızca %0.21 sapmaya sahip çözümler belirleyebildiğini ortaya koymuştur. Son olarak, PARP-ZP'ye görsel elverişlilik ve sürücü tutarlılığı kısıtlarının eklenmesiyle elde edilen genişletilmiş problem (PARP-ZPGEST) için bir karışık tamsayılı programlama formülasyonu geliştirilmiştir. Problemi etkin bir şekilde çözmek için bir adaptif geniş komşuluk arama algoritması ve buna dayalı bir paralel temperleme yöntemi önerilmiş ve önerilen algoritmanın performansı, yapılan hesaplama çalışmaları ile doğrulanmıştır.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisors, Prof. Dr. Güvenç Şahin and Assist. prof. Dr. Gizem Özbaygın, for their unwavering guidance, support, and encouragement throughout my PhD journey. Their insights and expertise have been invaluable in shaping this research.

I am also deeply grateful to Professors Prof. Dr. Bulent Çatay and Assist. prof. Dr. Esra Koca for the enlightening courses I had the privilege to attend. The knowledge and skills I gained from their classes were instrumental in the successful completion of this thesis.

My sincere thanks go to my jury members, Asst. prof. Dr. Duygu Taş, Assoc. prof. Dr. Çağrı Koç, Asst. prof. Dr. Esra Koca, and Asst. prof. Dr. Sina Rastani, for their insightful comments and suggestions, which significantly contributed to the improvement of this thesis.

My heartfelt thanks go to my parents and my sister for their endless love and support, and to Soosan and Hossein, who have embraced me like their own daughter, providing me with a second family.

With all my heart, I want to thank Yasaman, my dearest friend and confidante. Her unwavering friendship has been an endless well of strength and joy, filling every step of this journey with warmth and love.

Many thanks to my friends Amin, Sina, Siamak, and Nozir, for always supporting me. We have shared not only the Logistics Lab but also countless joyful moments together. Additionally, I have cherished the company of my friends at Sabanci University, far too numerous to be listed individually. However, I would like to specifically mention Ali, Hamid, Sepideh, Shahrzad, and Mahdi with whom I spent the most time; they provided me with unwavering support and immense joy.

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) Grant No. 120M358.

To Mohammad Hosseini & Mohammad Mehdi Karami

Women Life Freedom

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The periodic vehicle routing problem (PVRP) is a generalization of the vehicle routing problem (VRP) in which customers require one or several visits within a multi-period planning horizon. Visit dates are not fixed; instead, a list of allowable visiting options is associated with each customer. Without labeling the problem with a specific name, Beltrami and Bodin (1974) introduce it in a study about garbage collection in which some sites require service three times a week (Monday, Wednesday, and Friday; or Tuesday, Thursday, and Saturday), whereas other sites should be visited six times a week. Russell and Igo (1979) name the problem as the assignment routing problem and allow more flexible visiting options making all days of the week acceptable for each customer. Christofides and Beasley (1984), naming the problem as the period routing problem, consider predetermined allowable visiting options for each customer as in Beltrami and Bodin (1974). Gaudioso and Paletta (1992) use the term periodic vehicle routing for the first time. They enforce a minimum and a maximum number of days between two consecutive visits to a given customer.

The PVRP has many real-life applications; multi-period routing plans with customers or points of interest requiring regular and frequent visits are generic examples. As evidenced in Beltrami and Bodin (1974) as the very first study, the waste collection systems are straightforward applications of the PVRP. More recently, special purpose waste collection such as used oil and recyclable materials have become even more common (Elbek and Wøhlk, 2016). In urban freight logistics, planning for the distribution of Fast-moving consumer goods (FMCG) orders to retailers and sales points can also be considered as another solid example as the frequency of visits to customers, that is, the number of times each customer should be visited during the planning horizon, varies significantly (Çopur et al., 2020). In the care of elderly and disabled persons, the services are provided to users at different frequencies within a certain period of time depending on their needs and requests; the visit schedules of the care personnel can be constructed by solving a PVRP (Alves. et al., 2019).

The PVRP has a rich history of research since Beltrami and Bodin (1974). A variety of approaches have been proposed to model and solve the problem. A synopsis of alternative PVRP formulations up to 2008 is presented by Francis et al. (2008). Considering the various modeling approaches in the literature, our study aims to provide an overview and a detailed computational comparison of the most prominent PVRP formulations.

In particular, we investigate three alternative formulations: (1) a cut-based formulation involving vehicle flow variables and exponentially many subtour elimination constraints (SECs), (2) a commodity flow formulation referred to as the load-based formulation, involving both vehicle and commodity flow variables, and a polynomial number of SECs, and (3) a new mixed integer programming formulation, the so-called MTZ-based formulation, involving vehicle flow variables and a polynomial number of Miller-Tucker-Zemlin type SECs (MTZ-SECs), which combines several aspects of the existing modeling techniques. Our primary goal is to study the computational performance of these formulations. We also extend these formulations by adding time window constraints, which are commonly encountered in practice, and we investigate how the resulting formulations perform in solving the PVRP with time windows (PVRPTW). Moreover, we employ families of valid inequalities to tighten the formulations.

Our study also brings together various data sets from the literature. We conduct an extensive computational study on seven data sets with varying characteristics to provide a thorough comparison of the aforementioned formulations as well as to assess the benefits of using valid inequalities. The early PVRP benchmark instances are presented by Russell and Igo (1979), Christofides and Beasley (1984), Russell and Gribbin (1991), and Chao et al. (1995). Several studies use these instances as is while others adapt them based for their problem structures (Cordeau et al., 1997), or generate their own data sets with different characteristics (Rodriguez-Martin et al., 2019). Cordeau et al. (2001) adapt the ten basic PVRP instances of Cordeau et al. (1997) to the PVRPTW. Pirkwieser and Raidl (2009a) assign random allowable visit schedules to the customers in the famous Solomon data set for the VRP with time windows (Solomon, 1987).

Furthermore, the concept of "visual attractiveness" in routing, though not precisely defined (Constantino et al., 2015), has been increasingly recognized for its practical importance. It generally encompasses certain desirable features for routes:

- compactness (Hollis and Green, 2012; Matis, 2008; Matis and Kohání, 2011; Poot et al., 2002; Rossit et al., 2016; Tang and Miller-Hooks, 2006);

- non-overlapping or non-crossing routes (Hollis and Green, 2012; Poot et al., 2002; Kim et al., 2006; Rossit et al., 2016; Toth and Vigo, 2014; Lu and Dessouky, 2006; Tang and Miller-Hooks, 2006; Matis, 2008);

- not complex (Constantino et al., 2015; Mesa et al., 2024);
- driver consistency (Braekers and Kovacs, 2016a; Rodriguez-Martin et al., 2019; Rodríguez-Martín and Yaman, 2022)

Visual attractiveness in routing is highly relevant in practical applications. Routes that are compact and well-separated are often more acceptable to practitioners and facilitate the implementation of routing plans. In some cases, compactness is even a design requirement, such as in area-based distribution systems for parcel delivery (Schneider et al., 2015). The differences between solutions optimized for traditional objectives, like minimizing length or cost, and those optimized for visual attractiveness are stark. For instance, Figures 1.1 and 1.2 illustrate solutions for PVRPTW with and without considering non-crossing routes and driver consistency constraints.

Figure 1.1 Solution routes to the PVRPTW without considering visual attractiveness and driver consistency constraints

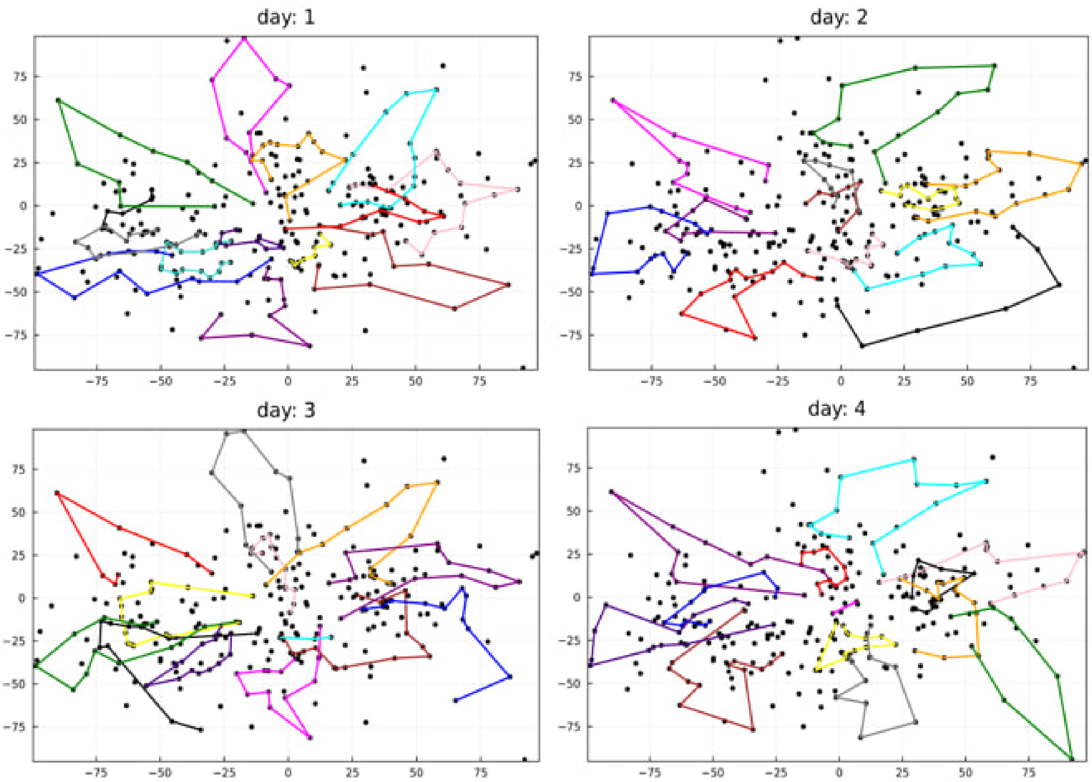
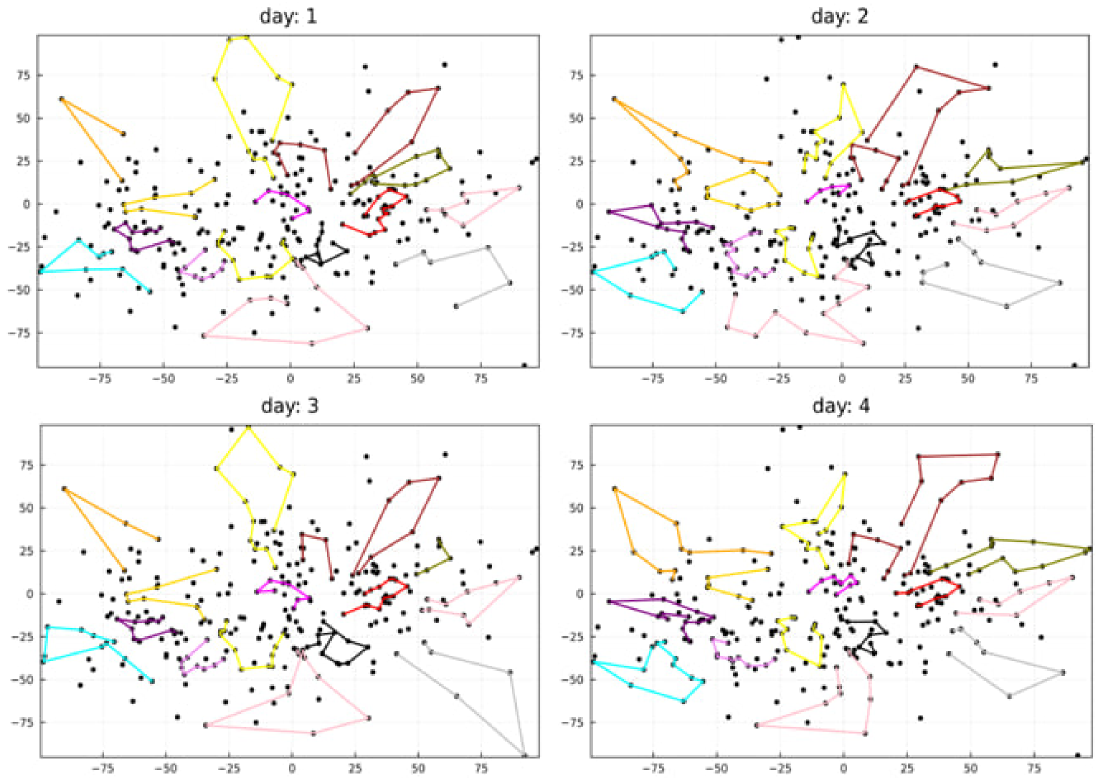


Figure 1.2 Solution routes to the PVRPTW considering visual attractiveness and driver consistency constraints



Despite the inherent vagueness in defining visual attractiveness, this concept has often been central in designing routing plans. To the best of our knowledge, Poot et al. (2002) were the first to use this term, reflecting their customers' requirements. They noted that some customers considered the outputs from the ORTEC vehicle routing software (http://www.ortec.com/) to be "poor." This dissatisfaction was not only due to traditional metrics such as cost, total number of vehicles used, or total distance traveled but also to non-standard indicators customers used to assess plan acceptability. They observed that these non-standard measures were not well-studied in the scientific literature and suggested that visually attractive plans seemed more logical and aligned with traditional working methods, thereby fostering trust among both drivers and planners, leading to quicker system acceptance (Poot et al., 2002).

Since then, the importance of considering visual attractiveness has been frequently emphasized in the literature, with its foundations largely based on practical applications. According to Matis (2008), overlapping routes can lead to complaints from drivers, who perceive such planning as inefficient. Practitioners generally dislike routes optimized solely for length that spread across different areas while intersecting each other (Mourgaya and Vanderbeck, 2007). "Nice" solutions are often easier to implement practically, reducing the time needed to instruct drivers and leading to more stable route durations due to their homogeneity in traffic conditions (Battarra et al., 2014; Schneider et al., 2015). Routes can be further refined by familiarity with the area and clients (Kant et al., 2008; Poot et al., 2002). In addition, if customer service cannot be provided at the preferred time, returning later is easier if the vehicle remains in the same area. Similarly, in the case of traffic jams or road disruptions, finding alternative routes is simpler when customers are in a compact area (Hollis and Green, 2012).

Battarra et al. (2014) described applications where route compactness is crucial, such as transporting elderly people to recreation centers, where users prefer being picked up with neighbors, or in "gated communities" where customers should be visited sequentially by the same vehicle to avoid time-consuming stops at checkpoints. Another example is household newspaper delivery, where it is undesirable to serve the same area with multiple carriers, as neighboring subscribers may receive their newspapers at very different times (Hasle et al., 2011).

In the home health care system, it is highly preferred that customers are visited by the same nurse throughout their treatment planning horizon. This continuity fosters trust and comfort, improves the quality of care, and ensures a better understanding of the patient's needs and medical history (Yang et al., 2021).

From our perspective, the importance of visual attractiveness lies mainly in its proven critical role in practical applications. Planning a near-minimum-cost routing plan that is unattractive and likely to be rejected by practitioners or modified according to their preferences can be a waste of time and effort. Therefore, generating visually appealing routing plans can enhance customer satisfaction and reduce implementation costs.

The contributions of this study are as follows:

- **Mixed Integer Linear Programming (MILP) Formulations:**
  - We investigate the mixed integer linear programming (MILP) formulations of the PVRP and the PVRPTW with a unified and comparative perspective,
  - We present a new formulation using vehicle flow variables, and
  - We conduct a comprehensive computational study in which we bring together seven sets of benchmark instances with different characteristics.

To the best of our knowledge, this is the first study exploring the behavior of traditional PVRP(TW) formulations on such a diverse set of problem instances. We believe that our findings will provide guidance to those wishing to study and develop solution methods for the PVRP(TW) and its variants as well as other closely related problems in the future.

- **Optimization Methodologies for PVRP:**
  - We propose Logic-Based Benders Decomposition (LBBD) algorithms to solve the PVRP.
  - We introduce a column generation-based heuristic to address the same problem.
- **Enhancements to PVRPTW Formulations:**
  - We extend the cut-based PVRPTW formulation to incorporate non-crossing routes and driver consistency restrictions.
  - We utilize PTALNS to solve the problem effectively.

  By addressing these contributions, our study not only advances the theoretical understanding of PVRP(TW) formulations but also provides practical insights for improving routing plans in real-world applications.

The structure of this thesis is organized as follows:

- Chapter 2 presents a comprehensive literature review, outlining the relevant studies and methodologies in the field.
- Chapter 3 is a comparative study of alternative PVRP(TW) formulations in which in Section 3.1, the PVRP is formally defined, four formulations are presented, and corresponding valid inequalities are introduced; then, all three formulations, defined on the directed network, are extended to the PVRPTW as well. In Section 3.2, the computational results on the data sets from the literature are reported and discussed. Finally, Section 3.3 presents our findings and concluding remarks.
- In Chapter 4, we address the PVRP using Logic-based Benders Decomposition (LBBD) algorithms and a column generation-based heuristic.
  - We propose LBBD algorithms in Section 4.1, which comprises five subsections. In Section 4.1.1, we introduce the Generalized Assignment Master Problem. Section 4.1.2 presents the Vehicle Routing Sub-Problems, followed by an introduction to Benders' cuts in Section 4.1.3. Section 4.1.4

explains the implementation of the proposed algorithms, and Section 4.1.5 covers the computational experiments.

  - Section 4.2 introduces a column generation-based heuristic to address the same problem, also divided into five subsections. Section 4.2.1 presents the problem definition and formulation. Section 4.2.2 details the column generation approach. Section 4.2.3 discusses the pricing subproblem. Section 4.2.4 describes the Integer solution of the problem. Section 4.2.5 provides the computational results. Finally, Section 4.3 offers the conclusion of the chapter.

- In Chapter 5 we take into account the visual attractiveness and driver consistency constraints and define PVRPTW with visual attractiveness and driver consistency (PVRPTWVADC). In Section 5.1, we define the problem and provide detailed formulations. In Section 5.2, we introduce an adaptive large neighborhood search (ALNS) algorithm, while Section 5.3 delves into the Parallel Tempering technique. Section 5.4 integrates these methodologies, describing the parallel tempering-based adaptive large neighborhood search (PTALNS) Algorithm. The effectiveness and performance of the proposed approach are evaluated in Section 5.5 through a thorough computational study. Finally, Section 5.6 concludes the chapter, summarizing key findings and suggesting directions for future research.

2. Literature Review

The PVRP is introduced in Beltrami and Bodin (1974), where, for the first time, periodicity of customer deliveries is taken into account in addition to routing decisions. However Beltrami and Bodin (1974) do not present a mathematical formulation of the problem. Another influential study in the early history of the PVRP by Russell and Igo (1979) approaches the PVRP as the assignment routing problem and provides a verbal description of the parameters and the constraints of the problem in detail, yet again without a formulation. Christofides and Beasley (1984) are credited for providing the first mathematical formulation of the PVRP in many studies, but in fact, Foster and Ryan (1976) made the earliest attempt to formulate the PVRP as a vehicle scheduling problem incorporating the periodic delivery requirements. Although the work by Foster and Ryan (1976) predates both Christofides and Beasley (1984) and Russell and Igo (1979), only Russell and Igo (1979) cite Foster and Ryan (1976). In general, most of these studies have been cited by the majority of the subsequent studies due to the pioneering roles they played in the evolution of the PVRP literature. In a more recent study, Campbell and Wilson (2014) discuss the wide applicability of the PVRP and describe the development of solution methods. In line with the focus of our study, in sections 2.1 and 2.2 we particularly focus on the modeling approaches and formulations as well as other VRP variants that can be considered as extensions or relatives of the PVRP. In Section 2.3, we explore the importance of visual attractiveness in routing problems from various perspectives.

2.1 Modelling Approaches

Considering the mathematical modeling approaches available in the PVRP literature, vehicle-indexed formulations seem to be relatively more popular (Cordeau et al., 1997; Christofides and Beasley, 1984; Huerta Muñoz, 2018; Alves. et al., 2019). Christofides and Beasley (1984) define the PVRP as the problem of assigning customers to schedules to meet the required visit frequencies and designing a set of routes for the customers assigned to be visited on each day of the planning horizon in a separate subproblem. They present an integer programming (IP) formulation for the problem using two sets of decision variables, one for assigning customers to visit combinations and another for routing on each day. To circumvent computational difficulties of routing constraints, Tan and Beasley (1984) use a simplified VRP formulation of Fisher and Jaikumar (1981) in which the routing constraints are not explicitly specified. They define a cost measure θ that represents the cost of visiting a specific node with a specific vehicle on a specific day and use two sets of binary decision variables to formulate the problem.

Christofides and Beasley (1984) propose an aggregate vehicle-indexed formulation for the PVRP (PVRP-A) by using aggregated binary variables which determines if a customer is visited on a specific day without specifying the vehicle identity. Huerta Muñoz (2018) develops a disaggregate vehicle-indexed formulation for the PVRP (PVRP-D) and compares it with the PVRP-A. According to the results, the PVRP-D produces better quality solutions than the PVRP-A.

Another approach to model the PVRP is by using commodity flow variables, instead of vehicle-indexed variables, to identify the total flow (i.e. the total quantity transported) on a given arc for each day of the planning horizon without addressing the specific vehicle that traverses the arc; this approach is referred to as the load-based formulation. Archetti et al. (2017) propose the first so-called load-based formulation for the problem. Huerta Muñoz (2018) compare it with the PVRP-D and the PVRP-A. The results show that the load-based formulation outperforms both the PVRP-D and the PVRP-A.

In vehicle-indexed formulations, mostly to guarantee the connectivity of the vehicle routes, standard SECs are used (Cordeau et al., 1997; Christofides and Beasley, 1984). Huerta Muñoz (2018) reinforces the classical SECs to have a stronger linear programming relaxation. Archetti et al. (2017) define a set of three-indexed commodity flow variables to keep track of the vehicles' load as they traverse the arcs to avoid subtours and capacity violation. To prevent the formation of subtours, Alves. et al. (2019) use a set of three-indexed integer variables which correspond to the place of each customer in the sequence of visits of each vehicle on each day.

As it is common with other variants of the VRP, some studies present set covering and set partitioning formulations of the PVRP (Foster and Ryan, 1976; Baldacci et al., 2011; Cacchiani et al., 2014). It is interesting to observe that the only exact solution algorithm was proposed ten years ago by Baldacci et al. (2011), and yet, it is still credited for being the state-of-the-art methodology to solve the PVRP. This algorithm is based on a set partitioning formulation of the PVRP. The authors propose three different relaxations of their set partitioning formulation, which are then used to derive strong lower bounds for the problem. Based on these lower bounds and the information collected through a related dual solution, they reduce the number of variables in the formulation without eliminating any optimal integer solution. The reduced integer program is solved exactly by a commercial solver. Computational results show that the proposed method can achieve lower bounds within 1% of optimality on the average, improves the best-known upper bound for five instances, and solves several test instances to optimality for the first time. Rothenbächer (2019) uses their exact branch-and-price-and-cut algorithm, originally proposed to solve the PVRPTW, also to solve some PVRP instances from the literature.

2.2 Relatives and Extensions of the PVRP

The multi-depot VRP (MDVRP) is a single-period VRP with multiple depots in which each vehicle is assigned to one of the given depots. Cordeau et al. (1997) show that the MDVRP can be formulated as a special case of the PVRP by associating the depots in the MDVRP with the periods in the PVRP, and by enforcing a single visit to every customer throughout the planning horizon. Consequently, any method that can solve the PVRP can also solve the MDVRP. Bettinelli et al. (2011) introduce a set-covering formulation for the multi-depot heterogeneous VRP with time windows and develop a branch-and-cut-and-price algorithm to solve the problem. The algorithm adopts different combinations of cutting planes and pricing strategies. It can also be used as a column generation-based heuristic for large-scale instances that cannot be solved to proven optimality. Computational results show that the proposed methodology is competitive and often better in comparison with local search heuristics from the literature.

Contardo and Martinelli (2014) provide an ad-hoc compact two-indexed IP formulation, namely the vehicle-flow formulation, and a set-partitioning formulation for the MDVRP. Several families of valid inequalities are used to strengthen both formulations. They present an exact algorithm for the MDVRP which employs the ad-hoc vehicle-flow and set-partitioning formulations at different stages. The first one is used to derive a lower bound and implement variable fixing. The second is solved by column-and-cut generation considering the reduced network obtained through the variable fixing procedure. The algorithm proves optimality for some previously open instances. For the instances that the algorithm is not able to solve, the terminal lower bounds are stronger than those obtained by earlier methods.

Francis et al. (2006) generalize the definition of periodicity and introduce the PVRP with service choice in which service frequency is a decision of the problem. They present an IP formulation for the PVRP-SC by extending the VRP formulation of Fisher and Jaikumar (1981) to accommodate service choice and develop an exact procedure to solve the problem. They deploy a Lagrangian relaxation scheme to separate the decisions on scheduling customer visits from the routing decisions, and create two subproblems. Then, they apply a subgradient optimization procedure on both subproblems to obtain lower bounds. At the end of the Lagrangian relaxation phase, if the solution, constructed along the way as the best-known upper bound, equals to the lower bound, an optimal solution is found. Otherwise, a branch and bound algorithm is used to close the remaining gap between the upper and lower bounds. They also present a heuristic variation of their exact method for larger problem instances.

Archetti et al. (2015a) propose MILP formulations for the multi-period VRP with due dates (MVRPD), where customers have to be served between predefined release and due dates. Their first formulation is a flow-based formulation (a vehicle flow formulation) containing a set of four-indexed binary decision variables indicating whether a given arc is traversed by a particular vehicle on a particular day. The flow-based formulation is then extended by defining additional variables to assign customers to vehicles and days, which yields another formulation. Finally, a load-based formulation, where aggregated vehicle flow variables are used, is obtained by adding commodity flow variables representing the quantity transported via each arc in the network. The authors present a series of valid inequalities for each formulation, and a branch-and-cut algorithm to solve the formulations. The load-based model outperforms the flow-based formulations according to the results of a computational study. Larrain et al. (2019) introduce two new classes of valid inequalities for the load-based model of Archetti et al. (2015a) to improve the performance of the branch-and-bound algorithm proposed by Archetti et al. (2015a). The enhanced branch-and-bound algorithm is further improved by a variable MIP neighborhood descent search algorithm (VMND) with local search operators in which whenever the exact phase of the algorithm reaches a new upper bound, the local search procedure is executed to speed up the search for high-quality solutions around the current solution. Upon termination of the local search phase, the exact solution algorithm resumes execution.

Rodriguez-Martin et al. (2019) propose a MILP model of the PVRP with driver consistency constraints which enforce all visits to a given customer within the planning horizon to be made by the same vehicle. They present several families of valid inequalities to strengthen the linear programming relaxation of their formulation and develop an exact branch-and-cut algorithm to solve the problem. On a new set of randomly generated instances, the authors demonstrate the efficiency of their algorithm as well as the benefits of the suggested valid inequalities and several algorithmic enhancement procedures implemented. The algorithm is able to solve to optimality all the feasible instances with 10, 20, and 30 customers and several instances with 50, 60, and 70 customers.

Rothenbächer (2019) addresses the PVRPTW and its variant with fully flexible schedule sets. The author presents a route-based extended set-partitioning formulation for the PVRPTW and an exact branch-and-price-and-cut algorithm to solve both problems. Results of a computational study with the PVRPTW benchmark instances of Cordeau et al. (2001) demonstrate that the suggested algorithm is able to identify lower bounds that are close to the best-known upper bounds for several large instances. However, tests on the PVRP instances indicate that the algorithm could not produce competitive results when compared to those reported by Baldacci et al. (2011).

2.3 Visual Attractiveness

As previously mentioned, the concept of visual attractiveness, introduced by Poot et al. (2002), is relatively new in the routing literature. However, earlier studies have addressed similar ideas, primarily focusing on route compactness.

Tang and Miller-Hooks (2006) develop an iterative heuristic for solving the VRP with maximum travel time constraints using a clustering-based algorithm. Initially, seed customers are selected, and other customers are assigned to the nearest seed via a Semi-Assignment Problem (SAP). A TSP heuristic then determines the schedule and routing time for each cluster. If a route exceeds the maximum travel time, the SAP is resolved with a modified cost matrix, increasing distances to overloaded clusters and reducing distances to feasible ones. If SAP still does not yield a feasible solution after several iterations, a Multi-objective Assignment Problem is used to reassign customers, aiming to minimize the number of customers closer to another route's center and the total travel times while explicitly limiting each route's travel time. The algorithm was tested on real-life instances from FedEx and compared with Tang and Hu (2005), showing the expected trade-off between visual attractiveness and standard objectives.

Zhou et al. (2006) and Lu and Dessouky (2006) present an insertion heuristic to solve the multi-vehicle VRP with Pickup and Delivery with time windows that includes a crossing-avoidance penalty in the insertion cost calculation. Both studies note that it is relatively easy to find feasible and inexpensive insertions without crossings at the beginning of route construction due to the availability of many customers and the low initial route occupancy. However, as routes near capacity, the focus shifts from visual attractiveness to optimizing length and capacity utilization. To maintain some level of visual attractiveness, their algorithm allows for increasingly less attractive insertions by reducing the crossing-avoidance penalty as more customers are assigned. Their algorithm was tested on instances derived from Solomon (1987) and compared to the Sequential Insertion Algorithm and a parallel insertion heuristic, generally yielding better results in terms of both visual attractiveness and standard objectives. However, when compared with Li and Lim (2001), their solutions performed worse in terms of the number of vehicles used and travel time.

Kim et al. (2006) develop a clustering-based algorithm to enhance the visual attractiveness of VRPTW solutions for waste collection. The two-stage heuristic begins with a capacitated clustering algorithm to estimate and form clusters, followed by an extended insertion algorithm to route points within each cluster. Initially, customers are assigned to the nearest seed customer, and cluster centroids are calculated. Customers are then reassigned based on their distance from a "grand centroid" while maintaining capacity and travel time constraints. The process repeats until cluster compositions stabilize. The second stage sorts clusters and applies an insertion algorithm to determine routes, refining them further using a Simulated Annealing algorithm. Similarly, Sahoo et al. (2005) introduced a route-management application for waste collection that emphasizes route compactness. Their algorithm, resembling Kim et al. (2006) method, employs the K-means Variant Balanced Clustering Algorithm (Surya Sahoo PhD, 2004) instead of using a "grand centroid." Tested on Solomon (1987), their approach generally produced more visually appealing but less efficient solutions compared to the best-known ones, whereas Kim et al. (2006) focused on real-world waste collection problems.

Another area where companies prioritize generating visually attractive routes is product distribution. Kant et al. (2008) demonstrate a heuristic algorithm implemented via the ORTEC vehicle routing software, which resulted in substantial savings for the Coca-Cola Company. This algorithm combines an insertion method with local search, avoiding movements that cause route overlaps and incorporating a Clustering Penalty (CP) to discourage non-compact routes. The CP, proportional to the distance of customers from the route's central customer, is initially set high to create familiar routes for dispatchers and drivers. As they become accustomed to the new routes, the CP is lowered to optimize for cost. This gradual adjustment, suggested by Poot et al. (2002), facilitates a smoother transition to more efficient routes.

Poot et al. (2002) also collaborate with ORTEC clients to adapt a savings algorithm, originally by Clarke and Wright (1964), to emphasize route compactness by considering customer location, time windows, and vehicle type. Incorporating a "region" factor in the savings calculation improved route compactness, outperforming standard insertion algorithms. Bosch (2014) further modified the ORTEC software to include visual attractiveness constraints, based on Savelsbergh's circle covering method (Savelsbergh, 1990). This modification, inspired by experienced planners' manual improvements, led to cost reductions for the Zeeman chain store in the Netherlands.

Hollis and Green (2012) develop a complex heuristic algorithm to create visually attractive routes for Schweppes Australia Pty. Ltd. The algorithm operates in two stages: a novel variation of Solomon's Sequential Insertion Algorithm (Solomon, 1987), followed by a local search using the Guided Local Search Algorithm by Kilby et al. (1999). Recognizing that insertion techniques alone can result in elongated, non-compact routes, they implement an alternative insertion criterion. When a route nears its maximum duration, new customers are inserted based on proximity to already visited customers, avoiding elongated routes. Additionally, the local search algorithm aims to minimize both routing costs and the overlap of convex hulls associated with the routes. Testing on real-world instances from Melbourne and Solomon's VRPTW benchmark (Solomon, 1987) show that their algorithm while yielding more visually appealing routes, resulted in greater total distances and more routes compared to the best-known solutions.

Gretton et al. (2013) enhance the visual attractiveness of solutions generated by the Indigo software, designed by Kilby and Verden (2011). Their approach utilizes the ALNS method by Ropke and Pisinger (2006), which iteratively removes and reinserts a large set of customers using a heuristic. They incorporate an insertion algorithm that prioritizes visual attractiveness by considering the distance of customers to the route median (the customer nearest to the route's geometric centroid) and the sum of turn angles, referred to as bending energy. Their tests, conducted on literature benchmarks (Solomon, 1987; Gehring and Homberger, 1999) and real-world instances, demonstrate improved visual attractiveness of the routes.

Rossit et al. (2016) introduce a heuristic algorithm aimed at optimizing both visual attractiveness and standard costs in the capacitated VRP (CVRP). Their approach begins by generating an initial solution using a clustering-based algorithm similar to Surya Sahoo PhD (2004), followed by refinement through local search techniques. Testing on CVRP instances proposed by Uchoa et al. (2017) shows that their algorithm produced visually appealing solutions surpassing the best-known solutions in terms of aesthetic quality but with a trade-off of longer total route lengths.

Rocha et al. (2022) tackles the challenge of designing vehicle routes that are both cost-effective and visually attractive, suggesting that clustering can effectively proxy for visual attractiveness. The authors present a bi-objective CVRP model that aims to minimize travel costs and clustering criteria simultaneously. They employ a multi-objective evolutionary algorithm to approximate the Pareto frontier. Extensive computational experiments assess the impact of three clustering criteria: diameter minimization, min-sum of cliques, and minimum sum-of-squares. The results indicate that the latter two criteria yield high-quality, visually attractive solutions while maintaining low travel costs.

Recent studies have introduced new variants of PVRP with a focus on "consistency." Consistency can refer to the consistency of routes, as explored by Yao et al. (2021), or to the consistency of visit times and drivers. Time consistency ensures that visits to a customer occur at roughly the same time of day, which is crucial in scenarios such as administering medication or conducting tests in-home healthcare services. Driver consistency means that the same driver visits a customer each time, enhancing service quality and leveraging the driver's familiarity with customers, routes, and traffic conditions (Smilowitz et al., 2013).

The Consistent VRP, which considers both time and driver consistency, has been examined by Groër et al. (2009). In Consistent VRP, customer visit schedules are predetermined, with consistency requirements linking different periods. Goeke et al. (2019) proposed exact and heuristic algorithms to solve this problem, achieving optimal solutions for instances with 30 customers over five periods within reasonable time frames.

Other studies, such as those by Braekers and Kovacs (2016b), Zhu et al. (2008), and Luo et al. (2015), explore a broader concept of driver consistency by limiting the number of drivers visiting each customer. Kovacs et al. (2015a) introduce the generalized ConVRP, where a customer is visited by a limited number of drivers, and variations in arrival times are penalized. A multi-objective approach to this problem was studied by Kovacs et al. (2015b), while Campelo et al. (2019) examined a variant involving multiple daily deliveries and service level agreements.

Rodriguez-Martin et al. (2019) develop a branch-and-cut algorithm for the PVRP with Driver Consistency, where each customer must be visited by the same vehicle at each visit. This study does not consider consistency in visit times or vehicle travel time limits but requires the model to determine the visit schedules. Rodríguez-Martín and Yaman (2022) address a variant called the PVRP with Driver Consistency and Service Time Optimization, which not only determines routes and visit schedules but also optimizes service times to maximize service utility for the company. The authors present a MILP formulation for this problem and propose three branch-and-cut methods to solve it, including two methods based on Benders decomposition.

For a more comprehensive insight into visual attractiveness in routing problems, see Rossit et al. (2019). Their review paper provides a detailed exploration of the topic.

2.4 Solution Approaches

To solve the PVRP and its variants, various solution methods have been developed, capable of producing good solutions within a reasonable amount of time. The evolution of these methods can be traced from early heuristic approaches to more advanced metaheuristics and exact methods.

2.4.1 Early Approaches

Initial methods for solving the PVRP primarily relied on construction and improvement heuristics. Beltrami and Bodin (1974) introduce key heuristics involving the Clarke and Wright procedure and random assignment of customers to delivery days. In the heuristic presented by Russell and Igo (1979) customers are clustered based on delivery requirements, followed by three heuristics: initial scheduling based on cost estimates, link exchanges for improvement, and a modified Clarke and Wright method. Christofides and Beasley (1984) present a heuristic for solving VRPs by initially assigning customers based on delivery constraints and quantities, then refining through cost-minimizing combinations. They improve solutions via customer swaps and solve new problems as median problems or TSPs using heuristics, testing with the dataset from Russell and Igo (1979) and 10 new instances.

2.4.2 Metaheuristics

As computational power increased, metaheuristics gained popularity due to their ability to handle larger instances effectively.

Tabu Search: Cordeau et al. (1997), introduce the first tabu search algorithm for PVRP, Periodic TSP, and MDVRP. It randomly assigns customers to delivery days, then optimizes routes using insertion heuristics and neighborhood moves, with penalties for infeasible solutions to encourage diversity. Their method, requiring fewer user parameters, produced new best solutions on many instances. This approach inspired numerous adaptations for related problems, such as PVRPTW, PVRP incorporates intermediate facilities, and applications in various industries, enhancing computational power and operational flexibility.

Variable Neighborhood Search: Hemmelmayr et al. (2009) improve best-known solutions using Variable Neighborhood Search (VNS), which alternated neighborhoods to escape local optima. They allowed inferior solutions similar to simulated annealing, which proved effective on canonical datasets. Pirkwieser and Raidl (2010) extended this approach with multilevel refinement, abstracting and solving simpler versions of the problem.

Adaptive Large Neighborhood Search: Dayarian et al. (2016) investigate the design of tactical plans for a transportation problem inspired by real-world milk collection in Quebec. To account for seasonal variations, they modeled the problem as a multi-period VRP and developed an ALNS algorithm incorporating several heuristics. Testing the algorithm on a large set of instances of different sizes, they compared results for smaller instances with existing exact solutions and computed lower and upper bounds for larger instances.

Other Metaheuristics: Several other metaheuristics have been developed for the PVRP. Ochi and Rocha (2000) propose a genetic algorithm combined with local search. Vidal et al. (2012) enhanced genetic search by increasing population diversity. Matos and Oliveira (2004) applied ant colony optimization, while Gonçalves et al. (2005) used a greedy-randomized adaptive search procedure.

2.4.3 Exact Solution Approaches

Christofides and Beasley (1984) is well cited by PVRP papers, likely since they provide the first IP formulation for the PVRP. Although many publications reference or extend that formulation or provide their own mathematical formulation, most admit that solving these formulations is limited to at most moderate-size problems and resort to other solution approaches. Mostly, Heuristics and metaheuristics are applied, some based on the information provided by the IPs. However, a few publications exploit deeply the flexibility of mathematical programming approaches to solve the PVRP or its variations.

Francis et al. (2006) generalize the definition of periodicity in PVRP and introduce the PVRP with service choice. They present an IP formulation for the PVRP-SC by extension of Fisher and Jaikumar (1981) VRP formulation in several dimensions to accommodate service choice and propose an exact solution procedure to solve the problem. The first component is the Lagrangian relaxation of one constraint and separates the decisions on scheduling customers from routing to create two subproblems. Then, they apply a subgradient optimization procedure on a Lagrangian function to develop lower bounds on the solution. At the end of the Lagrangian relaxation phase, if the solution, constructed along the way as the best-known upper bound, equals the lower bound, the optimal solution is reached. Otherwise, a branch and bound algorithm is used to close the remaining gap between the upper and lower bounds. They also present a heuristic variation of the exact method. These solution methods are limited to problem instances of medium size.

Francis and Smilowitz (2006) present a continuous model by approximating discrete variables and parameters of the formulation presented in Francis et al. (2006) with continuous functions. They propose a solution approach based on geographic decomposition and variable substitution. Utilizing this approach leads to a simple problem that can be solved quickly. Therefore, it can be useful for larger problems. They apply this approach to the 100b PVRP dataset from Christofides and Beasley (1984). Although this approach does not improve the best-known value, it is competitive. Its quick results can also be used to help design valuable service options.

Mourgaya and Vanderbeck (2007) develop a model with a twofold objective: optimization of regionalization and balancing the workload across vehicles. To solve the relaxed problem, they employ a Dantzig–Wolfe reformulation and column generation. They apply insertion heuristics to price out columns with considering balancing the two objectives. A feasible solution for the PVRP is produced by rounding the solution, obtained at the end of the LP solution phase, by exploring the branch and bound tree. They use some instances from the Cordeau dataset (Cordeau et al., 1997) to evaluate the solution approach. Although this approach produces costly routes, it can compete with the approach of Cordeau et al. (1997) when regionalization and workload balancing are taken into account.

Rodriguez-Martin et al. (2019) propose an integer linear mathematical formulation for the PVRP with driver consistency in which each customer should be visited by the same vehicle within the planning horizon. They presented several families of valid inequalities to strengthen the LP relaxation of the problem. They developed an exact algorithm, which is a combination of a branch and bound method and a cutting-plane method, to solve the problem. The latter is used to derive lower bounds by solving the improved LP relaxation of the problem. They generated their own benchmark instances to evaluate the performance of the algorithm. Comparing the branch and bound algorithm with other simplified versions of it shows the efficiency of applied valid inequalities and other procedures in the algorithm. The algorithm is able to solve to optimality all the feasible instances with 10, 20, and 30 customers and several instances with 50, 60, and 70 customers.

Baldacci et al. (2011) propose a new IP for the PVRP and three different relaxations of this formulation, which are used to derive strong lower bounds for the problem. Without eliminating any optimal integer solution, they reduce the number of variables in the formulation, using the incorporation of these lower bounds and the information gained from a related dual solution. The reduced integer problem is solved exactly. Computational results show that the exact method can achieve the lower bound on average within 1% of optimality. The algorithm improves the best-known upper bound for five instances and solves several test instances to optimality for the first time. Their proposed algorithm currently is credited as the state-of-the-art methodology for the exact solution of the PVRP.

Archetti et al. (2015b) propose three alternative IP and MIP formulations for the multi-period vehicle routing problem with due dates (MVRPD), where customers have to be served between a predefined release and a due date. They present a series of valid inequalities for each formulation. The formulations were solved with a branch and cut algorithm. According to computational results aggregated load-based model (MIP) outperformed the other two flow-based formulations (IPs).

Larrain et al. (2019) formulate MVRPD as a mixed-integer programming problem. They improve the performance of the branch-and-bound algorithm ($B\&B-1$) proposed by Archetti et al. (2015b) by introducing two new classes of valid inequalities. They compare the enhanced algorithm ($B\&B-2$) with $B\&B-1$ through computational experiments. They also present a hybrid algorithm for the MVRPD, which combines variable MIP neighborhood decent search (VMND) and branch-and-bound algorithm. The idea is to speed up the search for the high-quality solution in the local search heuristic embedded in $B\&B-2$ through VMND and define search operators through variable fixing on MILP problems. Computational results show that the new branch-and-bound algorithm ($B\&B-2$) beats the $B\&B-1$ in 47 instances, ties in 26, loses in 7 and reduces the average optimality gap on each group of instances. The $B\&B-2$ increases the number of instances solved to optimality from 21 to 24 out of 80 and reduces the optimality gap from 12.1% to 5.1% on the instances with one vehicle. The hybrid algorithm improves the best-known solution for 35 out of 80 instances and reduced the average optimality gap from 5.1% to 3.6%.

Cordeau et al. (1997) show the MDVRP can be formulated as a special case of PVRP. Consequently, the same methodology can be applied to solve both problems. According to the survey on the MDVRP provided by Montoya-Torres et al. (2015), exact solution techniques are employed in 25% of reviewed papers on MDVRP. Baldacci and Mingozzi (2009) develop an exact method for solving different classes of VRP that is also capable of solving MDVRP. The algorithm is based on the set-partitioning formulation and on dual heuristics. Firstly, the algorithm applies a procedure to generate routes. Then, three bounding procedures based on the LP-relaxation and Lagrangian relaxation of the formulation are applied to reduce the number of variables. The reduced problem can be solved by an ILP solver. The computational results show that the exact algorithm was able to improve the best-known upper bound of some instances from the literature. They also improve the lower bound for the main instances from the literature.

Bettinelli et al. (2011) present a branch-and-cut-and-price algorithm for optimization of the multi-depot heterogeneous vehicle routing problem with time windows. The method is allowed to use different combinations of cutting and pricing strategies. When the scale of the instances makes the algorithm unable to find provably optimal solutions, the algorithm can be used as a column generation-based heuristic. Computational results show that these procedures are competitive and often better in comparison with local search heuristics from the literature.

Contardo and Martinelli (2014) formulate MDVRP through a vehicle-flow and a set-partitioning formulation. Several families of valid inequalities are used to strengthen both formulations. They present an exact algorithm for the MDVRP in which the formulations are exploited at different stages. The algorithm includes variable fixing, column-and-cut generation, and column enumeration. To find a tight lower bound, they eliminate non-promising edges by using the flow-based formulation. They solve the set partitioning formulation by using the reduced network obtained through the variable fixing procedure. The exact algorithm is able to prove optimality for some previously open instances. For the instances that the algorithm is not able to solve, the terminal lower bounds prove stronger than those presented by earlier state-of-the-art methods.

Rodríguez-Martín and Yaman (2022) explore PVRPDC, an extension of the classic VRP where routes for several vehicles must be determined over a multi-day horizon. Each customer has possible visit schedules and must always be visited by the same vehicle. They studied a variant incorporating service time optimization, aiming to maximize the utility of the service. They proposed a mixed-integer linear programming formulation and three branch-and-cut methods, two based on Benders reformulations, and reported computational results on diverse benchmark instances.

These advancements, particularly in metaheuristics and exact methods, have significantly improved the ability to solve complex PVRP instances efficiently, with ongoing research continuing to refine and extend these approaches.

3. A Comparative Study of Alternative Formulations for the Periodic Vehicle Routing Problem

3.1 Problem Definition and Formulation

The PVRP can be defined on a complete directed graph $G = (N, A)$ with $N = \{0, 1, \ldots, n\}$ being the set of nodes, where node 0 represents the depot and the nodes in the set $N_c = \{1, \ldots, n\}$ correspond to the customers. The set of arcs is given by $A = \{(i, j) : i, j \in N, i \neq j\}$. Each arc (i, j) is associated with a non-negative cost c_{ij}. Let $T = \{1, \ldots, \tau\}$ denote the set of time periods defining the planning horizon. A fleet of homogeneous vehicles $K = \{1, \ldots, \kappa\}$, each having a capacity of Q units, is available to serve the customers. Associated with every customer $i \in N_c$ is a demand d_i for each visit, and a predefined set of possible visit schedules P_i. A given schedule $p \in P_i$ consists of the specific days on which the customer should be visited, i.e., $p \subseteq T$. The problem is to select a visit schedule for each customer and design the vehicle routes in order to minimize the total routing cost. A feasible solution to the PVRP involves τ sets of vehicle routes that satisfy customer visit frequency and demand constraints while respecting the vehicle capacities.

We discuss alternative formulation approaches for the PVRP to compare against each other with respect to computational performance. Two of these formulations are already studied: the load-based formulation (Archetti et al., 2017) and the cut-based formulation (Rodriguez-Martin et al., 2019). In addition, we develop a formulation of the problem using vehicle flow variables and the well known MTZ-SECs (Miller et al., 1960). This formulation is referred as the MTZ-based formulation. All three formulations are extended to the PVRPTW as well. We also present some valid inequalities adapted from the literature to strengthen the proposed formulations.

3.1.1 MTZ-based Formulation

We use the following decision variables in the MTZ-based formulation; x_{ijt} is a binary variable which is equal to 1 when arc $(i,j) \in A$ is traversed on day $t \in T$, and 0 otherwise, s_{ip} is a binary variable which is equal to 1 if schedule $p \in P_i$ is chosen to visit customer $i \in N_c$, and 0 otherwise, y_{it} is binary variable which is equal to 1 if customer i is visited at time period t, and 0 otherwise, and continuous variable u_{it} which denotes load of the vehicle when it arrives at node $i \in N$ on day $t \in T$.

Accordingly, the PVRP is formulated with the model M1 as

$$\text{minimize} \quad \sum_{(i,j)\in A}\sum_{t\in T} c_{ij}x_{ijt}, \tag{3.1}$$

$$\text{subject to} \quad \sum_{p\in P_i} s_{ip} = 1 \qquad \forall i \in N_c, \tag{3.2}$$

$$\sum_{p\in P_i:t\in p} s_{ip} = y_{it} \qquad \forall i \in N_c, \forall t \in T, \tag{3.3}$$

$$x_{ijt} \le \frac{y_{it}+y_{jt}}{2} \qquad \forall i,j \in N_c, \forall t \in T, \tag{3.4}$$

$$\sum_{j\in N_c} x_{0jt} \le \kappa \qquad \forall t \in T, \tag{3.5}$$

$$\sum_{j\in N\setminus\{i\}} x_{ijt} = y_{it} \qquad \forall i \in N_c, \forall t \in T, \tag{3.6}$$

$$\sum_{j\in N\setminus\{i\}} x_{jit} = y_{it} \qquad \forall i \in N_c, \forall t \in T, \tag{3.7}$$

$$u_{jt} \le u_{it} - d_i y_{it} + Q(1-x_{ijt}) \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T, \tag{3.8}$$

$$d_i \le u_{it} \le Q \qquad \forall i \in N_c, \forall t \in T, \tag{3.9}$$

$$x_{ijt} \in \{0,1\} \qquad \forall (i,j) \in A, \forall t \in T, \tag{3.10}$$

$$s_{ip} \in \{0,1\} \qquad \forall i \in N_c, \forall p \in P_i, \tag{3.11}$$

$$y_{it} \in \{0,1\} \qquad \forall i \in N_c, \forall t \in T. \tag{3.12}$$

The objective function (3.1) minimizes the total routing cost. Constraints (3.2) ensure that an allowable visit schedule is selected for each customer. Constraints (3.3) relate the customer visit variables and the schedule selection variables. Constraints (3.4) guarantee that only those arcs connecting the customer pairs assigned to the same day are used. Constraints (3.5) impose that the number of vehicles that can be used on any day of the planning horizon cannot exceed the fleet size κ. Any customer to be served on a given day will be visited exactly once due to constraints (3.6) and (3.7), which, together, imply vehicle flow conservation. Subtours are prevented through constraints (3.8). Finally, vehicle capacity and demand satisfaction constraints are given by (3.9). Constraints (3.10)–(3.12) specify the domain restrictions on the variables.

The model M1 also allows the following family of valid inequalities inspired by the valid inequalities originally developed for the capacitated vehicle routing problem by Kara et al. (2004).

Inequalities I1. The inequalities

$$u_{jt} - u_{it} + Qx_{ijt} + (Q - d_j - d_i)x_{jit} \leq Q - d_i \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T \tag{3.13}$$

are valid for the MTZ-based formulation.

Proof. The inequalities

$$u_{jt} \leq u_{it} - d_i + Q(1 - x_{ijt}) \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T \tag{3.14}$$

is the logical interpretation of constraints (3.8). Rearranging constraints (3.14) yields

$$u_{jt} - u_{it} + Qx_{ijt} \leq Q - d_i \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T. \tag{3.15}$$

Adding an extra term $\alpha_{jit}x_{jit}$ to the left-hand side as,

$$u_{jt} - u_{it} + Qx_{ijt} + \alpha_{jit}x_{jit} \leq Q - d_i \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T, \tag{3.16}$$

we seek the largest value of α_{jit} such that (3.16) is valid. If $x_{jit} = 0$, then α_{jit} can take any value. If $x_{jit} = 1$, then $x_{ijt} = 0$ and we obtain

$$\alpha_{jit} \leq Q + (u_{it} - u_{jt}) - d_i \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T. \tag{3.17}$$

Constraints (3.15) for $x_{jit} = 1$ imply that

$$(u_{it} - u_{jt}) \leq -d_j \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T. \tag{3.18}$$

Then, we obtain (3.19) from (3.17) and (3.18)

$$\alpha_{jit} \leq Q + (u_{it} - u_{jt}) - d_i \leq Q - d_j - d_i \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T. \tag{3.19}$$

Finally, by substituting the largest value of α_{jit} in (3.16), we get (3.13). □

Inequalities I2. **Symmetry-Breaking of Routes:** By exploiting the fact that arc costs are symmetric, each route is associated with a cost regardless of its orientation. we impose

$$x_{i0t} \leq \sum_{r \leq i} x_{0rt} \qquad \forall i \in N_c, \forall t \in T \tag{3.20}$$

to break the symmetry of the routes and choose the orientation that starts with the customer with the lowest index among two possible orientations for a route. The inequalities (3.20) are optimality cuts, but not valid inequalities.

Inequalities I3. **Valid inequalities:** The relation between variables x and y is imposed by constraints (3.4)

$$x_{ijt} \leq \frac{y_{it}+y_{jt}}{2} \qquad \forall i \in N_c, \forall j \in N_c, \forall t \in T,$$

which state that no arc (i,j) can be used between two customers i and j on a particular day t unless they are both scheduled for delivery on day t. Taking into account that no arc will be traversed in both directions in the same time period, the inequalities

$$x_{ijt} + x_{jit} \leq \frac{y_{it}+y_{jt}}{2} \qquad \forall i < j \in N_c, \forall t \in T, \tag{3.21}$$

are valid for MTZ-based formulation.

3.1.2 Load-based Formulation

The load-based formulation proposed by Archetti et al. (2017) includes, in addition to the previously defined x, y, and s variables used in the MTZ-based formulation, a set of continuous (commodity flow) variables l_{ijt} representing the load of the vehicle when traversing arc $(i,j) \in A$ in time period $t \in T$, and a set of integer variables v_t corresponding to the number of vehicles used in time period $t \in T$. The model L1 corresponding to the load-based formulation of the PVRP is as follows:

$$\text{(3.1)} \qquad \text{minimize} \quad \sum_{(i,j)\in A}\sum_{t\in T} c_{ij}x_{ijt},$$

$$\text{subject to} \quad (3.2)-(3.6),(3.10)-(3.12),$$

$$\text{(3.22)} \qquad \sum_{j\in N\setminus\{i\}} x_{ijt} = \sum_{j\in N\setminus\{i\}} x_{jit} \qquad \forall i\in N,\ \forall t\in T,$$

$$\text{(3.23)} \qquad \sum_{j\in N\setminus\{i\}} l_{ijt} - \sum_{j\in N\setminus\{i\}} l_{jit} = \begin{cases} -d_i y_{it}, & i\in N_c \\ \sum_{j\in N_c} d_j y_{jt}, & i=0 \end{cases} \qquad \forall t\in T,$$

$$\text{(3.24)} \qquad \sum_{i\in N_c} d_i y_{it} \le Q v_t \qquad \forall t\in T,$$

$$\text{(3.25)} \qquad l_{ijt} \le Q x_{ijt} \qquad \forall (i,j)\in A,\ \forall t\in T,$$

$$\text{(3.26)} \qquad l_{ijt} \ge 0 \qquad \forall (i,j)\in A,\ \forall t\in T,$$

$$\text{(3.27)} \qquad v_t \in \mathbb{Z}_+ \qquad \forall t\in T.$$

The objective function and constraints (3.1) to (3.6) and constraints (3.10) to (3.12) are already discussed for M1. The vehicle flow and the commodity flow conservation constraints are given by (3.22) and (3.23), respectively. Constraints (3.24) guarantee that the total amount delivered within a given period does not exceed the total capacity of the vehicles used in that time period. Constraints (3.25) link the vehicle flow and the commodity flow variables. Constraints (3.26) and (3.27) define the domains of the new variables.

Equality I4. **Sum of final loads:** All vehicles come back to the depot without carrying any load. This equality is not valid because there exist valid solutions to the problem that do not adhere to this restriction. Instead, it is an optimality cut, meaning that at least one optimal solution meets this condition. It can be utilized to tighten the solution space.

$$\text{(3.28)} \qquad \sum_{t\in T}\sum_{j\in N_c} l_{j0t} = 0$$

Equality (3.28) are exploited in order to strengthen the Load-based formulation. Optimality cuts (3.20) and valid inequalities (3.21) also can be considered for the Load-based formulation. Valid inequalities (3.21) and optimality cuts (3.20) and (3.28) are borrowed from Archetti et al. (2017).

3.1.3 Directed Cut-based Formulation

The cut-based formulation presented here is the directed version of the model proposed by Rodriguez-Martin et al. (2019) on an undirected graph, in which driver consistency restrictions are relaxed and customers have arbitrary demands (different than the unit demand assumption made by the authors). The newly defined and modified variables used in the cut-based formulation are as follows; binary variable x_{ijtk} which is equal to 1 if vehicle $k \in K$ traverses arc $(i,j) \in A$ on day $t \in T$, and 0 otherwise, binary variable z_{itk} which is equal to 1 if vehicle $k \in K$ visits customer $i \in N_c$ on day $t \in T$, and 0 otherwise, and also binary variable v_{tk} which takes value 1 if vehicle $k \in K$ is used on day $t \in T$, and 0 otherwise.

For any subset $S \subseteq N$, we define $\delta^+(S)$ as the set of outgoing arcs, i.e., arcs (i,j) with $i \in S$ and $j \in N \setminus S$, and $\delta^-(S)$ as the set of incoming arcs, i.e., arcs (i,j) with $i \in N \setminus S$ and $j \in S$. The model DC1 corresponding to the directed cut-based formulation is:

$$\text{minimize} \quad \sum_{(i,j)\in A} \sum_{t\in T} \sum_{k\in K} c_{ij} x_{ijtk}, \tag{3.29}$$

$$\text{subject to} \quad (3.2), (3.11),$$

$$\sum_{p\in P_i : t\in P} s_{ip} = \sum_{k\in K} z_{itk} \qquad \forall i \in N_c, \forall t \in T, \tag{3.30}$$

$$\sum_{i\in N_c} z_{itk}\, d_i \leq Q v_{tk} \qquad \forall k \in K, \forall t \in T, \tag{3.31}$$

$$\sum_{j\in N\setminus\{i\}} x_{ijtk} = z_{itk} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T, \tag{3.32}$$

$$\sum_{j\in N\setminus\{i\}} x_{jitk} = z_{itk} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T, \tag{3.33}$$

$$\sum_{(h,m)\in\delta^+(S)} x_{hmtk} \geq z_{itk} \qquad \forall S \subseteq N_c, \forall i \in S, \forall k \in K, \forall t \in T, \tag{3.34}$$

$$x_{ijtk} \in \{0,1\} \qquad \forall i,j \in N, \forall k \in K, \forall t \in T, \tag{3.35}$$

$$z_{itk} \in \{0,1\} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T, \tag{3.36}$$

$$v_{tk} \in \{0,1\} \qquad \forall k \in K, \forall t \in T. \tag{3.37}$$

The objective function (3.29) minimizes the total routing cost. Since $\sum_{k\in K} z_{itk} = y_{it}$ for all $i \in N_c$ and $t \in T$, constraints (3.30) and (3.32)–(3.33) are equivalent to (3.3) and (3.6)–(3.7), respectively. Constraints (3.31) are the vehicle capacity restrictions. Constraints (3.34) prevent subtours. Variable domain restrictions are given by (3.35)–(3.37).

3.1.4 Undirected Cut-based Formulation

The periodic vehicle routing problem (PVRP) can be defined on a complete undirected graph $G = (N, E)$ with $N = \{0, 1, \dots, n\}$ being the set of nodes, where node 0 represents the depot and the nodes in the set $N_c = \{1, \dots, n\}$ correspond to the customers. The set of edges is given by $E = \{e \subset N : |e| = 2\}$. Each edge e is associated with a non-negative cost c_e. Integer variable x_{etk} may take values $\{0,1\}$ $\forall \{i,j\} \in E \backslash \{\{0,j\} : j \in N_c\}$ and values $\{0,1,2\}$ $\forall \{0,j\}, j \in N_c$. For any subset $S \subseteq N$, let $\delta(S) = \{e \in E : |S \cap e| = 1\}$. If $S = \{i\}$, we write $\delta(i)$ instead of $\delta(\{i\})$. In addition, for a given subset of edges $E' \subseteq E$, we define $x_{tk}(E') = \sum_{e \in E'} x_{etk}$. The model UC1 corresponding to the undirected cut-based formulation is:

(3.38)

minimize $\displaystyle\sum_{e \in E} \sum_{t \in T} \sum_{k \in K} c_e x_{etk},$

subject to (3.2), (3.30), (3.31), (3.11), (3.36), (3.37),

(3.39)	$x_{tk}(\delta(i)) = 2z_{itk}$	$\forall i \in N, \forall k \in K, \forall t \in T,$
(3.40)	$x_{tk}(\delta(S)) \geq 2z_{itk}$	$\forall S \subseteq N_c, \forall i \in S, \forall k \in K, \forall t \in T,$
(3.41)	$x_{etk} \in \{0,1\}$	$\forall e \in E \backslash \{\{0,j\} : j \in N_c\}, \forall k \in K,$ $\forall t \in T,$
(3.42)	$x_{0jtk} \in \{0,1,2\}$	$\forall \{0,j\}, j \in N_c, \forall k \in K, \forall t \in T.$

The objective function (3.38) minimizes the total routing cost. Constraints (3.39) are the degree constraints for the depot and the customers. Constraints (3.40) prevent subtours. Variable domain restrictions are given by (3.41)–(3.42).

3.1.5 Extension to the PVRPTW

The PVRPTW is the extension of the PVRP in which each customer i must be served within a specified time interval $[E_i, L_i]$, called a time window. The vehicle must stop at the customer location for r_i units of time for service. In the case of early arrival at a customer, it must wait until E_i when the time window opens.

3.1.5.1 *MTZ-based PVRPTW Formulation*

Let w_{it} be the start time of serving customer i on day t and M be a large constant. We extend the MTZ-based PVRP model M1 to obtain a model M2 for the PVRPTW as follows:

$$
\begin{aligned}
&\text{minimize} && (3.1), && \\
&\text{subject to} && (3.2)-(3.12), && \\
(3.43)\quad & && w_{it}+r_i+c_{ij}-w_{jt} \leq (1-x_{ijt})M && \forall i \in N, \forall j \in N_c, \forall t \in T, \\
(3.44)\quad & && w_{it}+r_i+c_{i0}-L_0 \leq (1-x_{i0t})M && \forall i \in N_c, \forall t \in T, \\
(3.45)\quad & && E_i \leq w_{it} \leq L_i && \forall i \in N, \forall t \in T.
\end{aligned}
$$

Constraints (3.43)–(3.45) guarantee the feasibility of the tours with respect to time window restrictions. The large constant M may be equal to $\max_{(i,j)\in A}\{L_i + r_i + c_{ij} - E_j\}$.

3.1.5.2 *Load-based PVRPTW Formulation*

Using the new variables, w_{it} as defined in the MTZ-based PVRPTW model and the corresponding set of constraints (3.43)–(3.45), we extend the load-based PVRP model L1 to the model L2 to account for time window constraints as follows:

$$
\begin{aligned}
&\text{minimize} && (3.1), \\
&\text{subject to} && (3.2)-(3.6), (3.10)-(3.12), (3.22)-(3.27), (3.43)-(3.45).
\end{aligned}
$$

For both M2 and L2, we adapt the following valid inequalities, well-known in the literature without the multi-period planning horizon aspect, for the PVRPTW without any proof.

Inequalities I5. The constraints

$$E_0 + c_{0i} \leq w_{it} \leq L_0 - c_{i0} \qquad \forall i \in N_c, \forall t \in T, \tag{3.46}$$

are valid inequalities for models M2 and L2. It states that the initiation of service at node $i \in N_c$ is within the range of the earliest possible departure time from the depot plus the time it takes to travel from the depot to node i, and the latest allowable arrival time of a vehicle at the depot minus the time it takes to travel from node i to the depot.

3.1.5.3 *Cut-based PVRPTW Formulation*

We now define w_{itk} as the service start time of customer i with vehicle k on day t. We extend the directed cut-based PVRP model DC1 to the model C2 for the PVRPTW formulation as follows:

$$\begin{aligned}
&\text{minimize} && (3.29), && \\
&\text{subject to} && (3.2), (3.11), (3.30) - (3.37), && \\
&(3.47) && w_{itk} + r_i + c_{ij} - w_{jtk} \leq (1 - x_{ijtk})M && \forall i \in N, \forall j \in N_c, \forall k \in K, \forall t \in T, \\
&(3.48) && w_{itk} + r_i + c_{i0} - L_0 \leq (1 - x_{i0tk})M && \forall i \in N_c, \forall k \in K, \forall t \in T, \\
&(3.49) && E_i \sum_{j \in N} x_{ijkt} \leq w_{itk} \leq L_i \sum_{j \in N} x_{ijkt} && \forall i \in N, \forall k \in K, \forall t \in T.
\end{aligned}$$

Constraints (3.47) to (3.49) guarantee the feasibility of the tours with respect to time window restrictions.

3.2 Computational Experiments

We investigate the computational performance of the alternative formulations for both the PVRP and the PVRPTW as well as the contribution of the proposed valid inequalities. Our first set of experiments focuses on the comparison among the MTZ-based PVRP model (M1) and the load-based PVRP model (L1), in the presence and absence of valid inequalities, and the undirected cut-based PVRP model (UC1). In the second set of experiments, we compare the MTZ-based PVRPTW formulation (M2) and the load-based PVRPTW formulation (L2), in the presence and absence of the presented valid inequalities, and the directed cut-based PVRPTW formulation (C2). All computational experiments are carried out on a virtual machine with Intel Xeon CPU E5-2640 processor with 2.60 GHz speed, 16 GB RAM, and 64-bit, using Gurobi Optimizer 8.1.1 as the commercial solver with Python 3.7.4.

3.2.1 Benchmark Instances

One of the contributions of this study is to bring together all benchmark problem instances in the literature; we have identified eight sets: five for the PVRP and three for the PVRPTW. Their features are summarized below.

First data set (S1) The PVRP instances in S1 consists of 32 instances with up to 417 customers and $\tau \in \{2,4,5,6,7,8,10\}$; it is composed of instances originally proposed by Christofides and Beasley (1984), Russell and Igo (1979), Russell and Gribbin (1991), and Chao et al. (1995). Visit schedules of customers are symmetric, i.e., allowable visit schedules for the customers with the same visit frequency are the same. In some instances, the spatial distribution of nodes is random, and for the rest, it is symmetric with respect to the origin.

Cordeau (1997) data set (S2) The PVRP instances in S2 consists of ten instances with up to 288 customers and $\tau \in \{4,6\}$; it is composed of randomly generated instances by Cordeau et al. (1997). Visit schedules of customers are also symmetric.

Archetti (2017) data set (S3) The PVRP data set S3 is provided by Archetti et al. (2017) which consists of 25 instances with $\tau = 5$ and $|N_c| \in \{7,9,11,15,49\}$. They generated the instances in a similar way to those used by Francis et al. (2006). In these instances visit schedules of customers are also symmetric.

Archetti (2017) data set (S4) This set consists of 35 PVRP instances with clustered customers in which $\tau = 5$ and $|N_c| \in \{10,15,20\}$. Other parameters are: the number of clusters p, a radius $|r| \in \{0.15,0.30,0.50\}$ which determines the coverage area of each cluster, and β which is used in determining the minimum distance $\beta \times r$ among the centers of the clusters. For $|N_c| = 10$, the number of clusters has been set to $p = 2$, whereas for instances with $|N_c| \in 15,20$ has been set to $p = 3$. To avoid clusters overlapping the value of β has been set to 2 when $|r| \in \{0.15, 0.30\}$. For the instances with $|r| = 0.50$, β has been set to 1 to allow that a customer can belong to more than one cluster.

Rodriguez-Martin (2019) data set (S5) The PVRP instances in S5 are proposed by Rodriguez-Martin et al. (2019), originally for the PVRP with driver consistency. The data set consists of 80 combinations of $n \in N_c$, $k \in K$, and $t \in T$ where $|N_c| \in \{10, 20, ..., 70\}$, $\kappa \in \{2, 3, 4\}$ and $\tau \in \{2, 3, 4, 5\}$. There are three instances for each combination which results in 240 instances. The spatial distribution and visit schedules of nodes are randomly generated. The nodes with the same visit frequency have various options of visit schedules. Due to the violation of the capacity constraints, some instances had to be modified by increasing the vehicle capacity.

Cordeau (2001) data set (S6) The PVRPTW instances in S6 are proposed by Cordeau et al. (2001). This set of 20 instances was generated by adding time windows of different widths to ten basic PVRP instances introduced by Cordeau et al. (1997). The basic instances were created randomly by clustering the customers around a given number of seed points and the visit schedules of customers are symmetric. The new set consists of two groups of ten instances with $|N| \in \{48, 72, 96, 144, 192, 216, 240, 288\}$ and $\tau \in \{4, 6\}$. The first group has tight time windows which are created by choosing the uniform random numbers E_i and L_i respectively in the intervals $[60, 480]$ and $[E_i + 90, E_i + 180]$. For the second group, wide time windows are generated by using the intervals $[60, 300]$ and $[E_i + 180, E_i + 360]$ respectively for E_i and L_i. The depot is set to have a time window of $[0, 1000]$.

Pirkwieser (2009) data set (S7) The PVRPTW instances in S7 are proposed by Pirkwieser and Raidl (2009a). They derive 45 PVRPTW instances from the Solomon VRPTW data set by evenly assigning the possible visit combinations to the customers at random. They use the first five instances of each of types with the tight time windows, of random (R), clustered (C), and mixed random and clustered (RC). They use three planning horizons: four (P4), six (P6), and eight (P8) days. The distribution of nodes are identical for all instances within one type (i.e., R, C and RC). The instances differ with respect to the time window density (i.e., the percentage of customers with time windows).

Vidal (2013) data set (S8) PVRPTW instances in S8 are introduced by Vidal et al. (2013). S8 consists of 28 instances involving 360 to 960 customers with $\tau \in \{4, 6, 12\}$. Due to the size of instances, they can only be attempted with heuristic methods; we do not conduct any computational experiments on S8.

3.2.2 Evaluation of Alternative PVRP Formulations

Data sets S1, S2, S3, S4 and S5 are used for the comparison among the MTZ-based PVRP model (M1), the load-based PVRP model (L1), and the cut-based PVRP model (UC1). Each instance in S1 and S2 is solved with M1, UC1, and L1 within a time limit of 14400 seconds, while for the less complex instances in S3, S4, and S5, a time limit of 7200 seconds is imposed. We also include a variant of M1 by replacing constraints (3.8) in M1 with valid inequalities (3.13); the resulting model is called M1+(3.13). Then, we add constraints (3.20) and (3.21) to M1+(3.13), the resulting model is called M1+(3.13, 3.20, 3.21). Model L1+(3.20, 3.21, 3.28) is obtained by adding valid inequalities (3.21) and optimality cuts (3.20) and (3.28) to model L1.

The results for different data sets are shown in Tables 3.1, 3.2, 3.3, 3.5 and 3.7 where the first column shows the instance name. The second one is the best objective function value obtained by the alternative models (Best OFV). The next six blocks of five columns each correspond to models M1, L1, UC1, M1+(3.13), M1+(3.13, 3.20, 3.21) and L1+(3.20, 3.21, 3.28), respectively. In these tables, for a given model, OFV is the objective function value of the best solution obtained, $\Delta\%$ shows the percentage deviation of OFV from the Best OFV, Time reports the computing time (in seconds), Root is the objective function value obtained at the root node of the branch-and-bound tree, i.e., the root relaxation bound and LP is the LP-relaxation objective function value obtained by lifting the integrality requirement on the variables. “NA” means that no feasible solution could be identified by the model within the given time limit for the instances in the corresponding rows. “T-UP” shows that the time limit is reached before a solution is certified as optimal. When the objective function value is typed in bold, it indicates the best (i.e., the lowest) OFV for the instance.

If the same OFV is obtained by more than one model, only the one that is found within a shorter amount of time is typed in bold. In the case that the same OFV is obtained by more than one model and all models hit the time limit, all those OFVs are typed in bold.

Table 3.1 reports the results with the alternative PVRP models for S1. It should be noted that instances p06-N75p10m1 and p09-N100p8m1 respectively have a planning horizon of ten and eight days and instance p13-N417p7m9 consists of 417 customers. Due to the size of these instances, we do not conduct any computational experiments on them and Table 3.1 shows the results for the remaining 29 instances in date set S1. According to Table 3.1, L1 and UC1 are able to find a feasible solu-

Table 3.1 Results of the alternative PVRP models on data set S1

Instance	Best OFV	M1					L1					UC1				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
p01	529.34	597.65	12.90	T-UP	379.32	379.32	**529.34**	0.00	T-UP	473.60	473.60	584.93	10.50	T-UP	422.18	422.18
p02	1356.68	1739.10	28.19	T-UP	881.64	881.64	NA	_	T-UP	1105.03	1105.03	1763.07	29.95	T-UP	995.72	995.72
p03	567.83	NA	_	T-UP	379.32	379.32	722.01	27.15	T-UP	473.60	473.60	**567.83**	0.00	T-UP	422.18	422.18
p04	993.56	NA	_	T-UP	485.49	485.49	NA	_	T-UP	714.07	714.07	NA	_	T-UP	587.70	587.70
p05	2386.74	2623.32	9.91	T-UP	1129.53	1129.53	NA	_	T-UP	1719.67	1719.67	NA	_	T-UP	1355.90	1355.90
p07	862.31	1328.65	54.08	T-UP	579.86	579.86	**862.31**	0.00	T-UP	736.28	736.28	NA	_	T-UP	654.91	654.91
p08	2372.56	2674.38	12.72	T-UP	1305.36	1305.36	2689.80	13.37	T-UP	1776.07	1776.07	NA	_	T-UP	1547.62	1547.62
p10	2143.79	NA	_	T-UP	1071.50	1071.50	NA	_	T-UP	1421.17	1421.17	2567.88	19.78	T-UP	1234.37	1234.37
p11	969.16	1030.45	6.32	T-UP	369.79	369.79	1042.07	7.52	T-UP	580.82	580.82	NA	_	T-UP	421.64	421.64
p12	1666.23	NA	_	T-UP	425.41	425.41	**1666.23**	0.00	T-UP	1011.44	1011.44	NA	_	T-UP	692.43	692.43
p14	954.81	954.81	0.00	T-UP	687.12	687.12	954.81	0.00	2715	755.50	755.50	**954.81**	0.00	452.31	736.86	736.86
p15	1862.63	1920.18	3.09	T-UP	1507.32	1507.32	1862.63	0.00	T-UP	1634.06	1634.05	1862.63	0.00	T-UP	1576.77	1576.77
p16	2875.24	2994.67	4.15	T-UP	2401.56	2401.56	2875.24	0.00	T-UP	2580.16	2580.16	3033.16	5.49	T-UP	2475.22	2475.22
p17	1660.75	1717.33	3.41	T-UP	958.24	958.24	1686.18	1.53	T-UP	1172.98	1172.98	1883.88	13.44	T-UP	1066.60	1066.60
p18	3362.94	3439.34	2.27	T-UP	2155.36	2155.36	**3362.94**	0.00	T-UP	2485.30	2485.30	3754.15	11.63	T-UP	2275.76	2275.76
p19	4947.97	5804.68	17.31	T-UP	3698.88	3698.88	NA	_	T-UP	4147.58	4147.58	NA	_	T-UP	3825.30	3825.30
p20	11616.36	**11616.36**	0.00	T-UP	6966.24	6966.24	NA	_	T-UP	7634.93	7634.93	NA	_	T-UP	7098.68	7098.68
p21	2239.16	2432.83	8.65	T-UP	1140.96	1140.96	**2239.16**	0.00	T-UP	1610.14	1610.14	NA	_	T-UP	1320.96	1320.96
p22	4720.18	5249.89	11.22	T-UP	2549.28	2549.28	NA	_	T-UP	3226.86	3226.86	NA	_	T-UP	2749.28	2749.28
p23	8894.41	**8894.41**	0.00	T-UP	4349.76	4349.76	NA	_	T-UP	5212.10	5212.10	NA	_	T-UP	4559.76	4559.76
p24	3772.96	3841.08	1.81	T-UP	2500.02	2500.02	3865.32	2.45	T-UP	3350.82	3350.82	**3772.96**	0.00	T-UP	3350.25	3350.25
p25	3865.32	4067.12	5.22	T-UP	2500.02	2500.02	**3865.32**	0.00	T-UP	3412.42	3412.42	NA	_	T-UP	3410.71	3410.71
p26	3854.01	**3854.01**	0.00	T-UP	2500.02	2500.02	NA	_	T-UP	3473.45	3473.45	4446.28	15.37	T-UP	3471.17	3471.17
p27	24986.53	25336.74	1.40	T-UP	15000.00	15000.00	NA	_	T-UP	18910.48	18910.48	NA	_	T-UP	16765.71	16765.71
p28	24586.47	**24586.47**	0.00	T-UP	15000.00	15000.00	NA	_	T-UP	19252.51	19252.51	NA	_	T-UP	16888.34	16888.34
p29	26268.73	**26268.73**	0.00	T-UP	15000.00	15000.00	NA	_	T-UP	19653.55	19653.56	NA	_	T-UP	17022.30	17022.30
p30	89597.65	93319.61	4.15	T-UP	51001.98	51001.98	103091.46	15.06	T-UP	63082.68	63082.69	NA	_	T-UP	53885.58	53885.58
p31	89208.36	92643.30	3.85	T-UP	51001.98	51001.98	NA	_	T-UP	64208.87	64208.87	NA	_	T-UP	54126.34	54126.34
p32	91636.00	97286.12	6.17	T-UP	51001.98	51001.98	NA	_	T-UP	65684.02	65684.02	NA	_	T-UP	54374.41	54374.41

Table 3.1 Results of the alternative PVRP models on data set S1 (continued)

Instance	Best OFV	M1+(3.13)					M1+(3.13, 3.20, 3.21)					L1+(3.20, 3.21, 3.28)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
p01	529.34	571.02	7.87	T-UP	379.36	379.36	549.00	3.71	T-UP	414.98	414.98	531.02	0.32	T-UP	491.52	491.52
p02	1356.68	1492.82	10.03	T-UP	890.03	890.03	1471.17	8.44	T-UP	996.91	917.57	**1356.68**	0.00	T-UP	1186.02	1122.49
p03	567.83	NA	_	T-UP	379.36	379.36	NA	_	T-UP	414.98	414.98	615.22	8.35	T-UP	491.52	491.52
p04	993.56	**993.56**	0.00	T-UP	485.53	485.53	NA	_	T-UP	536.53	536.53	NA	_	T-UP	740.07	740.07
p05	2386.74	**2386.74**	0.00	T-UP	1142.33	1142.33	2570.17	7.69	T-UP	1262.60	1194.61	NA	_	T-UP	1779.54	1749.83
p07	862.31	997.52	15.68	T-UP	579.90	579.90	982.16	13.90	T-UP	625.82	625.82	905.03	4.95	T-UP	762.08	762.08
p08	2372.56	2667.89	12.45	T-UP	1310.69	1310.69	2596.71	9.45	T-UP	1433.64	1350.91	**2372.56**	0.00	T-UP	1840.66	1789.31
p10	2143.79	2227.31	3.90	T-UP	1072.77	1072.77	2331.41	8.75	T-UP	1120.09	1120.09	**2143.79**	0.00	T-UP	1439.35	1439.35
p11	969.16	1072.39	10.65	T-UP	374.60	374.60	**969.16**	0.00	T-UP	435.47	403.43	1062.67	9.65	T-UP	627.98	594.90
p12	1666.23	2111.87	26.75	T-UP	425.42	425.42	2009.17	20.58	T-UP	473.30	473.30	NA	_	T-UP	1017.42	1017.42
p14	954.81	954.81	0.00	T-UP	687.17	687.17	959.09	0.45	T-UP	762.56	703.72	954.81	0.00	1131.55	824.44	755.50
p15	1862.63	1864.56	0.10	T-UP	1507.37	1507.36	1884.52	1.18	T-UP	1604.08	1557.84	**1862.63**	0.00	T-UP	1711.55	1660.73
p16	2875.24	3030.15	5.39	T-UP	2423.59	2423.59	2879.52	0.15	T-UP	2585.73	2484.17	**2875.24**	0.00	T-UP	2725.24	2628.65
p17	1660.75	1683.67	1.38	T-UP	958.34	958.34	1716.40	3.35	T-UP	1153.73	1038.29	**1660.75**	0.00	T-UP	1339.07	1193.23
p18	3362.94	3550.72	5.58	T-UP	2155.57	2155.57	3436.80	2.20	T-UP	2312.74	2202.53	NA	_	T-UP	2597.02	2485.44
p19	4947.97	6061.36	22.50	T-UP	3699.10	3699.10	5602.73	13.23	T-UP	3804.11	3732.27	**4947.97**	0.00	T-UP	4203.52	4150.94
p20	11616.36	13708.61	18.01	T-UP	6997.27	6997.27	13867.90	19.38	T-UP	7303.51	7089.76	NA	_	T-UP	7887.11	7704.29
p21	2239.16	2493.91	11.38	T-UP	1140.96	1140.96	2325.10	3.84	T-UP	1496.63	1338.51	2242.28	0.14	T-UP	1829.25	1650.84
p22	4720.18	5583.01	18.28	T-UP	2549.28	2549.28	4917.75	4.19	T-UP	2887.88	2705.22	**4720.18**	0.00	T-UP	3437.99	3246.25
p23	8894.41	9031.39	1.54	T-UP	4349.86	4349.86	9408.53	5.78	T-UP	4624.09	4468.68	NA	_	T-UP	5370.85	5212.45
p24	3772.96	3796.03	0.61	T-UP	2500.02	2500.02	3785.89	0.34	T-UP	2696.10	2696.10	3823.39	1.34	T-UP	3489.70	3350.82
p25	3865.32	4087.05	5.74	T-UP	2500.02	2500.02	4110.11	6.33	T-UP	2696.10	2696.10	3865.41	0.00	T-UP	3550.16	3412.43
p26	3854.01	3969.50	3.00	T-UP	2500.02	2500.02	3867.50	0.35	T-UP	2696.10	2696.10	3865.41	0.30	T-UP	3610.63	3473.47
p27	24986.53	**24986.53**	0.00	T-UP	15000.00	15000.00	25595.88	2.44	T-UP	16176.61	16176.61	26903.50	7.67	T-UP	19676.25	18910.43
p28	24586.47	25967.67	5.62	T-UP	15000.00	15000.00	25989.52	5.71	T-UP	16176.61	16176.61	27706.11	12.69	T-UP	20000.21	19252.46
p29	26268.73	26306.24	0.14	T-UP	15000.00	15000.00	27147.83	3.35	T-UP	16176.61	16176.61	NA	_	T-UP	20375.21	19653.51
p30	89597.65	**89597.65**	0.00	T-UP	51001.98	51001.98	91480.02	2.10	T-UP	55385.94	55385.94	NA	_	T-UP	0	63147.63
p31	89208.36	**89208.36**	0.00	T-UP	51001.98	51001.98	93363.48	4.66	T-UP	55385.94	55385.94	NA	_	T-UP	0	64278.56
p32	91636.00	**91636.00**	0.00	T-UP	51001.98	51001.98	97574.97	6.48	T-UP	55385.94	55385.94	NA	_	T-UP	0	65749.04

Table 3.2 Results of the alternative PVRP models on data set S2

Instance	Best OFV	M1					L1					UC1				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
pr01-N48m2p4	2187.98	2413.49	10.31	T-UP	1451.45	1451.45	2200.36	0.57	T-UP	1628.18	1628.18	2189.27	0.06	T-UP	1529.24	1529.24
pr02-N96m4p4	4120.58	4336.09	5.23	T-UP	2255.79	2255.79	4298.07	4.31	T-UP	2770.16	2770.16	NA	_	T-UP	2347.14	2347.14
pr03-N144m6p4	6165.23	6230.21	1.05	T-UP	3032.08	3032.08	**6165.23**	0.00	T-UP	4083.99	4083.99	NA	_	T-UP	3389.02	3389.02
pr04-N192m8p4	7661.55	7688.84	0.36	T-UP	3377.91	3377.91	NA	_	T-UP	4693.53	4693.53	NA	_	T-UP	3836.42	3836.42
pr05-N240m10p4	NA	NA	_	T-UP	3525.92	3525.92	NA	_	T-UP	5340.67	5340.67	NA	_	T-UP	4065.58	4065.58
pr06-N288m12p4	10451.02	**10451.02**	0.00	T-UP	4275.34	4275.34	NA	_	T-UP	6468.34	6468.34	NA	_	T-UP	5147.57	5147.57
pr07-N72m3p6	5051.59	5750.27	13.83	T-UP	3398.81	3398.81	5423.38	7.36	T-UP	3884.69	3884.69	NA	_	T-UP	3504.43	3504.43
pr08-N144m6p6	9976.73	NA	_	T-UP	3842.02	3842.02	NA	_	T-UP	5467.25	5467.25	NA	_	T-UP	4416.01	4416.01
pr09-N216m9p6	13646.28	13755.84	0.80	T-UP	5502.17	5502.17	NA	_	T-UP	8315.47	8315.47	NA	_	T-UP	6764.80	6764.80
pr10-N288m12p6	NA	NA	_	T-UP	6069.61	6069.61	NA	_	T-UP	9965.74	9965.74	NA	_	T-UP	7583.18	7583.18

Table 3.2 Results of the alternative PVRP models on data set S2 (continued)

Instance	Best OFV	M1+(3.13)					M1+(3.13, 3.20, 3.21)					L1+(3.20, 3.21, 3.28)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
pr01-N48m2p4	2187.98	2448.74	11.92	T-UP	1544.35	1544.35	2480.00	13.35	T-UP	1729.31	1634.07	**2187.98**	0.00	T-UP	1828.41	1771.68
pr02-N96m4p4	4120.58	4189.39	1.67	T-UP	2707.14	2707.14	4261.60	3.42	T-UP	2958.04	2787.77	**4120.58**	0.00	T-UP	3350.05	3224.29
pr03-N144m6p4	6165.23	6890.63	11.77	T-UP	3149.51	3149.51	6520.53	5.76	T-UP	3654.97	3360.97	6386.52	3.59	T-UP	4442.56	4247.93
pr04-N192m8p4	7661.55	**7661.55**	0.00	T-UP	3636.17	3636.17	8500.16	10.95	T-UP	4120.77	3766.05	NA	_	T-UP	5203.34	4947.27
pr05-N240m10p4	NA	NA	_	T-UP	3602.33	3602.33	NA	_	T-UP	4154.72	3789.55	NA	_	T-UP	5670.61	5446.26
pr06-N288m12p4	10451.02	10866.57	3.98	T-UP	4447.53	4447.53	10607.87	1.50	T-UP	5025.33	4631.96	NA	_	T-UP	6908.12	6647.70
pr07-N72m3p6	5051.59	5580.80	10.48	T-UP	3440.25	3440.25	6434.33	27.37	T-UP	3837.12	3527.18	**5051.59**	0.00	T-UP	4236.39	3980.95
pr08-N144m6p6	9976.73	**9976.73**	0.00	T-UP	3919.68	3919.68	NA	_	T-UP	4528.91	4111.79	NA	_	T-UP	5927.56	5651.66
pr09-N216m9p6	13646.28	13827.72	1.33	T-UP	5687.37	5687.37	**13646.28**	0.00	T-UP	6440.76	5953.24	NA	_	T-UP	8743.78	8501.65
pr10-N288m12p6	NA	NA	_	T-UP	6283.45	6283.45	NA	_	T-UP	7158.28	6671.55	NA	_	T-UP	10511.60	10234.41

tion respectively for 15 and 11 out of 29 instances. This number increases to 25, 28 and 27 for M1, M1+(3.13) and M1+(3.13, 3.20, 3.21), respectively.

- For two out of four instances (p03-N50p5m1 and p12-N163p5m3) for which a feasible solution cannot be found by M1, a feasible solution is found by L1 within the time limit. On the other hand, M1 attains a feasible solution for 12 instances that remained unsolvable by L1. Both M1 and L1 return no feasible solution for two instances, namely p04-N75p2m5 and p10-N100p5m4. Among the 13 instances solved by M1 and L1, M1 beats L1 in four instances, ties in one and loses in eight with respect to solution quality.

- The only instance solved to optimality by L1 and UC1 is p14-N20p4m2, respectively in 2715 and 452.31 seconds. Although M1 and M1+(3.13) reach the same OFV for this instance, they cannot prove its optimality within the specified time limit. UC1 is not able to identify any feasible solution for 18 instances while it finds the best solution for four instances. UC1 finds a feasible solution for p03-N50p5m1 and p10-N100p5m4 which remained unsolvable by M1.

- M1+(3.13) is capable of improving the solution quality in 12 instances compared to M1 by 6.14% on average, and six instances overall. Compared to M1, M1+(3.13) does not produce better solutions for 12 instances; however, the deviation of OFVs has an average of 4.21%. M1+(3.13) also returns a feasible solution for three out of four instances for which M1 does not return any feasible solution within the time limit. Hence, M1+(3.13), which uses the stronger version of the SECs, seems more effective and robust than M1.

- M1+(3.13, 3.20, 3.21) is capable of improving the solution quality in 13 instances compared to M1 , and one instance overall. M1+(3.13, 3.20, 3.21) also returns a feasible solution for two out of four instances for which M1 does not return any feasible solution within the time limit.

- L1+(3.20, 3.21, 3.28) is capable of improving the solution quality in 4 instances compared to L1 by 7.29% on average, and nine instances overall. L1+(3.20, 3.21, 3.28) also returns a feasible solution for seven instances which remained unsolvable by L1.

- According to the results on data set S1, in terms of the quality of the root relaxation bounds, models L1 and L1+(3.20, 3.21, 3.28) yield significantly better objective function values at the root node than other models.

Table 3.2 shows the results for data set S2. Accordingly, L1 and UC1 are able to find a feasible solution respectively for four and one out of ten instances. This number increases to seven and eight for M1 and M1+(3.13), respectively. M1 attains a feasible solution for three instances that remained unsolvable by L1. Both M1 and L1 return no feasible solution for three instances, namely pr05-N240p4m10, pr08-N144p6m6 and pr10-N288p6m12. M1+(3.13) found a feasible solution for eight instances including pr08-N144p6m6 which remained unsolved by other models. While M1+(3.13) returned superior solutions in two instances, it does not produce the best solutions for six instances with an average deviation of 6.85% of the best solution. While L1+(3.20, 3.21, 3.28) does not reach a feasible solution for unsolvable instances by L1, it is capable of finding a best solution for three instances overall. Generally, M1+(3.13) can be considered as the most reliable model among other models with respect to instances also in S2 with 5.14% deviation from the best OFV and finding a feasible solution for eight instances out of ten.

Table 3.3 Results of the alternative PVRP models on data set S3

Instace	Best OFV	M1					L1					UC1				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
Instn8t5k3_1-Q871	5.81	5.81	0.00	9.83	3.89	3.87	5.81	0.00	2.80	3.99	3.99	**5.81**	0.00	0.36	3.75	3.58
Instn8t5k3_2-Q705	5.81	5.81	0.00	5.12	4.04	4.01	5.81	0.00	2.69	4.16	4.16	**5.81**	0.00	0.42	3.75	3.74
Instn8t5k3_3-Q765	5.81	5.81	0.00	10.38	3.98	3.95	5.81	0.00	6.43	4.09	4.09	**5.81**	0.00	0.34	3.75	3.65
Instn8t5k3_4-Q496	6.32	6.32	0.00	27.77	4.36	4.32	6.32	0.00	3.09	4.53	4.53	6.32	0.00	2.20	4.30	4.22
Instn10t5k3_1-Q1058	6.84	6.84	0.00	1023.98	3.24	3.15	6.84	0.00	11.30	4.41	4.41	**6.84**	0.00	4.66	3.22	3.21
Instn10t5k3_2-Q1045	6.84	6.84	0.00	472.16	3.25	3.16	6.84	0.00	18.39	4.44	4.44	**6.84**	0.00	0.86	3.23	3.22
Instn10t5k3_3-Q1208	6.70	6.70	0.00	118.16	3.09	3.04	6.70	0.00	5.45	4.15	4.15	**6.70**	0.00	0.51	3.12	3.12
Instn10t5k3_4-Q1037	6.84	6.84	0.00	609.86	3.26	3.17	6.84	0.00	9.76	4.45	4.45	**6.84**	0.00	0.83	3.23	3.23
Instn12t5k3_1-Q546	4.81	4.81	0.00	T-UP	3.05	3.05	4.81	0.00	1031.51	3.89	3.89	4.81	0.00	149.54	3.18	3.18
Instn12t5k3_2-Q802	4.48	4.48	0.00	504.88	3.05	3.05	4.48	0.00	31.31	3.53	3.53	4.48	0.00	11.65	3.14	3.14
Instn12t5k3_3-Q748	4.51	4.51	0.00	1528.81	3.05	3.05	4.51	0.00	88.05	3.58	3.58	4.51	0.00	23.48	3.15	3.15
Instn12t5k3_4-Q925	4.42	4.42	0.00	132.24	3.05	3.05	4.42	0.00	16.15	3.45	3.45	**4.42**	0.00	1.62	3.14	3.13
Instn12t5k3_5-Q1491	4.42	4.42	0.00	102.77	3.05	3.05	4.42	0.00	12.70	3.30	3.30	**4.42**	0.00	1.61	3.14	3.10
Instn12t5k3_6-Q1521	4.42	4.42	0.00	267.38	3.05	3.05	4.42	0.00	19.01	3.29	3.29	**4.42**	0.00	1.61	3.14	3.10
Instn12t5k3_7-Q1399	4.42	4.42	0.00	136.03	3.05	3.05	4.42	0.00	12.28	3.32	3.32	**4.42**	0.00	1.62	3.14	3.10
Instn12t5k3_8-Q1146	4.42	4.42	0.00	125.52	3.05	3.05	4.42	0.00	13.29	3.37	3.37	**4.42**	0.00	1.64	3.14	3.11
Instn16t5k3_1-Q1056	5.62	5.66	0.71	T-UP	2.67	2.67	5.62	0.00	310.52	3.29	3.29	**5.62**	0.00	14.27	2.73	2.73
Instn16t5k3_2-Q1030	5.65	5.67	0.35	T-UP	2.68	2.68	5.65	0.00	985.50	3.32	3.32	**5.65**	0.00	181.66	2.74	2.74
Instn16t5k3_3-Q1240	5.62	5.62	0.00	T-UP	2.65	2.65	5.62	0.00	414.80	3.16	3.16	**5.62**	0.00	13.43	2.69	2.69
Instn16t5k3_4-Q1232	5.62	5.64	0.36	T-UP	2.65	2.65	5.62	0.00	147.06	3.16	3.16	**5.62**	0.00	11.19	2.69	2.69
Instn16t5k3_5-Q802	5.77	5.77	0.00	T-UP	2.72	2.72	5.77	0.00	424.29	3.65	3.65	**5.77**	0.00	185.88	2.85	2.85
Instn16t5k3_6-Q757	5.77	5.77	0.00	T-UP	2.73	2.73	5.77	0.00	699.22	3.74	3.74	5.77	0.00	174.77	2.88	2.88
Instn16t5k3_7-Q851	5.76	5.77	0.17	T-UP	2.71	2.71	5.76	0.00	1254.57	3.56	3.56	5.76	0.00	639.33	2.82	2.82
Instn16t5k3_8-Q1027	5.65	5.67	0.35	T-UP	2.68	2.68	5.65	0.00	1096.99	3.32	3.32	5.65	0.00	142.01	2.74	2.74
Instn50t5k4-Q1111	14.44	15.53	7.55	T-UP	8.55	8.55	14.90	3.19	T-UP	11.10	11.10	NA	_	T-UP	9.23	9.23

Table 3.3 shows the results for data set S3; Table 3.4 reports the number of instances in S3 solved to optimality by the alternative PVRP models. The instances are grouped based on the number of customers ($|N_c|$) as given in the first column, and the number of instances in each group is provided in the second column plus the number of instances solved to optimality by alternative models. The next six

Table 3.3 Results of the alternative PVRP models on data set S3 (continued)

Instace	Best OFV	M1+(3.13)					M1+(3.13, 3.20, 3.21)					L1+(3.20, 3.21, 3.28)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
Instn8t5k3_1-Q871	5.81	5.81	0.00	13.59	4.94	4.48	5.81	0.00	12.06	4.94	4.61	5.81	0.00	1.21	4.80	4.65
Instn8t5k3_2-Q705	5.81	5.81	0.00	23.73	4.94	4.48	5.81	0.00	8.48	4.94	4.61	5.81	0.00	0.99	4.83	4.71
Instn8t5k3_3-Q765	5.81	5.81	0.00	10.33	4.94	4.48	5.81	0.00	10.81	4.94	4.61	5.81	0.00	0.79	4.82	4.69
Instn8t5k3_4-Q496	6.32	6.32	0.00	106.38	4.97	4.48	6.32	0.00	38.59	4.97	4.61	**6.32**	0.00	0.71	5.05	5.00
Instn10t5k3_1-Q1058	6.84	6.84	0.00	2160.02	5.31	5.11	6.84	0.00	T-UP	5.31	5.20	6.84	0.00	5.96	6.01	5.93
Instn10t5k3_2-Q1045	6.84	6.84	0.00	2095.00	5.31	5.11	6.84	0.00	T-UP	5.31	5.20	6.84	0.00	5.46	6.02	5.95
Instn10t5k3_3-Q1208	6.70	6.70	0.00	433.42	5.31	5.11	6.70	0.00	526.61	5.31	5.20	6.70	0.00	1.50	5.88	5.79
Instn10t5k3_4-Q1037	6.84	6.84	0.00	940.22	5.31	5.11	6.84	0.00	T-UP	5.31	5.20	6.84	0.00	2.86	6.03	5.95
Instn12t5k3_1-Q546	4.81	4.81	0.00	T-UP	3.55	3.13	4.81	0.00	5721.48	3.55	3.48	**4.81**	0.00	40.83	4.12	4.12
Instn12t5k3_2-Q802	4.48	4.48	0.00	90.00	3.55	3.13	4.48	0.00	59.03	3.55	3.48	**4.48**	0.00	8.42	3.77	3.77
Instn12t5k3_3-Q748	4.51	4.51	0.00	203.07	3.55	3.13	4.51	0.00	137.03	3.55	3.48	**4.51**	0.00	7.94	3.81	3.81
Instn12t5k3_4-Q925	4.42	4.42	0.00	45.00	3.55	3.13	4.42	0.00	26.16	3.55	3.48	4.42	0.00	2.06	3.71	3.70
Instn12t5k3_5-Q1491	4.42	4.42	0.00	28.31	3.55	3.13	4.42	0.00	34.41	3.55	3.48	4.42	0.00	2.13	3.61	3.60
Instn12t5k3_6-Q1521	4.42	4.42	0.00	18.67	3.55	3.13	4.42	0.00	31.05	3.55	3.48	4.42	0.00	3.53	3.61	3.60
Instn12t5k3_7-Q1399	4.42	4.42	0.00	19.66	3.55	3.13	4.42	0.00	41.19	3.55	3.48	4.42	0.00	3.28	3.62	3.61
Instn12t5k3_8-Q1146	4.42	4.42	0.00	31.45	3.55	3.13	4.42	0.00	22.50	3.55	3.48	4.42	0.00	2.70	3.66	3.64
Instn16t5k3_1-Q1056	5.62	5.64	0.36	T-UP	3.90	3.34	5.68	1.07	T-UP	3.90	3.78	5.62	0.00	34.36	4.37	4.18
Instn16t5k3_2-Q1030	5.65	5.71	1.06	T-UP	3.90	3.34	5.7	0.88	T-UP	3.90	3.78	5.65	0.00	186.19	4.39	4.20
Instn16t5k3_3-Q1240	5.62	5.62	0.00	T-UP	3.90	3.34	5.63	0.18	T-UP	3.90	3.78	5.62	0.00	31.16	4.27	4.07
Instn16t5k3_4-Q1232	5.62	5.62	0.00	T-UP	3.90	3.34	5.62	0.00	T-UP	3.90	3.78	5.62	0.00	31.73	4.28	4.07
Instn16t5k3_5-Q802	5.77	5.77	0.00	T-UP	3.90	3.34	5.77	0.00	T-UP	3.90	3.78	5.77	0.00	128.80	4.59	4.43
Instn16t5k3_6-Q757	5.77	5.77	0.00	T-UP	3.90	3.34	5.83	1.04	T-UP	3.90	3.78	**5.77**	0.00	16.45	4.65	4.50
Instn16t5k3_7-Q851	5.76	5.76	0.00	T-UP	3.90	3.34	5.77	0.17	T-UP	3.90	3.78	**5.76**	0.00	621.33	4.54	4.37
Instn16t5k3_8-Q1027	5.65	5.66	0.18	T-UP	3.90	3.34	5.67	0.35	T-UP	3.90	3.78	**5.65**	0.00	121.53	4.39	4.20
Instn50t5k4-Q1111	14.44	16.61	15.03	T-UP	8.66	8.66	16	10.80	T-UP	9.83	9.36	**14.44**	0.00	T-UP	11.61	11.31

Table 3.4 The number of instances in data set S3 solved to optimality by alternative models

$\|N_c\|$	No	M1	L1	UC1	M1+(3.13)	M1+(3.13, 3.20, 3.21)	L1+(3.20, 3.21, 3.28)
7	4/4	4/4	4/4	4/4	4/4	4/4	4/4
9	4/4	4/4	4/4	4/4	4/4	4/1	4/4
11	8/8	8/7	8/8	8/8	8/7	8/8	8/8
15	8/8	3/0	8/8	8/8	5/0	3/0	8/8
49	1/0	0/0	0/0	0/0	0/0	0/0	0/0

Table 3.5 Results of the alternative PVRP models on data set S4

	Instance	Best OFV	M1					L1					UC1				
			OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
radius 15	FPVRP_n10k4t5_2	13.02	13.02	0.00	T-UP	7.56	1.76	**13.02**	0.00	37.45	10.15	10.15	13.02	0.00	T-UP	9.45	9.39
	FPVRP_n10k4t5_4	15.09	15.09	0.00	T-UP	7.78	3.80	**15.09**	0.00	29.30	11.19	11.19	15.09	0.00	T-UP	10.85	10.71
	FPVRP_n10k5t5_1	_	inf	_	27.28	19.05	4.54	inf	_	242.64	17.08	17.08	inf	_	0.70	18.97	16.02
	FPVRP_n10k5t5_3	13.79	13.79	0.00	50.69	9.69	3.36	**13.79**	0.00	6.13	11.42	11.42	13.79	0.00	T-UP	9.95	7.85
	FPVRP_n10k8t5_5	26.07	27.79	6.60	0.73	23.87	3.76	27.79	6.60	3.16	23.65	23.65	27.79	6.60	3.30	25.29	23.22
	FPVRP_n15k6t5_2	19.82	**19.82**	0.00	T-UP	7.91	4.87	**19.82**	0.00	T-UP	15.05	15.05	**19.82**	0.00	T-UP	9.79	9.75
	FPVRP_n15k7t5_5	25.66	**25.66**	0.00	T-UP	14.39	4.49	**25.66**	0.00	T-UP	20.81	20.81	**25.66**	0.00	T-UP	18.37	17.11
	FPVRP_n15k8t5_4	34.18	34.18	0.00	T-UP	27.67	6.48	34.18	0.00	5304.79	27.46	27.46	34.18	0.00	T-UP	28.52	25.22
	FPVRP_n15k10t5_1	36.26	**36.26**	0.00	11.69	33.36	10.83	36.26	0.00	90.03	30.72	30.72	36.26	0.00	T-UP	31.85	19.01
	FPVRP_n15k10t5_3	28.06	28.06	0.00	1.81	25.38	4.89	28.06	0.00	65.61	23.11	23.11	28.06	0.00	2185.41	23.38	21.24
	FPVRP_n20k10t5_1	25.88	**25.88**	0.00	T-UP	15.60	5.03	**25.88**	0.00	T-UP	21.81	21.81	25.90	0.08	T-UP	16.40	13.16
	FPVRP_n20k10t5_4	39.55	39.57	0.05	T-UP	24.62	6.24	**39.55**	0.00	T-UP	31.10	31.10	39.57	0.05	T-UP	29.28	23.79
	FPVRP_n20k10t5_5	28.38	31.55	11.17	T-UP	12.86	5.10	31.50	10.99	T-UP	26.28	26.28	31.50	10.99	T-UP	23.50	20.45
	FPVRP_n20k11t5_3	26.70	**26.70**	0.00	T-UP	17.47	5.53	**26.70**	0.00	T-UP	20.95	20.95	**26.70**	0.00	T-UP	18.33	14.63
	FPVRP_n20k12t5_2	39.08	39.08	0.00	T-UP	30.88	6.99	39.08	0.00	1114.64	31.85	31.85	39.08	0.00	T-UP	32.52	22.87
radius 30	FPVRP_n10k5t5_3	15.58	15.58	0.00	89.42	10.58	4.12	**15.58**	0.00	26.47	11.84	11.84	15.58	0.00	T-UP	10.85	9.50
	FPVRP_n10k5t5_4	15.06	15.06	0.00	59.52	10.50	5.99	**15.06**	0.00	11.36	12.53	12.53	15.06	0.00	T-UP	11.52	11.12
	FPVRP_n10k6t5_1	20.15	20.15	0.00	2.44	17.79	7.42	20.15	0.00	21.95	17.26	17.26	20.15	0.00	1301.65	15.81	12.53
	FPVRP_n10k6t5_2	15.62	15.62	0.00	0.64	14.38	5.37	15.62	0.00	35.55	12.14	12.14	15.62	0.00	41.03	12.87	11.30
	FPVRP_n10k8t5_5	21.53	21.53	0.00	0.38	21.47	6.68	21.53	0.00	11.08	17.99	17.99	21.53	0.00	28.25	21.00	15.91
	FPVRP_n15k6t5_5	22.25	**22.25**	0.00	T-UP	14.54	6.95	**22.25**	0.00	T-UP	17.06	17.06	**22.25**	0.00	T-UP	14.97	14.70
	FPVRP_n15k7t5_3	29.17	29.17	0.00	T-UP	21.65	9.91	29.17	0.00	T-UP	23.76	23.76	29.17	0.00	T-UP	24.99	21.01
	FPVRP_n15k7t5_4	18.27	18.27	0.00	T-UP	11.72	6.85	18.27	0.00	T-UP	14.88	14.88	18.27	0.00	T-UP	13.94	13.82
	FPVRP_n15k9t5_1	27.98	27.98	0.00	27.44	23.73	11.80	27.98	0.00	187.52	25.12	25.12	27.98	0.00	T-UP	23.54	21.54
	FPVRP_n15k9t5_2	32.89	**32.89**	0.00	15.66	29.84	7.17	32.89	0.00	182.69	26.84	26.84	32.89	0.00	T-UP	28.90	22.72
	FPVRP_n20k10t5_1	32.56	**32.56**	0.00	T-UP	23.65	10.74	**32.56**	0.00	T-UP	25.86	25.86	32.58	0.06	T-UP	24.96	19.54
	FPVRP_n20k10t5_3	28.62	**28.62**	0.00	T-UP	16.34	8.06	**28.62**	0.00	T-UP	22.72	22.72	**28.62**	0.00	T-UP	21.07	18.55
	FPVRP_n20k12t5_2	32.00	33.44	4.50	443.20	27.71	11.04	33.44	4.50	1241.03	28.99	28.99	33.44	4.50	T-UP	29.65	26.32
	FPVRP_n20k12t5_5	37.24	37.24	0.00	28.83	25.35	12.27	37.24	0.00	2589.06	30.80	30.80	37.24	0.00	T-UP	27.45	25.01
	FPVRP_n20k13t5_4	46.22	46.22	0.00	525.44	39.39	11.72	46.22	0.00	2259.42	38.32	38.32	46.22	0.00	T-UP	41.25	31.11
radius 50	FPVRP_n20k7t5_3	24.43	**24.43**	0.00	T-UP	17.26	12.78	**24.43**	0.00	T-UP	20.39	20.39	24.48	0.20	T-UP	19.61	16.55
	FPVRP_n20k10t5_2	32.42	**32.42**	0.00	T-UP	21.05	12.17	**32.42**	0.00	T-UP	25.11	25.11	**32.42**	0.00	T-UP	23.60	20.71
	FPVRP_n20k10t5_4	27.93	**27.93**	0.00	T-UP	17.16	8.94	28.02	0.32	T-UP	21.31	21.31	27.95	0.07	T-UP	17.86	15.18
	FPVRP_n20k11t5_5	38.13	**38.13**	0.00	3791.09	28.13	13.20	38.13	0.00	T-UP	30.50	30.50	38.13	0.00	T-UP	29.20	26.24
	FPVRP_n20k14t5_1	34.45	**34.45**	0.00	190.46	27.76	10.64	34.45	0.00	5518.98	29.84	29.84	34.45	0.00	T-UP	30.57	27.21

Table 3.5 Results of the alternative PVRP models on data set S4 (continued)

	Instance	Best OFV	M1+(3.13)					M1+(3.13, 3.20, 3.21)					L1+(3.20, 3.21, 3.28)				
			OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
	FPVRP_n10k4t5_2	13.02	13.02	0.00	T-UP	7.91	1.78	13.02	0.00	T-UP	7.99	2.41	13.02	0.00	58.58	10.88	10.88
	FPVRP_n10k4t5_4	15.09	15.09	0.00	T-UP	9.32	4.03	15.09	0.00	T-UP	9.77	4.07	15.09	0.00	103.08	11.42	11.26
	FPVRP_n10k5t5_1	_	inf	_	5.24	20.27	4.25	inf	_	5.80	20.50	4.36	inf	_	47.59	17.35	17.10
	FPVRP_n10k5t5_3	13.79	13.79	0.00	17.80	10.51	3.41	13.79	0.00	31.87	10.51	3.69	13.79	0.00	6.44	11.74	11.66
	FPVRP_n10k8t5_5	26.07	**26.07**	0.00	0.50	24.26	1.99	26.07	0.00	0.83	25.40	2.36	27.79	6.60	2.55	23.71	23.68
	FPVRP_n15k6t5_2	19.82	19.83	0.05	T-UP	8.43	4.98	**19.82**	0.00	T-UP	8.87	5.18	**19.82**	0.00	T-UP	15.51	15.20
	FPVRP_n15k7t5_5	25.66	**25.66**	0.00	T-UP	15.26	4.51	25.67	0.04	T-UP	15.26	4.96	**25.66**	0.00	T-UP	21.18	20.89
radius 15	FPVRP_n15k8t5_4	34.18	34.18	0.00	T-UP	27.77	4.65	34.18	0.00	T-UP	27.77	4.95	**34.18**	0.00	1698.33	27.73	27.49
	FPVRP_n15k10t5_1	36.26	36.26	0.00	12.27	33.68	6.09	36.26	0.00	15.86	33.68	6.26	36.26	0.00	108.14	30.82	30.74
	FPVRP_n15k10t5_3	28.06	**28.06**	0.00	1.53	25.66	4.38	28.06	0.00	2.38	25.86	5.03	28.06	0.00	59.25	23.25	23.11
	FPVRP_n20k10t5_1	25.88	**25.88**	0.00	T-UP	17.17	5.08	**25.88**	0.00	T-UP	18.76	5.45	**25.88**	0.00	T-UP	21.93	21.87
	FPVRP_n20k10t5_4	39.55	39.57	0.05	T-UP	26.32	5.47	**39.55**	0.00	T-UP	27.24	5.83	**39.55**	0.00	T-UP	31.51	31.13
	FPVRP_n20k10t5_5	28.38	**28.38**	0.00	T-UP	13.03	4.77	**28.38**	0.00	T-UP	16.92	5.05	31.50	10.99	T-UP	26.47	26.29
	FPVRP_n20k11t5_3	26.70	**26.70**	0.00	T-UP	19.19	5.49	**26.70**	0.00	T-UP	19.68	5.91	**26.70**	0.00	T-UP	21.08	20.95
	FPVRP_n20k12t5_2	39.08	39.08	0.00	T-UP	31.69	6.93	39.08	0.00	T-UP	32.52	7.31	**39.08**	0.00	853.30	31.93	31.87
	FPVRP_n10k5t5_3	15.58	15.58	0.00	57.55	10.83	4.08	15.58	0.00	50.06	10.85	5.66	15.58	0.00	27.77	12.94	12.76
	FPVRP_n10k5t5_4	15.06	15.06	0.00	54.21	10.97	6.32	15.06	0.00	115.04	11.23	6.36	15.06	0.00	19.72	13.17	13.17
	FPVRP_n10k6t5_1	20.15	**20.15**	0.00	2.05	17.92	7.52	20.15	0.00	2.44	18.07	8.05	20.15	0.00	36.75	17.70	17.38
	FPVRP_n10k6t5_2	15.62	**15.62**	0.00	0.45	14.38	5.26	15.62	0.00	0.72	14.38	5.55	15.62	0.00	43.11	12.39	12.14
	FPVRP_n10k8t5_5	21.53	21.53	0.00	0.30	21.47	5.42	**21.53**	0.00	0.28	21.53	5.96	21.53	0.00	10.33	18.09	17.99
	FPVRP_n15k6t5_5	22.25	**22.25**	0.00	T-UP	15.25	7.39	**22.25**	0.00	T-UP	15.25	8.13	**22.25**	0.00	T-UP	18.23	17.57
	FPVRP_n15k7t5_3	29.17	29.17	0.00	T-UP	22.85	10.07	**29.17**	0.00	6985.89	23.03	10.15	29.17	0.00	T-UP	23.83	23.81
radius 30	FPVRP_n15k7t5_4	18.27	18.27	0.00	T-UP	12.26	7.00	18.27	0.00	T-UP	12.26	7.57	**18.27**	0.00	4638.70	15.65	15.27
	FPVRP_n15k9t5_1	27.98	**27.98**	0.00	24.25	24.85	10.72	27.98	0.00	29.94	25.30	10.95	27.98	0.00	128.55	25.40	25.14
	FPVRP_n15k9t5_2	32.89	32.89	0.00	19.24	30.26	6.77	32.89	0.00	20.60	30.26	7.03	32.89	0.00	264.09	27.02	26.99
	FPVRP_n20k10t5_1	32.56	**32.56**	0.00	T-UP	23.93	10.76	**32.56**	0.00	T-UP	25.00	11.11	**32.56**	0.00	T-UP	26.21	26.14
	FPVRP_n20k10t5_3	28.62	**28.62**	0.00	T-UP	19.00	8.43	**28.62**	0.00	T-UP	19.04	9.22	**28.62**	0.00	T-UP	22.96	22.76
	FPVRP_n20k12t5_2	32.00	32.00	0.00	404.66	28.25	11.02	**32.00**	0.00	208.64	29.31	11.30	33.44	4.50	876.14	29.06	28.99
	FPVRP_n20k12t5_5	37.24	37.24	0.00	37.64	26.90	9.71	**37.24**	0.00	25.75	27.96	9.97	37.24	0.00	2898.42	30.88	30.88
	FPVRP_n20k13t5_4	46.22	46.22	0.00	166.98	40.42	9.48	**46.22**	0.00	52.08	42.31	10.92	46.22	0.00	3596.06	38.38	38.37
	FPVRP_n20k7t5_3	24.43	24.56	0.53	T-UP	17.43	12.85	**24.43**	0.00	T-UP	18.00	13.50	**24.43**	0.00	T-UP	20.61	20.48
	FPVRP_n20k10t5_2	32.42	**32.42**	0.00	T-UP	21.27	12.33	**32.42**	0.00	T-UP	22.89	13.16	**32.42**	0.00	T-UP	25.55	25.33
radius 50	FPVRP_n20k10t5_4	27.93	**27.93**	0.00	T-UP	18.22	9.55	**27.93**	0.00	T-UP	20.35	10.87	**27.93**	0.00	T-UP	22.06	21.90
	FPVRP_n20k11t5_5	38.13	38.13	0.00	T-UP	29.83	12.98	38.13	0.00	T-UP	32.99	13.74	38.13	0.00	T-UP	31.60	31.13
	FPVRP_n20k14t5_1	34.45	34.45	0.00	260.39	29.75	10.27	34.45	0.00	328.02	31.56	10.60	34.45	0.00	1946.27	29.84	29.84

columns indicate the number of instances solved to optimality and the number of instances solved to optimality by proof by each model. For example, 8/7 for model M1 means that M1 reach the optimal solution for eight instances. However, the optimality is proven for seven instances out of eight.

According to the results presented in Table 3.3 and Table 3.4, L1, L1+(3.20, 3.21, 3.28) and UC1 find optimal solutions to all of the instances with up to 15 customers. Models M1 and M1+(3.13) solve the instances having up to nine customers plus seven of the instances with 11 customers to proven optimality. M1, M1+(3.13) and M1+(3.13, 3.20, 3.21) fail to reach a proven optimal solution for instances with 14 customers. The non-optimal solutions obtained with M1, M1+(3.13) and M1+(3.13, 3.20, 3.21) have respectively an average 1.58%, 4.15% and 2.07% deviation from the best OFV. The best OFV is gained by L1+(3.20, 3.21, 3.28) for instance Instn50t5k4-Q1111. Table 3.3 suggests that M1, M1+(3.13) are more demanding than L1, L1+(3.20, 3.21, 3.28) and UC1 in terms of computation time. Finally, it should also be noted that UC1 cannot return a feasible solution for the instance with 49 customers within the time limit unlike the other models.

Table 3.5 and Table 3.6 show the results for data set S4. Table 3.6, similar to Table 3.4 in format, shows the number of instances solved to optimality by the alternative models. In this Table, instances are categorized with respect to radius. The results show that eight and 12 instances are solved to proven optimality out of each 15 instances categorized as radius 15 and 30 respectively. L1+(3.20, 3.21, 3.28) finds the optimal solution for all of them while not being able to prove optimality for one of them. Only two instances out of five instances with radius 50 are solved to proven optimality by alternative models. Despite UC1 not being able to prove optimality for both of them, M1 reaches the optimal solution with proof for both. Except for instance FPVRP_n10k5t5_1 which is infeasible due to the capacity violation, all models reach the reported best OFV for most of the instances.

Table 3.6 The number of instances in data set S4 solved to optimality by alternative models

Radius	No	M1	L1	UC1	M1+(3.13)	M1+(3.13, 3.20, 3.21)	L1+(3.20, 3.21, 3.28)
15	15/8	8/4	8/8	7/2	8/4	8/4	8/8
30	15/12	12/10	12/10	12/3	12/10	12/11	12/11
50	5/2	2/2	2/1	2/0	2/1	2/1	2/1

Table 3.7 shows the results for the first instance of each combination in the data set S4. L1 and C1 both find optimal solutions to all of the instances with 10 and 20 customers despite C1 not being able to prove optimality for three of them. For M1, M1+(3.13), M1+(3.13, 3.20, 3.21) and L1+(3.20, 3.21, 3.28) these numbers are seven, seven, six and one, respectively.

Considering the solution quality, the results of the experiments on data set S4 (Table 3.7) suggest that L1 and L1+(3.20, 3.21, 3.28) outperforms the other models. L1 and L1+(3.20, 3.21, 3.28) respectively solve to optimality 32 and 35 instances out of 80 and attains feasible solutions for the remaining. Nevertheless, M1 is able to identify an optimal solution for 12 and M1+(3.13) and M1+(3.13, 3.20, 3.21) to 13 instances. M1 and M1+(3.13) prove the optimality of seven of those instances and this number for M1+(3.13, 3.20, 3.21) is six. Models M1, M1+(3.13) and M1+(3.13, 3.20, 3.21) produce a feasible solution for the remaining instances except for one (test51-2-3-a-Q110) in which M1+(3.13) hits the time limit without returning a feasible solution.

Table 3.7 Results of the alternative PVRP models on data set S5

Instance	Best OFV	M1					L1					UC1				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
test11-2-2-a-Q51	620.48	620.48	0.00	2.35	471.71	470.70	620.48	0.00	0.97	553.78	553.78	**620.48**	0.00	0.81	556.52	530.99
test11-2-3-a-Q51	620.48	620.48	0.00	2.61	471.71	470.70	**620.48**	0.00	0.94	553.78	553.78	620.48	0.00	1.43	534.87	530.99
test11-3-2-a-Q51	834.96	834.96	0.00	344.80	608.29	607.75	834.96	0.00	2.27	741.16	741.16	**834.96**	0.00	1.26	724.82	701.75
test11-3-3-a-Q51	834.96	834.96	0.00	407.54	608.29	607.75	**834.96**	0.00	1.92	741.16	741.16	834.96	0.00	6.79	705.57	701.75
test11-4-2-a-Q51	1043.91	1043.91	0.00	90.14	740.19	738.60	1043.91	0.00	2.50	920.40	920.40	**1043.91**	0.00	0.93	888.17	874.92
test11-4-3-a-Q51	1043.91	1043.91	0.00	131.06	740.19	738.60	**1043.91**	0.00	2.00	920.40	920.40	1043.91	0.00	2.47	924.54	874.92
test11-5-2-a-Q51	1083.97	1083.97	0.00	T-UP	728.63	724.91	1083.97	0.00	37.67	928.18	928.18	**1083.97**	0.00	7.36	905.66	904.91
test11-5-3-a-Q51	1083.97	1083.97	0.00	T-UP	728.63	724.91	1083.97	0.00	55.53	928.18	928.18	1083.97	0.00	96.38	906.65	904.91
test21-2-2-a-Q71	828.98	828.98	0.00	7161.35	623.09	622.20	**828.98**	0.00	1.92	749.84	749.84	828.98	0.00	3.02	730.58	717.20
test21-2-3-a-Q71	828.98	828.98	0.00	T-UP	623.09	622.20	**828.98**	0.00	1.97	749.84	749.84	828.98	0.00	2.64	717.20	717.20
test21-2-4-a-Q71	828.98	828.98	0.00	T-UP	623.09	622.20	**828.98**	0.00	1.89	749.84	749.84	828.98	0.00	34.97	717.20	717.20
test21-3-2-a-Q71	1088.36	1088.36	0.00	T-UP	781.74	781.14	1088.36	0.00	16.54	931.36	931.36	**1088.36**	0.00	4.21	892.80	870.80
test21-3-3-a-Q71	1088.36	1095.95	0.70	T-UP	781.74	781.14	**1088.36**	0.00	19.73	931.36	931.36	1088.36	0.00	42.95	877.80	870.80
test21-3-4-a-Q71	1088.36	1098.60	0.94	T-UP	781.74	781.14	**1088.36**	0.00	13.81	931.36	931.36	1088.36	0.00	108.85	870.80	870.80
test21-4-2-a-Q71	1253.47	1278.42	1.99	T-UP	828.44	827.78	1253.47	0.00	208.05	985.93	985.93	**1253.47**	0.00	76.29	935.32	934.75
test21-4-3-a-Q71	1253.47	1262.64	0.73	T-UP	828.44	827.78	1253.47	0.00	476.11	985.93	985.93	1253.47	0.00	327.73	935.32	934.75
test21-4-4-a-Q71	1253.47	1259.01	0.44	T-UP	828.44	827.78	1253.47	0.00	456.24	985.93	985.93	1253.47	0.00	T-UP	934.75	934.75
test21-5-2-a-Q71	1729.00	1779.87	2.94	T-UP	984.77	984.77	1729.00	0.00	2341.55	1268.65	1268.65	**1729.00**	0.00	1293.83	1185.11	1122.59
test21-5-3-a-Q71	1729.00	1774.25	2.62	T-UP	984.77	984.77	**1729.00**	0.00	4926.02	1268.65	1268.65	1729.00	0.00	T-UP	1122.59	1122.59
test21-5-4-a-Q71	1729.00	1787.14	3.36	T-UP	984.77	984.77	**1729.00**	0.00	3772.19	1268.65	1268.65	1729.00	0.00	T-UP	1122.59	1122.59
test31-2-2-a-Q80	841.63	876.56	4.15	T-UP	551.97	551.96	841.63	0.00	124.06	665.77	665.77	**841.63**	0.00	15.72	620.02	615.47
test31-2-3-a-Q64	917.28	934.17	1.84	T-UP	553.87	553.85	917.28	0.00	T-UP	689.39	689.39	**917.28**	0.00	1757.67	623.97	623.97
test31-2-4-a-Q64	911.89	918.13	0.68	T-UP	553.87	553.85	911.89	0.00	T-UP	689.39	689.39	911.89	0.00	T-UP	623.97	623.97
test31-3-2-a-Q80	1190.97	1255.21	5.39	T-UP	765.86	765.74	1190.97	0.00	496.63	936.48	936.48	**1190.97**	0.00	70.13	848.47	848.47
test31-3-3-a-Q64	1285.54	1317.64	2.50	T-UP	767.50	767.31	1287.10	0.12	T-UP	1004.17	1004.17	**1285.54**	0.00	T-UP	877.07	877.07
test31-3-4-a-Q64	1287.10	1307.25	1.57	T-UP	767.50	767.31	**1287.10**	0.00	T-UP	1004.17	1004.17	1288.27	0.09	T-UP	877.07	877.07
test31-4-2-a-Q80	1625.85	1704.06	4.81	T-UP	981.08	980.57	1625.85	0.00	1815.83	1250.23	1250.23	1625.85	0.00	T-UP	1144.77	1113.89
test31-4-3-a-Q64	1742.31	1879.50	7.87	T-UP	985.57	984.78	1753.21	0.63	T-UP	1348.56	1348.56	**1742.31**	0.00	T-UP	1152.33	1152.33
test31-4-4-a-Q64	1741.92	1853.06	6.38	T-UP	985.57	984.78	1750.24	0.48	T-UP	1348.56	1348.56	1752.70	0.62	T-UP	1152.33	1152.33
test31-5-2-a-Q80	1673.06	1761.89	5.31	T-UP	1013.46	1013.15	1676.28	0.19	T-UP	1208.85	1208.84	**1673.06**	0.00	T-UP	1110.70	1110.70
test31-5-3-a-Q64	1773.10	1808.14	1.98	T-UP	1015.51	1015.02	1780.12	0.40	T-UP	1301.45	1301.45	1780.61	0.42	T-UP	1147.22	1147.22
test31-5-4-a-Q64	1771.75	1942.78	9.65	T-UP	1015.51	1015.02	1777.24	0.31	T-UP	1301.45	1301.45	1809.47	2.13	T-UP	1147.22	1147.22
test41-2-2-a-Q180	871.32	886.41	1.73	T-UP	653.15	653.15	871.32	0.00	357.30	681.36	681.36	871.32	0.00	144.61	669.79	659.95
test41-2-3-a-Q150	889.32	896.95	0.86	T-UP	653.98	653.98	889.32	0.00	225.06	694.89	694.89	889.32	0.00	2988.48	664.67	664.67
test41-2-4-a-Q60	1073.75	1157.75	7.82	T-UP	661.44	661.32	1080.06	0.59	T-UP	851.96	851.96	1089.37	1.45	T-UP	736.37	736.37
test41-3-2-a-Q180	1185.19	1198.00	1.08	T-UP	952.19	952.16	1185.19	0.00	1137.67	984.03	984.03	1185.19	0.00	408.95	958.07	958.07
test41-3-3-a-Q150	1214.05	1274.48	4.98	T-UP	953.36	953.31	**1214.05**	0.00	754.25	999.55	999.55	1214.05	0.00	T-UP	964.78	964.78
test41-3-4-a-Q60	1489.58	1558.55	4.63	T-UP	963.85	963.57	1493.45	0.26	T-UP	1209.36	1209.36	1543.21	3.60	T-UP	1067.04	1067.04
test41-4-2-a-Q180	1460.46	1514.93	3.73	T-UP	1104.28	1104.17	**1460.46**	0.00	962.98	1161.52	1161.52	1460.46	0.00	6396.19	1124.58	1124.58
test41-4-3-a-Q150	1476.33	1583.14	7.23	T-UP	1105.35	1105.18	1476.33	0.00	2835.68	1182.72	1182.72	1478.80	0.17	T-UP	1132.30	1132.30
test41-4-4-a-Q60	1750.77	1860.68	6.28	T-UP	1114.97	1113.98	**1750.77**	0.00	T-UP	1420.78	1420.78	1942.06	10.93	T-UP	1258.19	1258.19
test41-5-2-a-Q190	1921.38	2040.06	6.18	T-UP	1388.04	1387.97	1924.87	0.18	T-UP	1473.15	1473.15	1971.05	2.59	T-UP	1431.82	1431.82
test41-5-3-a-Q160	1952.17	2117.74	8.48	T-UP	1390.76	1390.66	1952.47	0.02	T-UP	1499.35	1499.35	2001.29	2.52	T-UP	1445.72	1445.72
test41-5-4-a-Q60	2468.46	2712.85	9.90	T-UP	1419.81	1419.06	**2468.46**	0.00	T-UP	1872.36	1872.36	NA	_	T-UP	1614.43	1614.43
test51-2-2-a-Q190	1085.59	1148.34	5.78	T-UP	735.00	734.97	1085.59	0.00	1343.70	855.49	855.49	1085.59	0.00	2978.97	813.44	797.40
test51-2-3-a-Q110	1217.06	1414.89	16.25	T-UP	740.65	740.54	**1217.06**	0.00	T-UP	984.51	984.51	1226.23	0.75	T-UP	884.28	884.28
test51-2-4-a-Q100	1248.60	1352.14	8.29	T-UP	741.98	741.86	**1248.60**	0.00	T-UP	1006.63	1006.63	1345.40	7.75	T-UP	882.75	882.75
test51-3-2-a-Q160	1677.65	1875.03	11.77	T-UP	1125.69	1125.68	**1677.65**	0.00	T-UP	1307.59	1307.59	1703.23	1.52	T-UP	1204.70	1202.87
test51-3-3-a-Q110	1804.88	2165.23	19.97	T-UP	1132.73	1132.72	**1804.88**	0.00	T-UP	1437.71	1437.71	2086.25	15.59	T-UP	1252.03	1252.03
test51-3-4-a-Q100	1888.78	1990.94	5.41	T-UP	1134.98	1134.97	**1888.78**	0.00	T-UP	1483.59	1483.59	2176.38	15.23	T-UP	1272.46	1272.46
test51-4-2-a-Q160	1936.78	2226.11	14.94	T-UP	1251.36	1251.35	1943.83	0.36	T-UP	1450.85	1450.85	1954.72	0.93	T-UP	1360.80	1360.80
test51-4-3-a-Q120	2020.04	2384.30	18.03	T-UP	1253.06	1253.05	2028.29	0.41	T-UP	1544.90	1544.90	2153.69	6.62	T-UP	1405.81	1405.81
test51-4-4-a-Q100	2128.49	2413.85	13.41	T-UP	1254.42	1254.40	2184.53	2.63	T-UP	1633.10	1633.09	2464.00	15.76	T-UP	1444.24	1444.24
test51-5-2-a-Q200	2401.49	2647.04	10.22	T-UP	1501.01	1501.01	2415.55	0.59	T-UP	1715.89	1715.89	2540.61	5.79	T-UP	1622.14	1622.14
test51-5-3-a-Q190	2425.05	2655.09	9.49	T-UP	1501.52	1501.52	2460.19	1.45	T-UP	1732.38	1732.38	NA	_	T-UP	1629.43	1629.43
test51-5-4-a-Q170	2469.66	2756.79	11.63	T-UP	1502.72	1502.72	2470.63	0.04	T-UP	1771.96	1771.96	2917.50	18.13	T-UP	1651.51	1651.51
test61-2-2-a-Q180	1168.95	1374.12	17.55	T-UP	902.62	902.62	1175.40	0.55	T-UP	974.33	974.33	**1168.95**	0.00	5037.86	933.13	931.05
test61-2-3-a-Q120	1267.71	1718.50	35.56	T-UP	905.29	905.28	1269.04	0.10	T-UP	1041.40	1041.40	1383.84	9.16	T-UP	954.83	954.83
test61-2-4-a-Q97	1323.03	1646.27	24.43	T-UP	907.18	907.18	**1323.03**	0.00	T-UP	1095.56	1095.56	1662.45	25.65	T-UP	973.97	973.97
test61-3-2-a-Q190	1741.95	2095.39	20.29	T-UP	1270.74	1270.72	**1741.95**	0.00	T-UP	1388.80	1388.80	1796.95	3.16	T-UP	1312.51	1307.85
test61-3-3-a-Q150	1790.93	2113.95	18.04	T-UP	1273.53	1273.49	1795.87	0.28	T-UP	1444.52	1444.52	1911.15	6.71	T-UP	1328.07	1328.07
test61-3-4-a-Q97	1999.44	2398.03	19.94	T-UP	1280.75	1280.66	3015.23	50.80	T-UP	1617.87	1617.87	2388.40	19.45	T-UP	1409.43	1409.43
test61-4-2-a-Q180	2200.08	2956.72	34.39	T-UP	1537.08	1537.03	2231.16	1.41	T-UP	1710.10	1710.10	2340.70	6.39	T-UP	1626.38	1615.36
test61-4-3-a-Q150	2319.59	2855.08	23.09	T-UP	1539.75	1539.68	**2319.59**	0.00	T-UP	1766.04	1766.03	NA	_	T-UP	1634.97	1634.97
test61-4-4-a-Q97	2507.13	3064.86	22.25	T-UP	1548.50	1548.34	2529.01	0.87	T-UP	1974.49	1974.49	NA	_	T-UP	1708.33	1708.33
test61-5-2-a-Q180	2450.39	3374.70	37.72	T-UP	1914.11	1914.07	2494.84	1.81	T-UP	2040.39	2040.39	2490.96	1.66	T-UP	1943.99	1936.91
test61-5-3-a-Q150	2513.36	3360.93	33.72	T-UP	1916.00	1915.94	2546.63	1.32	T-UP	2094.16	2094.16	4804.95	91.18	T-UP	1957.19	1957.19
test61-5-4-a-Q97	2842.33	3682.97	29.58	T-UP	1922.19	1922.05	**2842.33**	0.00	T-UP	2300.71	2300.71	NA	_	T-UP	2055.85	2055.85
test71-2-2-a-Q200	1243.98	1376.58	10.66	T-UP	935.13	935.13	1243.98	0.00	1781.23	1022.89	1022.89	1243.98	0.00	1336.43	958.23	955.63
test71-2-3-a-Q150	1326.69	1420.85	7.10	T-UP	937.38	937.38	1333.43	0.51	T-UP	1071.03	1071.03	1351.36	1.86	T-UP	974.02	974.02
test71-2-4-a-Q120	1395.07	1727.76	23.85	T-UP	939.63	939.62	**1395.07**	0.00	T-UP	1125.70	1125.70	1583.92	13.54	T-UP	994.50	994.50
test71-3-2-a-Q200	1641.94	1716.65	4.55	T-UP	1153.30	1153.25	1641.94	0.00	3735.14	1299.03	1299.03	1664.46	1.37	T-UP	1220.41	1206.69
test71-3-3-a-Q150	1733.11	2086.44	20.39	T-UP	1155.99	1155.92	**1733.11**	0.00	T-UP	1361.35	1361.35	NA	_	T-UP	1231.63	1231.63
test71-3-4-a-Q120	1790.09	2328.41	30.07	T-UP	1158.69	1158.57	**1790.09**	0.00	T-UP	1432.07	1432.07	NA	_	T-UP	1260.21	1260.21
test71-4-2-a-Q200	2167.52	2480.62	14.45	T-UP	1499.30	1499.27	2188.79	0.98	T-UP	1656.74	1656.74	2386.41	10.10	T-UP	1553.00	1541.67
test71-4-3-a-Q150	2211.85	3149.34	42.38	T-UP	1502.55	1502.50	2268.86	2.58	T-UP	1740.15	1740.15	NA	_	T-UP	1579.03	1579.03
test71-4-4-a-Q120	2322.54	3011.96	29.68	T-UP	1505.81	1505.73	2349.10	1.14	T-UP	1830.92	1830.92	NA	_	T-UP	1632.71	1632.71
test71-5-2-a-Q200	3009.01	3937.49	30.86	T-UP	2094.35	2094.34	**3009.01**	0.00	T-UP	2276.81	2276.81	4513.85	50.01	T-UP	2148.82	2148.82
test71-5-3-a-Q150	2919.21	3322.29	13.81	T-UP	2094.54	2094.53	**2919.21**	0.00	T-UP	2283.52	2283.52	NA	_	T-UP	2151.94	2151.94
test71-5-4-a-Q120	3240.84	4232.61	30.60	T-UP	2099.29	2099.26	**3240.84**	0.00	T-UP	2503.94	2503.94	NA	_	T-UP	2235.17	2235.17

Table 3.7 Results of the alternative PVRP models on data set S5 (continued)

Instance	Best OFV	M1+(3.13)					M1+(3.13, 3.20, 3.21)					L1+(3.20, 3.21, 3.28)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
test11-2-2-a-Q51	620.48	620.48	0.00	14.54	524.04	505.74	620.48	0.00	10.56	524.04	512.85	620.48	0.00	1.64	579.68	553.78
test11-2-3-a-Q51	620.48	620.48	0.00	20.62	524.04	505.74	620.48	0.00	8.53	524.04	512.85	620.48	0.00	1.67	579.68	553.78
test11-3-2-a-Q51	834.96	834.96	0.00	1716.34	676.66	639.31	834.96	0.00	740.97	676.66	665.28	834.96	0.00	3.85	756.51	741.16
test11-3-3-a-Q51	834.96	834.96	0.00	1177.94	676.66	639.31	834.96	0.00	612.58	676.66	665.28	834.96	0.00	3.59	756.51	741.16
test11-4-2-a-Q51	1043.91	1043.91	0.00	334.02	899.02	807.76	1043.91	0.00	195.27	899.02	852.44	1043.91	0.00	3.89	986.19	920.40
test11-4-3-a-Q51	1043.91	1043.91	0.00	331.02	899.02	807.76	1043.91	0.00	319.56	899.02	852.44	1043.91	0.00	2.67	986.19	920.40
test11-5-2-a-Q51	1083.97	1083.97	0.00	T-UP	893.51	762.79	1083.97	0.00	T-UP	893.51	806.21	1083.97	0.00	9.12	984.54	928.18
test11-5-3-a-Q51	1083.97	1083.97	0.00	T-UP	893.51	762.79	1084.12	0.01	T-UP	893.51	806.21	**1083.97**	0.00	9.55	984.54	928.18
test21-2-2-a-Q71	828.98	828.98	0.00	T-UP	708.92	698.34	828.98	0.00	T-UP	708.92	705.91	828.98	0.00	3.19	793.81	749.84
test21-2-3-a-Q71	828.98	828.98	0.00	T-UP	708.92	698.34	828.98	0.00	T-UP	708.92	705.91	828.98	0.00	3.09	793.81	749.84
test21-2-4-a-Q71	828.98	828.98	0.00	4400.69	856.15	698.34	828.98	0.00	T-UP	856.15	705.91	828.98	0.00	2.05	793.81	749.84
test21-3-2-a-Q71	1088.36	1088.36	0.00	T-UP	856.15	840.16	1095.95	0.70	T-UP	856.15	850.23	1088.36	0.00	16.95	979.25	931.36
test21-3-3-a-Q71	1088.36	1088.36	0.00	T-UP	856.15	840.16	1088.36	0.00	T-UP	856.15	850.23	1088.36	0.00	26.38	979.25	931.36
test21-3-4-a-Q71	1088.36	1098.79	0.96	T-UP	856.15	840.16	1088.36	0.00	T-UP	856.15	850.23	1088.36	0.00	30.92	979.25	931.36
test21-4-2-a-Q71	1253.47	1259.01	0.44	T-UP	981.63	905.89	1260.97	0.60	T-UP	981.63	965.40	1253.47	0.00	297.20	1086.28	985.93
test21-4-3-a-Q71	1253.47	1256.30	0.23	T-UP	981.63	905.89	1260.17	0.53	T-UP	981.63	965.40	**1253.47**	0.00	200.70	1086.28	985.93
test21-4-4-a-Q71	1253.47	1259.01	0.44	T-UP	981.63	905.89	1260.97	0.60	T-UP	981.63	965.40	**1253.47**	0.00	199.83	1086.28	985.93
test21-5-2-a-Q71	1729.00	1759.32	1.75	T-UP	1100.54	1100.54	1756.79	1.61	T-UP	1242.50	1166.21	1729.00	0.00	T-UP	1445.47	1268.65
test21-5-3-a-Q71	1729.00	1769.56	2.35	T-UP	1100.54	1100.54	1758.72	1.72	T-UP	1242.50	1166.21	1729.00	0.00	5563.17	1445.47	1268.65
test21-5-4-a-Q71	1729.00	1766.37	2.16	T-UP	1100.54	1100.54	1744.60	0.90	T-UP	1242.50	1166.21	1729.00	0.00	4525.73	1445.47	1268.65
test31-2-2-a-Q80	841.63	845.76	0.49	T-UP	597.58	597.58	880.00	4.56	T-UP	644.21	617.22	841.63	0.00	20.81	745.94	665.77
test31-2-3-a-Q64	917.28	942.37	2.74	T-UP	597.58	597.58	928.77	1.25	T-UP	644.21	617.22	917.28	0.00	T-UP	772.60	689.39
test31-2-4-a-Q64	911.89	944.58	3.58	T-UP	597.58	597.58	911.89	0.00	T-UP	644.21	617.22	**911.89**	0.00	3423.20	772.60	689.39
test31-3-2-a-Q80	1190.97	1207.17	1.36	T-UP	826.70	826.70	1261.81	5.95	T-UP	857.09	836.66	1190.97	0.00	267.69	1011.77	936.48
test31-3-3-a-Q64	1285.54	1319.19	2.62	T-UP	826.70	826.70	1303.86	1.43	T-UP	857.09	836.66	**1285.54**	0.00	T-UP	1071.06	1004.17
test31-3-4-a-Q64	1287.10	1320.77	2.62	T-UP	826.70	826.70	1344.45	4.46	T-UP	857.09	836.66	**1287.10**	0.00	T-UP	1071.06	1004.17
test31-4-2-a-Q80	1625.85	1690.84	4.00	T-UP	1049.37	1049.37	1724.27	6.05	T-UP	1167.62	1101.80	**1625.85**	0.00	846.62	1355.32	1250.23
test31-4-3-a-Q64	1742.31	1801.26	3.38	T-UP	1049.37	1049.37	1805.51	3.63	T-UP	1167.62	1101.80	1752.70	0.60	T-UP	1436.87	1348.56
test31-4-4-a-Q64	1741.92	1857.93	6.66	T-UP	1049.37	1049.37	1799.29	3.29	T-UP	1167.62	1101.80	**1741.92**	0.00	T-UP	1436.87	1348.56
test31-5-2-a-Q80	1673.06	1747.16	4.43	T-UP	1075.18	1075.18	1774.74	6.08	T-UP	1170.13	1139.20	**1673.06**	0.00	T-UP	1328.67	1208.84
test31-5-3-a-Q64	1773.10	1823.27	2.83	T-UP	1075.18	1075.18	1826.94	3.04	T-UP	1170.13	1139.20	**1773.10**	0.00	T-UP	1413.16	1301.45
test31-5-4-a-Q64	1771.75	1822.25	2.85	T-UP	1075.18	1075.18	1853.50	4.61	T-UP	1170.13	1139.20	**1771.75**	0.00	T-UP	1413.16	1301.45
test41-2-2-a-Q180	871.32	883.07	1.35	T-UP	735.14	735.14	877.82	0.75	T-UP	774.02	756.49	**871.32**	0.00	114.72	787.47	681.36
test41-2-3-a-Q150	889.32	893.82	0.51	T-UP	735.14	735.14	889.32	0.00	T-UP	774.02	756.49	**889.32**	0.00	70.19	796.41	694.89
test41-2-4-a-Q60	1073.75	1121.17	4.42	T-UP	735.14	735.14	1130.82	5.32	T-UP	774.02	756.49	**1073.75**	0.00	T-UP	932.40	851.96
test41-3-2-a-Q180	1185.19	1193.56	0.71	T-UP	1030.93	1030.93	1185.19	0.00	T-UP	1094.19	1052.66	**1185.19**	0.00	272.55	1108.25	984.03
test41-3-3-a-Q150	1214.05	1247.92	2.79	T-UP	1030.93	1030.93	1262.03	3.95	T-UP	1094.19	1052.66	1214.05	0.00	980.41	1117.85	999.55
test41-3-4-a-Q60	1489.58	1575.70	5.78	T-UP	1030.93	1030.93	1529.96	2.71	T-UP	1094.19	1052.66	**1489.58**	0.00	T-UP	1303.64	1209.36
test41-4-2-a-Q180	1460.46	1486.47	1.78	T-UP	1217.35	1217.35	1472.58	0.83	T-UP	1315.64	1266.52	1460.46	0.00	3241.75	1335.38	1161.52
test41-4-3-a-Q150	1476.33	1540.26	4.33	T-UP	1217.35	1217.35	1550.49	5.02	T-UP	1315.64	1266.52	**1476.33**	0.00	1804.66	1345.73	1182.72
test41-4-4-a-Q60	1750.77	1869.94	6.81	T-UP	1217.35	1217.35	1840.26	5.11	T-UP	1315.64	1266.52	1752.93	0.12	T-UP	1541.98	1420.78
test41-5-2-a-Q190	1921.38	1946.52	1.31	T-UP	1586.09	1586.09	1991.39	3.64	T-UP	1688.68	1630.51	**1921.38**	0.00	1396.48	1711.70	1473.15
test41-5-3-a-Q160	1952.17	1960.33	0.42	T-UP	1586.09	1586.09	2079.13	6.50	T-UP	1688.68	1630.51	**1952.17**	0.00	4425.48	1727.72	1499.35
test41-5-4-a-Q60	2468.46	2770.25	12.23	T-UP	1586.09	1586.09	2585.36	4.74	T-UP	1688.68	1630.51	2481.09	0.51	T-UP	2057.81	1872.36
test51-2-2-a-Q190	1085.59	1100.05	1.33	T-UP	815.04	815.04	1120.14	3.18	T-UP	870.02	839.68	**1085.59**	0.00	730.91	953.64	855.49
test51-2-3-a-Q110	1217.06	NA	_	T-UP	815.04	815.04	1302.45	7.02	T-UP	870.02	839.68	**1217.06**	0.00	T-UP	1069.25	984.51
test51-2-4-a-Q100	1248.60	1336.75	7.06	T-UP	815.04	815.04	1280.01	2.52	T-UP	870.02	839.68	**1248.60**	0.00	T-UP	1088.11	1006.63
test51-3-2-a-Q160	1677.65	1887.24	12.49	T-UP	1278.65	1278.65	1728.08	3.01	T-UP	1341.48	1311.88	1692.62	0.89	T-UP	1483.78	1307.59
test51-3-3-a-Q110	1804.88	2002.40	10.94	T-UP	1278.65	1278.65	1820.33	0.86	T-UP	1341.48	1311.88	1809.87	0.28	T-UP	1595.24	1437.71
test51-3-4-a-Q100	1888.78	1977.61	4.70	T-UP	1278.65	1278.65	1955.59	3.54	T-UP	1341.48	1311.88	1897.03	0.44	T-UP	1635.83	1483.59
test51-4-2-a-Q160	1936.78	2064.00	6.57	T-UP	1320.66	1320.66	1980.37	2.25	T-UP	1558.95	1440.24	**1936.78**	0.00	T-UP	1671.39	1450.85
test51-4-3-a-Q120	2020.04	2150.20	6.44	T-UP	1320.66	1320.66	2104.09	4.16	T-UP	1558.95	1440.24	**2020.04**	0.00	T-UP	1751.17	1544.90
test51-4-4-a-Q100	2128.49	2292.63	7.71	T-UP	1320.66	1320.66	2306.36	8.36	T-UP	1558.95	1440.24	**2128.49**	0.00	T-UP	1825.96	1633.09
test51-5-2-a-Q200	2401.49	2479.14	3.23	T-UP	1683.66	1683.66	2559.94	6.60	T-UP	1849.94	1747.64	**2401.49**	0.00	T-UP	1996.47	1715.89
test51-5-3-a-Q190	2425.05	2647.58	9.18	T-UP	1683.66	1683.66	2511.43	3.56	T-UP	1849.94	1747.64	**2425.05**	0.00	T-UP	2009.72	1732.38
test51-5-4-a-Q170	2469.66	2575.71	4.29	T-UP	1683.66	1683.66	2575.42	4.28	T-UP	1849.94	1747.64	**2469.66**	0.00	T-UP	2041.60	1771.96
test61-2-2-a-Q180	1168.95	1229.01	5.14	T-UP	965.14	965.14	1281.99	9.67	T-UP	1029.00	996.58	1168.96	0.00	T-UP	1075.91	974.33
test61-2-3-a-Q120	1267.71	1378.80	8.76	T-UP	965.14	965.14	1317.96	3.96	T-UP	1029.00	996.58	**1267.71**	0.00	T-UP	1130.30	1041.40
test61-2-4-a-Q97	1323.03	1393.00	5.29	T-UP	965.14	965.14	1420.12	7.34	T-UP	1029.00	996.58	1346.50	1.77	T-UP	1179.07	1095.56
test61-3-2-a-Q190	1741.95	1781.49	2.27	T-UP	1406.10	1406.10	1782.64	2.34	T-UP	1473.94	1435.96	1787.27	2.60	T-UP	1559.28	1388.80
test61-3-3-a-Q150	1790.93	1929.96	7.76	T-UP	1406.10	1406.10	1889.92	5.53	T-UP	1473.94	1435.96	**1790.93**	0.00	T-UP	1611.55	1444.52
test61-3-4-a-Q97	1999.44	2179.47	9.00	T-UP	1406.10	1406.10	2295.33	14.80	T-UP	1473.94	1435.96	**1999.44**	0.00	T-UP	1765.10	1617.87
test61-4-2-a-Q180	2200.08	2298.84	4.49	T-UP	1687.55	1687.55	2463.29	11.96	T-UP	1794.33	1742.57	**2200.08**	0.00	T-UP	1896.55	1710.10
test61-4-3-a-Q150	2319.59	2394.52	3.23	T-UP	1687.55	1687.55	2446.83	5.49	T-UP	1794.33	1742.57	2328.57	0.39	T-UP	1944.76	1766.03
test61-4-4-a-Q97	2507.13	2767.81	10.40	T-UP	1687.55	1687.55	2658.43	6.03	T-UP	1794.33	1742.57	**2507.13**	0.00	T-UP	2128.00	1974.49
test61-5-2-a-Q180	2450.39	2619.46	6.90	T-UP	2019.71	2019.71	2553.63	4.21	T-UP	2131.95	2058.92	**2450.39**	0.00	T-UP	2228.89	2040.39
test61-5-3-a-Q150	2513.36	2598.59	3.39	T-UP	2019.71	2019.71	2714.55	8.00	T-UP	2131.95	2058.92	**2513.36**	0.00	T-UP	2275.74	2094.16
test61-5-4-a-Q97	2842.33	3062.36	7.74	T-UP	2019.71	2019.71	3121.05	9.81	T-UP	2131.95	2058.92	2854.06	0.41	T-UP	2449.01	2300.71
test71-2-2-a-Q200	1243.98	1281.09	2.98	T-UP	1039.73	1039.73	1254.07	0.81	T-UP	1107.44	1072.65	**1243.98**	0.00	128.48	1175.63	1022.89
test71-2-3-a-Q150	1326.69	1387.89	4.61	T-UP	1039.73	1039.73	1377.74	3.85	T-UP	1107.44	1072.65	**1326.69**	0.00	6163.58	1211.43	1071.03
test71-2-4-a-Q120	1395.07	1544.24	10.69	T-UP	1039.73	1039.73	1496.61	7.28	T-UP	1107.44	1072.65	1421.80	1.92	T-UP	1257.82	1125.70
test71-3-2-a-Q200	1641.94	1738.56	5.88	T-UP	1283.98	1283.98	1702.57	3.69	T-UP	1397.68	1326.36	**1641.94**	0.00	1428.56	1491.67	1299.03
test71-3-3-a-Q150	1733.11	1829.00	5.53	T-UP	1283.98	1283.98	1845.15	6.46	T-UP	1397.68	1326.36	1745.10	0.69	T-UP	1541.59	1361.35
test71-3-4-a-Q120	1790.09	2000.33	11.74	T-UP	1283.98	1283.98	1961.17	9.56	T-UP	1397.68	1326.36	1817.94	1.56	T-UP	1602.65	1432.07
test71-4-2-a-Q200	2167.52	2294.96	5.88	T-UP	1686.77	1686.77	2531.14	16.78	T-UP	1818.31	1761.36	**2167.52**	0.00	T-UP	1922.11	1656.74
test71-4-3-a-Q150	2211.85	2474.41	11.87	T-UP	1686.77	1686.77	2605.37	17.79	T-UP	1818.31	1761.36	**2211.85**	0.00	T-UP	1995.20	1740.15
test71-4-4-a-Q120	2322.54	2679.90	15.39	T-UP	1686.77	1686.77	2568.38	10.58	T-UP	1818.31	1761.36	**2322.54**	0.00	T-UP	2081.11	1830.92
test71-5-2-a-Q200	3009.01	3265.11	8.51	T-UP	2225.74	2225.74	3495.66	16.17	T-UP	2377.38	2317.38	3025.68	0.55	T-UP	2537.06	2276.81
test71-5-3-a-Q150	2919.21	3282.53	12.45	T-UP	2225.74	2225.74	3189.08	9.24	T-UP	2377.38	2317.38	2969.63	1.73	T-UP	2543.76	2283.52
test71-5-4-a-Q120	3240.84	3757.32	15.94	T-UP	2225.74	2225.74	3710.61	14.50	T-UP	2377.38	2317.38	3252.40	0.36	T-UP	2750.39	2503.94

Table 3.8 Summary of results for alternative PVRP formulations

size	data set	No	M1				L1				UC1			
			# Best OFV	#opt	avg Gap	time	# Best OFV	#opt	avg Gap	time	# Best OFV	#opt	avg Gap	time
Small	S1	3	1	0	0.19	0	3	1	0.05	0	2	1	0.10	1
	S3	24	19	15	0.04	0	24	24	0	0	24	24	0	17
	S4	35	30	16	0.05	3	30	19	0.01	6	27	5	0.12	0
	S5	32	12	7	0.13	0	31	23	0.01	10	28	20	0.02	10
	sum\avg	94	62	38	0.08	3	88	67	0.01	16	81	50	0.05	28

size	data set	No	# Best OFV	#opt	# feaSol	avg LB	time	# Best OFV	#opt	# feaSol	avg LB	time	# Best OFV	#opt	# feaSol	avg LB	time
Medium	S1	8	1	0	8	1765.68	0	4	0	5	2153.56	0	2	0	7	2157.89	0
	S2	1	0	0	1	1778.49	0	0	0	1	2038.04	0	0	0	1	2070.17	0
	S3	1	0	0	1	10.23	0	0	0	1	12.41	0	0	0	0	10.70	0
	S5	48	0	0	48	1182.81	0	26	9	48	1774.12	2	0	7	37	1684.71	1
	sum\avg	58	1	0	58	1253.26	0	30	9	55	1800.6	2	2	7	45	1727.76	1

size	data set	No	# Best OFV	# feaSol	avg LB	# Best OFV	# feaSol	avg LB	# Best OFV	# feaSol	avg LB
Large	S1	18	4	15	13431.53	3	6	16139.46	0	2	13946.61
	S2	9	1	6	4675.25	1	3	6160.58	0	0	4801.71
	sum\avg	27	5	21	10512.77	4	9	12813.17	0	2	10898.31

size	data set	No	M1+(3.13)				M1+(3.13, 3.20, 3.21)				L1+(3.20, 3.21, 3.28)			
			# Best OFV	#opt	avg Gap	time	# Best OFV	#opt	avg Gap	time	# Best OFV	#opt	avg Gap	time
Small	S1	3	1	0	0.18	0	0	0	0.18	0	3	1	0.03	0
	S3	24	21	15	0.03	0	18	13	0.05	0	25	24	0	8
	S4	35	31	15	0.05	5	33	16	0.04	5	31	20	0.01	5
	S5	32	13	7	0.12	0	13	6	0.12	0	31	23	0.01	5
	sum\avg	94	66	37	0.07	5	64	35	0.08	5	90	68	0.01	18

size	data set	No	# Best OFV	#opt	# feaSol	avg LB	time	# Best OFV	#opt	# feaSol	avg LB	time	# Best OFV	#opt	# feaSol	avg LB	time
Medium	S1	8	0	0	7	1775.5	0	0	0	7	1771.23	0	2	0	8	2264.25	0
	S2	1	0	0	1	1787.72	0	0	0	1	1802.52	0	1	0	1	2078.58	0
	S3	1	0	0	1	10.25	0	0	0	1	10.25	0	1	0	1	12.41	0
	S5	48	0	0	47	1582.06	0	2	0	48	1580.72	0	33	12	48	1786.65	10
	sum\avg	58	0	0	56	1585.18	0	2	0	57	1583.75	0	37	12	58	1826.97	10

size	data set	No	# Best OFV	# feaSol	avg LB	# Best OFV	# feaSol	avg LB	# Best OFV	# feaSol	avg LB
Large	S1	18	6	18	13222.26	1	17	13219.62	4	8	5190.18
	S2	9	2	7	4698.13	1	6	4778.66	2	3	6234.50
	sum\avg	27	8	25	10380.89	2	23	10405.97	6	11	5538.29

Replacing constraints (3.8) with the stronger versions of SECs (valid inequalities (3.13)) leads to an improvement of up to 22.68% in the solutions of 54 instances. C1 competes with L1 and L1+(3.20, 3.21, 3.28) in the most of small instances containing up to 30 customers. Nevertheless, it is unable to find a feasible solution for 11 relatively larger instances. M1+(3.13) yields significantly better objective function values at the root node than does M1: achieving a maximum of 37.40% improvement. Yet again, L1+(3.20, 3.21, 3.28) is superior to the other models with respect to the quality of the root node objective values.

Across all five data sets, it is consistently observed that model L1+(3.20, 3.21, 3.28) provides the best LP-relaxation objective function values. Model L1 follows closely behind L1+(3.20, 3.21, 3.28) in terms of LP-relaxation quality, often being the second-best performer across all data sets. For data sets S1, S2, and S4, the rankings are consistent, with L1+(3.20, 3.21, 3.28) and L1 leading the LP-relaxation quality, followed by C1, M1+(3.13, 3.20, 3.21), M1+(3.13), and M1. For data set S3, the inclusion of valid inequalities (3.13) and (3.21) and optimality cut (3.20) enhances the LP-relaxation quality of MTZ-formulation to such an extent that M1+(3.13, 3.20, 3.21) and M1+(3.13) surpass both L1 and C1 in terms of tightness. In data set S5, L1+(3.20, 3.21, 3.28) continues to excel, while L1 takes the lead in most instances compared to M1+(3.13, 3.20, 3.21). Models C1 competes with M1+(3.13), and M1+(3.13, 3.20, 3.21) on data set S5. This suggests that the quality of the LP-relaxation objective function can be dependent on the specific data set characteristics.

It should also be noted that, in some instances, the values of the root node relaxation and LP-relaxation are the same. To investigate this further, we incorporated a callback function into the optimization process to count the number of cuts added to the root node relaxation in these instances. It becomes evident that no cuts are added to the root node relaxation, and consequently, the root node and the LP relaxation coincide.

Table 3.8 narrows down the evaluation of the alternative PVRP models into a comparison of OFV, number of optimal solutions, average gap, run time, number of instances for which at least an upper bound is obtained and average best lower bounds. We categorize the instances into three groups as follows; instances with up to 40 customers are considered small, instances with 41 to 71 customers are considered medium, and instances with 72 to 288 customers are considered large. The second column of the table shows the data set while the next column indicates the number of instances in the corresponding set that belongs to the specified instance size category.

For the small category, the next blocks of four columns report the number of instances for which the corresponding model reaches the best OFV, the number of instances solved to optimality, the average gap, and the number of instances solved to optimality faster, respectively. For medium and large categories, due to the existence of instances for which some of the models fail to find a feasible solution, we report the number of instances where at least an upper bound is obtained by the corresponding model (#feaSol) as well as average best obtained lower bounds (avg LB) instead of the average gap. For the large category, the first two columns of the block report the number of instances for which the corresponding model reaches the best OFV and the number of instances solved to feasibility, respectively. The third column presents the average of the best lower bounds attained by the corresponding model.

- Results based on the 94 small instances illustrate the superior performance of L1+(3.20, 3.21, 3.28) in comparison with other models. L1+(3.20, 3.21, 3.28) finds the best OFV for 90 out of 94 instances, among which 68 instances are solved to optimality. L1 and L1+(3.20, 3.21, 3.28) exhibit an average gap of 0.01 across the 94 instances. In terms of solution time, 75 small instances are solved to optimality by the alternative models. For the remaining instances the models hit the time limit. The time it takes for UC1 to find an optimal solution is shorter compared to the other models in 28 of small instances, followed by L1+(3.20, 3.21, 3.28) and L1 in 18 and 16 instances respectively.

- Results based on the 58 medium instances also exhibit the superior performance of L1+(3.20, 3.21, 3.28) with regard to quality of OFV, solution time, average best lower bounds and number of the optimal solutions. Models M1 and L1+(3.20, 3.21, 3.28) successfully find an upper bound for all medium instances followed by M1+(3.13, 3.20, 3.21)). L1+(3.20, 3.21, 3.28) finds the optimal solution faster in 10 out of 13 medium instances solved to optimality by alternative formulations.

- Results based on the 27 large instances demonstrate that M1+(3.13) outperformed the other alternative models. Generally, in large instances the models based on the MTZ-SECs (M1, M1+(3.13), M1+(3.13, 3.20, 3.21)), despite the L1, L1+(3.20, 3.21, 3.28) and UC1, are able to find a feasible solution for most of the instances. Out of 27 large instances, M1-13 finds a feasible solution for 25 instances, while this number is 11 for L1+(3.20, 3.21, 3.28). In terms of the quality of OFV, MTZ-based formulations are still superior to load-based and cut-based models. However, Model L1 obtains the greatest average lower bounds.

The aforementioned results imply that Load-based models are faster than MTZ-based formulations in small and medium instances whereas MTZ-based formulations, particularly M1+(3.13), produce more favorable results in solving large instances.

MTZ-based formulations, use a set of continuous two-indexed variables to guarantee the connectivity of the vehicle routes and to avoid vehicle capacity violations. To this end, the load-based formulations, use a set of continuous three-indexed (commodity flow) variables that makes the formulation stronger. Hence, in terms of solution quality and run time, the load-based models outperform the MTZ-based models in the small and medium instances. However, in large instances, the models cannot even attain a feasible solution for some of the instances.

3.2.3 Evaluation of Alternative PVRPTW Formulations

In order to compare the alternative PVRPTW models, data sets S6 and S7 are used. First, we test M2, L2 and C2; then, we evaluate the impact of valid inequalities and optimality cuts as follows.

- Model M2+(3.13) is obtained by replacing constraints (3.8) with (3.13).
- We add valid inequalities (3.46) to models M2 and L2 and obtain models M2+(3.46) and L2+(3.46), respectively.
- The effect of applying valid inequalities (3.13) and (3.46) together on model M2 is also investigated with model M2+(3.13, 3.46),
- Model M2+(3.13, 3.20, 3.21, 3.46) is obtained by adding optimality cuts (3.20) and valid inequalities (3.21) to model M2+(3.13, 3.46)
- Model L2+(3.20, 3.21, 3.28, 3.46) is obtained by adding valid inequalities (3.21) and optimality cuts (3.20) and (3.28) to model L2+(3.46).

Overall, nine alternative formulations are tested. The time limit is set to 86400 seconds for the instances in the data set S6 and 14400 seconds for the instances in the data set S7.

The results in Table 3.9 highlight the difficulty of the instances in the data set S6. Out of the 20 instances, none can be solved to feasibility by C2, whereas L2 and M2 are able to identify a feasible solution to seven and 15 instances, respectively. More precisely, out of the first ten instances with narrow time windows,

Table 3.9 Results of the alternative PVRPTW models on data set S6

Instance	Best OFV	M2					L2					C2				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
pr01-N48m2p4	2909.27	2911.51	0.08	T-UP	2134.73	1451.45	**2909.27**	0.00	T-UP	2175.85	1628.18	NA	_	T-UP	2140.77	1529.24
pr02-N96m6p4	5090.38	**5090.38**	0.00	T-UP	3887.02	2255.79	5258.48	3.30	T-UP	4047.29	2770.16	NA	_	T-UP	3955.41	2347.14
pr03-N144m9p4	8005.54	8390.26	4.81	T-UP	4471.58	3032.08	8532.49	6.58	T-UP	5084.20	4083.99	NA	_	T-UP	4718.59	3389.02
pr04-N192m12p4	8485.23	9428.59	11.12	T-UP	4713.84	3377.91	**8485.23**	0.00	T-UP	5595.27	4693.53	NA	_	T-UP	5050.71	3836.42
pr05-N240m15p4	9916.44	**9916.44**	0.00	T-UP	4947.11	3525.92	NA	_	T-UP	6181.79	5340.67	NA	_	T-UP	_	4065.58
pr06-N288m18p4	12481.09	12749.94	2.15	T-UP	5857.50	4275.34	12812.89	2.66	T-UP	7546.70	6468.34	NA	_	T-UP	_	5147.57
pr07-N72m5p6	6811.75	6882.41	1.04	T-UP	4748.77	3398.81	NA	_	T-UP	4927.67	3884.69	NA	_	T-UP	4743.33	3504.42
pr08-N144m10p6	11066.50	**11066.50**	0.00	T-UP	5952.98	3842.02	NA	_	T-UP	6824.63	5467.25	NA	_	T-UP	6378.16	4416.01
pr09-N216m15p6	17253.82	NA	_	T-UP	7924.27	5502.17	NA	_	T-UP	9671.64	8315.47	NA	_	T-UP	_	6764.80
pr10-N288m20p6	20054.45	20842.08	3.93	T-UP	9007.69	6069.61	NA	_	T-UP	11793.08	9965.74	NA	_	T-UP	_	7583.17
pr11-N48m2p4	2327.35	2559.99	10.00	T-UP	1451.45	1451.45	**2327.35**	0.00	T-UP	1628.18	1628.18	NA	_	T-UP	1529.24	1529.24
pr12-N96m6p4	4767.58	4960.67	4.05	T-UP	2300.56	2255.79	NA	_	T-UP	2803.51	2770.16	NA	_	T-UP	2347.14	2347.14
pr13-N144m9p4	6574.22	7648.39	16.34	T-UP	3032.30	3032.08	7030.50	6.94	T-UP	4083.99	4083.99	NA	_	T-UP	3389.02	3389.02
pr14-N192m12p4	7242.10	8852.02	22.23	T-UP	3398.52	3377.91	NA	_	T-UP	4710.86	4693.53	NA	_	T-UP	3853.06	3836.42
pr15-N240m15p4	7864.38	9447.75	20.13	T-UP	3626.88	3525.92	NA	_	T-UP	5372.85	5340.67	NA	_	T-UP	4161.96	4065.58
pr16-N288m18p4	9807.60	12327.05	25.69	T-UP	4286.69	4275.34	NA	_	T-UP	6468.88	6468.34	NA	_	T-UP	_	5147.57
pr17-N72m4p6	NA	NA	_	T-UP	3414.30	3398.81	NA	_	T-UP	3897.92	3884.69	NA	_	T-UP	3518.66	3504.43
pr18-N144m8p6	NA	NA	_	T-UP	3848.74	3842.02	NA	_	T-UP	5480.23	5467.25	NA	_	T-UP	4422.73	4416.01
pr19-N216m12p6	NA	NA	_	T-UP	5557.29	5502.17	NA	_	T-UP	8321.87	8315.47	NA	_	T-UP	6814.56	6764.80
pr20-N288m16p6	NA	NA	_	T-UP	6100.82	6069.61	NA	_	T-UP	9970.50	9965.74	NA	_	T-UP	_	7583.18

- the first six instances have a planning horizon of four days and the rest has a planning horizon of six days,
- M2 finds a feasible solution for nine instances, and the only instance (pr09) for which M2 is not able to find a feasible solution has a planning horizon of six days,
- L2 attains a feasible solution for five out of six instances with a planning horizon of four days while it fails to detect a solution for the instances with a planning horizon of six days.

The second ten instances have wide time windows. None of the models reach a feasible solution for the last four instances (pr17 to pr20) with wide time windows and a planning horizon of six days. M2 solves to feasibility all the remaining instances with four days of planning horizon, while L2 is able to solve only two of them.

Table 3.10 Results of the alternative PVRPTW models with valid inequalities on data set S6

Instance	Best OFV	M2+(3.13)					M2+(3.46)					M2+(3.13, 3.46)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
pr01-N48m2p4	2909.27	**2909.27**	0.00	T-UP	2230.87	1544.35	2916.61	0.25	T-UP	2134.73	1451.45	2919.61	0.36	T-UP	2230.87	1544.35
pr02-N96m6p4	5090.38	5194.20	2.04	T-UP	3902.58	2707.14	5176.99	1.70	T-UP	3887.01	2255.79	5092.96	0.05	T-UP	3902.58	2707.14
pr03-N144m9p4	8005.54	8038.46	0.41	T-UP	4543.41	3149.51	8390.26	4.81	T-UP	4471.57	3032.08	8394.93	4.86	T-UP	4543.41	3149.51
pr04-N192m12p4	8485.23	9677.30	14.05	T-UP	4917.03	3636.17	9364.72	10.36	T-UP	4715.61	3377.91	9941.58	17.16	T-UP	4917.03	3636.17
pr05-N240m15p4	9916.44	10028.94	1.13	T-UP	5069.41	3602.33	9964.73	0.49	T-UP	4947.10	3525.92	9937.49	0.21	T-UP	5069.41	3602.33
pr06-N288m18p4	12481.09	12783.49	2.42	T-UP	5991.17	4447.53	12898.60	3.35	T-UP	5857.50	4275.34	12792.55	2.50	T-UP	5991.17	4447.53
pr07-N72m5p6	6811.75	6934.93	1.81	T-UP	4915.18	3440.25	6855.83	0.65	T-UP	4748.77	3398.81	6926.99	1.69	T-UP	4915.18	3440.25
pr08-N144m10p6	11066.50	11177.43	1.00	T-UP	6032.73	3919.68	11851.12	7.09	T-UP	5952.98	3842.02	11825.99	6.86	T-UP	6032.73	3919.68
pr09-N216m15p6	17253.82	**17253.82**	0.00	T-UP	8021.50	5687.37	17551.92	1.73	T-UP	7924.26	5502.17	17531.32	1.61	T-UP	8021.50	5687.37
pr10-N288m20p6	20054.45	20227.26	0.86	T-UP	9243.02	6283.45	21384.83	6.63	T-UP	9007.68	6069.61	21467.51	7.05	T-UP	9243.02	6283.45
pr11-N48m2p4	2327.35	2482.65	6.67	T-UP	1544.35	1544.35	2349.78	0.96	T-UP	1451.45	1451.45	2473.94	6.30	T-UP	1544.35	1544.35
pr12-N96m6p4	4767.58	NA	_	T-UP	2710.51	2707.14	NA	_	T-UP	2300.56	2255.79	**4767.58**	0.00	T-UP	2710.51	2707.14
pr13-N144m9p4	6574.22	7206.87	9.62	T-UP	3149.51	3149.51	**6574.22**	0.00	T-UP	3032.30	3032.08	7251.57	10.30	T-UP	3149.51	3149.51
pr14-N192m12p4	7242.10	9219.59	27.31	T-UP	3649.66	3636.17	**7242.10**	0.00	T-UP	3398.57	3377.91	9200.60	27.04	T-UP	3649.66	3636.17
pr15-N240m15p4	7864.38	9524.95	21.12	T-UP	3695.26	3602.33	**7864.38**	0.00	T-UP	3626.87	3525.92	9697.24	23.31	T-UP	3695.26	3602.33
pr16-N288m18p4	9807.60	12396.29	26.39	T-UP	4456.44	4447.53	**9807.60**	0.00	T-UP	4286.85	4275.34	12054.66	22.91	T-UP	4456.44	4447.53
pr17-N72m4p6	NA	NA	_	T-UP	3457.96	3440.25	NA	_	T-UP	3414.30	3398.81	NA	_	T-UP	3457.96	3440.25
pr18-N144m8p6	NA	NA	_	T-UP	3923.04	3919.68	NA	_	T-UP	3848.73	3842.02	NA	_	T-UP	3923.04	3919.68
pr19-N216m12p6	NA	NA	_	T-UP	5734.61	5687.37	NA	_	T-UP	5557.29	5502.17	NA	_	T-UP	5734.61	5687.37
pr20-N288m16p6	NA	NA	_	T-UP	6312.31	6283.45	NA	_	T-UP	6100.82	6069.61	NA	_	T-UP	6312.31	6283.45

Instance	Best OFV	M2+(3.13, 3.20, 3.21, 3.46)					L2+(3.46)					L2+(3.20, 3.21, 3.28, 3.46)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
pr01-N48m2p4	2909.27	2911.27	0.07	T-UP	2340.76	1634.07	2911.27	0.07	T-UP	2175.84	1628.18	**2909.27**	0.00	T-UP	2356.31	1771.68
pr02-N96m6p4	5090.38	5105.18	0.29	T-UP	4040.27	2787.77	5142.78	1.03	T-UP	4047.29	2770.16	NA	_	T-UP	4157.12	3224.29
pr03-N144m9p4	8005.54	**8005.54**	0.00	T-UP	4814.67	3360.97	9090.67	13.55	T-UP	5084.20	4083.99	NA	_	T-UP	5356.07	4247.93
pr04-N192m12p4	8485.23	9974.17	17.55	T-UP	5267.48	3766.05	8580.41	1.12	T-UP	5595.63	4693.53	NA	_	T-UP	5993.69	4947.27
pr05-N240m15p4	9916.44	10154.66	2.40	T-UP	5547.88	3789.55	NA	_	T-UP	6181.78	5340.67	10706.33	7.97	T-UP	6629.18	5446.26
pr06-N288m18p4	12481.09	12644.75	1.31	T-UP	6367.24	4631.96	**12481.09**	0.00	T-UP	7546.69	6468.34	NA	_	T-UP	7848.18	6647.70
pr07-N72m5p6	6811.75	**6811.75**	0.00	T-UP	5327.63	3527.18	NA	_	T-UP	4927.66	3884.69	NA	_	T-UP	5425.46	3980.95
pr08-N144m10p6	11066.50	NA	_	T-UP	6514.51	4111.79	11524.93	4.14	T-UP	6824.62	5467.25	11552.58	4.39	T-UP	7251.47	5651.66
pr09-N216m15p6	17253.82	NA	_	T-UP	8539.44	5953.24	NA	_	T-UP	9671.64	8315.47	NA	_	T-UP	10048.18	8501.65
pr10-N288m20p6	20054.45	**20054.45**	0.00	T-UP	9923.97	6671.55	NA	_	T-UP	11793.07	9965.74	NA	_	T-UP	12384.71	10234.41
pr11-N48m2p4	2327.35	2544.69	9.34	T-UP	1729.31	1634.07	2379.70	2.25	T-UP	1628.17	1628.18	2353.35	1.12	T-UP	1836.24	1771.68
pr12-N96m6p4	4767.58	NA	_	T-UP	2958.60	2787.77	NA	_	T-UP	2803.50	2770.16	NA	_	T-UP	3349.33	3224.29
pr13-N144m9p4	6574.22	8020.96	22.01	T-UP	3654.97	3360.97	NA	_	T-UP	4083.98	4083.99	NA	_	T-UP	4442.56	4247.93
pr14-N192m12p4	7242.10	9254.08	27.78	T-UP	4128.78	3766.05	NA	_	T-UP	4710.91	4693.53	NA	_	T-UP	5203.58	4947.27
pr15-N240m15p4	7864.38	9766.15	24.18	T-UP	4173.53	3789.55	NA	_	T-UP	5372.84	5340.67	NA	_	T-UP	5675.56	5446.26
pr16-N288m18p4	9807.60	12490.37	27.35	T-UP	5029.47	4631.96	NA	_	T-UP	6469.00	6468.34	NA	_	T-UP	6908.12	6647.70
pr17-N72m4p6	NA	NA	_	T-UP	3843.52	3527.18	NA	_	T-UP	3897.92	3884.69	NA	_	T-UP	4242.72	3980.95
pr18-N144m8p6	NA	NA	_	T-UP	4532.26	4111.79	NA	_	T-UP	5480.23	5467.25	NA	_	T-UP	5931.96	5651.66
pr19-N216m12p6	NA	NA	_	T-UP	6454.47	5953.24	NA	_	T-UP	8321.87	8315.47	NA	_	T-UP	8749.55	8501.65
pr20-N288m16p6	NA	NA	_	T-UP	7168.68	6671.55	NA	_	T-UP	_	9965.74	NA	_	T-UP	10512.21	10234.41

According to Table 3.10, replacing constraints (3.8) with valid inequalities (3.13) leads to an improvement in the solution of five instances with respect to M2, and two instances overall. Moreover, a feasible solution is identified for the instance pr09 with narrow time windows. M2+(3.46) produces better quality solutions in eight and nine instances compared to M2 and M2+(3.13), respectively, and four instances overall. Improvements in the results of M2+(3.46) highlight the effectiveness of the valid inequalities (3.46) in tightening the problems, especially the ones with wide time windows. M2+(3.13, 3.46) yields a better solution for an instance (pr12) for which M2 also returns a feasible solution while the individual addition of valid inequalities even fail to find a feasible solution for this instance. Model M2+(3.13, 3.20, 3.21, 3.46) improves the solution for three instances with narrow time windows overall. However, for instances with wide time windows, the solutions obtained by M2+(3.13, 3.20, 3.21, 3.46) have a deviation of 22.13% on average from reported Best OFV. The performance of L2 is not improved consistently with the addition of the valid inequalities. It should be noticed that, in terms of the root node solution, L2+(3.20, 3.21, 3.28, 3.46) outperforms other formulations, and is followed by L2+(3.46).

Table 3.11 Time window density and width of data set S7

	C1		R1		RC1	
#	TWD	TWW	TWD	TWW	TWD	TWW
01	100%	60.76	100%	10.00	100%	30.00
02	75%	61.27	75%	10.00	75%	30.00
05	100%	121.61	100%	30.00	100%	54.33

Table 3.12 Results of the alternative PVRPTW models on data set S7

Instance	Best OFV	M2					L2					C2				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
P4C101	2913.94	**2913.94**	0.00	43.90	2910.87	786.31	2913.94	0.00	72.91	2910.87	1716.35	2913.94	0.00	8064.03	2910.87	1274.11
P4R101	4162.33	**4162.33**	0.00	120.16	4109.89	1391.86	4162.33	0.00	305.28	4109.89	1773.92	NA	_	T-UP	4109.89	1589.05
P4RC101	3983.30	**3983.30**	0.00	T-UP	3039.33	1313.82	3998.22	0.37	T-UP	3224.68	2050.50	NA	_	T-UP	3014.05	1577.30
P4C102	3079.89	NA	_	T-UP	1954.08	865.92	NA	_	T-UP	2096.00	1767.87	NA	_	T-UP	2019.72	1364.26
P4R102	3763.76	NA	_	T-UP	2414.20	1455.58	NA	_	T-UP	2588.72	1818.07	NA	_	T-UP	2456.53	1623.92
P4RC102	NA	NA	_	T-UP	1950.79	1346.10	NA	_	T-UP	2467.40	2144.50	NA	_	T-UP	2159.56	1670.22
P4C105	2889.92	2889.92	0.00	384.05	2788.82	832.51	2889.92	0.00	1043.81	2788.83	1742.67	NA	_	T-UP	2788.82	1356.68
P4R105	3697.07	3710.12	0.35	T-UP	2963.16	1549.84	NA	_	T-UP	3003.10	1859.74	NA	_	T-UP	2891.48	1703.30
P4RC105	NA	NA	_	T-UP	2272.33	1411.08	NA	_	T-UP	2541.52	2086.72	NA	_	T-UP	2352.25	1686.18
P6C101	3989.48	3989.48	0.00	2846.14	3808.02	1096.97	3989.48	0.00	2226.65	3808.02	2302.67	3998	0.21	T-UP	3808.02	1716.39
P6R101	5393.19	5393.19	0.00	379.66	5337.68	1856.92	5393.19	0.00	611.76	5337.68	2332.18	NA	_	T-UP	5337.68	2029.85
P6RC101	5834.62	5967.34	2.27	T-UP	4279.68	1764.04	NA	_	T-UP	4299.05	2524.85	NA	_	T-UP	4267.18	2020.08
P6C105	4059.25	4062.10	0.07	T-UP	3833.48	1048.12	4069.38	0.25	T-UP	3834.12	2333.17	NA	_	T-UP	3833.48	1745.85
P6R105	NA	NA	_	T-UP	3533.48	1764.35	NA	_	T-UP	3543.15	2224.57	NA	_	T-UP	3505.34	2003.17
P6RC105	NA	NA	_	T-UP	3248.26	1844.41	NA	_	T-UP	3550.29	2799.08	NA	_	T-UP	3350.38	2212.60

Table 3.13 Results of the alternative PVRPTW models with valid inequalities on data set S7

Instance	Best OFV	M2+(3.13)					M2+(3.46)					M2+(3.13, 3.46)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
P4C101	2913.94	2913.94	0.00	96.61	2910.87	816.27	2913.94	0.00	65.00	2910.87	786.31	2913.94	0.00	107.14	2910.87	816.27
P4R101	4162.33	4162.33	0.00	177.78	4109.89	1427.18	4162.33	0.00	195.33	4109.89	1391.86	4162.33	0.00	214.20	4109.89	1427.18
P4RC101	3983.30	3988.87	0.14	T-UP	3089.90	1344.05	**3983.30**	0.00	T-UP	3039.37	1313.82	3989.65	0.16	T-UP	3089.94	1344.05
P4C102	3079.89	NA	_	T-UP	1959.24	898.5	**3079.89**	0.00	T-UP	1954.15	865.92	NA	_	T-UP	1959.25	898.50
P4R102	3763.76	NA	_	T-UP	2422.55	1478.95	3888.21	3.31	T-UP	2499.13	1455.58	**3763.76**	0.00	T-UP	2507.54	1478.95
P4RC102	NA	NA	_	T-UP	1963.84	1406.64	NA	_	T-UP	1950.90	1346.10	NA	_	T-UP	1963.84	1406.64
P4C105	2889.92	2889.92	0.00	658.71	2788.82	861.94	**2889.92**	0.00	175.72	2788.82	832.51	2889.92	0.00	270.18	2788.82	861.94
P4R105	3697.07	3721.18	0.65	T-UP	3009.37	1601.89	3709.58	0.34	T-UP	2963.25	1549.84	3725.34	0.76	T-UP	3009.38	1601.89
P4RC105	NA	NA	_	T-UP	2314.56	1444.11	NA	_	T-UP	2272.33	1411.08	NA	_	T-UP	2314.56	1444.11
P6C101	3989.48	3989.48	0.00	1527.72	3808.02	1162.04	**3989.48**	0.00	1212.13	3808.02	1096.97	3989.48	0.00	1913.72	3808.02	1162.04
P6R101	5393.19	5393.19	0.00	361.31	5337.68	1912.10	**5393.19**	0.00	195.38	5337.68	1856.92	5393.19	0.00	299.24	5337.68	1912.10
P6RC101	5834.62	5994.47	2.74	T-UP	4301.44	1854.79	**5834.62**	0.00	T-UP	4279.67	1764.04	5897.39	1.08	T-UP	4301.44	1854.79
P6C105	4059.25	**4059.25**	0.00	14309.01	3833.48	1064.40	4059.55	0.01	T-UP	3833.48	1048.12	4066.12	0.17	T-UP	3833.48	1064.40
P6R105	NA	NA	_	T-UP	3558.77	1790.65	NA	_	T-UP	3533.47	1764.35	NA	_	T-UP	3558.77	1790.65
P6RC105	NA	NA	_	T-UP	3281.73	1887.40	NA	_	T-UP	3248.26	1844.41	NA	_	T-UP	3281.74	1887.40

Instance	Best OFV	M2+(3.13, 3.20, 3.21, 3.46)					L2+(3.46)					L2+(3.20, 3.21, 3.28, 3.46)				
		OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP	OFV	Δ%	Time	Root	LP
P4C101	2913.94	2913.94	0.00	154.45	2910.87	871.21	2913.94	0.00	51.01	2910.87	1716.35	2913.94	0.00	106.23	2910.87	1754.96
P4R101	4162.33	4162.33	0.00	294.23	4109.89	1473.52	4162.33	0.00	330.01	4109.89	1773.92	4162.33	0.00	457.91	4109.89	1818.99
P4RC101	3983.30	4007.34	0.60	T-UP	3344.54	1408.82	3991.82	0.21	T-UP	3224.71	2050.50	3993.02	0.24	T-UP	3364.46	2097.42
P4C102	3079.89	NA	_	T-UP	1988.37	936.29	NA	_	T-UP	2097.20	1767.87	NA	_	T-UP	2115.49	1800.56
P4R102	3763.76	3849.59	2.28	T-UP	2511.96	1522.45	NA	_	T-UP	2606.24	1818.07	NA	_	T-UP	2615.08	1858.86
P4RC102	NA	NA	_	T-UP	2034.89	1448.52	NA	_	T-UP	2459.31	2144.50	NA	_	T-UP	2521.11	2178.38
P4C105	2889.92	2889.92	0.00	495.28	2830.80	895.16	2889.92	0.00	1114.65	2788.83	1742.67	2889.92	0.00	537.34	2836.83	1772.66
P4R105	3697.07	**3697.07**	0.00	T-UP	3112.99	1647.88	3703.55	0.18	T-UP	3001.13	1859.74	NA	_	T-UP	3112.73	1917.37
P4RC105	NA	NA	_	T-UP	2370.54	1500.26	NA	_	T-UP	2541.52	2086.72	NA	_	T-UP	2598.43	2121.59
P6C101	3989.48	3989.48	0.00	1372.20	3808.02	1212.92	3989.48	0.00	2605.38	3808.02	2302.67	3989.48	0.00	11476.51	3808.02	2368.90
P6R101	5393.19	5393.19	0.00	524.06	5337.68	1956.82	5393.19	0.00	600.44	5337.68	2332.18	5393.19	0.00	895.28	5337.68	2405.66
P6RC101	5834.62	5858.02	0.40	T-UP	4501.98	1894.13	NA	_	T-UP	4321.70	2524.85	NA	_	T-UP	4530.65	2605.76
P6C105	4059.25	4069.60	0.25	T-UP	3909.37	1118.57	**4059.25**	0.00	T-UP	3834.11	2333.17	4166.35	2.64	T-UP	3909.37	2364.74
P6R105	NA	NA	_	T-UP	3699.98	1843.93	NA	_	T-UP	3540.84	2224.57	NA	_	T-UP	3700.34	2272.00
P6RC105	NA	NA	_	T-UP	3411.82	1973.60	NA	_	T-UP	3550.28	2799.08	NA	_	T-UP	3637.76	2877.14

In data set S7, the customer coordinates are identical for all instances with the same type (i.e., R, C and RC). The instances differ with respect to the width of the time windows and time window density (TWD), that is, the percentage of customers with time windows. Among 45 instances in S7, we choose 15 instances, with TWD of 100% and 75%, which are easier to solve with the commercial solver. The corresponding instances are indexed with 01, 02 and 05 in each subset of instances C1, R1 and RC1. The information about the TWD and the average width of the time window (TWW) for those instances are reported in Table 3.11. For example, value 100% in the first column and the first row of the table shows the TWD of the instances containing C101 in their names.

Based on the results in Table 3.12, C2 only solves two instances out of 15: instance P4C101 is solved to optimality and instance P6C101 is solved to feasibility. None of the models return a feasible solution for the instances with 75% TWD. M2 and L2 solve nine and seven instances out of 15, respectively. More precisely, out of 12 instances with 100% TWD,

- M2 and L2 both reach the optimal solution for five instances. M2 identifies the optimal solutions more quickly than L2 in all instances except P6C101,
- M2 solves four instances to feasibility. L2 is not able to find a feasible solution for two of them, P4R105 and P6RC101. For P4RC101 and P6C105, M2 returns a better solution than L2 in the given time limit.

Replacing constraints (3.8) with valid inequalities (3.13) makes M2+(3.13) the only model to solve P6C105 to optimality. By adding valid inequalities (3.46) to M2 and L2,

- M2 improves the computation time in three instances overall, and finds a feasible solution for two of the three instances with 75% TWD;
- L2 improves the quality of the solution only for one instance (P4R105) overall and reaches the optimal solution for instance P6C105 although not being able to prove its optimality within the specified time limit.

According to Table 3.13, adding valid inequalities (3.46) does not help L2 in producing a feasible solution for instances with 75% TWD. Replacing constraints (3.8) with valid inequalities (3.13) and adding (3.46) to M2 simultaneously only improves the solution quality of instance P4R102. M2+(3.13, 3.20, 3.21, 3.46) also improves the solution quality of only one instance (P4R105) overall.

The results based on data sets S6 and S7 indicate that valid inequalities (3.46) do not affect LP relaxation quality, and the same objective function values of LP relaxations are observed between models M2 and M2+(3.46) and between models M2+(3.13) and M2+(3.13, 3.46), as well as L2 and L2+(3.46). Across data sets S6 and S7, model L2+(3.20, 3.21, 3.28, 3.46) consistently delivers the highest quality LP relaxation followed by model L2 (L2+(3.46)). For data set S7, the ranking follows as C2, M2+(3.13, 3.20, 3.21, 3.46), M2+(3.13) (M2+(3.13, 3.46)), and M2 (M2+(3.46)), respectively. For data set S6, models C2, M2+(3.13, 3.20, 3.21, 3.46) and M2+(3.13) (M2+(3.13, 3.46)) compete with each other, followed by model M2 (M2+(3.46)).

To extend the PVRP formulations M1 and L1 to model the PVRPTW, identical (time window) constraints are used. We observe that load-based formulations yield stronger linear relaxation bounds in comparison with the MTZ-based formulations in the case of the PVRPTW. Furthermore, despite having weaker linear relaxation bounds, MTZ-based formulations produce mostly better results in terms of the objective function value and run time than load-based formulations in PVRPTW instances. This may be attributed to the fact that MTZ-based formulations, i.e., M2 and its extensions, have a smaller number of variables in comparison with the load-based formulations, L2 and its extensions.

3.3 Conclusions

In this study, we overview the PVRP focusing on modeling approaches and exact solutions to be obtained by a commercial solver. We develop a new MILP formulation of the PVRP using vehicle flow variables and employ families of valid inequalities and optimality cuts to tighten the formulation. Two existing prominent PVRP formulations in the literature referred to as the load-based formulation and the cut-based formulation are also investigated. The main differences among the formulations are the representation of vehicle information and the SECs. The cut-based formulations use decision variables explicitly including a vehicle index. The SECs of the cut-based formulations have an exponential size and need to be separated. In the load-based formulations and the MTZ-based formulations, auxiliary continuous variables are used to define the constraints preventing the formation of subtours and both formulations are compact, i.e., they do not involve exponentially many constraints. Since the continuous variables defined in the MTZ-based formulation have one dimension less than those used in the load-based formulation, the MTZ-based formulation has smaller number of variables compared to the load-based formulation. We also extend these formulations to model the PVRPTW and apply families of valid inequalities to tighten the formulations. Comprehensive computational experiments are conducted on seven sets of benchmark instances with different characteristics to compare the performances of the alternative formulations.

The results of the computational experiments on the PVRP data sets show that the cut-based model and the load-based model are competitive in solving small instances. When it comes to solving medium instances, the load-based model overtakes the cut-based model. While the results indicate the difficulty of solving most of the PVRP instances of large size for the cut-based model and the load-based model, the MTZ-based formulations are robust and consistent in producing good-quality solutions.

Considering the PVRPTW, the results obtained by alternative models with data sets S6 and S7 show that the cut-based model is only successful in solving two out of 35 instances. In general, the MTZ-based formulations outperform the load-based formulations. For the instances solved to optimality, they reach the solution faster, and for those instances for which an optimal solution cannot be identified within the specified time limit, they return solutions of higher quality when compared to the load-based formulations. The superiority of the MTZ-based formulations are more evident in solving larger instances.

The goal of this research is to compare and provide insights into the computational performance of widely used formulation approaches in the literature that can be implemented and solved directly by state-of-the-art commercial solver capabilities. Other formulation approaches are available (e.g. set partitioning) for which an exact solution cannot be obtained directly by a solver, but only by employing efficient optimization algorithms (e.g. column generation and branch-and-price). Extending the computational analysis to cover a broader spectrum of exact solution methodologies may provide additional insights into the performance of different formulation and solution techniques with respect to run times and solution quality. Moreover, (parts of) the formulation approaches considered in this study are easily applicable to other variants of the PVRP and exploring the computational performance on different problem variants may also be an interesting direction for future research.

4. Optimization Methodologies for PVRP

In this chapter, we tackle the PVRP using two state-of-the-art methodologies. Section 4.1 details the application of the LBBD approach to solving the PVRP, while Section 4.2 explores the use of the column generation approach for the same problem.

4.1 LBBD Approach in Solving PVRP

Benders decomposition (Benders, 1962) was initially developed for MIP problems, utilizing projection, outer linearization, and relaxation techniques to transform the problem into a more tractable form. This method divides an MIP problem into two more manageable problems: the master problem, a relaxed version of the original that includes the integer variables, and the subproblem, a linear program parameterized by the temporarily fixed integer variables from the master problem. During each iteration of the Benders method, the master problem is solved, and its solution is used to define the subproblems. The subproblem is then dualized, with its solution providing coefficients for inequalities known as Benders optimality cuts or feasibility cuts. If the dual subproblem is optimal, it generates an optimality cut; if it is infeasible, a feasibility cut is produced. This method converges to the optimal solution as long as the Benders master problem solution continues to improve the lower bound by solving to optimality and excluding the current master problem solution through added cuts. The process repeats until an optimal solution to the original problem is found or an early stopping criterion is met. In the latter case, the solution from the master problem combined with a feasible subproblem solution yields a globally feasible MIP solution, while the optimally solved master problem provides a lower bound for the original objective function value. Benders decomposition is effective when it concludes well before reaching the total number of potential master problem solutions.

A modified version of UC1 is well-suited for the application of Benders decomposition, resulting in a master problem resembling a generalized assignment problem (GAP), where customers are assigned to days, and independent VRP subproblems. This approach, known as cluster-first and route-second, generates MIP subproblems instead of linear ones, rendering the classical Benders method inapplicable. Instead, LBBD, developed by Hooker (2000) and Hooker and Ottosson (2003), is used to handle a broader range of optimization problems, including those with Benders' subproblems containing one or more sets of integer variables.

For the sake of completeness, we bring the expansion of UC1 formulation here;

$$\begin{align}
&\text{minimize} \quad \sum_{e\in E}\sum_{t\in T}\sum_{k\in K} c_e x_{etk}, && \tag{4.1}\\
&\text{subject to} \quad \sum_{p\in P_i} s_{ip} = 1 && \forall i \in N_c, \tag{4.2}\\
&\sum_{p\in P_i : t\in p} s_{ip} = \sum_{k\in K} z_{itk} && \forall i \in N_c, \forall t \in T, \tag{4.3}\\
&\sum_{i\in N_c} z_{itk}\, d_i \le Q v_{tk} && \forall k \in K, \forall t \in T, \tag{4.4}\\
&x_{tk}(\delta(i)) = 2z_{itk} && \forall i \in N, \forall k \in K, \forall t \in T, \tag{4.5}\\
&x_{tk}(\delta(S)) \ge 2z_{itk} && \forall S \subseteq N_c, \forall i \in S, \forall k \in K, \forall t \in T, \tag{4.6}\\
&s_{ip} \in \{0,1\} && \forall i \in N_c, \forall p \in P_i, \tag{4.7}\\
&y_{itk} \in \{0,1\} && \forall i \in N_c, \forall k \in K, \forall t \in T, \tag{4.8}\\
&v_{tk} \in \{0,1\} && \forall k \in K, \forall t \in T. \tag{4.9}\\
&x_{etk} \in \{0,1\} && \forall e \in E \backslash \{\{0,j\} : j \in N_c\}, \forall k \in K, \forall t \in T, \tag{4.10}\\
&x_{0jtk} \in \{0,1,2\} && \forall \{0,j\}, j \in N_c, \forall k \in K, \forall t \in T. \tag{4.11}
\end{align}$$

LBBD generalizes the classical Benders decomposition by similarly dividing a problem into a master problem and subproblems. Unlike the classical Benders method, which uses linear programming duality to derive cuts from subproblems, LBBD employs the more general inference dual. The inference dual solution provides a bounding function on the cost of the master problem that is tight for its current solution. However, unlike classical Benders decomposition, there is no standardized procedure for deriving these cuts in LBBD. Instead, cuts must be developed specific to the problem.

4.1.1 Generalized Assignment Master Problem

Let σ_t represent the lower bound on the routing cost for each period $t \in T$, constrained by the Benders cuts. The proposed LBBD addresses the following GAP master problem:

$$
\begin{aligned}
&(4.12) && \text{minimize} && \sum_{t \in T} \sigma_t, && \\
&(4.13) && \text{subject to} && \sum_{p \in P_i} s_{ip} = 1 && \forall i \in N_c, \\
&(4.14) && && \sum_{p \in P_i : t \in p} s_{ip} = y_{it} && \forall i \in N_c, \forall t \in T, \\
&(4.15) && && \sum_{i \in N_c} y_{it} d_i \leq KQ y_{0t} && \forall t \in T, \\
&(4.16) && && \sigma_t \geq \text{bounding functions of optimality cuts} && \forall t \in T, \\
&(4.17) && && \text{feasibility cuts}, && \\
&(4.18) && && s_{ip} \in \{0, 1\} && \forall i \in N_c, \forall p \in P_i, \\
&(4.19) && && y_{it} \in \{0, 1\} && \forall i \in N, \forall t \in T,
\end{aligned}
$$

Here, constraints (4.16) and (4.17) consist of the optimality and feasibility cuts derived in Section 4.1.3.

4.1.2 Vehicle Routing Sub-Problems

For each period $t \in T$, let $\bar{H}(t)$ represent the clusters of nodes assigned to day t by the solution of the master problem (4.12)–(4.19). For each day t with $\bar{H}_{(t)} \neq \emptyset$, the following independent VRP sub-problems originate from fixed values of y variables:

$$
\begin{aligned}
&(4.20) \qquad && \text{minimize} \quad && \sum_{e\in E} c_e x_{ek}^{(t)}, && \\
&(4.21) && \text{subject to} && \sum_{k\in K} z_{ik}^{(t)} = 1 && \forall i \in \bar{H}_{(t)}\backslash\{0\}, \\
&(4.22) && && x_k^{(t)}(\delta(i)) = 2z_{ik}^{(t)} && \forall i \in \bar{H}_{(t)}, \forall k \in K \\
&(4.23) && && x_k^{(t)}(\delta(S)) \geq 2z_{ik}^{(t)} && \forall S \subseteq \bar{H}_{(t)}, \forall i \in S, \forall k \in K, \\
&(4.24) && && \sum_{i\in \bar{H}_{(t)}} z_{ik}^{(t)}\, d_i \leq Q v_k && \forall k \in K, \\
&(4.25) && && x_{ek}^{(t)} \in \{0,1\} && \forall e \in E\backslash\{\{0,j\} : j \in \bar{H}_{(t)}\}, \forall k \in K, \\
&(4.26) && && x_{0jk}^{(t)} \in \{0,1,2\} && \forall \{0,j\}, j \in \bar{H}_{(t)}, \forall k \in K. \\
&(4.27) && && z_{ik}^{(t)} \in \{0,1\} && \forall i \in \bar{H}_{(t)}, \forall k \in K \\
&(4.28) && && v_k^{(t)} \in \{0,1\} && \forall k \in K
\end{aligned}
$$

The objective function (4.20) minimizes the total routing cost. Constraints (4.21) assign each customer to a vehicle. Constraints (4.22) are the degree constraints for the depot and the customers. Constraints (4.23) prevent subtours. Constraints (4.24) are the capacity constraints. Variable domain restrictions are given by (4.25) to (4.28).

4.1.3 Benders Cuts

The solution of each independent VRP sub-problem produces either an optimality cut or a feasibility cut, which is then incorporated into the GAP master problem. These cuts are described as follows.

4.1.3.1 Optimality Cut

Let $C^{(t)}$ denote the optimal cost for a VRP solution associated with cluster $\bar{H}_{(t)}$. We apply the following optimality cuts for the Benders master problem

$$\sigma_t \geq C^{(t)}\Big(\sum_{i\in\bar{H}_{(t)}} y_{it} - |\bar{H}_{(t)}| + 1\Big). \tag{4.29}$$

where $C(t)$ is the optimal objective function value of the t^{th} subproblem, and $|\bar{H}_{(t)}|$ is the number of customers assigned to day t. It means that if exactly the same set of customers are assigned to day t, the cost of this particular subproblem will be at least $C(t)$. As a result, the solver should avoid allocating the same set of customers to day t to obtain less cost. The optimality cut applied is the adapted optimality cut from Riazi et al. (2013).

4.1.3.2 Feasibility Cut

Whenever a cluster $\bar{H}_{(t)}$ assigned to day t yields an infeasible VRP subproblem, the inequality

$$\sum_{i\in\bar{H}_{(t)}} (1 - y_{it}) \geq 1 \tag{4.30}$$

is a feasibility cut for the master problem.

4.1.4 LBBD implementation

The LBBD algorithm builds upon the classical Benders method as outlined in Algorithm 1, where itr represents the iteration counter, LB is the lower bound, and UB is the upper bound on the optimal cost for the PVRP (lines 1-3). The iterative process is detailed in lines 4-21. Initially, the solution to the master problem assigns values y^*_{itr} to the variables y, forms the customer clusters $\bar{H}_{(t)}, t = 1, \ldots, |T|$, and updates the lower bound LB, as this problem is a relaxation of the PVRP (lines 4-5). Subsequently, the VRP subproblems are solved (lines 7-18), producing optimality (line 12) and/or feasibility cuts (line 15), and potentially updating the upper bound UB (line 19). The loop from lines 4-21 continues until the optimality gap $(UB - LB)/UB$ is less than the specified tolerance, ϵ.

Algorithm 1 LBBD

$itr \leftarrow 0$
$LB \leftarrow 0$
$UB \leftarrow \infty$
while $(UB - LB)/UB > \epsilon$ **do**
    Solve the master problem to optimality, obtaining $y^*_{(itr)} and\ \bar{H}_{(t)}$
    $LB \leftarrow \sum_{t \in T} \sigma_t^{(itr)}$
    for $(t);\ \ t \in T$ **do**
        if $\bar{H}_{(t)} \neq \emptyset$ **then**
            Solve the VRP sub-problem associated with cluster $\bar{H}_{(t)}$
            if the VRP sub-problem is feasible **then**
                $C(t) \leftarrow$ VRP optimal cost
                Add a new optimality cut (4.29) to the master problem
            else
                $C(t) \leftarrow \infty$
                Add a new feasibility cut (4.30) to the master problem
            end if
        end if
    end for
    $UB \leftarrow \min(UB, \sum_t \bar{C}(t))$
    $itr \leftarrow itr + 1$
end while

4.1.4.1 Branch and check

Algorithm 2 describes a Branch and Check (BAC) approach for solving the PVRP. This method involves solving a single master problem using a general branch-and-cut MIP solver. A callback function is utilized to handle each potentially feasible solution $\bar{y}$ identified by the master problem, which is then used to formulate the sub-problems. This leverages the functionality of modern MIP solvers that allow user intervention during the solution process through incumbent callbacks. Such callbacks, known in Gurobi Optimization as MIPSOL callbacks are triggered whenever a new potential incumbent solution is discovered.

If the inequality $\sum_{t \in T} \bar{C}^{(t)} \leq \sum_{t \in T} \sigma_t$ holds true for the optimal solutions $\bar{C}^{(t)}$ of the subproblems for all $t \in T$, then a globally feasible solution for the PVRP has been identified. Otherwise, Benders cuts generated from the subproblems are added to the master problem as lazy constraints, thereby excluding the current solution $\bar{y}$ from the feasible solution space.

Algorithm 2 BAC

Solve the master problem using an MIP solver
for all potential incumbent solution $\bar{y}$ with cost $\sum_{t \in T} \sigma_t$ **do**
    Obtain the associated clusters $\bar{H}_{(t)}, \forall t \in T$
    for (t); $t \in T$ **do**
        if $\bar{H}_{(t)} \neq \emptyset$ **then**
            Solve the VRP sub-problem associated with cluster $\bar{H}_{(t)}$
            if the VRP sub-problem is feasible **then**
                $C(t) \leftarrow$ VRP optimal cost
                Generate a new optimality cut (4.29) to the master problem
            else
                $C(t) \leftarrow \infty$
                Generate a new feasibility cut (4.30) to the master problem
            end if
        end if
    end for
    if $\sum_{t \in T} \bar{C}(t) \leq \sum_{t \in T} \sigma_t$ **then**
        Accept the current globally feasible solution $\bar{y}$
    else
        Add the generated Benders cuts to the master problem excluding the
        current solution $\bar{y}$
    end if
end for

4.1.5 Computational Results of the LBBD Algorithms

To evaluate the performance of the proposed algorithms, we utilize dataset S5. This dataset categorizes instances based on the number of customers and sorts them according to the planning horizon length. We selected dataset S5 because it allows us to begin with the category containing 11 customers and progress until the algorithms demonstrate efficient performance. Each algorithm is executed with a 2-hour time limit. All computational experiments are performed on a virtual machine equipped with an Intel Xeon CPU E5-2640 processor running at 2.60 GHz, 16 GB of RAM, and a 64-bit operating system. The Gurobi Optimizer 8.1.1, integrated with Python

3.7.4, is used as the commercial solver.

Table 4.1 provides a comparative analysis of the BAC and LBBD algorithms applied to dataset S5. The comparison includes various metrics such as "OFV" (best obtained objective function value), "Gap" (reported gap by Gurobi), "RunTime" (computing time), "$\#Opt-cut$" (number of generated optimality cuts), "Average VRP Time" (average time to solve each VRP subproblem), "Total VRP Time" (total computing time dedicated to solving VRPs), "$\#VRP$" (number of solved VRPs), "$\#callback$" (number of times the callback function is called), "RunTime-Total VRP Time" (remaining computing time after extracting total VRP time), "UB" (upper bound), "LB" (lower bound), "$\#it$" (number of iterations), "Total Master Time" (total time spent solving master problems), and "Average Master Time" (average master solution time).

According to Table 4.1, as complexity increases with more customers and longer planning horizons, the BAC algorithm generally outperforms LBBD, regarding the solution time. For instance, in test21-4-3-a-Q71, BAC completes in 2047.64 seconds compared to LBBD's 6050.05 seconds. Consequently, we proceed with the BAC algorithm and evaluate its performance by comparing it with model UC1.

Table 4.2 presents the results of UC1 and BAC for selected instances in S5. The first column lists the instance names. The following two columns display the best objective function value obtained by UC1 and its corresponding solution time. The next block of eight columns provides detailed information about the BAC implementation. Instances with up to 30 customers are solved within a 7200-second time limit, while instances with 40 customers are given a 14400-second limit. BAC was executed without any algorithmic enhancements. According to the BAC results, convergence to an optimal solution is slow. Furthermore, as the problem size increases, the time required to solve the VRP subproblem also increases, rendering the algorithm inefficient.

Table 4.1 LBBD versus BAC

Instance	BAC									LBBD								
	OFV	Gap	RunTime	# Opt_cut	Avgerage VRP Time	Total VRP Time	# VRP	# callback	Run Time - Total VRP Time	UB	LB	Run Time	Avgerage VRP Time	Total VRP Time	# VRP	# it	Total Master time	Avgerage Master Time
test11-2-2-a-Q51	620,48	0	0,49	14	0,03	0,41	14	7	0,08	620,48	620,48	0,26	0,02	0,23	10	5	0,024	0,005
test11-2-3-a-Q51	620,48	0	0,99	14	0,07	0,94	14	7	0,06	620,48	620,48	0,51	0,05	0,49	10	5	0,018	0,004
test11-3-2-a-Q51	834,96	0	2,97	81	0,03	2,76	81	27	0,21	834,96	834,96	1,84	0,02	1,70	69	23	0,103	0,004
test11-3-3-a-Q51	834,96	0	4,72	78	0,06	4,53	78	26	0,19	834,96	834,96	3,44	0,05	3,29	69	23	0,105	0,005
test11-4-2-a-Q51	1043,91	0	3,42	156	0,02	3,16	156	39	0,26	1043,91	1043,91	2,31	0,02	2,05	136	34	0,199	0,006
test11-4-3-a-Q51	1043,91	0	6,21	156	0,04	5,94	156	39	0,27	1043,91	1043,91	4,37	0,03	4,07	136	34	0,215	0,006
test11-5-2-a-Q51	1083,97	0	34,31	1300	0,02	30,44	1300	260	3,87	1083,97	1083,97	83,71	0,01	16,40	1335	267	66,631	0,250
test11-5-3-a-Q51	1083,97	0	51,34	1170	0,04	47,19	1170	234	4,14	1083,97	1083,97	91,96	0,02	23,10	1405	281	68,188	0,243
test21-2-2-a-Q71	828,98	0	7,26	10	0,71	7,07	10	5	0,19	828,98	828,98	1,04	0,13	1,01	8	4	0,018	0,004
test21-2-3-a-Q71	828,98	0	16,88	14	1,19	16,68	14	7	0,20	828,98	828,98	2,74	0,27	2,69	10	5	0,024	0,005
test21-2-4-a-Q71	828,98	0	20,63	14	1,46	20,43	14	7	0,20	828,98	828,98	5,44	0,54	5,40	10	5	0,025	0,005
test21-3-2-a-Q71	1088,36	0	94,43	267	0,35	92,64	267	89	1,79	1088,36	1088,36	21,96	0,08	20,83	255	85	0,882	0,010
test21-3-3-a-Q71	1088,36	0	146,53	255	0,57	145,09	255	85	1,44	1088,36	1088,36	34,86	0,13	33,83	252	84	0,782	0,009
test21-3-4-a-Q71	1088,36	0	232,19	273	0,84	230,51	273	91	1,68	1088,36	1088,36	62,54	0,24	61,30	255	85	0,934	0,011
test21-4-2-a-Q71	1253,47	0	874,89	3568	0,24	840,11	3568	892	34,77	1253,47	1253,47	2068,56	0,06	601,90	9568	2392	1456,518	0,609
test21-4-3-a-Q71	1253,47	0	2047,64	4380	0,46	2004,83	4380	1095	42,81	1253,47	1253,47	6050,05	0,13	1842,35	14044	3511	4189,595	1,193
test21-4-4-a-Q71	1253,47	0	3117,08	4056	0,76	3080,22	4056	1014	36,86	1253,47	1253,47	6277,35	0,20	2814,19	14164	3541	3443,654	0,973
test31-5-2-a-Q80	1775,57	1	7222,39	7630	0,93	7071,72	7630	1526	150,67	1737,06	0	7207,36	0,60	4478,60	7465	1493	2695,302	1,805

Therefore, to expedite the algorithm, we attempted to enhance the LB by incorporating cuts derived from the dual of the LP relaxation of the subproblem at each iteration. However, this approach did not yield any significant improvement in the LB and the cuts appeared to be redundant. we also aimed to enhance the speed of solving the VRP subproblem. Initially, we applied an exact column generation algorithm for this purpose. However, our proposed column generation approach did not yield significant improvements in solution time. Consequently, we opted to refine and adapt it specifically for the PVRP problem, aiming to solve the PVRP directly using the column generation algorithm.

Table 4.2 Comparison of results obtained by UC1 and BAC

Instance	UC1		BAC							
	OFV	RunTime (s)	OFV	Gap	RunTime (s)	# Opt_cut	# feas_cut	Avg VRP Time	Total VRP time	# callback
test11-2-2-a-Q51	620,48	0,97	620,48	0	**0,49**	14	0	0,03	0,41	7
test11-2-3-a-Q51	620,48	1,73	620,48	0	**0,99**	14	0	0,07	0,94	7
test11-3-2-a-Q51	834,96	1,56	834,96	0	2,97	81	0	0,03	2,76	27
test11-3-3-a-Q51	834,96	8,22	834,96	0	**4,72**	78	0	0,06	4,53	26
test11-4-2-a-Q51	1043,91	1,08	1043,91	0	3,42	156	0	0,02	3,16	39
test11-4-3-a-Q51	1043,91	2,58	1043,91	0	6,21	156	0	0,04	5,94	39
test11-5-2-a-Q51	1083,97	7,84	1083,97	0	34,31	1300	0	0,02	30,44	260
test11-5-3-a-Q51	1083,97	99,11	1083,97	0	**51,34**	1170	0	0,04	47,19	234
test21-2-2-a-Q71	828,98	3,50	828,98	0	7,26	10	0	0,71	7,07	5
test21-2-3-a-Q71	828,98	2,77	828,98	0	16,88	14	0	1,19	16,68	7
test21-2-4-a-Q71	828,98	34,81	828,98	0	**20,63**	14	0	1,46	20,43	7
test21-3-2-a-Q71	1088,36	4,52	1088,36	0	94,43	267	0	0,35	92,64	89
test21-3-3-a-Q71	1088,36	42,84	1088,36	0	146,53	255	0	0,57	145,09	85
test21-3-4-a-Q71	1088,36	106,75	1088,36	0	232,19	273	0	0,84	230,51	91
test21-4-2-a-Q71	1253,47	75,94	1253,47	0	874,89	3568	0	0,24	840,11	892
test21-4-3-a-Q71	1253,47	333,03	1253,47	0	2047,64	4380	0	0,46	2004,83	1095
test21-4-4-a-Q71	1253,47	7200,28	1253,47	0	**3117,08**	4056	0	0,76	3080,22	1014
test21-5-2-a-Q71	1729,00	1274,06	1729,00	0	5455,50	4230	0	0,47	1995,94	846
test21-5-3-a-Q71	1729,00	7200,33	1729,00	0,01	7200,11	3930	0	0,92	3605,06	786
test21-5-4-a-Q71	1729,00	7200,42	1729,19	0,03	7200,08	3645	0	1,53	5591,05	729
test31-2-2-a-Q80	841,63	15,02	841,63	0	67,92	28	0	2,40	67,33	14
test31-2-3-a-Q64	917,28	1735,72	917,28	0	3473,66	340	0	10,21	3469,77	170
test31-2-4-a-Q64	911,89	7200,34	915,76	0,49	7441,98	274	0	27,15	7438,55	137
test31-3-2-a-Q80	1190,97	69,66	1194,00	0,33	7234,07	5352	0	1,34	7167,08	1784
test31-3-3-a-Q64	1285,54	7200,36	1308,86	0,67	7208,20	1719	0	4,18	7190,13	573
test31-3-4-a-Q64	1288,27	7200,55	1348,55	0,68	7777,21	495	0	15,70	7770,73	165
test31-4-2-a-Q80	1625,85	7200,41	1626,00	0,73	7257,50	5032	0	1,43	7183,47	1258
test31-4-3-a-Q64	1742,31	7200,55	1785,83	0,76	7210,22	1168	0	6,16	7191,30	292
test31-4-4-a-Q64	1752,70	7200,63	1803,94	0,88	7549,22	452	0	16,69	7542,50	113
test31-5-2-a-Q80	1673,06	7200,58	1775,57	1	7222,39	7630	0	0,93	7071,72	1526
test31-5-3-a-Q64	1780,61	7200,75	1885,27	1	7327,73	2705	0	2,70	7294,22	541
test31-5-4-a-Q64	1809,47	7200,73	1909,31	1	7259,48	1250	0	5,80	7245,22	250
test41-2-2-a-Q180	871,32	162,58	871,32	0	11068,60	2252	0	4,90	11029,60	1126
test41-2-3-a-Q150	889,32	3329,77	889,76	0,52	14837,66	636	0	23,31	14824,23	318
test41-2-4-a-Q60	1089,37	14400,63	_	_	_	15	0	2016.66	30249.98	8
test41-3-2-a-Q180	1185,19	454,86	1247,09	1	14439,33	4149	0	3,46	14339,64	1383
test41-3-3-a-Q150	1214,05	14400,59	1267,27	1	14658,58	939	0	15,59	14643,01	313
test41-3-4-a-Q60	1542,96	14400,96	_	_	_	12	0	1604.34	19252,09	4
test41-4-2-a-Q180	1460,46	6959,16	1527,26	0,78	14446,12	6168	0	2,31	14270,43	1542
test41-4-3-a-Q150	1476,33	14401,06	1606,61	1	14478,00	1684	0	8,58	14441,42	421

4.2 Heuristic Column Generation Approach in Solving Periodic Vehicle

4.2.1 Problem Definition and Formulation

PVRP can be defined on a complete undirected graph $G = (N, E)$ with $N = \{0, 1, \ldots, n\}$ being the set of nodes, where node 0 represents the depot and the nodes in the set $N_c = \{1, \ldots, n\}$ correspond to the customers. The set of edges is given by $E = \{e \subset N : |e| = 2\}$. Each edge e is associated with a non-negative cost c_e. Let $T = \{1, \ldots, \tau\}$ denote the set of time periods defining the planning horizon. A fleet of homogeneous vehicles $K = \{1, \ldots, \kappa\}$, each having a capacity of Q units, is available to serve the customers. Associated with every customer $i \in N_c$ is a demand d_i for each visit, and a predefined set of possible visit schedules P_i. A given schedule $p \in P_i$ consists of the specific days on which the customer should be visited, i.e., $p \subseteq T$.

Let Ω be the set of all feasible routes, which is exponentially large. For each route $r \in \Omega$ its cost is ζ_r. Let also α_{ir} and β_{pt} be binary constants and indicate whether customer i is visited by route r and whether day t is in combination $p \in P_i$, respectively.

The following decision variables are used in the integer master problem (IMP) for the PVRP; x_{rt} is a binary variable which is equal to 1 if route r is traversed on day t, otherwise 0, s_{ip} is a binary variable which is equal to 1 if combination $p \in P_i$ is chosen for visiting customer $i \in N_c$, and 0 otherwise.

We formulate MIP for the PVRP as a set-covering model as follows:

$$\text{minimize} \quad \sum_{r\in\Omega}\sum_{t\in T}\zeta_r x_{rt}, \tag{4.31}$$

$$\text{subject to} \quad \sum_{p\in P_i} s_{ip} \geq 1 \qquad \forall i \in N_c, \tag{4.32}$$

$$\sum_{r\in\Omega}\alpha_{ir}x_{rt} - \sum_{p\in P_i}\beta_{pt}s_{ip} \geq 0 \qquad \forall i \in N_c, \forall t \in T, \tag{4.33}$$

$$\sum_{r\in\Omega} x_{rt} \leq \kappa \qquad \forall t \in T, \tag{4.34}$$

$$x_{rt} \in \{0,1\} \qquad \forall r \in \Omega, \forall t \in T, \tag{4.35}$$

$$s_{ip} \in \{0,1\} \qquad \forall i \in N_c, \forall p \in P_i. \tag{4.36}$$

The objective function (4.31) minimizes the total routing cost. Constraints (4.32) ensure that at least one allowable visit schedule is selected for each customer. Constraints (4.33), which relate the routes and the visit combinations, ensure that every customer is visited by at least one route on each day belonging to its allowable visit schedules. Constraints (4.34) guarantee that the number of vehicles that can be used on any day of the planning horizon cannot exceed the fleet size κ. Constraints (4.35) and (4.36) specify the domain restrictions on the variables.

4.2.2 Column Generation

As there are exponentially many variables (columns) corresponding to routes, it is not tractable to solve the set covering problem (4.31-4.36) directly. Therefore, we use a Column generation technique based on the following principle. We introduced the Restricted Master Problem (RMP) as the linear relaxation of (4.31-4.36) in which the initial columns set is $\Omega' \subseteq \Omega$.

$$
\begin{array}{llll}
(4.37) & \text{minimize} & \sum_{r \in \Omega'} \sum_{t \in T} \zeta_r x_{rt}, & \\
(4.38) & \text{subject to} & \sum_{p \in P_i} s_{ip} \geq 1 & \forall i \in N_c, \\
(4.39) & & \sum_{r \in \Omega'} \alpha_{ir} x_{rt} - \sum_{p \in P_i} \beta_{pt} s_{ip} \geq 0 & \forall i \in N_c, \forall t \in T, \\
(4.40) & & \sum_{r \in \Omega'} x_{rt} \leq \kappa & \forall t \in T, \\
(4.41) & & x_{rt} \geq 0 & \forall r \in \Omega', \forall t \in T, \\
(4.42) & & s_{ip} \geq 0 & \forall i \in N_c, \forall p \in P_i.
\end{array}
$$

Let D-RMP(Ω') be the dual program of the RMP(Ω'):

$$\text{(4.43)} \qquad \text{maximize} \quad \sum_{i\in C}\alpha_i + \sum_{t\in T}\kappa\lambda_{0t},$$

$$\text{(4.44)} \qquad \text{subject to} \quad \alpha_i - \sum_{t\in T}\beta_{p't}\lambda_{it} \leq 0 \qquad \forall i \in N_c, \forall p' \in P_i,$$

$$\text{(4.45)} \qquad \sum_{i\in N_c}\alpha_{ir}\lambda_{it} + \lambda_{0t} \leq \zeta_r \quad \forall r \in \Omega', \forall t \in T,$$

$$\text{(4.46)} \qquad \alpha_i \geq 0 \qquad \forall i \in N_c,,$$

$$\text{(4.47)} \qquad \lambda_{it} \geq 0 \qquad \forall i \in N_c, \forall t \in T,$$

$$\text{(4.48)} \qquad \lambda_{0t} \leq 0 \qquad \forall t \in T,$$

in which α_i, $\forall i \in N_c$ are the non-negative dual variables associated to constraints (4.38), λ_{it}, $\forall i \in N_c, \forall t \in T$ are the non-negative dual variables associated to constraints (4.39), and λ_{0t}, $\forall t \in T$ are the non-positive dual variables associated to constraints (4.40).

4.2.3 Pricing Subproblem

The subproblem aims to find routes $r \in \Omega \setminus \Omega'$ such that

$$\text{(4.49)} \qquad \zeta_r - \sum_{i\in N_c}\alpha_{ir}\lambda^*_{it} - \lambda^*_{0t} \leq 0.$$

In other words, this condition is:

$$\text{(4.50)} \qquad \sum_{(i,j)\in A} b_{ijrt}(c_{ij} - \lambda^*_{it}) \leq 0,$$

where $b_{ijrt} = 1$ if on day t route r uses arc (i,j), otherwise 0. Inequality (4.50) is equivalent to

$$\text{(4.51)} \qquad \sum_{(i,j)\in A} b_{ijrt}(c_{ij} - \frac{\lambda^*_{it} + \lambda^*_{jt}}{2}) \leq 0.$$

According to inequality (4.51) the subproblem reduces to an elementary shortest-path problem with resource constraints (ESPPRC). The ESPPRC holds for each day $t \in T$. One has to find a path without cycles from the depot to the depot, respecting the maximum vehicle capacity limit with a negative cost, where costs are defined as

$$\hat{C}_{ijt} = c_{ij} - \frac{\lambda^*_{it} + \lambda^*_{jt}}{2} \qquad \forall i \in N, \forall j \in N, \forall t \in T. \tag{4.52}$$

We solve the ESPPRC using a dynamic programming-based approach, ng-route decremental state-space relaxation (DSSR), and completion bounds (Martinelli et al., 2014). In DP-based methods, new paths, represented by labels, are constructed from the depot to its duplicate. For each customer $i \in N_c$, let $N_i \subseteq N_c$ be a subset of customers associated with i. This association can be defined by proximity, meaning N_i includes the nearest customers to i, including i itself. These subsets are known as ng-sets, representing the customers that customer i can "remember." The size of each ng-set N_i is constrained by $\Delta(N_i)$, a predefined parameter.

Consider a path $P = (0, i_1, \ldots, i_{p-1}, i_p)$. The components of a label $L(P) = (i_p, d(P), \Pi_p, C_p)$ corresponding to path P are defined as follows:

- i_p: the last customer assigned to path P,
- $d(P)$: the total demand serviced by path P,
- Π_p: the prohibited extensions (the "memory") of path P, defined as

$$\Pi_p = \{i_k \in C(P) \setminus \{i_p\} : i_k \in \bigcap_{s=k+1}^{p} N_{i_s}\} \cup \{i_p\} \tag{4.53}$$

where

- $C(P)$ is the set of customers visited by path P,
- C_p: the total cost of path P.

A label $L(P)$ can be extended to include a customer i_{p+1} if $i_{p+1} \notin \Pi_p$ and $d(P) + d_{i_{p+1}} \leq Q$. When extended, i_{p+1} becomes the last customer in the new path $P' = (0, i_1, \ldots, i_p, i_{p+1})$, and the new label $L(P')$ is derived from $L(P)$ through the following updates:

$$L(P') = (i_{p+1},\ d(P) + d_{i_{p+1}},\ \Pi_p \cap N_{i_{p+1}} \cup \{i_{p+1}\},\ C_p + C_{i_p i_{p+1}}) \tag{4.54}$$

The adapted DSSR is an iterative algorithm that initially relaxes the state space of the original ng-sets N_i. During each iteration k, it replaces N_i with subsets $\Gamma_i^k \subseteq N_c$. These subsets Γ_i^k are used in place of N_i in the definition from Eq. 4.53 and in creating new labels as specified in Eq. 4.54. Initially, the algorithm starts with Γ_i^0 as an empty set and runs the dynamic programming algorithm. Since the optimal routes identified by this process may not be ng-routes with respect to the original ng-sets N_i, their feasibility must be verified before accepting them as the pricing solution. This verification is conducted, and the subsets Γ_i^k are updated if necessary. If any subset Γ_i^k is modified at the end of iteration k, the dynamic programming algorithm is rerun with the updated subsets Γ_i^{k+1}.

DSSR and the ng-route relaxation aim to keep this set as small as possible without losing the elementarity of the routes. DSSR ensures elementarity, whereas ng-route relaxation does not guarantee it.

To preserve elementarity, Martinelli et al. (2014) propose a modified approach where the sets, though functioning similarly, are not strictly subsets of the original neighborhoods. The update procedure is triggered whenever a cycle is detected. This method does not utilize ng-routes, resulting exclusively in elementary routes.

In another study, Dayarian et al. (2015) addresses the prevention of non-elementary optimal paths while employing ng-routes differently. If the algorithm identifies a non-elementary ng-route, it does not allow it to terminate conventionally; instead, an additional layer of DSSR is applied. Consequently, any repeating vertex i within a valid ng-cycle is designated as critical and incorporated into all applicable neighborhoods Γ_j, $j \in N$ as if $j \in N_j$. This supplementary step ensures the elementarity of all columns, thereby enhancing the lower bound. Algorithm 1 employs this strategy to uphold elementarity.

As we observed, when the Δ (the size of each ng-set) is small, the size of Γ_i, $i \in N$ increases rapidly. So, we extend Γ only according to invalid ng-cycles and keep critical customers in set S. Set S are the customers for which a customer-visit resource is maintained. We start with $S = \emptyset$ and update S at the end of each label setting iteration that returns a non-elementary ng-route. Suppose that the state corresponding to a given path p is described by the label vector $U = (U_1, \ldots, U_{|S|})$. if c_k is the k^{th} customer added to S, then U_k shows whether customer c_k is included in the path p or not. We also have to respect these limits while constructing resource-feasible paths. We refer to this approach as a hybrid NG-customer-visit resource approach and use it in Algorithm 2 to maintain elementarity.

We employ a matrix-based data structure of dimensions $|N| \times Q$ to store the labels on the graph. Each cell (i, q) in this matrix contains a list of all labels ending at node $i \in N$ with an accumulated load of q, where $0 \leq q \leq Q$. This structure's benefit lies in the fact that, since each cell contains labels with identical loads, the labels can be compared using classical dominance rules. However, the drawback is the extensive number of labels stored. At any given node i, some labels in cell (i, q) might dominate labels in cells (i, q') for $q' \geq q$. This dominance is not detected because the verification process only considers labels within the same cell.

To decrease the number of possible paths, we incorporate specific dominance rules in Algorithm 1, described as follows. Suppose we have two paths with labels $L(P_1)$ and $L(P_2)$. Path P_1 dominates path P_2 if it can achieve every possible extension of P_2 at an equal or lower total cost. For this to be valid, two conditions must be met:

- $C_{P_1} \leq C_{P_2}$ and
- $\Pi(P_1) \subseteq \Pi(P_2)$.

In Algorithm 2 besides the mentioned dominance rules an additional rule is also needed to be checked. Path P_1 is said to dominate path P_2 if $U_1 \neq U_2$ and $U_1^k \leq U_2^k$ for $k = 1, \ldots, |S|$, where k is the number of resources.

4.2.3.1 Acceleration Methods

In each iteration, it is not necessary to return the column(s) with the most negative reduced cost. It is sufficient to return (at least one) column(s) with a negative reduced cost if one exists. Adding any negative reduced cost column instead of the most negative reduced cost column may lead to a smaller change in the value of the solution and consequently increase the number of pricing iterations to reach the optimal solution. However, the reduction in the solution time of each pricing iteration can lead to less overall computation time. This concept can be applied simplistically by observing that many negative reduced-cost paths are usually found during the first few label-setting iterations.

Therefore, we implement the algorithm in such a way that we collect elementary negative reduced cost paths found in each iteration and sometimes terminate the algorithm prematurely, i.e., before there has been a finding of the most negative reduced cost column, and instead return the top specific number of collected elementary negative reduced cost columns (Ozbaygin et al., 2017).

The second useful idea comes from another observation: duals are updated after each pricing iteration, which means that some elementary paths may have non-negative reduced costs in one pricing iteration, but negative reduced costs in the next iterations. Thus, after the last label setting iteration, non-dominated elementary paths with non-negative reduced costs are kept in a column pool (Ozbaygin et al., 2017). Each pricing iteration starts with an evaluation of the columns in the pool to confirm if they (now) have a negative reduced cost. Keeping a maximum number of columns prevents the column pool from becoming too expensive to explore. If the maximum pool size is exceeded after a pricing iteration, the oldest columns are removed and the new ones are added until the desired pool size is reached.

It is also possible to detect negative reduced cost columns quickly by using heuristics before invoking the exact pricing algorithm (Ozbaygin et al., 2017). To this end, we implement a truncated version of the dynamic programming-based approach in which each cell $(i;q)$ of the matrix can keep a limited number of efficient labels (we refer to it as bucket length). Each time a new label is added to the list of efficient labels, we discard the one with the largest cost if the number of labels exceeds the limit. In this way, fewer labels are treated at each label setting iteration which can facilitate faster detection of negative cost elementary paths. Again, excluding a part of the solution space may come at the expense of performing more iterations, i.e., there is a trade-off between the number of efficient labels maintained and the number of label setting iterations performed.

Briefly, we attempt to identify negative reduced cost columns by first exploring the column pool, then implementing the truncated-search version of the dynamic programming algorithm, and finally implementing the full-search version of the dynamic programming algorithm if the other approaches fail. After finding a predetermined number of elementary negative reduced cost columns, we terminate any pricing iteration further to control time.

4.2.4 Integer solution

When the optimal solution to the master problem is fractional, We solve the following RMP where Ω' consists of all columns generated during the column generation phase.

$$
\begin{aligned}
&(4.55) && \text{minimize} && \sum_{r\in\Omega'}\sum_{t\in T}\zeta_r x_{rt}, && \\
&(4.56) && \text{subject to} && \sum_{p\in P_i} s_{ip} = 1 && \forall i \in N_c, \\
&(4.57) && && \sum_{r\in\Omega'} \alpha_{ir}x_{rt} - \sum_{p\in P_i} \beta_{pt}s_{ip} \geq 0 && \forall i \in N_c, \forall t \in T, \\
&(4.58) && && \sum_{r\in\Omega'} x_{rt} \leq \kappa && \forall t \in T, \\
&(4.59) && && x_{rt} \in \{0,1\} && \forall r \in \Omega', \forall t \in T, \\
&(4.60) && && s_{ip} \in \{0,1\} && \forall i \in N_c, \forall p \in P_i.
\end{aligned}
$$

Due to the inequality (4.57), the model has the chance of selecting routes for each day with less cost although some routes may contain customers that are not assigned to those days. We get the solution from the model and create a new Ω containing the routes in the solution plus some extra routes which are generated by editing the routes containing customers who are visited more than their corresponding frequencies. We consider all the customers' combinations, who are visited more than their corresponding frequencies while generating new routes. Then, we change the sign of (4.57) to equality and solve the model one more time with new Ω to obtain a feasible solution for the problem.

4.2.5 Computational Results of the Column Generation Algorithm

15 benchmark instances from the S1 dataset were used to test our algorithms since vehicles have a small capacity and these algorithms are suitable for solving them. All computational experiments are carried out on a virtual machine with Intel Xeon CPU E5-2640 processor with 2.60 GHz speed, 16 GB RAM, and 64-bit, using Julia 1.8.3.

Table 4.3 describes the chosen instances of data set S1 for the PVRP, where n is the number of customers, m is the number of vehicles that can be used, t is the number of days in the planning horizon, Q is the maximum capacity of the vehicles, and f_i is the number of customers that must be visited i times.

In the implementation, we set the bucket length to 2. The run time is limited to 4 hours. In Table 4.4, the first column lists the instance names. The second

Table 4.3 Instance description

Instance	n	m	t	Q	Service frequency			
					f1	f2	f4	f6
14	20	2	4	20	8	8	4	_
15	38	2	4	30	16	16	6	_
16	56	2	4	40	24	24	8	_
17	40	4	4	20	16	16	8	_
18	76	4	4	30	32	32	12	_
21	60	6	4	20	24	24	12	_
24	51	3	6	20	36	9	_	6
25	51	3	6	20	36	9	_	6
26	51	3	6	20	36	9	_	6
27	102	6	6	20	72	18	_	12
28	102	6	6	20	72	18	_	12
29	102	6	6	20	72	18	_	12
30	153	9	6	20	108	27	_	18
31	153	9	6	20	108	27	_	18
32	153	9	6	20	108	27	_	18

and third columns display the results reported by Baldacci et al. (2011) and Vidal et al. (2012) for these instances, with Vidal's results representing the best-known solutions in the literature.

Table 4.4 Results of Algorithm 1 and Algorithm 2 for Various Δ Values with Bucket Length of 2

			bucket length = 2								
			Δ = 4			Δ = 8			Δ = 16		
Instance	Baldacci et al. (2011)	Vidal et al. (2012)	OFV	Time Alg.1	Time Alg.2	OFV	Time Alg.1	Time Alg.2	OFV	Time Alg.1	Time Alg.2
P14	954.81	954.81	**954.81**	15.49	14.71	**954.81**	22.47	19.32	**954.81**	23.54	24.387
p15	1862.63	1862.63	**1862.63**	188.82	178.67	**1862.63**	651.99	598.97	**1862.63**	546.17	530.025
P16	2875.24	2875.24	**2875.24**	530.14	393.09	**2875.24**	856.73	934.96	**2875.24**	1001.02	1113.52
P17	1597.75	1597.75	**1597.75**	158.38	121.73	**1597.75**	125.00	123.91	**1597.75**	167.93	172.626
P18	3136.69	3131.09	3169.05	1973.81	1894.89	3169.05	4372.64	3932.763	3177.71	5342.19	5117.37
P21	2170.61	2170.61	2172.74	655.67	520.15	2172.74	654.19	674.088	2199.837	693.65	734.423
P24	3687.46	3687.46	**3687.46**	234.91	192.77	**3687.46**	254.05	229.30	**3687.46**	276.887	304.653
P25	3777.15	3777.15	3781.38	174.33	135.24	3781.38	174.67	163.073	3789.004	202.038	238.55
P26	3795.32	3795.32	**3795.32**	115.54	116.37	**3795.32**	193.77	170.45	**3795.32**	201.10	193.873
P27	21912.85	21833.87	21941.30	6332.57	5282.01	21954.37	3641.29	3205.785	21948.45	3710.25	3411.398
P28	22242.51	22242.51	**22242.51**	1612.38	1551.74	**22242.51**	2369.47	2050.863	**22242.51**	2003.55	1931.457
P29	22543.76	22543.76	**22543.76**	1998.62	2076.83	**22543.76**	2536.33	2167.619	**22543.76**	2456.98	2010.475
P30	74464.26	73875.19	74944.79	14400.00		74728.26	14400.00	14400.00	74848.32	14400.00	14400.00
			74393.49		14400.00						
P31	76322.04	76001.57	**76287.97**	14400.00	14400.00	76391.87	14400.00		**76248.73**	14400.00	11342.06
						76333.32		14400.00			
P32	78072.88	77598.00	**77833.57**	14400.00		**77794.69**	14400.00	14400.00	**77755.90**	14400.00	11309.20
			77759.79		14400.00						

Following these, there are three blocks of three columns each, corresponding to specific Δ values (4, 8, and 16). Within each block, the first column shows the objective function value (OFV), the second column reports the runtime for Algorithm 1, and the third column reports the runtime for Algorithm 2. For problem instances p30, p31, and p32, if the solutions of Algorithm 1 and Algorithm 2 differ, the solution from Algorithm 2 would be reported in a separate row below the original. Bold values indicate that the results match the best-known solutions in the literature. Bold and Underlined values indicate that our algorithm outperforms Baldacci's results but does not surpass Vidal's results.

As it is evident from the Table 4.4, Algorithm 2 generally outperforms Algorithm 1 in terms of runtime. The Ω set at the end of the Column Generation phase is identical for both algorithms. Since Algorithm 2 is frequently faster during the Column Generation phase, this advantage allows Algorithm 2 more time after the Column Generation phase to find the integer solution.

Considering the best solution obtained from the algorithms for each instance, the average percentage deviation of them from Vidal's results is approximately 0.21%. The relatively low average deviation (0.21%) indicates that the solutions produced by the algorithms are very close to the best-known solutions provided by Vidal, showing a high level of performance and accuracy.

- For 9 out of the 15 instances, the OFV exactly matches Vidal's results, resulting in a 0% deviation. These instances are p14, p15, p16, p17, p24, p26, p28, and p29.

- Most deviations from Vidal's results are small, typically under 0.5%. This suggests that the solutions are very close to the best-known solutions in the literature. Specifically p21 (0.098%), p25 (0.112%), p27 (0.492%), p31 (0.325%), p32 (0.203%).

- There are two instances with deviations greater than 0.5%, p30 (0.702%) and p18 (1.212%).

In summary, the proposed algorithms yield results that are very close to the best-known solutions in the literature, with an average deviation of only 0.21%. This underscores the algorithm's effectiveness, as most instances exhibit either no deviation or only slight deviations from the best-known solution in the literature

4.3 Conclusions

In the first part of the chapter, we proposed the LBBD and BAC algorithms to solve the PVRP. Computational results demonstrate that BAC outperforms LBBD in medium-sized instances. However, a comparison between the BAC algorithm and the UC1 formulation reveals that the convergence of BAC to the optimal solution is slow, and the time required to solve the VRP subproblem is considerable.

To expedite the algorithm, we aimed to enhance the speed of solving the VRP subproblem. Initially, we applied an exact column generation algorithm for this purpose. Unfortunately, our proposed column generation approach did not yield significant improvements in solution time. Consequently, we refined and adapted this approach specifically for the PVRP problem, aiming to solve the PVRP directly using the column generation algorithm.

In the second part of the chapter, we propose a column generation-based heuristic to solve the PVRP. In the pricing problem, we applied Martinelli et al. (2014) algorithm. However, to maintain elementarity in solving the pricing sub-problem, we applied two different approaches: one proposed by Dayarian et al. (2015) and a hybrid approach. At the end of the algorithm, if the optimal solution to the master problem is fractional, we solve the integer RMP, allowing visits to customers more frequently than their specified frequencies. Based on this solution, we generate a new Ω and enrich it with additional columns. We then resolve the problem using the updated Ω.

The results show that the proposed algorithm produces results very close to the best-known solutions in the literature, with an average deviation of only 0.21%. This highlights the effectiveness of the algorithm, especially given that most instances show either no deviation or very minor deviations from the best-known solution in the literature.

Time window and non-crossing intra-route constraints can be seamlessly integrated into the proposed algorithm by incorporating them directly into the pricing problem. This approach ensures that these constraints are enforced during the column generation process. However, integrating non-crossing inter-route and driver consistency constraints requires a different approach. These constraints need to be included in the master problem, where the master problem verifies the absence of arc intersections between different routes. As the size of Ω increases, this verification process can become computationally intensive.

Future research can explore various algorithmic enhancements to improve the performance of the column generation approach. These include the implementation of cutting planes, advanced branching strategies, and sophisticated data structures for label storage during the pricing stage. Additionally, the integration of metaheuristics could further enhance the efficiency and effectiveness of the method.

5. A Parallel Tempering-based Adaptive Large Neighborhood Search Algorithm for a Periodic Vehicle Routing Problem with Time Windows and Visual Attractiveness constraints

5.1 Problem Definition and Formulations

In this section, we define and formulate PVRPTW and PVRPTWDCVA problems. PVRPTW formulation is the extension of the directed cut-based formulation for the PVRPTW in which we also consider the tour duration constraint. Then we extend the PVRPTW formulation to the PVRPTWDCVA formulation by adding constraints that prevent both inter-route and intra-route crossings, while also ensuring driver consistency.

5.1.1 Mathematical Formulation for PVRPTW

Assume D is the maximum allowable route duration. The PVRPTW considering route duration restriction can be formulated as follows:

$$
\begin{aligned}
&(5.1) && \text{minimize} && \sum_{(i,j)\in A}\sum_{t\in T}\sum_{k\in K} c_{ij}x_{ijtk}, && \\
&(5.2) && \text{subject to} && \sum_{p\in P_i} s_{ip} = 1 && \forall i \in N_c, \\
&(5.3) && && \sum_{p\in P_i : t\in P} s_{ip} = \sum_{k\in K} z_{itk} && \forall i \in N_c, \forall t \in T, \\
&(5.4) && && \sum_{i\in N_c} z_{itk}\, d_i \le Q v_{tk} && \forall k \in K, \forall t \in T, \\
&(5.5) && && \sum_{j\in N\setminus\{i\}} x_{ijtk} = z_{itk} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.6) && && \sum_{j\in N\setminus\{i\}} x_{jitk} = z_{itk} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.7) && && w_{itk} + r_i + c_{ij} - w_{jtk} \le (1 - x_{ijtk})M && \forall i \in N, \forall j \in N_c, \\
& && && && \forall k \in K, \forall t \in T, \\
&(5.8) && && w_{itk} + r_i + c_{i0} - L_0 \le (1 - x_{i0tk})M && \forall i \in N_c, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.9) && && E_i \sum_{j\in N} x_{ijkt} \le w_{itk} \le L_i \sum_{j\in N} x_{ijkt} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.10) && && w_{itk} + r_i + c_{i0} - w_{0tk} \le D && \forall i \in N_c, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.11) && && s_{ip} \in \{0,1\} && \forall i \in N_c, \forall p \in P_i, \\
&(5.12) && && x_{ijtk} \in \{0,1\} && \forall i,j \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.13) && && z_{itk} \in \{0,1\} && \forall i \in N_c, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.14) && && v_{tk} \in \{0,1\} && \forall k \in K, \forall t \in T, \\
&(5.15) && && w_{itk} \ge 0 && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T.
\end{aligned}
$$

The objective function (5.1) minimizes the total routing cost. Constraints (5.2) ensure that an allowable visit schedule is selected for each customer. Constraints (5.3) relate the customer visit variables and the schedule selection variables. Constraints

(5.4) are the vehicle capacity restrictions. Any customer to be served on a given day will be visited exactly once due to constraints (5.5) and (5.6), which, together, imply vehicle flow conservation. Constraints (5.7)–(5.9) guarantee the feasibility of the tours with respect to time window restrictions. Constraints (5.10) are the route duration restriction. Constraints (5.11)–(5.15) specify the domain restrictions on the variables.

5.1.2 Mathematical Formulation for PVRPTWDCVA

In this section, we aim to propose a new formulation that focuses on enhancing the visual appeal of solution routes by avoiding inter and intra-route crossings and ensuring driver consistency. In addition to the previously defined x, w, z, and v variables used in the PVRPTW formulation, we define s_{ipk} which is equal to 1 if schedule $p \in P_i$ is chosen to visit customer $i \in N_c$, and if all visits in that schedule are done by vehicle $k \in K$, and 0 otherwise. In the following model, the additional input data $\delta(i,j,m,n)$ is considered, which takes the value 1 if arcs (i,j) and (m,n) intersect. To ensure non-crossing routes, at most one of any pair of intersecting arcs can be included in the solution.

The PVRPTWDCVA formulation is as follows;

$$
\begin{aligned}
&(5.16) && \text{minimize} && \sum_{(i,j)\in A}\sum_{t\in T}\sum_{k\in K} c_{ij}x_{ijtk}, && \\
&(5.17) && \text{subject to} && \sum_{p\in P_i}\sum_{k\in K} s_{ipk} = 1 && \forall i \in N_c, \\
&(5.18) && && \sum_{p\in P_i: t\in P} s_{ipk} = z_{itk} && \forall i \in N_c, \forall t \in T, \\
& && && && \forall k \in K, \\
&(5.19) && && \sum_{i\in N_c} z_{itk}\, d_i \le Q v_{tk} && \forall k \in K, \forall t \in T, \\
&(5.20) && && \sum_{j\in N\setminus\{i\}} x_{ijtk} = z_{itk} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.21) && && \sum_{j\in N\setminus\{i\}} x_{jitk} = z_{itk} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.22) && && \sum_{k\in K} x_{ijtk} + \sum_{k\in K} x_{jitk} + && \\
& && && \sum_{k\in K} x_{mntk} + \sum_{k\in K} x_{nmtk} \le 1 && \forall i,j,m,n \in N_c, \\
& && && && \forall t \in T, \\
& && && && if\ \delta(i,j,m,n) = 1, \\
&(5.23) && && w_{itk} + r_i + c_{ij} - w_{jtk} \le (1 - x_{ijtk})M && \forall i \in N, \forall j \in N_c, \\
& && && && \forall k \in K, \forall t \in T, \\
&(5.24) && && w_{itk} + r_i + c_{i0} - L_0 \le (1 - x_{i0tk})M && \forall i \in N_c, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.25) && && E_i \sum_{j\in N} x_{ijkt} \le w_{itk} \le L_i \sum_{j\in N} x_{ijkt} && \forall i \in N, \forall k \in K, \\
& && && && \forall t \in T, \\
&(5.26) && && w_{itk} + r_i + c_{i0} - w_{0tk} \le D && \forall i \in N_c, \forall k \in K, \forall t \in T, \\
&(5.27) && && s_{ipk} \in \{0,1\} && \forall i \in N_c, \forall p \in P_i, \forall k \in K, \\
&(5.28) && && x_{ijtk} \in \{0,1\} && \forall i,j \in N, \forall k \in K, \forall t \in T, \\
&(5.29) && && z_{itk} \in \{0,1\}, w_{itk} \ge 0 && \forall i \in N_c, \forall k \in K, \forall t \in T, \\
&(5.30) && && v_{tk} \in \{0,1\} && \forall k \in K, \forall t \in T.
\end{aligned}
$$

The objective function (5.16) minimizes the total routing cost. Constraints (5.17) ensure that each customer is serviced by one vehicle, and following one of its allowable visit schedules. Constraints (5.18) relate the customer visit variables and the schedule selection variables. Constraints (5.19) are the vehicle capacity restrictions. Any customer to be served on a given day will be visited exactly once due to constraints (5.20) and (5.21), which, together, imply vehicle flow conservation. The constraints (5.22) indicate that at most one of the arcs (i,j), (j,i), (n,m), and (m,n) can be in the solution if they intersect. Constraints (5.23)–(5.25) guarantee the feasibility of the tours with respect to time window restrictions. Constraints (5.26) are the route duration restriction. Constraints (5.27)–(5.30) specify the domain restrictions on the variables.

5.2 Adaptive Large Neighborhood Search

Given the intractability of the problem for large instances, we develop an ALNS-based metaheuristic approach to solve it. ALNS a metaheuristic algorithm utilized for solving combinatorial optimization problems, takes an initial solution and in an iterative process explores different neighborhoods of a given solution in the hope of finding an enhanced solution (Ropke and Pisinger, 2006). ALNS explores the search space by adjusting the current solution at each iteration using a strategy called the "Destruction and Construction" principle. Destruction pertains to removing a part of the current solution, while construction involves generating a new solution by rebuilding the destroyed portion. Appropriate heuristics are utilized to carry out destruction and construction tasks. These heuristics are selected at each iteration using a biased random mechanism known as the roulette-wheel. This mechanism favors the heuristics that have demonstrated success in recent iterations based on specific criteria, such as improvement in solution quality.

Here are some more common characteristics that can vary between ALNS implementations (Voigt, 2024).

- **Search space**
  - Feasible solutions only: The approach only considers solutions that meet all constraints and requirements.
  - Penalized infeasible solutions*: Infeasible solutions are allowed but penalized by a generalized cost function, allowing the algorithm to explore a wider range of solutions.

Algorithm 3 ALNS

```
s, s* ← initial solution
Initialize the weights and Set parameters T, φ, δ, γ, c, MaxIter
itr ← 0
segmentIter, seg ← 1
Flag ← false
while iter < MaxIter do
    ŝ ← s
    counter ← 1
    while counter % φ ≠ 0 do
        q ← select the maximum number of customers to remove
        Opr ← select an operator
        s′ ← Opr(s, q)
        if f(s′) < f(ŝ) then
            ŝ ← s′
        end if
        counter ← counter + 1
    end while
    if f(ŝ) < f(s*) & ŝ is feasible then
        s, s* ← ŝ
        segmentIter ← 1
        Flag ← true
    else if f(ŝ) < f(s*) & ŝ is not feasible then
        s ← ŝ
    else
        if unif(0,1) < e^(-(f(ŝ)-f(s))/T) then
            s ← ŝ
        end if
    end if
    Update the score of operation Opr
    if segmentIter == δ then
        if Flag == true then
            s″, IsReduced ← VehicleNumberReduction(s*)
            if IsReduced then
                s, s* ← s″
                segmentIter ← 1
            else
                T ← cT
                seg ← seg + 1
            end if
        else
            T ← cT
            seg ← seg + 1
        end if
    end if
    if seg%γ == 0 then
        Update the weights
    end if
    iter ← iter + 1
    segmentIter ← segmentIter + 1
end while
Return s*
```

  - Single solution: In each iteration, only one new solution is generated and it is accepted as the current solution based on the acceptance criteria.
  - Parallel solutions*: In each iteration, multiple solutions are generated from the current solution, and the best solution among them is accepted as the current solution based on the acceptance criteria.

- **Initial solution**
  - Feasible initial solution*: The algorithm starts with a feasible initial solution.
  - Infeasible initial solution: The algorithm starts with an infeasible initial solution.

- **stopping criteria**
  - Time limit: The algorithm stops after a specified amount of time.
  - Maximum number of iterations*: The algorithm stops after reaching a certain number of iterations.
  - Maximum number of iterations without improvement: The algorithm stops if no improvement is made after a certain number of iterations.
  - Temperature in simulated annealing: The algorithm stops when the temperature in simulated annealing reaches a certain threshold.

- **operation selection**
  - Independent selection: The operators may be selected independently (i.e., there are two roulette wheels, one for destroy and one for repair operators)
  - Pair-wise selection*: There is only one roulette wheel selecting a destroy-repair pair.

- **Number and type of destruction/construction operators** Specifies the number and type of operators used to explore and exploit the search space.

- **Local search**
  - None: No local search is performed.
  - When criteria are met*: Local search is performed when specific criteria are met to improve the current solution.

- All solutions: Local search is performed on all solutions generated by the algorithm.

- **Acceptance criteria**
  - Simulated annealing*: The algorithm uses simulated annealing to accept or reject new solutions based on a temperature parameter.
  - Record to record: A new solution is accepted if the absolute gap between the candidate and the best or current solution is smaller than a threshold.
  - Hill climbing: It accepts only progressively better solutions, discarding the ones that result in a worse objective value.

- **Parameter update**
  - After every iteration: Parameters are updated after each algorithm iteration.
  - After a search segment*: Parameters are updated after completing a search segment within the algorithm.

The characteristics that we applied are shown with $*$. The overall framework of our algorithm is reported in Algorithm 3. First, we initialize the hyper-parameters for the ALNS. Then, we create an initial solution (Section 5.2.1) and set it as the current solution. At each iteration, we explore the neighborhood of the current solution, generating potentially ϕ new solutions. New solutions are obtained by applying an operator $opr \in \Omega$ to the current solution, where Ω is the set of all operators. Contrary to classical ALNS, the operators are built through coupling each combination of destruction and construction heuristics. (A similar idea of pairing heuristics was used by Dayarian et al. (2016) for MVRP) The main advantage is that we can weigh the performance of each (destruction-construction) pair. We select the operator to apply to the solution of the current iteration via a roulette-wheel mechanism.

Following this, the best solution ($\hat{s}$) among the ϕ solutions found is accepted as the current solution (s) depending on acceptance criteria. In case a new best feasible solution has been found the best feasible solution s^* and current solution s are updated and *segmentIter*, which counts the number of consecutive iterations in which the best feasible solution is not improved, is set to 0. In other cases, we always accept the superior solution right away and accept worse solutions under the Metropolis criterion like in simulated annealing (Kirkpatrick et al., 1983) with a probability of $e^{-(f(\hat{s})-f(s))/T}$, depending on the cost difference to the current solution

s and the temperature T. Then, we update the score of operations. Each time the best feasible solution is not improved in δ consecutive iterations one segment in the algorithm is completed. At the end of each segment, if a superior feasible solution is found the algorithm enters the route reduction phase and accepts the solution with less number of routes as the best feasible solution. Otherwise, the cooling procedure is applied. We update the weights of the operators every γ segment.

5.2.1 Initial Solution

To generate an initial solution, customers are initially allocated to days based on their available schedules. Subsequently, they are sorted by proximity to the depot, from nearest to farthest. The process commences with an empty route for each day. It iterates through the list of customers, selecting the closest one to the depot. If this customer can be placed within the same route on their assigned days while adhering to all constraints (such as time windows, vehicle capacity, route duration, and avoiding inter-route crossings), we iterate through the list to place the next customer. If not, an empty route is created for each day within the planning horizon, and the selected customer is assigned to these new routes on the corresponding days. This process continues until all customers are assigned to routes. Subsequently, a reduction phase ensues, wherein routes with fewer than five customers are dissolved, and their customers are reallocated to other routes while respecting all constraints.

An alternative procedure involves iteratively attempting to assign all customers to a specific route on their respective days until adding a new customer violates any constraints. Subsequently, empty routes are generated for each day, and unassigned customers are placed on these new routes, taking into account all restrictions and their designated days. This process repeats until all customers are allocated to routes, followed by the reduction phase.

Additionally, different sorting strategies for customers are tested, including sorting them by distance to the depot from farthest to nearest, as well as by their angles to the depot. Among the various permutations of sorting strategies and route creation procedures tested, the initially described method demonstrates superior performance.

5.2.2 Search Space

It is well known in the metaheuristic literature that allowing the search into infeasible regions may lead to good solutions (Expósito et al., 2018; Dayarian et al., 2016), thus smoothing the search space, by relaxing time windows, vehicle capacity, and tour duration conditions of the problem definition. For a given solution s, the total travel cost is represented by $c(s)$, while the total violations of time windows, load, and duration constraints are denoted by w(s), q(s), and d(s), respectively. The calculations for $q(s)$ and $d(s)$ are based on individual routes, taking into account the values of Q_k and D_k. On the other hand, $w(s)$ is determined by the formula $\sum_i \max\{0, a_i - L_i\}$, where a_i represents the arrival time at customer i. The cost function $f(s)$ is defined as $c(s)+\theta w(s)+\eta q(s)+\zeta d(s)$, where θ, η and ζ are positive weights that can be adjusted dynamically during the search process. Nevertheless, based on initial experiments, we have decided to use a fixed value of 100, which was also the approach taken in Polacek et al. (2004) for an MDVRP with time windows.

5.2.3 Local Search

At the end of each segment, a reduction procedure is performed on the best solution found. This procedure aims to reduce the number of vehicles used, while staying within the feasible region. If successful, the current and best feasible solutions are updated accordingly.

In this procedure, routes are sorted by the number of customers they serve, from fewest to most, within the planning horizon. It begins by dismantling the first route in the sorted list from all days. The next step involves attempting to reallocate the removed customers to other routes without violating any constraints. If all removed customers can be successfully placed into other routes, the original route is permanently removed, and the procedure proceeds to the next route in the list. If reallocation is not feasible, the route is restored to its original state, and the procedure moves on to the next route.

5.2.4 Adaptive Weight Adjustment Procedure

The selection of paired removal-insertion operators is managed by a roulette-wheel mechanism. Initially, all pairs are equally likely to be chosen. Each pair *opr* is assigned a weight Ψ_{opr}, which starts at 1 and is updated after each γ segments based on their performance history. The probability of selecting pair *opr* is given by: $\frac{\Psi_{opr}}{\sum_{j\in\Omega}\Psi_{opr}}$

Each pair *opr* is also assigned a score Π_{opr}, which starts at zero and is reset after every γ segments. During each iteration, the scores are updated by adding a bonus factor σ_i, $i \in \{1,\ldots,4\}$, where $\sigma_1 \le \sigma_2 \le \sigma_3 \le \sigma_4$, according to the following criteria:

- σ_4: if a new superior feasible solution is found,
- σ_3: if the new solution improves the current solution but is not the best feasible solution,
- σ_2: if the new solution meets the acceptance criterion and is a feasible solution,
- σ_1: if the new solution meets the acceptance criterion but is not a feasible solution,
- 0: otherwise.

The weights are updated during the algorithm using the formula:

$$\Psi_{opr,i+1} = \Psi_{opr,i}(1-\alpha) + \alpha\frac{\Pi_{opr}}{\Xi_{opr}}$$

where $\Psi_{opr,i+1}$ is the updated weight of operator *opr* after the ith block of γ segments, Π_{opr} is the score of *opr*, Ξ_{opr} is the number of times *opr* was used during the last γ segments, and α is the reaction factor.

5.2.5 Destruction Heuristics

Listed below are the destruction heuristics employed in the ALNS algorithm.

Customer with Maximum Distance from Geometric Center Removal: This heuristic identifies the customer that is farthest from the geometric center of the route to which they are assigned, across all periods. It then removes this customer from the assigned route over the planning horizon, provided that their removal does not cause inter-route or intra-route crossings.

Random Removal: This heuristic randomly selects and removes a customer from the assigned route over the planning horizon, ensuring that the removal does not result in inter-route or intra-route crossings.

Customer Closer to Another Route's Geometric Center than Its Own: This heuristic identifies a customer whose maximum distance from the geometric center of their assigned route during the planning horizon is greater than their maximum distance from the geometric centers of other routes during the same period. It then removes this customer from the assigned route over the planning horizon, provided that their removal does not cause inter-route or intra-route crossings.

Shaw Removal: This heuristic evaluates similar customers based on their distance, earliest service time, route assignment, days of service, and demand. It randomly selects customer i and calculates the relatedness measure R_{ij} to identify similar customers. The relatedness measure is defined as:

$$R_{ij} = \phi_1 SD_{ij} + \phi_2 |E_j - E_i| + \phi_3 SR_{ij} + \phi_4 |D_j - D_i| + \phi_5 c_{ij}$$

where $\phi_1 - \phi_5$ are the Shaw parameters, $SR_{ij} = -1$ if i and j are in the same route, 1 otherwise, c_{ij} is the distance between customers i and j, SD_{ij} is the number of common days among chosen days for customers i and j. A smaller R_{ij} indicates higher similarity. Customers are ordered in non-decreasing order of their relatedness values with customer i, and the first q customers in this list are sequentially removed from the solution. We adapted the heuristic from Rastani and Çatay (2023) to suit our problem.

5.2.6 Construction Heuristics

After the destruction heuristic, the removed customer is considered for reinsertion into routes. The following construction heuristics are considered:

Greedy Insertion with Geometric Center Proximity Priority (GCPP): This heuristic assigns the removed customer to a randomly allowed schedule. It sorts the routes according to the total distance of the customer from the geometric center of the routes. Then, it goes through the sorted route list until it finds a route that can place the customer on all chosen days without violating visual attractiveness. The customer is then placed in the best positions (considering visual attractiveness) within the route on each chosen day.

Greedy Insertion on a Random Route: This heuristic assigns the removed customer to a randomly allowed schedule. It goes through the shuffled route list until it finds a route that can place the customer on all chosen days without violating visual attractiveness. The customer is then placed in the best position (considering visual attractiveness) within the route on each chosen day. We adapted the heuristic from Rastani and Çatay (2023) to suit our problem.

Regret Greedy Insertion with GCPP: This heuristic assigns the removed customer to a randomly allowed schedule. It sorts the routes according to the total distance of the customer from the geometric center of the routes. Then, it goes through the sorted route list until it finds a route that can place the customer on all chosen days without violating visual attractiveness. The customer is then placed in the second-best position (considering visual attractiveness) within the route on each chosen day.

Regret Greedy Insertion on a Random Route: This heuristic assigns the removed customer to a randomly allowed schedule. It iterates through the shuffled route list until it finds a route that can cover that customer on all chosen days without violating visual attractiveness. The customer is then placed in the second-best position (considering visual attractiveness) within the route on each chosen day. We adapted the heuristic from Rastani and Çatay (2023) to suit our problem.

Random Insertion on a Random Route: This heuristic assigns the removed customer to a randomly allowed schedule. It iterates through the shuffled route list until it finds a route that can cover that customer on all chosen days without violating visual attractiveness. The customer is then placed in a random position (considering visual attractiveness) within the route on each chosen day.

Insertion with the Least Infeasibility Violation and GCPP: This heuristic assigns the removed customer to a randomly allowed schedule. It sorts the routes according to the total distance of the customer from the geometric center of the routes. Then, it iterates through the route list until it finds a route that can cover that customer on all chosen days without violating visual attractiveness. The customer is then placed in the position on the route (considering visual attractiveness) that leads to the least infeasibility violation on each chosen day.

Insertion with the Least Infeasibility Violation on a Random Route: This heuristic assigns the removed customer to a randomly allowed schedule.

It iterates through the shuffled route list until it finds a route that can cover that customer on all chosen days without violating visual attractiveness. The customer is then placed in the position on the route (considering visual attractiveness) that leads to the least infeasibility violation on each chosen day.

5.3 Parallel Tempering

Parallel tempering, also referred to as replica exchange MCMC sampling, is a simulation technique designed to enhance the dynamic characteristics of Monte Carlo method simulations of physical systems, as well as of Markov chain Monte Carlo (MCMC) sampling methods. The main idea of the PT is to run multiple copies (replicas) of the system, randomly initialized, each at a different temperature. High-temperature replicas have the capability of exploring extensive portions of the solution space, while low-temperature replicas can examine local areas accurately.

A good sampling of the solution search space is obtained by exchanging configurations at different temperatures based on the Metropolis criterion. In other words, this method makes configurations at high temperatures available to the replicas at low temperatures and vice versa. It is worth noting that the temperature of each replica remains constant, setting it different from Simulated Annealing where the temperature decreases gradually.

5.4 Parallel Tempering Based ALNS

In the proposed PTALNS algorithm a number M of the ALNS-based replicas are executed in parallel. The replicas differ from the ALNS in the temperature changes. In the PTALNS, the temperature of each ALNS-based replica remains constant while in the ALNS the temperature is gradually lowered. The overall framework of our PTALNS algorithm is reported in Algorithm 4. PTALNS begins with the creation of M replicas each with a unique temperature level $T_i, i \in \{1, \ldots, M\}$(lines 1-4) where $T_1 < T_2 < \cdots < T_M$. Afterward, each replica is allowed to do $MaxRepIter$ iterations (lines 6–8). Following this, a trial is conducted to exchange the replica's configuration at Tj (where j is selected at random, line 10) with the replica's configuration at T_{j+1} (lines 9–15). We can just switch the replicas' temperature levels rather than modifying their internal states (line 13). The equation contained the reciprocals of T_j and T_{j+1} and the cost difference of the replicas' current solutions (line 12) determines whether to switch between T_j and T_{j+1} levels. Line 13 compares the probability that is obtained with a random variable that is evenly selected from the interval (0, 1). It is important to remember that the probability equals one if the cost of the current solution of the (higher temperature) replica T_{j+1} is less than the corresponding cost of the (lower temperature) replica T_j. When calculating ΔE, the cost difference is divided by the cost of the best solution so far for all replicas (line 11). This is a normalization step that aims to keep the computation independent of the edge weights of the problem instance.

Algorithm 4 PTALNS

1: **for** `i from 1 to M` **do**
2:     $T_i \leftarrow \text{SetTemperature}(i)$
3:     $Replica_i \leftarrow$ new copy of the ALNS with T_i
4: **end for**
5: **while** Not reached stopping criterion **do**
6:     **for** `i from 1 to M` **do**
7:         Run $replica_i$ for $MaxRepIter$ iterations
8:     **end for**
9:     $BGC \leftarrow$ Minimum cost among best feasible solutions of M replicas
10:     $j \leftarrow unif\{1, M-1\}$
11:     $\Delta E \leftarrow \frac{Cost(CurrentSolution_{replica_j}) - Cost(CurrentSolution_{replica_{j+1}})}{BGC}$
12:     **if** $unif(0,1) \leq min(1, e^{\frac{\Delta E}{T_j} - \frac{\Delta E}{T_{j+1}}})$ **then**
13:         Set the temperature of $replica_j$ and $replica_{j+1}$ respectively to T_{j+1} and T_j
14:         swap $replica_j$ and $replica_{j+1}$ to keep replicas ordered by temperature
15:     **end if**
16: **end while**

5.5 Computational study

Our computational experiments are detailed as follows. Section 5.5.1 introduces the benchmark instances used in this chapter. In Section 5.5.2, we present the results of an extensive sensitivity analysis conducted to calibrate the parameter values. Section 5.5.3 evaluates the performance of the operators through a series of tests. In section 5.5.4, we analyze the proposed formulations. Finally, sections 5.5.5 and 5.5.6 present the computational results for the test problems. All computational experiments are carried out on a virtual machine with Intel(R) Core(TM) i9-9900K CPU processor with 3.60 GHz speed, 64 GB RAM, and 64-bit, with Julia 1.9.4 and Gurobi 9.1.2.

5.5.1 Benchmark Instances

We rely on two classical benchmark instances: the PVRPTW instances of Cordeau et al. (2001) and Pirkwieser and Raidl (2009b). These datasets are described in Chapter 2. It should be noted that the previous authors used instances from Pirkwieser and Raidl (2009b) without duration constraints, and the distances were truncated to the first digit. We followed this convention exclusively in this case to ensure a fair comparison. Additionally, we generated 11 smaller instances named datasets S9 to evaluate the proposed formulations and the performance of the PTALNS algorithm. These were derived from instances pr01 and pr11 of dataset S6, and from instances P4c10, p4r101, and p4rc101 of dataset S7. The first three instances are generated from S7, with the geographical data being clustered, random, and random-clustered, respectively. The remaining instances are generated from S6, where the first three have tight time windows, and the rest have wide time windows.

5.5.2 Parameter Tuning

We employ a two-phase procedure to fine-tune the parameters of the PTALNS algorithm to enhance its performance. The first phase leverages the black box optimizer HyperTuning to adjust the most impactful parameters: "M", "$MaxRepIter$", "δ", "ϕ", and "γ". HyperTuning systematically explores the parameter space defined by specified ranges, using a scenario configuration with a $MedianPruner$ to prune underperforming trials early and a maximum of 200 trials. The objective function integrates these parameters into the PTALNS algorithm and evaluates the results, aiming to identify the optimal parameter set that maximizes the algorithm's effectiveness.

In the second phase, we manually tune the less sensitive parameters, "T" and "q", through trial and error. This approach is adopted due to the diminishing returns in accuracy with the increasing number of parameters handled by the black box optimizer. By first optimizing the critical parameters automatically and then fine-tuning the remaining ones manually, we ensure a comprehensive and efficient parameter-tuning process. This methodology balances the thorough exploration of the parameter space with practical adjustments, leading to improved performance of the PTALNS algorithm for our problem instances.

In parameter tuning, splitting the problem instances into training and testing sets is standard practice to ensure robust performance evaluation. Our study generated 20 train problem instances derived from the data sets S6, S7, and S8. This approach allows us to fine-tune the algorithm's parameters effectively, ensuring that the optimized settings generalize well across various problem instances.

Table 5.1 Parameter values found using HyperTuning for PTALNS

	Parameter	**Range**	**Value**	
			Up to 50 customers	**Above 50 customers**
δ	Segment length	[50,100]	20	10
ϕ	Inner loop length in ALNS	[1,20]	10	20
γ	Number of segments to update operator weights	[1,4]	3	3
α	Reaction factor in weight update	[0,1]	0.25	0.25
M	Number of replicas	[2,8]	4	4
$RepMaxIter$	Maximum number of iterations for each replica	[100,1000]	300	300

Table 5.2 Parameter values found by trial and error for PTALNS

	Parameter	**Value**
q	Maximum number of nodes to remove	$[0.05\|N_c\|]$
T_1, T_2, T_3, T_4	Temperature levels for replicas	$0, 2, 4, 6$
$\sigma_1, \sigma_2, \sigma_3, \sigma_4$	Bonus factors for adaptive weight adjustment	$1, 1, 1, 2$
$\phi_1, \phi_2, \phi_3, \phi_4, \phi_5$	Shaw removal coefficients	$-0.25, 0.25, 0.15, 0.25, 0.5$

In Tables 5.1 and 5.2, you will find the values determined for various parameters. The first and second columns present the name and description of each parameter, respectively. Table 5.1 displays the range for each parameter, as determined through preliminary tests, while the last block of two columns shows the best value for each parameter for instances with less than 50 and more than 50 customers.

We use the same procedure to tune the ALNS parameters. In Tables 5.3 and 5.4, you find the values determined for various parameters by the HyperTuning black box and through trial and error, respectively.

Table 5.3 Parameter values found using HyperTuning for ALNS

	Parameter	Range	Value
δ	Segment length	[50,100]	60
ϕ	Inner loop length in ALNS	[1,20]	15
γ	Number of segments to update operator weights	[1,4]	2
α	Reaction factor in weight update	[0,1]	0.25

Table 5.4 Parameter values found by trial and error for ALNS

name	Parameter	Optimal Value
q	Maximum number of nodes to remove	$[0.05\|N_c\|]$
T	initial temperature	10
c	Cooling rate	0.9987
$\sigma_1, \sigma_2, \sigma_3, \sigma_4$	Bonus factors for adaptive weight adjustment	$1, 1, 1, 2$
$\phi_1, \phi_2, \phi_3, \phi_4$, ϕ_5	Shaw removal coefficients	$-0.25, 0.25, 0.15, 0.25, 0.5$

The maximum number of iterations in the PTALNS is set to 240,000. We use the same value for the maximum number of iterations in the ALNS.

5.5.3 Evaluating the Performances of the Operators

Table 5.5 summarizes the statistics on the selection probabilities of different operators, based on a sample of 20 instances. Each operator, defined as a pair of destroy-repair heuristics, is evaluated using three key metrics:

- min%: The lowest probability of being chosen at the end of the solution process across all 20 instances,
- avg.%: The mean probability of being chosen at the end of the solution process across all 20 instances,
- max%: The highest probability of being chosen at the end of the solution process across all 20 instances.

The range of probabilities for the minimum, average, and maximum metrics spans $[0.00, 1.79]$, $[0.06, 7.58]$, and $[1.19, 26.24]$, respectively. Moreover, the average values for the min, avg., and max columns are 0.21%, 2.86%, and 7.88%, respectively.

These findings reveal that each operator pair demonstrates utility at different points across all instances, highlighting the adaptability and effectiveness of the heuristic combinations used.

Table 5.5 Final probabilities of choosing different destroy-repair pair.

Repair Heuristic	Destroy Heuristic	min%	avg%	max%
Greedy Insertion- GCPP	Max-Dist-GC	0.00	7.11	16.32
	Closer-GC	0.05	3.79	9.15
	Shaw removal	0.00	2.32	4.51
	Random removal	1.16	6.27	11.14
Greedy Insertion- Random Route	Max-Dist-GC	0.00	2.08	5.22
	Closer-GC	0.04	1.08	4.12
	Shaw removal	0.04	2.07	11.21
	Random removal	0.02	3.08	15.14
Regret Greedy Insertion- GCPP	Max-Dist-GC	0.00	1.97	3.92
	Closer-GC	0.01	0.06	1.10
	Shaw removal	0.03	0.71	2.09
	Random removal	0.53	1.05	3.12
Regret Greedy Insertion- Random Route	Max-Dist-GC	0.00	1.13	3.44
	Closer-GC	0.03	0.45	1.19
	Shaw removal	0.02	4.05	12.18
	Random removal	0.11	1.23	2.75
Random Insertion- Random Route	Max-Dist-GC	0.00	1.29	3.10
	Closer-GC	0.00	1.50	4.03
	Shaw removal	0.47	7.26	23.31
	Random removal	0.92	5.81	18.18
Least Infeasibility Violation- GCPP	Max-Dist-GC	0.08	7.08	26.24
	Closer-GC	0.14	1.29	3.15
	Shaw removal	0.03	1.57	4.12
	Random removal	1.42	6.69	17.93
Least Infeasibility Violation- Random Route	Max-Dist-GC	0.00	6.07	11.23
	Closer-GC	0.04	1.07	4.13
	Shaw removal	0.01	2.42	7.04
	Random removal	1.79	7.58	18.63
	Average	0.21	2.86	7.88

The operator pair **Least Infeasibility Violation - GCPP and Max-Dist-GC** stands out with the highest maximum probability (26.24%), indicating a strong preference in certain instances, followed by **Random Insertion - Random Route and Shaw removal**. The operator pair **Least Infeasibility Violation - Random Route and Random removal** exhibits the highest average probability (7.58%), demonstrating consistent preference across different instances, with **Random Insertion - Random Route and Shaw removal** also performing well.

5.5.4 Analysis of the Proposed Formulations

To test the proposed formulations, data set S9 is used. A description of the different instances of the data set S9 can be found in Table 5.6 where n is the number of customers, m is the number of vehicles that can be used, t is the number of days in the planning horizon, D is the maximum duration of a route, Q is the maximum capacity of the vehicles, and visit frequency is the number of customers that must be visited i times.

Table 5.6 Description of data set S9

Instances	n	t	m	D	Q	visit frequency		
						4	2	1
pr01_Cus20c_p4_Q200	20	4	5	_	200	7	4	9
pr02_Cus20r_p4_Q200	20	4	5	_	200	6	8	6
pr03_Cus20rc_p4_Q200	20	4	5	_	200	10	4	6
pr04_Cus10_p4_Q200_dur500	10	4	5	500	200	5	5	0
pr05_Cus15_p4_Q200_dur500	15	4	5	500	200	7	8	0
pr06_Cus20_p4_Q200_dur500	20	4	5	500	200	5	5	10
pr07_Cus15_p4_Q200_dur500	15	4	5	500	200	12	3	0
pr08_Cus15_p4_Q200_dur500	15	4	5	500	200	5	3	7
pr09_Cus20_p4_Q200_dur500	20	4	5	500	500	5	5	10
pr10_Cus25_p4_Q195_dur480	25	4	5	480	195	14	6	5
pr11_Cus30_p4_Q195_dur480	30	4	5	480	195	14	10	6

The problem instances of dataset S9 were solved using the PVRPTW and PVRPTWDCVA formulations with Gurobi, each given a time limit of 7200 seconds. The results are presented in Table 5.7.

An analysis of the PVRPTW formulation results reveals that some solutions exhibit inefficiencies in terms of route crossings and driver consistency. For these instances, incorporating visual attractiveness and driver consistency constraints increases the routing cost by 10.91%, as evidenced by instance pr03_Cus20rc_p4_Q200.

Table 5.7 PVRPTW vs. PVRPTWDCVA on data set S9

Instances	PVRPTW					PVRPTWDCVA		
	OFV	Gap	Time	# crossings	nonDC	OFV	Gap	Time
pr01_Cus20c_p4_Q200	566.69	0	4.58	0	0	566.69	0	3.10
pr02_Cus20r_p4_Q200	1191.65	0	5.65	1	4	1201.62	0	10.91
pr03_Cus20rc_p4_Q200	1152.31	0	121.22	6	0	1277.98	0.009	7200
pr04_Cus10_p4_Q200_dur500	1872.34	0	1.42	2	3	1889.76	0	3.03
pr05_Cus15_p4_Q200_dur500	2112.92	0	30.04	0	2	2132.88	0	64.83
pr06_Cus20_p4_Q200_dur500	1925.08	0	1731.38	1	3	1943.06	0	709.14
pr07_Cus15_p4_Q200_dur500	1761.84	0.101	7200	0	2	1806.60	0.064	7200
pr08_Cus15_p4_Q200_dur500	1525.30	0.125	7200	0	1	1527.30	0.024	7200
pr09_Cus20_p4_Q200_dur500	1616.04	0.141	7200	0	0	1616.04	0.070	7200
pr10_Cus25_p4_Q195_dur480	NA	NA	7200	_	_	NA	NA	7200
pr11_Cus30_p4_Q195_dur480	NA	NA	7200	_	_	NA	NA	7200

Figure 5.1 Solution routes to problem instance pr06 without considering visual attractiveness and driver consistency constraints

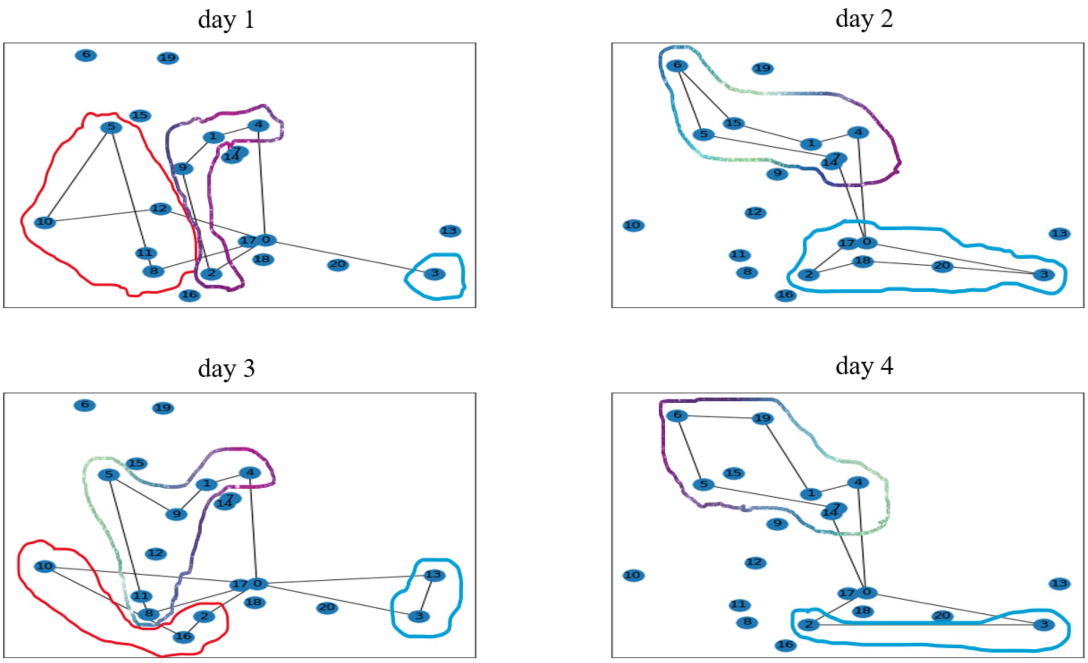
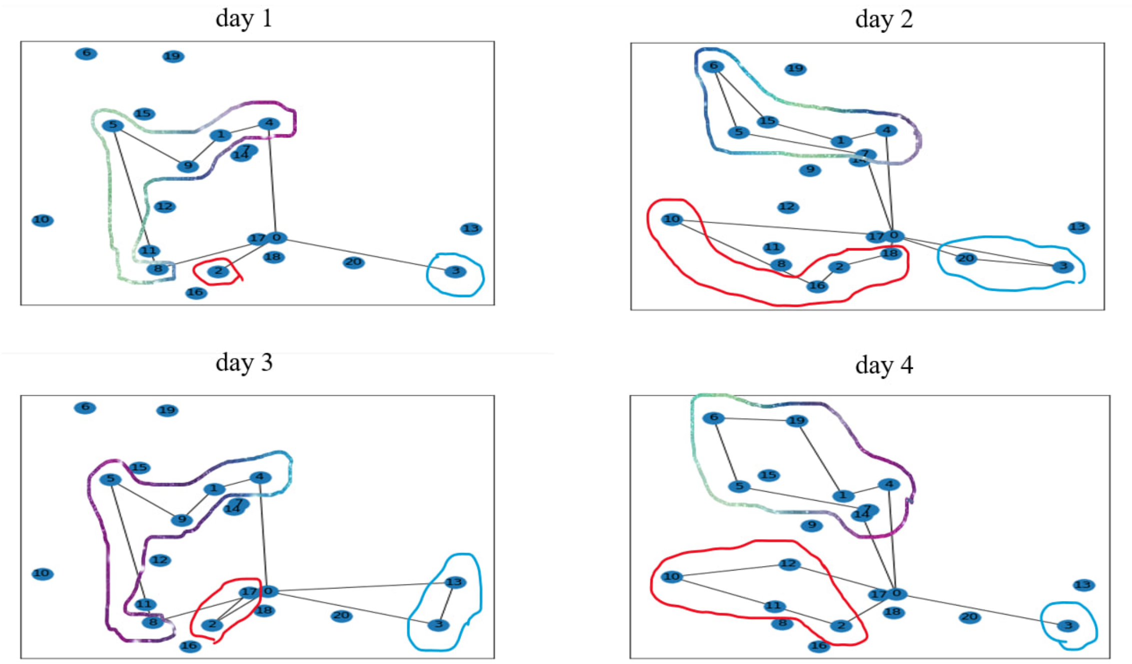


Figure 5.2 Solution routes to problem instance pr06 considering visual attractiveness and driver consistency constraints



Tables 5.1 and 5.2 present the solution routes for the problem instance $pr06$ with and without the inclusion of visual attractiveness and driver consistency constraints. As clearly shown in the figures, when visual attractiveness and driver consistency are not considered, there is a crossing between arcs $(10, 12)$ and $(5, 8)$. It is important to note that crossings involving the depot are excluded from this count. Additionally, customers 2, 5, and 8 are visited by more than one driver in their visits.

5.5.5 Results for Small-size Instances

We consider data set S9 to validate the performance of the proposed PTALNS algorithm. Each problem instance was solved using Gurobi and PTALNS algorithm. Given that the primary goal of PTALNS is to minimize the number of vehicles used, we adopt the following strategy to compare the performance of PTALNS with the Gurobi implementation:

- Allocate 5 vehicles to each instance and solve them using Gurobi,
- Reduce the number of available vehicles to the smallest possible number and solve the instances again using Gurobi,

- Solve each instance with PTALNS twice: once incorporating the local search procedure and once without it.

The results of these comparisons are presented in Table 5.8

Table 5.8 Results of Gurobi vs. PTALNS on Data set S9

Instances	Gurobi				PTALNS (without Local search)		
	5 available vehicles						
	OFV	Gap	Time	veh	OFV	Time	veh
pr01_Cus20c_p4_Q200	566.69	0	3.1	2	566.69	62.71	2
pr02_Cus20r_p4_Q200	1201.62	0	10.91	5	1201.62	74.87	5
pr03_Cus20rc_p4_Q200	1277.98	0.01	3000	4	1277.98	90.32	4
pr04_Cus10_p4_Q200_dur500	1889.76	0	3.03	3	1889.76	54.23	3
pr05_Cus15_p4_Q200_dur500	2132.88	0	64.83	3	2132.88	60.45	3
pr06_Cus20_p4_Q200_dur500	1943.06	0	709.14	3	1943.06	84.54	3
pr07_Cus15_p4_Q200_dur500	1827.30	0.13	3000	2	1806.60	68.73	2
pr08_Cus15_p4_Q200_dur500	1527.30	0.13	1284/3000	2	1527.30	73.79	2
pr09_Cus20_p4_Q200_dur500	1625.57	0.15	3000	2	1616.04	83.04	2
pr10_Cus25_p4_Q195_dur480	NA	NA	3000	NA	3045.42	81.13	3
pr11_Cus30_p4_Q195_dur480	NA	NA	3000	NA	3205.65	90.39	3

Instances	Gurobi				PTALNS (with local search)		
	reduced # of vehicles						
	OFV	Gap	Time	veh	OFV	Time	veh
pr01_Cus20c_p4_Q200	566.69	0	3.1	2	566.69	68.35	2
pr02_Cus20r_p4_Q200	1201.62	0	10.91	5	1201.62	91.83	5
pr03_Cus20rc_p4_Q200	1277.98	0.01	3000	4	1277.98	111.12	4
pr04_Cus10_p4_Q200_dur500	1901.36	0	0.84	2	1901.36	58.76	2
pr05_Cus15_p4_Q200_dur500	2154.76	0	22.62	2	2154.76	70.25	2
pr06_Cus20_p4_Q200_dur500	1965.82	0	112.72	2	1965.82	93.01	2
pr07_Cus15_p4_Q200_dur500	1806.60	0.09	1009/3000	2	1806.60	65.33	2
pr08_Cus15_p4_Q200_dur500	1527.30	0.09	64/3000	2	1527.30	71.09	2
pr09_Cus20_p4_Q200_dur500	1616.04	0.12	2010/3000	2	1616.04	84.27	2
pr10_Cus25_p4_Q195_dur480	NA	NA	3000	NA	3045.42	123.18	3
pr11_Cus30_p4_Q195_dur480	NA	NA	3000	NA	3205.65	112.76	3

In this table, the blocks of 4 columns present the results obtained by Gurobi with 5 and the reduced numbers for available vehicles, while the blocks of 3 columns correspond to PTALNS results. Columns '#veh' indicate the number of vehicles used in obtained solutions. In the columns labeled 'Time', an entry such as '1284/3000' means that Gurobi found the reported solution in 1284 seconds for the first time. The term "NA" indicates that Gurobi could not find any feasible solution for the instance corresponding to that row within the time limit. We performed PTALNS for 2400 iterations. The remaining parameters of PTALNS are as follows; $M = 4$, $\delta = 20$, $\gamma = 3$, $\phi = 20$, and $\alpha = 0.25$. The number of iterations for each replica is set to 100.

According to Table 5.8,

- For the first three instances, Gurobi returned the optimal solutions for the first two and failed to prove optimality for the third instance. PTALNS, both with and without incorporating the local search procedure, also returned the same solutions for these instances in a reasonable computational time.
- Gurobi solved the next three instances with tight time windows to optimality, considering both five and a reduced number of available vehicles. PTALNS, with and without incorporating the local search procedure, also returned the same solutions for these instances in a reasonable computational time.
- With five available vehicles, Gurobi was unable to reach optimal solutions for instances pr07 and pr09 within the time limit. However, PTALNS, without the local search procedure, achieved better solutions for these instances than Gurobi. For problem instance pr08, both Gurobi and PTALNS found the same solution. Although Gurobi could not prove optimality and hit the time limit, PTALNS reached the same solution in a reasonable runtime. Gurobi also failed to solve instances pr10 and pr11 within the time limit, while PTALNS solved them with three vehicles, with and without using the local search procedure.
- With the reduced number of available vehicles, Gurobi and PTALNS, both with and without the local search procedure, solved instances pr07 to pr09 to the same solution. However, PTALNS outperformed Gurobi in terms of runtime. Reducing the number of available vehicles rendered the first three instances and instances pr10 and pr11 infeasible.

These results demonstrate that PTALNS is a robust and efficient algorithm, capable of finding high-quality solutions within a reasonable timeframe, even for complex instances where Gurobi struggles.

5.5.6 Results for Large-size Instances

We apply the VNS algorithm proposed by Pirkwieser and Raidl (2009c) to data set S6. The results are reported in Table 5.9 where the first column shows the instance name. The second and third columns are the number of intra-route and inter-route crossings in the solution, respectively. In the next column, we have the number of customers who are not visited by the same vehicle on each visit. The last column is the best-obtained objective function value.

Table 5.9 shows that while the solutions generated exhibit zero intra-route crossings,

Table 5.9 Results of PVRPTW on data set S6

Instance	PVRPTW			
	intra-route	inter-route	nonDC	OFV
pr01_Cus48_m3_p4_Q200_dur500	0	12	16	3125.51
pr02_Cus96_m6_p4_Q195_dur480	0	16	38	5488.79
pr03_Cus144_m9_p4_Q190_dur460	0	37	57	7584.01
pr04_Cus192_m12_p4_Q185_dur440	0	47	78	8484.84
pr05_Cus240_m15_p4_Q180_dur420	0	73	101	9433.73
pr06_Cus288_m18_p4_Q175_dur400	0	75	129	11966.40
pr07_Cus72_m5_p6_Q200_dur500	0	18	36	7421.56
pr08_Cus144_m10_p6_Q190_dur475	0	66	72	10581.04
pr09_Cus216_m15_p6_Q180_dur450	0	111	108	15047.56
pr10_Cus288_m20_p6_Q170_dur425	0	236	144	19927.22
pr11_Cus48_m3_p4_Q200_dur500	0	1	9	2284.74
pr12_Cus96_m6_p4_Q195_dur480	0	9	31	4281.02
pr13_Cus144_m9_p4_Q190_dur460	0	13	53	6030.80
pr14_Cus192_m12_p4_Q185_dur440	0	21	70	7107.71
pr15_Cus240_m15_p4_Q180_dur420	0	30	96	7743.30
pr16_Cus288_m18_p4_Q175_dur400	0	53	120	9770.44
pr17_Cus72_m4_p6_Q200_dur500	0	5	24	5687.71
pr18_Cus144_m8_p6_Q190_dur475	0	21	72	8374.35
pr19_Cus216_m12_p6_Q180_dur450	0	46	107	11937.44
pr20_Cus288_m16_p6_Q170_dur425	0	117	144	15378.72

there are inefficiencies in terms of inter-route crossings and driver consistency. Each routing solution has about 13% inter-route crossings on average and approximately 45% of customers visit more than one driver during their visits.

To have visually attractive solution routes and driver consistency, we applied ALNS and PTALNS algorithms to solve the problems of data set S6. The results are reported in Table 5.10, where the columns are as follows: Instance, which denotes the name of the instances; PTALNS Avg., which is the average result obtained by the PTALNS algorithm across five runs; PTALNS Best UB, representing the best upper bound (UB) found by the PTALNS algorithm; ALNS Avg., which is the average result obtained by the ALNS algorithm across five runs; ALNS Best UB, indicating the best UB found by the ALNS algorithm; Best UB^*, which is the best upper bound overall among all the results for that instance; and Imp%, the percentage improvement of PTALNS Avg. over ALNS Avg., calculated as

$$\text{Imp\%} = \left(\frac{\text{ALNS Avg.} - \text{PTALNS Avg.}}{\text{ALNS Avg.}} \right) \times 100$$

According to Table 5.10, for each instance, the PTALNS algorithm consistently achieves lower average results compared to the ALNS algorithm. This indicates that PTALNS generally performs better on average, leading to solutions closer to

Table 5.10 Results of ALNS vs. PTALNS on Data set S6

Instance	PTALNS		ALNS		Best UB^*	Imp%
	Avg.	Best UB	Avg.	Best UB		
pr01-N48m3p4	3255.43	3234.90	3330.82	3247.86	3234.90	2.26
pr02-N96m6p4	5812.76	5695.36	6029.44	5909.68	5695.36	3.59
pr03-N144m9p4	8851.22	8729.97	9080.99	8913.63	8729.97	2.53
pr04-N192m12p4	10609.65	10194.16	10946.80	10676.57	10194.16	3.08
pr05-N240m15p4	11422.73	11097.57	11533.03	11174.00	11097.57	0.96
pr06-N288m18p4	14559.91	14333.24	15068.53	14773.41	14333.24	3.38
pr07-N72m5p6	7895.45	7736.63	8294.13	7981.46	7736.63	4.81
pr08-N144m10p6	13673.36	13530.98	13914.22	13723.12	13530.98	1.73
pr09-N216m15p6	19115.30	18330.03	19887.57	19452.72	18330.03	3.88
pr10-N288m20p6	24609.84	24389.64	25422.40	25322.77	24389.64	3.20
pr11-N48m3p4	2408.91	2400.23	2427.98	2410.18	2400.23	0.79
pr12-N96m6p4	4760.86	4612.43	5236.97	4927.04	4612.43	9.09
pr13-N144m9p4	6814.50	6705.51	7006.06	6901.98	6705.51	2.73
pr14-N192m12p4	7910.67	7793.33	8291.12	8095.04	7793.33	4.59
pr15-N240m15p4	8333.48	8194.43	8787.72	8489.66	8194.43	5.17
pr16-N288m18p4	10463.71	10367.31	11037.51	10836.63	10367.31	5.20
pr17-N72m4p6	6337.97	6138.60	6523.75	6243.38	6138.60	2.85
pr18-N144m8p6	9821.09	9810.71	10002.95	9814.80	9810.71	1.82
pr19-N216m12p6	13474.98	13374.23	13947.04	13562.17	13374.23	3.38
pr20-N288m16p6	17907.60	17485.31	18531.94	18358.78	17485.31	3.37

the best solution across multiple runs.

The PTALNS algorithm finds better best upper bounds compared to ALNS in every instance. The "Best UB" column, which records the best solution found among all runs, always matches the PTALNS Best UB, showing the robustness and effectiveness of PTALNS in the better solutions.

The "Imp%" column shows the percentage improvement of PTALNS Avg. over ALNS Avg. for each instance. This percentage varies across instances, but PTALNS consistently shows a positive improvement, with some instances showing substantial improvements (e.g., pr12-N96m6p4 with 9.09% improvement)

Table 5.11 Results of ALNS vs. PTALNS on Data set S7

PVRPTW		PVRPTWDCVA					
Instance	BKS	PTALNS		ALNS		Best UB^*	Imp%
		Avg.	Best UB	Avg.	Best UB		
P4c101	2766.22	3071.55	3067.93	3228.46	3067.93	3067.93	4.86
P4r101	3434.18	4812.03	4735.65	4945.14	4900.59	4735.65	2.69
P4rc101	3620.70	4843.74	4797.87	4903.26	4810.05	4797.87	1.21
P6c101	3723.20	4291.82	4286.95	4397.48	4287.00	4286.95	2.40
P6r101	4428.88	6436.60	6389.30	6523.44	6460.32	6389.30	1.33
P6rc101	4952.94	7458.18	7424.00	8049.49	7428.40	7424.00	7.35
P8c101	4809.46	5350.57	5267.70	5564.19	5381.24	5267.70	3.84
P8r101	5428.80	7764.14	7737.20	7950.70	7777.70	7737.20	2.35
P8rc101	5840.84	8375.06	8303.69	8675.86	8361.52	8303.69	3.47

Table 5.11 presents the results of the ALNS and PTALNS algorithms on data set

S7. The "BKS" column lists the best-known solutions reported for the PVRPTW. The table follows the same format as Table 5.10. Consistent with the findings for data set S6, the PTALNS algorithm consistently achieves lower average results and better Best UB compared to the ALNS algorithm. According to the "Imp%" column, the percentage improvement of PTALNS Avg. over ALNS Avg. varies across instances, but PTALNS consistently shows a positive improvement, with some instances exhibiting substantial improvements.

For example, the P6rc101 instance demonstrates a notable improvement of 7.35%, indicating that PTALNS performs significantly better in this case. Similarly, other instances such as P4c101 and P8c101 also show notable improvements, with increases of 4.86% and 3.84%, respectively.

Overall, the results suggest that PTALNS is a more effective algorithm compared to ALNS, particularly in achieving lower average results and better best upper bounds, thereby demonstrating its superiority in solving PVRPTWDCVA problems.

It is noteworthy that for problem instances with clustered geographical data, when visual attractiveness and driver consistency constraints are added, there is an average 11.86% increase in the OFV. This increase is 41.56% for random geographical data and 41.52% for random-clustered geographical data.

5.6 Conclusions

We presented an MILP formulation and a PTALNS algorithm to efficiently solve the PVRPTW instances considering visual attractiveness and driver consistency restrictions. We evaluated the performance of the PVRPTWDCVA formulation and assessed the efficiency of the PTALNS algorithm. For small-size instances, the PTALNS algorithm was tested and compared against the Gurobi solver with different numbers of available vehicles. PTALNS consistently found high-quality solutions within reasonable computational times, even for instances where Gurobi struggled to reach optimality or feasible solutions within the time limit.

Large-size instances from data set S6 were addressed using a VNS algorithm, revealing that routing has inefficiencies regarding route crossings and driver consistency. There are on average 50.35 inter-route crossings in each solution and approximately 45 percent of customers visit more than one driver during their visits. To have appealing routes and considering driver consistency the instances in data set S6 are also solved by ALNS and PTALNS algorithms. In the comparison between ALNS and PTALNS, the latter is more robust and outperforms in solving all instances. Furthermore, PTALNS also demonstrated superior performance over ALNS in solving instances from data set S7.

There are several potential directions for extending the current work. First, exploring different metrics for visual attractiveness could provide valuable insights into their impact on routing efficiency and overall effectiveness. Second, during the destruction and construction phases, the constraints related to visual attractiveness and driver consistency could be relaxed. Violations of these constraints could then be incorporated as penalty costs within the objective function, allowing for a more flexible approach to solution optimization. Finally, leveraging data mining techniques to analyze historical driver behavior could inform routing decisions, potentially leading to more efficient and driver-friendly routes based on past performance and preferences.

5.7 Chapter Conclusion

In Chapter 3, we overview the PVRP focusing on modeling approaches and exact solutions to be obtained by a commercial solver. We develop a new MILP formulation of the PVRP using vehicle flow variables and employ families of valid inequalities and optimality cuts to tighten the formulation. Two existing prominent PVRP formulations in the literature referred to as the load-based formulation and the cut-based formulation are also investigated. The main differences among the formulations are the representation of vehicle information and the SECs. The cut-based formulations use decision variables explicitly including a vehicle index. The SECs of the cut-based formulations have an exponential size and need to be separated. In the load-based formulations and the MTZ-based formulations, auxiliary continuous variables are used to define the constraints preventing the formation of subtours and both formulations are compact, i.e., they do not involve exponentially many constraints. Since the continuous variables defined in the MTZ-based formulation have one dimension less than those used in the load-based formulation, the MTZ-based formulation has smaller number of variables compared to the load-based formulation. We also extend these formulations to model the PVRPTW and apply families of valid inequalities to tighten the formulations. Comprehensive computational experiments are conducted on seven sets of benchmark instances with different characteristics to compare the performances of the alternative formulations. The results of the computational experiments on the PVRP data sets show that the cut-based model and the load-based model are competitive in solving small instances. When it comes to solving medium instances, the load-based model overtakes the cut-based model. While the results indicate the difficulty of solving most of the PVRP instances of large size for the cut-based model and the load-based model, the MTZ-based formulations are robust and consistent in producing good-quality solutions.Considering the PVRPTW, the results obtained by alternative models with data sets S6 and S7 show that the cut-based model is only successful in solving two out of 35 instances. In general, the MTZ-based formulations outperform the load-based formulations. For the instances solved to optimality, they reach the solution faster, and for those instances for which an optimal solution cannot be identified within the specified time limit, they return solutions of higher quality when compared to the load-based formulations. The superiority of the MTZ-based formulations are more evident in solving larger instances. The goal of this chapter is to compare and provide insights into the computational performance of widely used formulation approaches in the literature that can be implemented and solved directly by state-of-the-art commercial solver capabilities. Other formulation approaches are available (e.g. set partitioning) for which an exact solution cannot be obtained directly by a solver, but only by employing efficient optimization algorithms (e.g. column generation and branch-and-price). Extending the computational analysis to cover a broader spectrum of exact solution methodologies may provide additional insights into the performance of different formulation and solution techniques with respect to run times and solution quality. Moreover, (parts of) the formulation approaches considered in this study are easily applicable to other variants of the PVRP and exploring the computational performance on different problem variants may also be an interesting direction for future research.

In the first part of Chapter 4, we proposed the LBBD and BAC algorithms to solve the PVRP. Computational results demonstrate that BAC outperforms LBBD in medium-sized instances. However, a comparison between the BAC algorithm and the UC1 formulation reveals that the convergence of BAC to the optimal solution is slow, and the time required to solve the VRP subproblem is considerable. To expedite the BAC algorithm, we attempted to enhance the LB by incorporating cuts derived from the dual of the LP relaxation of the subproblem at each iteration. However, this approach did not yield any significant improvement in the LB and the cuts appeared to be redundant. we also aimed to enhance the speed of solving the VRP subproblem. Initially, we applied an exact column generation algorithm for this purpose. Unfortunately, our proposed column generation approach did not yield significant improvements in solution time. Consequently, we refined and adapted this approach specifically for the PVRP problem, aiming to solve the PVRP directly using the column generation algorithm. In the second part of Chapter 4, we propose a column generation-based heuristic to solve the PVRP. In the pricing problem, we applied Martinelli et al. (2014) algorithm. However, to maintain elementarity in solving the pricing sub-problem, we applied two different approaches: one proposed by Dayarian et al. (2015) and a hybrid approach. At the end of the algorithm, if the optimal solution to the master problem is fractional, we solve the integer RMP, allowing visits to customers more frequently than their specified frequencies. Based on this solution, we generate a new Ω and enrich it with additional columns. We then resolve the problem using the updated Ω. To expedite the column generation-based heuristic, we attempt to identify negative reduced cost columns by first exploring the column pool, then implementing the truncated-search version of the dynamic programming algorithm, and finally implementing the full-search version of the dynamic programming algorithm if the other approaches fail. After finding a predetermined number of elementary negative reduced cost columns, we terminate any pricing iteration further to control time. The results show that the proposed algorithm produces results very close to the best-known solutions in the literature, with an average deviation of only 0.21%. This highlights the effectiveness of the algorithm, especially given that most instances show either no deviation or very minor deviations from the best-known solution in the literature.

Time window and non-crossing intra-route constraints can be seamlessly integrated into the proposed column generation algorithm by incorporating them directly into the pricing problem. This approach ensures that these constraints are enforced during the column generation process. However, integrating non-crossing inter-route and driver consistency constraints requires a different approach. These constraints need to be included in the master problem, where the master problem verifies the absence of arc intersections between different routes. As the size of Ω increases, this verification process can become computationally intensive.

In Chapter 5, we presented an MILP formulation and a PTALNS algorithm to efficiently solve the PVRPTW instances considering visual attractiveness and driver consistency restrictions. We evaluated the performance of the PVRPTWDCVA formulation and assessed the efficiency of the PTALNS algorithm. For small-size instances, the PTALNS algorithm was tested and compared against the Gurobi solver with different numbers of available vehicles. PTALNS consistently found high-quality solutions within reasonable computational times, even for instances where Gurobi struggled to reach optimality or feasible solutions within the time limit. Large-size instances from data set S6 were addressed using a VNS algorithm, revealing that routing has inefficiencies regarding route crossings and driver consistency. There are on average 50.35 inter-route crossings in each solution and approximately 45 percent of customers visit more than one driver during their visits. To have appealing routes and considering driver consistency the instances in data set S6 are also solved by ALNS and PTALNS algorithms. In the comparison between ALNS and PTALNS, the latter is more robust and outperforms in solving all instances. Furthermore, PTALNS also demonstrated superior performance over ALNS in solving instances from data set S7.

Future research can explore various algorithmic enhancements to improve the performance of the column generation approach to solve PVRP. These include the implementation of cutting planes, advanced branching strategies, and sophisticated data structures for label storage during the pricing stage. Additionally, the integration of metaheuristics could further enhance the efficiency and effectiveness of the method. There are also several potential directions for extending the current work on PVRPTWVADC. First, exploring different metrics for visual attractiveness could provide valuable insights into their impact on routing efficiency and overall effectiveness. Second, during the destruction and construction phases, the constraints related to visual attractiveness and driver consistency could be relaxed. Violations of these constraints could then be incorporated as penalty costs within the objective function, allowing for a more flexible approach to solution optimization. Finally, leveraging data mining techniques to analyze historical driver behavior could inform routing decisions, potentially leading to more efficient and driver-friendly routes based on past performance and preferences.

6. Conclusions and Future Research

In Chapter 3, we overview the PVRP focusing on modeling approaches and exact solutions to be obtained by a commercial solver. We develop a new MILP formulation of the PVRP using vehicle flow variables and employ families of valid inequalities and optimality cuts to tighten the formulation. Two existing prominent PVRP formulations in the literature referred to as the load-based formulation and the cut-based formulation are also investigated. The main differences among the formulations are the representation of vehicle information and the SECs. The cut-based formulations use decision variables explicitly including a vehicle index. The SECs of the cut-based formulations have an exponential size and need to be separated. In the load-based formulations and the MTZ-based formulations, auxiliary continuous variables are used to define the constraints preventing the formation of subtours and both formulations are compact, i.e., they do not involve exponentially many constraints. Since the continuous variables defined in the MTZ-based formulation have one dimension less than those used in the load-based formulation, the MTZ-based formulation has smaller number of variables compared to the load-based formulation. We also extend these formulations to model the PVRPTW and apply families of valid inequalities to tighten the formulations. Comprehensive computational experiments are conducted on seven sets of benchmark instances with different characteristics to compare the performances of the alternative formulations. The results of the computational experiments on the PVRP data sets show that the cut-based model and the load-based model are competitive in solving small instances. When it comes to solving medium instances, the load-based model overtakes the cut-based model. While the results indicate the difficulty of solving most of the PVRP instances of large size for the cut-based model and the load-based model, the MTZ-based formulations are robust and consistent in producing good-quality solutions.Considering the PVRPTW, the results obtained by alternative models with data sets S6 and S7 show that the cut-based model is only successful in solving two out of 35 instances. In general, the MTZ-based formulations outperform the load-based formulations. For the instances solved to optimality, they reach the solution faster, and for those instances for which an optimal solution cannot be identified within the specified time limit, they return solutions of higher quality when compared to the load-based formulations. The superiority of the MTZ-based formulations are more evident in solving larger instances. The goal of this chapter is to compare and provide insights into the computational performance of widely used formulation approaches in the literature that can be implemented and solved directly by state-of-the-art commercial solver capabilities. Other formulation approaches are available (e.g. set partitioning) for which an exact solution cannot be obtained directly by a solver, but only by employing efficient optimization algorithms (e.g. column generation and branch-and-price). Extending the computational analysis to cover a broader spectrum of exact solution methodologies may provide additional insights into the performance of different formulation and solution techniques with respect to run times and solution quality. Moreover, (parts of) the formulation approaches considered in this study are easily applicable to other variants of the PVRP and exploring the computational performance on different problem variants may also be an interesting direction for future research.

In the first part of Chapter 4, we proposed the LBBD and BAC algorithms to solve the PVRP. Computational results demonstrate that BAC outperforms LBBD in medium-sized instances. However, a comparison between the BAC algorithm and the UC1 formulation reveals that the convergence of BAC to the optimal solution is slow, and the time required to solve the VRP subproblem is considerable. To expedite the BAC algorithm, we attempted to enhance the LB by incorporating cuts derived from the dual of the LP relaxation of the subproblem at each iteration. However, this approach did not yield any significant improvement in the LB and the cuts appeared to be redundant. we also aimed to enhance the speed of solving the VRP subproblem. Initially, we applied an exact column generation algorithm for this purpose. Unfortunately, our proposed column generation approach did not yield significant improvements in solution time. Consequently, we refined and adapted this approach specifically for the PVRP problem, aiming to solve the PVRP directly using the column generation algorithm. In the second part of Chapter 4, we propose a column generation-based heuristic to solve the PVRP. In the pricing problem, we applied Martinelli et al. (2014) algorithm. However, to maintain elementarity in solving the pricing sub-problem, we applied two different approaches: one proposed by Dayarian et al. (2015) and a hybrid approach. At the end of the algorithm, if the optimal solution to the master problem is fractional, we solve the integer RMP, allowing visits to customers more frequently than their specified frequencies. Based on this solution, we generate a new Ω and enrich it with additional columns. We then resolve the problem using the updated Ω. To expedite the column generation-based heuristic, we attempt to identify negative reduced cost columns by first exploring the column pool, then implementing the truncated-search version of the dynamic programming algorithm, and finally implementing the full-search version of the dynamic programming algorithm if the other approaches fail. After finding a predetermined number of elementary negative reduced cost columns, we terminate any pricing iteration further to control time. The results show that the proposed algorithm produces results very close to the best-known solutions in the literature, with an average deviation of only 0.21%. This highlights the effectiveness of the algorithm, especially given that most instances show either no deviation or very minor deviations from the best-known solution in the literature.

Time window and non-crossing intra-route constraints can be seamlessly integrated into the proposed column generation algorithm by incorporating them directly into the pricing problem. This approach ensures that these constraints are enforced during the column generation process. However, integrating non-crossing inter-route and driver consistency constraints requires a different approach. These constraints need to be included in the master problem, where the master problem verifies the absence of arc intersections between different routes. As the size of Ω increases, this verification process can become computationally intensive.

In Chapter 5, we presented an MILP formulation and a PTALNS algorithm to efficiently solve the PVRPTW instances considering visual attractiveness and driver consistency restrictions. We evaluated the performance of the PVRPTWDCVA formulation and assessed the efficiency of the PTALNS algorithm. For small-size instances, the PTALNS algorithm was tested and compared against the Gurobi solver with different numbers of available vehicles. PTALNS consistently found high-quality solutions within reasonable computational times, even for instances where Gurobi struggled to reach optimality or feasible solutions within the time limit. Large-size instances from data set S6 were addressed using a VNS algorithm, revealing that routing has inefficiencies regarding route crossings and driver consistency. There are on average 50.35 inter-route crossings in each solution and approximately 45 percent of customers visit more than one driver during their visits. To have appealing routes and considering driver consistency the instances in data set S6 are also solved by ALNS and PTALNS algorithms. In the comparison between ALNS and PTALNS, the latter is more robust and outperforms in solving all instances. Furthermore, PTALNS also demonstrated superior performance over ALNS in solving instances from data set S7.

Future research can explore various algorithmic enhancements to improve the performance of the column generation approach to solve PVRP. These include the implementation of cutting planes, advanced branching strategies, and sophisticated data structures for label storage during the pricing stage. Additionally, the integration of metaheuristics could further enhance the efficiency and effectiveness of the method. There are also several potential directions for extending the current work on PVRPTWVADC. First, exploring different metrics for visual attractiveness could provide valuable insights into their impact on routing efficiency and overall effectiveness. Second, during the destruction and construction phases, the constraints related to visual attractiveness and driver consistency could be relaxed. Violations of these constraints could then be incorporated as penalty costs within the objective function, allowing for a more flexible approach to solution optimization. Finally, leveraging data mining techniques to analyze historical driver behavior could inform routing decisions, potentially leading to more efficient and driver-friendly routes based on past performance and preferences.

BIBLIOGRAPHY

Alves., F., Alvelos., F., Rocha., A. M. A. C., Pereira., A. I., and Leitao., P. (2019). Periodic vehicle routing problem in a health unit. In *Proceedings of the 8th International Conference on Operations Research and Enterprise Systems - Volume 1: ICORES,*, pages 384–389. SciTePress.

Archetti, C., Fernández, E., and Huerta-Muñoz, D. L. (2017). The flexible periodic vehicle routing problem. *Computers & Operations Research*, 85:58–70.

Archetti, C., Jabali, O., and Speranza, M. G. (2015a). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134.

Archetti, C., Jabali, O., and Speranza, M. G. (2015b). Multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 61:122–134.

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. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380.

Battarra, M., Erdoğan, G., and Vigo, D. (2014). Exact algorithms for the clustered vehicle routing problem. *Operations Research*, 62(1):58–71.

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

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252. cited By 2296.

Bettinelli, A., Ceselli, A., and Righini, G. (2011). A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 19(5):723–740.

Bosch, A. (2014). Seed selection in multi-period planning with time windows. Master's thesis, University of Twente.

Braekers, K. and Kovacs, A. A. (2016a). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological*, 94:355–377.

Braekers, K. and Kovacs, A. A. (2016b). A multi-period dial-a-ride problem with driver consistency. *Transportation Research Part B: Methodological*, 94:355–377.

Cacchiani, V., Hemmelmayr, V., and Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, 163:53 – 64.

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

Campelo, P., Neves-Moreira, F., Amorim, P., and Almada-Lobo, B. (2019). Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. *European Journal of Operational Research*, 273(1):131–145.

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.

Constantino, M., Gouveia, L., Mourão, M. C., and Nunes, A. C. (2015). The mixed capacitated arc routing problem with non-overlapping routes. *European Journal of Operational Research*, 244(2):445–456.

Contardo, C. and Martinelli, R. (2014). A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146.

Çopur, O., Yıldız, M., Göven, S., Güneri, A. Ö., Yavuz, A. B., Gökçe, M. A., and Yurtseven, C. (2020). Periodic route optimization for fmcg distributors. In Durakbasa, N. M. and Gençyılmaz, M. G., editors, *Proceedings of the International Symposium for Production Research 2019*, pages 748–759, Cham. Springer International Publishing.

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., Laporte, G., and Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52(8):928–936.

Dayarian, I., Crainic, T. G., Gendreau, M., and Rei, W. (2015). A branch-and-price approach for a multi-period vehicle routing problem. *Computers & Operations Research*, 55:167–184.

Dayarian, I., Crainic, T. G., Gendreau, M., and Rei, W. (2016). An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95:95–123.

Elbek, M. and Wøhlk, S. (2016). A variable neighborhood search for the multi-period collection of recyclable materials. *European Journal of Operational Research*, 249(2):540–550.

Expósito, A., Raidl, G. R., Brito, J., and Moreno-Pérez, J. A. (2018). Grasp-vns for a periodic vrp with time windows to deal with milk collection. In *Computer Aided Systems Theory–EUROCAST 2017: 16th International Conference, Las Palmas de Gran Canaria, Spain, February 19-24, 2017, Revised Selected Papers, Part I 16*, pages 299–306. Springer.

Fisher, M. L. and Jaikumar, R. (1981). A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124.

Foster, B. A. and Ryan, D. M. (1976). An integer programming approach to the vehicle scheduling problem. *Journal of the Operational Research Society*, 27(2):367–384.

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. M., Smilowitz, K. R., and Tzur, M. (2008). The period vehicle routing problem and its extensions. In *The vehicle routing problem: latest advances and new challenges*, pages 73–102. Springer.

Gaudioso, M. and Paletta, G. (1992). A heuristic for the periodic vehicle routing problem. *Transportation Science*, 26(2):86–92.

Gehring, H. and Homberger, J. (1999). A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. In *Proceedings of EUROGEN99*, volume 2, pages 57–64. Citeseer.

Goeke, D., Roberti, R., and Schneider, M. (2019). Exact and heuristic solution of the consistent vehicle-routing problem. *Transportation Science*, 53(4):1023–1042.

Gonçalves, L. B., Ochi, L. S., and Martins, S. L. (2005). A grasp with adaptive memory for a period vehicle routing problem. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 1, pages 721–727. IEEE.

Gretton, C., Kilby, P., and Grzybowska, H. (2013). A study of shape penalties in vehicle routing.

Groër, C., Golden, B., and Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing & service operations management*, 11(4):630–643.

Hasle, G., Kloster, O., and Smedsrud, M. (2011). *A capacitated clustering-based method for newspaper delivery routing.*

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

Hollis, B. and Green, P. (2012). Real-life vehicle routing with time windows for visual attractiveness and operational robustness. *Asia-Pacific Journal of Operational Research*, 29(04):1250017.

Hooker, J. (2000). *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley.

Hooker, J. N. and Ottosson, G. (2003). Logic-based benders decomposition. *Mathematical Programming*, 96(1):33–60.

Huerta Muñoz, D. L. (2018). The flexible periodic vehicle routing problem: modeling alternatives and solution techniques. Master's thesis, Universitat Politècnica de Catalunya.

Kant, G., Jacks, M., and Aantjes, C. (2008). Coca-cola enterprises optimizes vehicle routes for efficient product delivery. *Interfaces*, 38(1):40–50.

Kara, I., Laporte, G., and Bektas, T. (2004). A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793 – 795.

Kilby, P., Prosser, P., and Shaw, P. (1999). *Guided local search for the vehicle routing problem with time windows*, pages 473–486. Springer.

Kilby, P. and Verden, A. (2011). Flexible routing combing constraint programming, large neighbourhood search, and feature-based insertion. In *Proceedings 2nd Workshop on Artificial Intelligence and Logistics (AILOG'11)*, pages 43–49.

Kim, B.-I., Kim, S., and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642.

Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2015a). The generalized consistent vehicle routing problem. *Transportation Science*, 49(4):796–816.

Kovacs, A. A., Parragh, S. N., and Hartl, R. F. (2015b). The multi-objective generalized consistent vehicle routing problem. *European Journal of Operational Research*, 247(2):441–458.

Larrain, H., Coelho, L. C., Archetti, C., and Speranza, M. G. (2019). Exact solution methods for the multi-period vehicle routing problem with due dates. *Computers & Operations Research*, 110:148–158.

Li, H. and Lim, A. (2001). A metaheuristic for the pickup and delivery problem with time windows. In *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, pages 160–167. IEEE.

Lu, Q. and Dessouky, M. M. (2006). A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, 175(2):672–687.

Luo, Z., Qin, H., Che, C., and Lim, A. (2015). On service consistency in multi-period vehicle routing. *European Journal of Operational Research*, 243(3):731–744.

Martinelli, R., Pecin, D., and Poggi, M. (2014). Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, 239(1):102–111.

Matis, P. (2008). Decision support system for solving the street routing problem. *Transport*, 23(3):230–235.

Matis, P. and Koháni, M. (2011). Very large street routing problem with mixed transportation mode. *Central European Journal of Operations Research*, 19:359–369.

Matos, A. C. and Oliveira, R. C. (2004). An experimental study of the ant colony system for the period vehicle routing problem. In *Ant Colony Optimization and Swarm Intelligence: 4th International Workshop, ANTS 2004, Brussels, Belgium, September 5-8, 2004. Proceedings 4*, pages 286–293. Springer.

Mesa, J. P., Montoya, A., Ramos-Pollán, R., and Toro, M. (2024). A bi-objective approach to last-mile delivery routing considering driver preferences.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329.

Montoya-Torres, J. R., López Franco, J., Nieto Isaza, S., Felizzola Jiménez, H., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

Mourgaya, M. and Vanderbeck, F. (2007). Column generation based heuristic for tactical planning in multi-period vehicle routing. *European Journal of Operational Research*, 183(3):1028–1041.

Ochi, L. S. and Rocha, M. L. (2000). A new hybrid evoltionary algorithm for the vehicle routing and scheduling problems. In *IS*, pages 135–138.

Ozbaygin, G., Karasan, O. E., Savelsbergh, M., and Yaman, H. (2017). A branch-and-price algorithm for the vehicle routing problem with roaming delivery locations. *Transportation Research Part B: Methodological*, 100:115–137.

Pirkwieser, S. and Raidl, G. (2009a). Boosting a variable neighborhood search for the periodic vehicle routing problem with time windows by ILP techniques. In *Preprints of the 8th Metaheuristic International Conference (MIC 2009), Hamburg, Germany.*

Pirkwieser, S. and Raidl, G. R. (2009b). Multiple variable neighborhood search enriched with ilp techniques for the periodic vehicle routing problem with time windows. In *International Workshop on Hybrid Metaheuristics*, pages 45–59. Springer.

Pirkwieser, S. and Raidl, G. R. (2009c). A variable neighborhood search for the periodic vehicle routing problem with time windows. In *Proceedings of the 9th EU/MEeting on Metaheuristics for Logistics and Vehicle Routing.*

Pirkwieser, S. and Raidl, G. R. (2010). Multilevel variable neighborhood search for periodic routing problems. In *Evolutionary Computation in Combinatorial Optimization: 10th European Conference, EvoCOP 2010, Istanbul, Turkey, April 7-9, 2010. Proceedings 10*, pages 226–238. Springer.

Polacek, M., Hartl, R. F., Doerner, K., and Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics*, 10:613–627.

Poot, A., Kant, G., and Wagelmans, A. P. M. (2002). A savings based method for real-life vehicle routing problems. *Journal of the Operational Research Society*, 53(1):57–68.

Rastani, S. and Çatay, B. (2023). A large neighborhood search-based matheuristic for the load-dependent electric vehicle routing problem with time windows. *Annals of Operations Research*, pages 1–33.

Riazi, S., Seatzu, C., Wigström, O., and Lennartson, B. (2013). Benders/gossip methods for heterogeneous multi-vehicle routing problems. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–6. IEEE.

Rocha, D., Aloise, D., Aloise, D. J., and Contardo, C. (2022). Visual attractiveness in vehicle routing via bi-objective optimization. *Computers & Operations Research*, 137:105507.

Rodriguez-Martin, I., Salazar-Gonzalez, J.-J., and Yaman, H. (2019). The periodic vehicle routing problem with driver consistency. *European Journal of Operational Research*, 273(2):575–584.

Rodríguez-Martín, I. and Yaman, H. (2022). Periodic vehicle routing problem with driver consistency and service time optimization. *Transportation Research Part B: Methodological*, 166:468–484.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472.

Rossit, D. G., Vigo, D., Tohmé, F., and Frutos, M. (2016). Improving visual attractiveness in capacitated vehicle routing problems: a heuristic algorithm. In *XVIII Latin-Iberoamerican Conference on Operations Research-CLAIO*, page 749.

Rossit, D. G., Vigo, D., Tohmé, F., and Frutos, M. (2019). Visual attractiveness in routing problems: A review. *Computers & Operations Research*, 103:13–34.

Rothenbächer, A.-K. (2019). Branch-and-price-and-cut for the periodic vehicle routing problem with flexible schedule structures. *Transportation Science*, 53(3):850–866.

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

Russell, R. A. and Gribbin, D. (1991). A multiphase approach to the period routing problem. *Networks*, 21(7):747–765.

Sahoo, S., Kim, S., Kim, B.-I., Kraas, B., and Popov Jr, A. (2005). Routing optimization for waste management. *Interfaces*, 35(1):24–36.

Savelsbergh, M. (1990). A parallel insertion heuristic for vehicle routing with side constraints. *Statistica Neerlandica*, 44(3):139–148.

Schneider, M., Stenger, A., Schwahn, F., and Vigo, D. (2015). Territory-based vehicle routing in the presence of time-window constraints. *Transportation Science*, 49(4):732–751.

Smilowitz, K., Nowak, M., and Jiang, T. (2013). Workforce management in periodic delivery operations. *Transportation Science*, 47(2):214–230.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.

Surya Sahoo PhD, P. (2004). Balanced clustering algorithms for improving shapes on vehicle routing problems. In *IIE Annual Conference. Proceedings*, page 1. Institute of Industrial and Systems Engineers (IISE).

Tan, C. and Beasley, J. (1984). A heuristic algorithm for the period vehicle routing problem. *Omega*, 12(5):497–504.

Tang, H. and Hu, M. (2005). Dynamic vehicle routing problem with multiple objectives: Solution framework and computational experiments. *Transportation research record*, 1923(1):199–207.

Tang, H. and Miller-Hooks, E. (2006). Interactive heuristic for practical vehicle routing problem with solution shape constraints. *Transportation research record*, 1964(1):9–18.

Toth, P. and Vigo, D. (2014). Software tools and emerging technologies for vehicle routing and intermodal transportation. *en Vehicle Routing: Problems, Methods, and Applications*, 2:355–384.

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.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2013). A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Comput. Oper. Res.*, 40(1):475–489.

Voigt, S. (2024). A review and ranking of operators in adaptive large neighborhood search for vehicle routing problems. *European Journal of Operational Research*.

Yang, M., Ni, Y., and Yang, L. (2021). A multi-objective consistent home healthcare routing and scheduling problem in an uncertain environment. *Computers & Industrial Engineering*, 160:107560.

Yao, Y., Van Woensel, T., Veelenturf, L. P., and Mo, P. (2021). The consistent vehicle routing problem considering path consistency in a road network. *Transportation Research Part B: Methodological*, 153:21–44.

Zhou, C., Tan, Y., Liao, L., and Liu, Y. (2006). Solving the multi-vehicle pick-up and delivery problem with time widows by new construction heuristic. In *Sixth International Conference on Intelligent Systems Design and Applications*, volume 2, pages 1035–1042. IEEE.

Zhu, J., Zhu, W., Che, C. H., and Lim, A. (2008). A vehicle routing system to solve a periodic vehicle routing problem for a food chain in hong kong. In *AAAI*, pages 1763–1768.

APPENDIX A