Improving Distribution Efficiency in Cash Supply Chains

Annelieke Baller

Reading committee:

prof.dr. J.A. Dos Santos Gromicho
prof.dr. R.D. van der Mei
prof.dr.ing. G.J. Woeginger
dr. C. Archetti
dr. R. Spliet

The author was supported by the Netherlands Organisation for Scientific Research (NWO), project number 407-13-050.

The book is printed by Drukkerij Haveka. The illustration on the cover is made by Remco Wetzels (www.remcowetzels.nl).

This dissertation is number 43 in the ABRI dissertation series.

ISBN: 978 90 361 0584 2

VRIJE UNIVERSITEIT

Improving Distribution Efficiency in Cash Supply Chains

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. V. Subramaniam,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de School of Business and Economics
op donderdag 28 november 2019 om 9.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Annemarie Cornelieke Baller

geboren te Purmerend

promotoren: prof.dr. W.E.H. Dullaert
prof.dr. L. Stougie
copromotor: dr.ir. S. Dabia

Dankwoord

You laugh and you cry, it's a hell of a ride[1]

Although the song that contains this sentence is about love, I think it is as much, or even more, applicable to doing a PhD. It has been such an incredible journey and I am so proud to be finishing my PhD trajectory with this dissertation.

First of all, I want to thank my supervisors Wout, Leen and Said. Wout, thank you for encouraging me to do the best job possible. While enthusiastically using your red pen to mark my drafts, you triggered me to think about the business implications of my work. That resulted in more elaborate introductions and conclusions. Thank you for your support in this process. Leen, thank you for being a very much appreciated second promotor who was always concerned about my well-being. You relied on my knowledge and experience. Asking me to explain some technique to you that I was familiar with, challenged me to explain myself clearly and consistently and this helped me to get my thoughts clear for myself. Said, how much did we grow together in our roles as supervisor and PhD candidate! You taught me to make my own choices and to defend them. Also, I am grateful that you encouraged me to sometimes say 'no' to some interesting addition to a research project or code implementation.

I am grateful to the reading committee members for investing time to read my dissertation and to give their approval of my work. Joaquim, thank you for being chair and thank you for introducing me to academic research many years ago. You challenged me to read quite complicated articles for my bachelor's and master's theses and you reassured me afterwards when I presented a page full of questions: 'I didn't expect you to fully understand these articles!'. I still wanted to understand.

I would like to thank all colleagues from the VU that contributed in some way to this dissertation, often indirectly. Maaike and Christos, the project wouldn't have been half the fun if we couldn't have shared our experiences on the cash supply chain topic, our working days at GSN and the trips to conferences. Maaike, I will happily remember our habit of getting lost, especially on our way to a highly secured cash center. Dung and Roman, you were always taking care of the group and having a personal connection with everyone, the group wasn't the same without you guys. Amir, thank you for all the deep conversations on culture, religion, life and everything else! A lot of gratitude to Nurmala for sharing her advice for my trip to Indonesia and to Julian for being my fellow vegetarian. To Bianca for the gezelligheid in the office and your interest in our research. And to Florence, for the practical questions and solutions, and the many nice talks! I learned to relax a bit more from Marije, and I now know so much about hockey, birds and, more recently, cheese. Thank you for the relaxing walks that we made just to watch some birds. Nienke, I am so thankful for your support in both good and less

[1] *River of Love*, The Shires

good times. I deeply valued, and still do, your very detailed considerations on research, houses, career and many other topics. I will definitely cherish your down-to-earth West-Frisian comments and reflections. Also, thanks for our amazing trip through Canada with a very wet train journey, shady motels and incredibly beautiful views. Thanks to Daniele, Sander, Yousef, Jelle, Eirini, Roberto, Gabor, David, Dirk, Denise, Nicola, Claudio, Rodrigo, Rosario and Teun for sharing your wisdom and experience.

Also, I am grateful to the colleagues from the Econometrics and Operations Research department for being my second base during my PhD. Especially, Joost and Martijn, thank you for all our discussions and fun lunches. Martijn, thank you for our collaboration, I always enjoyed our Friday afternoon 'brainstorm & staring at a whiteboard' sessions and I am very proud that the work did result in a chapter in this dissertation. Joost, thank you for our mindful walks in Lunteren and your bright words in any situation. Hedda, thank you for your personal interest and compassion, I was always happy to come by and have a nice chat.

This project would not have been possible without Roel and the other gentlemen at Geldmaat. Thank you for explaining the world of cash, showing us around and having conversations on the potential improvements that you see. As member of the 'wiskundemeisjes', I highly appreciated the insights that you have given me in practical aspects that are relevant in such a complex, fast-developing branch.

I am very grateful for the hospitality at GERAD in Montréal, Canada. Thank you Guy, for the collaboration and all learning opportunities, it was really an inspiration. Moritz, I have very fond memories of plowing through the snow together to visit Christmas markets, the beers we drank and the conversations on work, life and our hopes for the future. Thanks to Alex for being my office mate and showing me many German video's, and to all other GERAD and CIRRELT members who made my stay in Canada memorable.

Mirelle, zo ver weg, maar toch zo dichtbij, zo leuk dat we deze tocht ongeveer tegelijkertijd hebben gemaakt en dat we het einde hebben bereikt. Gelukkig zijn we maar een telefoontje van elkaar verwijderd! Anna, bedankt voor alle ontspannende yoga avondjes, de gezellige etentjes en de fijne wandelingen.

Mooie familie Baller, deze groeiende familie geeft veel gezelligheid en verbreedt mijn blik op de wereld. Lieve Vethaakjes, ik denk regelmatig aan onze mooie en gezellige avonden, ik ben blij jullie mijn familie te mogen noemen. Lieve Oma, ik heb altijd plezier in onze uitstapjes en ik ben blij u te zien genieten van de familie. Lieve Miekie, Meike en Sietske, fijn dat jullie er altijd zijn om plezier mee te hebben!

Liefste Mam, Pap, Menke en Inger, woorden zijn soms niet genoeg om gevoel uit te drukken. Lieve Pap en Mam, het is fijn om thuis te komen en jullie mee te nemen in wat ik doe en waar ik heen wil. Ik vind de steun en de knuffels die nodig zijn om verder te komen. Mam, geniet van de 'grote vloeken'. Lieve Menke, je bent zo betrokken en oprecht geïnteresseerd, en wat ik geniet ik van je gezinnetje met Sem en Yannick. Lieve Inger, je was en bent mijn vraagbaak als het aankomt op Engels, zinnen die niet lopen en de zoektocht naar mooie woorden, daarom veel dank voor jouw letterlijke bijdrage aan dit proefschrift. Daarnaast genoot ik van de weekendjes in Antwerpen en onze prachtige reis door Indonesië. Liefste Jeroen, wat heb ik een geluk gehad jou te mogen leren kennen. Je bent bereid mijn twijfels en stress aan te horen en je geeft me de kracht om door te gaan. Jouw steun in de laatste fase van mijn promotietraject was zo fijn en maakte het nog zoveel leuker.

Contents

1 Introduction

How to distribute goods is a question that is faced daily by many companies and, therefore, these questions are solved regularly in practice either with or without supporting technologies. A general aim is to keep costs low and customer service high, for example, by minimizing delivery costs and making sure that customer demand is satisfied. Increasing resource utilization and transportation efficiency can lead to cost savings and service improvements for all parties involved (e.g., manufacturers, transportation companies and customers). Efficiency can, for example, be enhanced by finding improved distribution plans, i.e., plans with lower costs, or by exploring new distribution strategies. This dissertation intends to provide insight in complex distribution problems, to find more efficient distribution plans, and to analyze the potential benefit of novel distribution strategies.

Academic research on distribution problems relates to industry in several ways. First, scientific studies identify basic optimization problems which are underlying real-life distribution networks. Research focuses on analyzing these basic problems and on developing solution methods to solve them. An example of a fundamental problem for the distribution of goods is the so-called Traveling Salesman Problem (TSP). In this problem there are a vehicle at a depot and some geographically spread customers that need to be visited, the question is to determine the tour with the shortest distance. A fundamental extension of this problem is the Vehicle Routing Problem (VRP) in which each customer requests a number of units of goods (demand) and there are multiple vehicles with a load capacity (i.e., number of units that can be loaded into the vehicle). Again, the customers need to receive their demand and the problem is to decide which vehicle serves which customers and to find the corresponding routes such that the total covered distance is minimized. Another fundamental problem, which does not include determining vehicle tours, is the Joint Replenishment Problem (JRP). In the JRP, deliveries have to be made to customers over a given time horizon. For each period in which a replenishment takes place and for each replenishment to a customer a fixed fee is incurred. The problem is to decide when to deliver goods to each customer and how many goods while minimizing the delivery costs.

Secondly, literature concerns more integrated problems and examines multiple types of solution approaches for these problems. For instance, consider the case in which there are multiple vehicles available (fleet) to replenish multiple locations that have to satisfy a certain demand every period over a given planning horizon. This problem is known as the Inventory Routing Problem (IRP). The IRP contains the following questions: when to replenish the locations, how much to deliver at each replenishment and how to make the deliveries with the given fleet (routing) such that all demand can be satisfied. Along with these research questions, the IRP usually considers multiple, contradicting, cost components, such as routing and inventory holding costs. Multiple types of solution approaches are present in academic literature for such integrated problems. For the IRP, it is possible to apply a sequential solution method in which first the delivery periods and quantities are decided upon and thereafter, the delivery routes are determined per period (i.e., a VRP is solved per period). This decomposition shows that the VRP is a subproblem of the IRP. Next to sequential solution methods, also iterative and integrated solution methods are proposed, among others, which make use of the knowledge on subproblems as well. The difference in efficiency between solving two problems sequentially or integrating the optimization of the decisions can be substantial (see for example Chandra and Fisher [1994] and Archetti and Speranza [2016]), since integrated methods address the trade-off between costs directly. Therefore, integration of more types of decisions in one optimization problem, towards a 'systemic focus', is one of the trends identified for the field of Operations Research [Speranza, 2018].

Finally, academic research investigates advanced distribution strategies which are possibly already applied in the industry. As an illustration, consider the case in which for the distribution of goods to a customer a choice can be made between a privately owned truck and an external carrier. Outsourcing the service can, for example, be beneficial when the total quantity to delivery is higher than the total capacity of the private fleet. Despite the practical applications and the actual use in industry, the relatively recent work by Chu [2005] can be considered the first to examine the VRP combined with the question which customer services to outsource (Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC)). Compared to the abundant literature on the VRP (over 50 years [Laporte, 2009]) and the IRP (over 30 years [Coelho et al., 2014]), studies that consider the VRPPC are emerging only recently.

Inspired by distribution challenges in cash supply chains, this dissertation aims to contribute to the above three research directions. In order to be able to define the research challenges which this dissertation addresses in more detail, first, short overviews are presented on the IRP, the JRP and the VRPPC in Sections 1.1, 1.2 and 1.3, respectively. The contributions of this dissertation are specified within those sections. Thereafter, an introduction to a main solution method is provided in Section 1.4. The practical motivation that served as inspiration for the research topics is discussed in Section 1.5 before concluding with an overview of this dissertation in Section 1.6.

1.1 Inventory Routing Problems

The Inventory Routing Problem (IRP) is a combination of an inventory management problem and a vehicle routing problem. The most common variant of the problem in the literature has the following characteristics. There are one or multiple vehicles located at a depot each with a certain load capacity. These vehicles replenish the inventory of a set of geographically spread customers over a given planning horizon of several periods. At the depot a quantity of the product to be distributed becomes available each period of the planning horizon. A customer has a limited inventory capacity, has to satisfy a given demand per period and cannot have a shortage. The IRP entails the decisions on when to replenish each customer, which quantity to deliver and how to combine the visits to the customers in feasible routes while minimizing the total routing and inventory holding cost. This problem arises when a supplier (often denoted by vendor) can decide on the replenishments of its customers which is known as a Vendor Managed Inventory (VMI) setting. It is assumed that the vendor incurs all costs that usually consist of routing costs and inventory holding costs at both the depot and the customers. A good solution to this problem addresses the trade-off between these two cost types as discussed before.

1.1.1 Literature

The IRP is a widely studied class of problems in the literature for over more than thirty years [Coelho et al., 2014]. Bertazzi and Speranza [2012] and Bertazzi and Speranza [2013] introduce the IRP by considering a single customer case with multiple products and a multiple customer case with a single product respectively. In the first case, transportation costs and inventory holding costs are minimized, but only direct routes from the depot to the single customer are possible, hence the transportation cost per replenishment is fixed. In the second case, the transportation cost depends on the combination of customers served by one vehicle. Therefore, a VRP is a subproblem of the IRP in this variant. Bertazzi et al. [2008] provide another introduction to IRPs in which the focus is to examine the influence of holdings costs and inventory holding capacities at the customers, among others.

The two most recent surveys on the IRP each have a different focus. First, Andersson et al. [2010] provide an extensive overview on the industrial aspects and applications of inventory routing. The authors also propose a classification based on seven characteristics that concern the length of the time horizon, demand (deterministic or stochastic), routing, inventory and fleet aspects. The paper contains an extensive literature review divided in three groups based on an instant, finite or infinite planning horizon. Secondly, Coelho et al. [2014] propose a slightly different classification of IRPs than Andersson et al. [2010]. Coelho et al. [2014] suggest to separate the problem structure from the information availability and therefore leave the demand component out. Next to classifying a large number of studies in their classification scheme, Coelho et al. [2014] focus on solution methods for both 'basic' IRP variants and for extensions such as the IRP with multiple products.

More recent literature on the IRP includes a novel problem formulation by Desaulniers et al. [2016] including a solution method that gives promising results for the multiple-vehicle IRP. Additionally, Archetti et al. [2017] and Alvarez et al. [2018] present new, competitive heuristic solution methods.

1.1.2 Solution methods

Solution methods for the IRP can be roughly divided into two groups: exact and heuristic solution methods. Exact solution methods result in the optimal solution of the given problem, while heuristic solution methods provide solutions for which there is no guarantee on the quality. Sequential and many iterative solution methods are heuristics by definition. Although these methods can result in the optimal solution, there is no guarantee for that.

For the main exact solution methods, first the problem is formulated as a mixed integer linear program (MILP). This mathematical representation of the problem contains variables that represent the decision to be taken (e.g., the delivery quantity to a customer in a certain period). Based on this formulation branch-and-cut and branch-and-price-and-cut solution methods can be applied. In both methods, the integer decision variables are first relaxed and subsequently forced to be integer sequentially by adding constraints ('branching'). In a branch-and-cut solution method, next to the branching, additional constraints (valid inequalities) are added in each branching step which are not necessary to find the optimal solution, but will help to reach the optimal solution quicker ('cutting'). Also sets of constraints that are necessary to find the optimal solution can be added with a similar procedure which is especially useful if the number of constraints is exponential (i.e., enlisting all of them would be cumbersome). Archetti et al. [2007] were the first to apply branch-and-cut to solve the IRP with a single vehicle. They identified multiple families of valid inequalities and also added one type of necessary constraints as cuts. Instances with up to fifty customers and three periods, and thirty customers and six periods were solved to optimality within two hours of running time. Branch-and-cut solution methods for the multi-vehicle IRP were proposed by Coelho and Laporte [2013] and Adulyasak et al. [2014]. The latter solves instances with up to 45 customers, three periods and three vehicles to optimality.

If a problem formulation contains an exponential number of a specific type of decision variables, one can apply branch-and-price-and-cut. This method starts with a limited number of decision variables, and iteratively adds more variables by validating which additional variables would result in a better solution ('pricing'). Branch-and-price-and-cut was applied to the IRP with multiple vehicles by Desaulniers et al. [2016]. The authors were able to solve instances with up to fifty customers, three periods and five vehicles to optimality. Since branch-and-price-and-cut is applied in multiple chapters in this dissertation, more details on the method will be provided in Section 1.4.

The heuristic solution methods developed to solve the IRP are numerous. Therefore, a few recent examples will be discussed without having the ambition to give a complete overview. For more examples and an extensive discussion on heuristic solution methods for the IRP see Coelho et al. [2014]. First, Campbell and Savelsbergh [2004] develop a sequential, two-phase heuristic. The first phase assigns customer replenishments to periods in the planning horizon. This assignment is based on clusters which allows customers only to be served in the same route if they are in the same cluster. The second phase optimizes the delivery routes with the output of the first phase as suggestion. Secondly, many metaheuristics which are based on local search procedures have been applied to the IRP. Local search means that by making small changes in a non-optimal solution of the problem a better solution is sought. Recent examples of such methods are the iterated local search and simulated annealing heuristics by Alvarez et al. [2018].

With these methods new best solutions were found for almost 300 benchmark instances from the literature with up to 200 customers. Finally, another type of heuristic solution methods are matheuristics which combine exact and heuristic solution methods. Archetti et al. [2017] recently proposed a matheuristic that combines tabu search (a metaheuristic) with the solutions of MILPs. First, a MILP is used to find an initial solution which is not necessarily integer in the routing variables. Then, tabu search is applied to improve the initial solution during which vehicle capacity can be violated and stock-outs at the supplier are allowed. Finally, another MILP is solved which is based on the solutions found during the tabu search. Applying this method to small benchmark instances (up to 50 customers and three periods or 30 customers and six periods) resulted in 48% optimal solutions and 125 improved upper bounds compared to existing literature; for larger benchmark instances (up to 200 customers and six periods) for 92% of the tested instances a better upper bound was established.

1.1.3 Contributions

This dissertation contributes to the area of IRP in the following ways.

Chapter 2 analyses the computational complexity of a variant of the IRP. It is well-known that the IRP as defined earlier is an NP-hard problem since the TSP is an underlying problem which is NP-hard [Karp, 1972]. Hence, one source of complexity in the IRP is the routing part. Therefore, Chapter 2 considers special cases of the studied IRP in which the underlying metric is such that routing does not cause immediate NP-hardness through the TSP. This allows for studying the influence of other aspects than routing on the complexity of the IRP. One main result is proving that the IRP on the half-line with uniform service times and a planning horizon of two periods can be solved in polynomial time. Moreover, it is shown that if the planning horizon is extended to more than two periods, the problem is harder than the Pinwheel Scheduling Problem [Holte et al., 1989], of which the complexity is unknown. Furthermore, almost any variant of the IRP with non-uniform service times is NP-hard and the study shows, equivalently, that the same results holds for an IRP with multiple vehicles each with a vehicle capacity constraint and different demand at the customers. Establishing other sources of complexity than routing could in the future aid the development of solution methods for IRPs.

Chapter 4 incorporates a novel replenishment strategy in the IRP. In the IRP it is assumed that each customer faces a certain demand in each period of the planning horizon which must be satisfied by the customer itself without running out of stock. In Chapter 4, inspired by practice, an extension of the IRP is studied which relaxes the assumption that demand must be satisfied by the customer itself. In the extension it is assumed that if a customer i is close enough to another customer j, customer i can satisfy (part of) the demand of customer j in each period. This implies that a customer does not always have sufficient stock to satisfy its demand, but that it can move part of the demand to another customer. Hence, the required replenishments differ compared to the traditional IRP and costs can be saved on the distribution of the goods. To include this extension in the IRP, the possibility of *demand moves* is introduced in the IRP, i.e., a customer i can satisfy the demand of another customer j, since j's demand is *moved* to i. This problem is defined as the Inventory Routing Problem with Demand Moves (IRPDM). To move a unit of demand a service cost is charged depending on the distance between the involved customers. The objective is to minimize the total inventory holding, routing and service costs. In Chapter 4 a branch-and-price-and-cut solution method is developed to solve the IRPDM. The costs of the solutions are compared to those of the IRP to evaluate the influence of allowing demand moves. Moreover, the impact of changing the service costs and of putting a maximum on the demand that can be moved per customer per period is assessed.

1.2 Joint Replenishment Problems

Joint Replenishment Problems (JRP) concern the distribution of goods without a routing aspect. In the JRP there are multiple customers that have to be replenished over a given time horizon of multiple periods such that the customers can satisfy the demand in each period. The costs consist of replenishment costs and inventory holding costs. The replenishment costs involve two components. First, if any customer is replenished in a period, independent of the locations involved, a fixed reorder/transportation cost (major cost) is incurred. Secondly, a customer specific cost (minor cost) is charged per replenished customer in a period. An example of such a cost structure is given by Anily and Haviv [2007] for the case that a number of retailers outsource their inventory management to an external carrier. If there would be no major cost, it would be most cost efficient to replenish all customers just before they run out of stock. When including major costs, it might be beneficial to replenish a customer earlier to serve it jointly with other customers in order to save on the major costs. Note that replenishing a customer earlier will give slightly higher inventory holding costs and possibly higher minor costs since in the long run the number of replenishments is higher. Another setting which is modeled as JRP is the case when multiple products have to be ordered for one customer and the question is how to combine the different products in one order [Khouja and Goyal, 2008].

1.2.1 Literature and solution methods

The literature on JRPs distinguishes three types of problems which differ in the assumptions on or availability of data on the demand [Khouja and Goyal, 2008]. First, the traditional JRP assumes that the demand is known and constant, i.e., for a customer the demand is the same in each period of the planning horizon. Secondly, the JRP with stochastic demand assumes that demand is stochastic but stationary in the mean. Finally, the Dynamic-Demand JRP (DJRP) considers the case in which the demand is known, but can be different in each period of the planning horizon. The solutions to JRP problems consist of either a plan for a given time horizon or a long term plan which minimizes long term average costs. As in the IRP, a solution method should find a good trade-off between the replenishment and inventory holding costs, however, note that the cost structure is fundamentally different than in the IRP.

For the traditional JRP it is possible to derive analytical expressions that define the minimal total costs for a given replenishment policy. Heuristics have been designed to determine the replenishment policies. These policies are often cyclic for each customer which means there is a fixed time between two replenishments. Khouja and Goyal [2008] provides a thorough overview of heuristics for the JRP including the RAND algorithm, the power-of-two policy, in which the time between replenishments is restricted to 2^p

times the cycle time with p a positive integer, and genetic algorithms. A systematic review of the literature published between 2006 and 2015 is given by Bastos et al. [2017].

Two main policies for the stochastic JRP are the periodic review policy and the can-order policy. In a can-order policy there is an indicated inventory level at which a replenishment must take place, but also a can-order level at which a replenishment is optional which provides flexibility in the ordering process. A periodic review policy implies a replenishment order at fixed moments in time, similar to the cyclic policies for the JRP. More details on the policies and heuristics to determine the parameters are provided in Khouja and Goyal [2008]; references to more recent literature can be found in Bastos et al. [2017].

For the JRP with dynamic demand, cyclic replenishment policies are not as applicable as for the other JRP variants. Still, cyclic policies have been studied for the DJRP because of the easiness of implementation [Webb et al., 1997]. Boctor et al. [2004] proposes multiple MILP formulations for the DJRP which are compared in performance by implementing them in CPLEX. Moreover, the authors compare the performance of eight heuristics which are, for instance, based on dynamic programming and greedy planning of replenishments. Exact algorithms are based on dynamic programming, branch-and-bound, branch-and-cut and column generation acording to Boctor et al. [2004]. Tighter DJRP problem formulations were proposed and tested by Narayanan and Robinson [2006]. Additional heuristics were tested by Robinson et al. [2007]. Robinson et al. [2009] study formulations for the DJRP but denote the problem by 'coordinated uncapacitated lot-sizing problem' and review heuristic solution methods classified as specialized heuristics, metaheuristics and mathematical programming-based heuristics.

1.2.2 Contributions

Chapter 3 proposes a model to incorporate customer locations in the replenishment decisions in the DJRP without aiming at optimizing the actual routing of the vehicles. This study follows the tendency to integrate optimization of multiple decisions. The locations of the customers are not taken into account in the decisions on replenishments in traditional JRPs. Hence, in an optimal distribution plan, it can occur that two customers that are in close proximity are served in subsequent periods. This might not be desirable when considering that an actual vehicle has to execute the deliveries. Hence, replenishments determined by the DJRP can lead to high actual transportation costs. Moreover, the number of customers that can be served per period can be unnecessarily low caused by higher travel time between served customers. Still, the cost structure in the JRP is a realistic one in practice, for example if all deliveries are outsourced to a common carrier. Given the JRP cost structure, the supplier of the customers has no incentive to consider customer locations when making the replenishment decisions. Thereafter, given the replenishment orders for one period, the common carrier can only solve the routing problem for this single period. Therefore, Chapter 3 introduces the Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs (DJRP-AT) which accounts for customer locations while solving the DJRP. Instead of optimizing the delivery routes, which is not a task in the DJRP, the transportation costs are approximated. The DJRP-AT is solved with a branch-and-price-and-cut solution method to provide insight in the problem structure and solution structure. Novel dominance conditions are introduced in order to discard labels in the exact labeling algorithm that is used to solve the pricing problem.

1.3 Vehicle Routing Problems with Outsourcing

If there are both a private fleet and a common fleet available to deliver goods to customers in the VRP, the literature refers to this feature as *outsourcing* or as problems with *private fleet and common carrier*. In the VRPPC there are multiple geographically dispersed customers that each have a certain demand and there is usually a limited number of privately owned vehicles available to deliver goods to a customer. Delivery can also be outsourced to a common carrier. Outsourcing services to a common carrier is suitable if the total demand exceeds the total vehicle capacity or if it is unprofitable to serve certain customers with the private fleet. A cost is incurred for outsourcing a customer service to the common carrier. This outsourcing cost can, for example, be a fixed fee per service, potentially customer dependent, a fixed fee per unit of demand, or a cost dependent on the total outsourced quantity with a discount structure. Routing costs and, often, a fixed setup cost per vehicle are incurred for the private fleet. The objective in the VRPPC is to minimize routing, setup and outsourcing costs.

1.3.1 Literature and solution methods

The VRPPC is first introduced by Chu [2005] in a practical case. The authors propose a heuristic based on a modified savings algorithm to solve the problem. More heuristics have been proposed for the problem subsequently. Bolduc et al. [2008] propose a perturbation-based metaheuristic and Côté and Potvin [2009] define a tabu search metaheuristic. Potvin and Naud [2011] use an ejection-chain neighborhood within a tabu search heuristic. Huijink [2016] develop large neighborhood search heuristics and introduce new local search moves. Several different outsourcing and private fleet cost structures are considered by Gahm et al. [2017] for the VRPPC with heterogeneous fleet and the authors present a variable neighborhood search to solve the VRPPCs. The VRPPC is also a variant of the VRP with profits for which Vidal et al. [2016] study a unified hybrid genetic search framework and local search metaheuristics. For a survey on the VRP with profits, see Archetti et al. [2014b]. Goeke et al. [2018] propose a large neighborhood search heuristic with a post-processing step which solves an ILP that selects the best routes from all routes encountered during the search. The multi-depot version of the VRPPC is introduced by Stenger et al. [2013] and the authors develop a variable neighborhood search algorithm. Only two studies that solve the VRPPC with an exact solution method have appeared very recently. Dabia et al. [2019] proposes a branch-and-price-and-cut approach for the VRPPC with heterogeneous fleet and a discount outsourcing cost structure. Independently, Goeke et al. [2018] proposes a similar method for the VRPPC with customer-dependent, fixed fees as outsourcing costs and a heterogeneous fleet.

1.3.2 Contributions

In the VRPPC it is assumed that service to a customer can either be performed by a private vehicle or that it can be outsourced to a common carrier. Chapter 5 proposes to relax this assumption and allows that the service to a customer can be split over one private vehicle and the common carrier. This problem is denoted by Vehicle Routing Problem with Partial Outsourcing (VRPPO). Combining outsourcing and split delivery aspects in VRPs has received increasing attention in the literature. There are several studies considering split delivery and outsourcing distribution strategies in practical cases for which heuristic solution methods are developed (e.g., Bolduc et al. [2010], Lee and Kim [2015]). Chapter 5 contributes to the literature by formally defining the VRPPO. Thereafter, branch-and-price-and-cut solution approaches are developed for two different problem formulations, including two different exact pricing mechanisms for each formulation. The aim is to explore the potential cost improvement of the VRPPO over the VRPPC and to examine how solutions change compared to the VRPPC in which a split of the service to a customer is not allowed.

1.4 Solution Methods

For the optimization of distribution decisions many approaches have been used in the literature. A distribution problem can be formulated as a MILP which provides insight in the structure of the problem and offers the starting point for an exact solution method. Although exact solution methods can usually only solve instances of limited size, the obtained optimal solutions do reveal the solution structure and moreover, the exact solution method can serve as a base for a matheuristic capable of solving larger instances. In this section it is not the intention to give a complete overview of all possible exact solution methods, but to highlight one method that is at the core of this dissertation. Therefore, Section 1.4.1 introduces branch-and-price-and-cut which is explained with the help of the VRP.

1.4.1 Branch-and-Price-and-Cut

Consider a VRP in which a set of vehicles is available to serve a set of customers that each have to receive a given number of units of goods in the one-day planning horizon. One type of MILP formulations for the VRP is based on the connections between the customers (arc flow formulations). This formulation explicitly considers whether a vehicle route visits customer j after customer i and hence, a decision variable representing this option is present in the model, for each pair of customers. Another type of MILP formulations for the VRP is not based on the arcs between customers, but on the set of vehicle routes (route-based formulation). This means that complete routes, which start and end at the depot and that satisfy all route constraints such as vehicle capacity, should already be available. Then, instead of modeling the construction of the routes in the linear program, one has to model the selection of the routes such that all customers are visited exactly once while respecting the fleet size. These two examples of VRP formulations show that there is often not just one possible MILP representation of a problem. Laporte [2009] provides several linear programming formulations for the VRP, Archetti et al. [2014a] compare several formulations for the IRP and Narayanan and Robinson [2006] analyze several JRP formulations.

It is possible to deduce a route-based VRP formulation from an arc flow formulation by applying the so-called Dantzig-Wolfe decomposition which was introduced by Dantzig and Wolfe [1960]. For a thorough technical description of the decomposition method see e.g., Lübbecke and Desrosiers [2005], and for some examples see Barnhart et al. [1998].

The number of routes in a route-based VRP formulation can become quite substantial since the model has to consider all possible routes (i.e., all customer sequences) that respect the route constraints (e.g., vehicle capacity). To avoid enumeration of all possible routes upfront, a method called *column generation* can be applied (Desaulniers et al. [2005]). First, one formulates the problem with a limited number of routes and solves this MILP. The solution of the program is not necessarily the overall optimal solution since not all possible routes were included in the program. Then, given the current solution of the program, additional routes (columns) are generated by solving a so-called *pricing problem*. The pricing problem identifies routes that are likely to steer the program in the direction of the overall optimal solution. This process is continued until no more routes are identified that will give a better solution. The additional routes can for example be generated with a labeling algorithm which iteratively extends a path from the depot to all possible subsequent customers (see for example Feillet et al. [2004], Righini and Salani [2006] and Tilk et al. [2017]). Heuristics can be used to generate multiple routes quickly before solving a labeling algorithm.

When solving MILPs, the integer variables make the problem much harder to solve since these variables must be integer. Therefore, when applying column generation, the integer variables are usually relaxed to be continuous variables and integrality is enforced later. Hence, a VRP solution that is obtained with the column generation procedure is not necessarily integral, for example, the solution is to perform a route half. This is not feasible in practice, therefore, the column generation method is incorporated in a *branch-and-bound* framework to enforce integrality. By applying *branching*, a variable with a fractional value is selected and two branches are created. In one branch a constraint is added which limits the value of the variable from above by its rounded down value, and in the other branch a constraint requires that the variable has at least a value higher than its rounded up value. Column generation is applied in each branch before repeating the branching step. The combination of column generation and branch-and-bound is denoted by *branch-and-price*.

Finally, in a MILP formulation for the VRP only necessary constraints are present initially. Leaving out one of these constraints, may result in an infeasible solution to the overall problem. Next to these necessary constraints, it is possible to identify constraints that are not required to find a feasible optimal solution, but that can be useful during the solution method since they eliminate fractional solutions. This type of constraints are called *cutting planes* or *valid inequalities*. Applying valid inequalities in the above described branch-and-price framework results in a solution approach called *branch-and-price-and-cut* (see e.g., Nemhauser and Park [1991] and Lübbecke and Desrosiers [2005]). When applying branch-and-price-and-cut, some technical considerations have be kept in mind. For example, branching can change the structure of the pricing problem and adding valid inequalities can imply that the pricing problem has to expressed differently, see for instance Jepsen et al. [2008] and Dabia et al. [2019]. Branch-and-price-and-cut is applied in Chapters 3, 4 and 5.

1.5 Practical Motivation

The research in this dissertation is motivated and inspired by cash management in the Netherlands and research questions have been established in consultation with the business partner Geldmaat (Geldservice Nederland until January 1, 2019). Although cash is still essential for the economy and access to cash for all inhabitants is required by the government, the use of cash is diminishing in recent years. In the Netherlands, three commercial banks are responsible for the distribution of bank notes to bank offices and Automated Teller Machines (ATMs). According to Van Anholt [2014], the banks were not fully aware of the impact of the costs of this task and they did not focus on efficient replenishment. In 2008 this view changed, influenced by the shift in use of payment methods (less cash money) and the financial crisis. In order to organize the supply of cash more efficiently Geldmaat was founded to support collaboration between the banks while securing the accessibility of cash throughout the Netherlands. Van Anholt [2014] thoroughly describes the cash supply chain and involved parties. Only the important aspects for this dissertation are summarized here.

In the current situation in the Netherlands, Geldmaat determines each day which ATMs to replenish during the next day based on a predictive model. A third-party transportation company, specifically referred to as Cash-in-Transit company (CIT) in this field, performs the actual ATM replenishments. The orders based on the prediction model are packed in the Geldmaat cash center, subsequently these packages are shipped to a CIT cash center. In the CIT cash center, the packages are distributed to armored vehicles that will perform the actual routes serving the ATMs and potentially other customers such as retailers. The CIT has full control over the performed routes and carries out its own optimization.

High costs are involved in the replenishment of the ATMs for Geldmaat. Therefore, in multiple ways, the company is in search for more efficient or more collaborative ways to replenish the ATMs. The quest for more efficiency is intensified by the fact that less and less cash is used and hence, the cost per banknote is increasing which is not desirable. In this dissertation several possible directions for more efficiency are explored with the purpose to support Geldmaat to improve their business. Chapter 3 facilitates the discussion between Geldmaat and the CITs by providing insight in the consequences of certain collaborative cost structures. To this end, in Chapter 3 transportation costs are taken into account when deciding on the replenishments of the ATMs. Chapter 4 supports the search for a more efficient distribution strategy by exploring the option to not replenish fully all ATMs but allowing for redirecting users between ATMs. Both research questions were established in close collaboration with Geldmaat and the latter one was a future research topic in the dissertation by Van Anholt [2014].

The dissertation by Van Anholt [2014] contains a thorough literature review on cash supply chains up to 2014. Studies discussed in Van Anholt [2014] which are directly relevant for the work in this dissertation and newer studies that contribute to the stream of research on cash supply chains are highlighted here. Van Anholt et al. [2016] consider a combined inventory management and routing problem for so-called Recirculation ATMs (RATM). At an RATM, an ATM-user can both withdraw and deposit money, hence, the IRP-like solutions contain both delivery and pick-up activities. Money that is picked up at one ATM can be used for a replenishment of another ATM. Batı and

Gözüpek [2017] study an IRP for a network containing both traditional ATMs and RATMs combined with the optimization of which ATMs to change to RATMs, given that withdrawal and deposit amounts are known. Larrain et al. [2017] consider an IRP that allows for stock-outs and the replenishment policy consists of swapping new cassettes of a chosen amount for the current cassettes that can still contain bank notes which are returned to the depot. Geismar et al. [2017] provide an overview on currency supply chains by reviewing studies that look into the cash supply chain from the supply side (national banks), the demand side (commercial banks and ATM networks), and the private sector logistics providers' side. In their analysis on ATM replenishment-related literature, Geismar et al. [2017] mention the study by Van Anholt et al. [2016] on RATMs and suggest for future research to investigate possible incentives to rebalance RATM inventories by steering users to a certain RATM (either withdraw from a full RATM or deposit at an empty RATM). They suggest a premium as incentive for making a deposit at a certain RATM and these premiums can be reviewed online by the user. Chapter 4 investigates the possible gain in supply chain costs when implementing a similar idea for regular ATMs. Other recent studies consider different issues in the cash supply chains such as vault location and size optimization by the central bank [Huang et al., 2017], combined optimization of inventory management and the denomination mix issued at a cash withdrawal [Van der Heide et al., 2017], and combined demand forecasting and replenishment policy optimization [Lázaro et al., 2018].

1.6 Dissertation Outline and Research Output

Chapter 2 studies a variant of the IRP in which routing is easy with the aim to study the computational complexity of the problem and to identify other sources of complexity than routing. Chapter 3 studies the DJRP-AT and proposes a branch-and-price-and-cut solution method. Incorporating demand moves in the IRP is explored in Chapter 4 and a branch-and-price-and-cut solution approach is developed to solve the IRPDM. Chapter 5 formally introduces the VRPPO in which it is not only possible to outsource some services to a common carrier, but also to split the service to one customer between a single private vehicle and the common carrier. To solve the problem, a branch-and-price-and-cut solution method is designed for two path-based formulations. In each chapter the notation is consistent with that in closely related literature of that chapter, resulting in different notation throughout the chapters in this dissertation. Finally, Chapter 6 concludes with an overview of the main findings and suggestions for future research.

An overview of the research output of this dissertation is presented in Table 1.1. For each of the chapters the table contains the title, the research questions and the status of the publication status (i.e., published, revision, submitted or in preparation for journal submission).

Table 1.1 Overview of Research Output

Chapter	Title	Research Questions	Journal Publication Status
2	On the complexity of Inventory Routing Problems when routing is easy	What factors influence the computational complexity of the IRP? Can a borderline between easy and hard problems be defined?	Accepted for publication in Networks [Baller et al., 2019d]
3	The Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs	What is the potential improvement by considering transportation costs in a DJRP setting? How close are the resulting costs to an IRP solution with optimal routes?	Published in European Journal of Operational Research [Baller et al., 2019b]
4	The Inventory Routing Problem with Demand Moves	What is the potential cost improvement of allowing for demand moves in the IRP? How do several factors influence the cost improvement?	In preparation for submission [Baller et al., 2019a]
5	The Vehicle Routing Problem with Partial Outsourcing	How to model a VRP in which a customer can either be served by a single private vehicle, by a common carrier, or by both a single private vehicle and a common carrier? What is the potential cost improvement compared with only allowing fully outsourcing customers and not allowing for splits?	Accepted for publication in Transportation Science [Baller et al., 2019c]

2

On the complexity of Inventory Routing Problems when routing is easy

2.1 Introduction

This paper studies the computational complexity of special cases of a variant of the Inventory Routing Problem (IRP), in which a set of customers is supplied over a given time horizon by identical vehicles from a central depot. Each customer has a storage capacity, a fixed demand per day, a latest delivery day at the start of the planning horizon and a service time. The metrics that underlie the customer locations do not immediately imply intractability because of routing aspects. In particular, we consider the problem in which all customers are located in a single point, on a half-line and in the Euclidean plane, but the latter under a specific approximation of the tour length. On a half-line, the depot is located in the origin, i.e., at one end of the half-line. The vehicles have a tour duration constraint which limits the number of time units per day (traveling plus service time). The objective is to minimize the total time spent by all vehicles over all days.

The motivation for this study stems from a business project in ATM replenishment in the Netherlands. ATMs need to be replenished regularly such that banknotes are sufficiently available to consumers. In practice, this is an involved problem in which service levels, safety regulations and uncertainty play a role. In this paper we study a stylized version of this problem.

In many vehicle routing problems (VRP) the vehicles are capacitated in terms of load, however, in the ATM replenishment problem, the time spent by a vehicle is often more binding [Baller et al., 2019b]. Time is also more binding than load in, e.g., online

This chapter is based on: A.C. Baller, M. van Ee, M. Hoogeboom, and L. Stougie. On the complexity of Inventory Routing Problems when routing is easy, Networks, 2019, to appear [Baller et al., 2019d] ordered package delivery and blood product distribution [Hemmelmayr et al., 2009]. Therefore, we consider a tour duration constraint.

Almost any version of the IRP is NP-hard since it contains the well-known NP-hard Traveling Salesman Problem (TSP) [Karp, 1972, Papadimitriou, 1977] as a special case. We investigate the complexity of an IRP on metrics for which routing does not cause immediate NP-hardness through TSP. In the first part of the paper, we consider the problem on a point, i.e., the depot and the customers are all at the same location, and on the half-line. Furthermore, the problem variants studied differ in the number of vehicles available, the length of the planning horizon and the type of service times. In the second part of the paper, we consider the respective problem in the Euclidean plane and choose a route length approximation function that avoids hardness through TSP. Still the problem is shown to be NP-hard.

Some variants of the studied IRP are easily shown to be solvable in polynomial time and some are easily shown to be NP-hard (Section 2.3). In order to identify which aspects determine the computational complexity of the respective IRP variant, we search for *borderline* problem variants on a point and on the half-line, which are either maximally easy or minimally hard. A maximally easy problem variant is a variant that is polynomial time solvable, but if one feature is generalized it becomes hard or has an open complexity. Similarly, a minimally hard problem becomes easy or open if one feature is further restricted.

IRPs form a class of widely studied and still challenging problems in the Operations Research literature. An introduction to the IRP is given in the tutorials Bertazzi and Speranza [2012] and Bertazzi and Speranza [2013]. In their introduction to IRPs, Bertazzi et al. [2008] provide examples that give insight in the influence of holding costs, inventory capacities at the customers, and continuous consumption of goods at the customers. The authors state informally that limited storage capacity causes extra complexity in IRPs because of the implied time required between two deliveries. We will formalize this statement in this paper.

Several literature surveys have been published since the 1990's [Federgruen and Simchi-Levi, 1995, Baita et al., 1998, Sarmiento and Nagi, 1999, Cordeau et al., 2007, Moin and Salhi, 2007, Andersson et al., 2010, Coelho et al., 2014]. The two most recent surveys each have a different focus. Andersson et al. [2010] focus on industrial aspects of inventory routing and they propose a classification based on seven aspects that concern time, demand (deterministic or stochastic), routing, inventory and fleet aspects. The survey is split into three parts based on the time horizon: instant, finite and infinite time horizon. Coelho et al. [2014] state in their recent survey that there "does not really exist a standard version of the problem", since many variants with changing aspects are present in the literature. They propose a classification based on seven criteria including time horizon, routing, inventory and fleet aspects, and call all variants of the IRP that fit these criteria as 'basic versions'. These seven criteria do not include demand aspects, since the authors want to separate the structure of the problem from information availability (i.e., stochasticity of demand). The criterion added compared to Andersson et al. [2010] is the inventory policy used in the problem (maximum level or order-up-to level). Coelho et al. [2014] give an in-depth overview of models and solution methods for both 'basic versions' of the IRP as well as extensions of this version. The solution methods are divided in exact methods and heuristic methods for the basic versions of the problem. Additionally, Desaulniers et al. [2016] propose a structurally different problem formulation and an exact solution method for the IRP that gives promising results for the multiple-vehicle IRP. Alvarez et al. [2018] and Archetti et al. [2017] present the most recent heuristic solution methods for the IRP.

The variant of the IRP considered in this paper fits into the classification of Coelho et al. [2014] as follows. It has a finite time horizon and a one-to-many structure, which means that one vehicle can visit multiple customers in one route. Multiple homogeneous vehicles are available and each vehicle can serve multiple customers in one route. Inventory is replenished via an order-up-to-level policy, which means that inventory is filled up to capacity at each replenishment and all demand has to be satisfied, i.e., back-orders or lost sales are not allowed. We assume that all demand information is available at the beginning of the planning horizon. Furthermore, suppose that in a given day all replenishments take place before demand occurs, which is a common assumption in IRP (see for example Archetti et al. [2014a]). Besides this classification, we take the service times of the customers as given and we do not consider inventory holding costs. Hence, our objective is to minimize the total traveling and service time. Moreover, instead of a vehicle capacity constraint in terms of units of goods, we consider a tour duration constraint which limits the time spent by a vehicle per day. In Section 2.4.4, we briefly discuss that our results with the tour duration constraint imply similar results for the case with a vehicle capacity constraint.

In this paper, we study problem variants of the IRP in which customers are located on a point or on the half-line. A similar study was executed for the VRP by Archetti et al. [2011]. Specifically, the authors consider the VRP with unsplittable demand and a limited fleet on a line, a star, a tree, and a cycle. They show several hardness results using the relation with the weakly NP-hard Partition Problem. Here we derive similar hardness results by relating our problem variants to the strongly NP-hard Bin Packing Problem (BPP).

Finally, we mention two papers that consider problems similar to the ones in this paper. Das et al. [2011] study the *Train Delivery Problem* with a single time period and multiple capacitated vehicles. The customers have weights and are located on the half-line. The goal is to assign customers to vehicles such that the vehicle capacity is not violated and the total distance traveled is minimized. The authors mention the NP-hardness of this problem, since it generalizes the BPP (cf. our Section 2.3.2.1). The main focus is on investigating approximation algorithms. Bosman et al. [2018] consider a replenishment problem on a tree and on general metrics over an arbitrary time horizon. Their main results also concern approximation algorithms, but some of their complexity results are similar to ours (cf. our Section 2.3.2.2).

The remainder of the paper is organized as follows. In Section 2 the studied IRP variant is formally described, and the relevant problem variants are presented. We also present related problems and their complexity which will play a role in the complexity analysis of our problems. In Section 2.3, we present complexity results of some problem variants that are easily seen polynomially solvable (in P), or intractable. The main result on a maximally easy borderline problem, for which the complexity is not trivial, is presented in Section 2.4. We show that the problem can be solved in polynomial time using dynamic programming. In Section 2.5, we present the hardness result for a variant of the problem in the Euclidean plane. Related problems and literature for that specific variant will be discussed in Section 2.5. Finally, Section 2.6 finishes the paper with concluding remarks.

2.2 Problem Description and Related Problems

In the problems we study we are given a metric space containing N customers and a depot r. Each customer has a service or replenishment time s_i, a latest delivery day at the start of the planning horizon and a period p_i, $i = 1, \ldots, N$, which is the maximum number of days between two replenishments. The periods are defined directly by the customer's storage capacity and the fixed daily demand. At the depot, M identical vehicles are present that can each spend at most L time units per day on traveling plus service time. The vehicles need to return to the depot at the end of a day. We assume that travel time is equal to travel distance, i.e., vehicles travel at unit speed. The length of the planning horizon is Z days. Customers can be replenished at most once per day, i.e., split deliveries are not allowed. A solution to this problem consists of an assignment of customers to vehicles, and a route for each vehicle, for each day in the planning horizon. A solution is feasible if each vehicle spends at most L time units per day and the time between two consecutive replenishments of customer i is at most p_i days. The objective is to minimize the total time spent by the vehicles.

2.2.1 Problem Variants

We provide a concise description of the problem variants we study in this paper, in much the same spirit as done for scheduling problems in Graham et al. [1979] and later for dial-a-ride problems in de Paepe et al. [2004]. For all variants, a vehicle can spend at most L time units a day. Remaining features of the problems are stated in a 4-field notation $\alpha_1|\alpha_2|\alpha_3|\alpha_4$. In this notation, α_1 denotes the number of identical vehicles which is equal to 1 or a given $M > 1$, hence $\alpha_1 \in \{1, M\}$. The problem is studied on a point and on the half-line, which is indicated by α_2, $\alpha_2 \in \{half\text{-}line, point\}$. We use $\alpha_2 = point$ to indicate that all distances are zero, i.e., both the depot and the customers are located in one point. Equivalently, we could say that the vehicles drive at infinite speed. We use $\alpha_2 = half\text{-}line$ to indicate that the customers are located on a half-line. On the half-line, we assume the customers are numbered in increasing distance to the depot; customer i is located at distance d_i from the depot r and $d_1 \leq \ldots \leq d_N$. The type of service times are denoted by α_3, $\alpha_3 \in \{s, s_i\}$, in which s indicates uniform service times and s_i indicates that service times can differ per customer (arbitrary service times). The planning horizon is denoted by α_4, which can be equal to 1, 2 or a given $Z > 2$ days, $\alpha_4 \in \{1, 2, Z\}$.

Table 2.1 provides an overview of the complexity results obtained in Sections 2.3 and 2.4. Each entry of the table corresponds to a configuration of type of service times, whether the customers are located on a point or on the half-line, the number of vehicles and the length of the time horizon. All these problems are characterized as polynomially solvable, NP-hard, strongly NP-hard or "PSP-hard" which is defined in the next section. Table 2.1 contains references to the corresponding sections for all borderline problems and, additionally, for the problems in Section 2.3.1.

Most of the problems in Sections 2.3 and 2.4 consider a planning horizon of two days. For these problems, we define three types of customers: day 1-customers (D1-customers), day 2-customers (D2-customers) and period 1-customers (P1-customers). D1- and D2-customers need service latest on day 1 and 2, respectively. Note that D2-customers can also be served on day 1. P1-customers need service on both days 1 and

Table 2.1 Overview complexity results, where problems are either in P (dots), NP-hard (vertical lines), strongly NP-hard (diagonal lines) or PSP-hard (horizontal lines).

		Point, $\alpha_2 = point$			Half-line, $\alpha_2 = half\text{-}line$		
		$\alpha_4 = 1$	$\alpha_4 = 2$	$\alpha_4 = Z$	$\alpha_4 = 1$	$\alpha_4 = 2$	$\alpha_4 = Z$
Unif. service times, $\alpha_3 = s$	$\alpha_1 = 1$			2.3.2.2			
	$\alpha_1 = M$		2.3.1.1		2.3.1.2	2.4	
Arb. service times, $\alpha_3 = s_i$	$\alpha_1 = 1$		2.3.2.1	2.3.2.1	2.3.2.1		
	$\alpha_1 = M$	2.3.2.1					

2. Similarly, define Dh-customers which need service latest on day h and Pm-customers need service every m days.

2.2.2 Related Problems

To assess the complexity of some variants in the set of problems, we use hardness results from the *Bin Packing Problem* and the *Pinwheel Scheduling Problem* (PSP).

Bin Packing Problem. Given are n items with weights $w_1, \ldots, w_n$ and bins with capacity B. Pack the items in a minimum number of bins such that each bin contains total item weight no more than B. BPP is strongly NP-hard [Garey and Johnson, 1979]. The decision problem whether all items can be packed in two bins, with bins of capacity $\frac{1}{2}\sum_{j=1}^{n} w_j$, is known as the *Partition Problem*, and is a weakly NP-complete problem [Karp, 1972]. We use these hardness results in Section 2.3.2.1.

Pinwheel Scheduling Problem. Given are n tasks with integer periods $p_1, \ldots, p_n$. Each time unit, one task can be scheduled. A schedule is feasible if the time between two consecutive moments at which task i is scheduled is at most p_i time units. The goal is to find a feasible schedule. A special feature of this problem is the following. If one would decide to schedule a task one time unit earlier, the next due date for the task is also shifted one time unit back. Hence, a decision for a given time unit influences directly the situation at a later moment in the schedule and can cause conflicts there.

The complexity of the PSP is a long-standing open question, which is mainly due to the compact input description. It was only shown to be in PSPACE by Holte et al. [1989]. Recently, it was shown by Jacobs and Longo [2014] that the PSP cannot be solved in pseudopolynomial time, unless there is a randomized algorithm for solving the well-known *Satisfiability Problem* in time $n^{\mathcal{O}(\log n \log\log n)}$. Since the latter is unlikely, the PSP is assumed to be intractable. Yet it is unclear if it is in NP or in co-NP. We define the class of *PSP-hard problems* as the problems that are at least as hard as the PSP. We use this hardness notion in Section 2.3.2.2.

2.3 Preliminary Results

In this section, we consider some problems for which its computational complexity is easily established. We first discuss two easy problems that are actually not borderline easy, but we think they will provide insight into the structure of optimal solutions of the problem $M|half\text{-}line|s|2$, which is discussed in Section 2.4. We finish this section with presenting two classes of intractable problems.

2.3.1 Easy Problems

The two easy problems are equivalent to special cases of problem $M|half\text{-}line|s|2$. The first easy problem is the problem on a point, instead of on a half-line. In the second easy problem, the planning horizon is restricted to one day, instead of two days.

2.3.1.1 Problem $M|point|s|2$

There are M vehicles, the depot and all the customers are located in one point, the customers have uniform service times ($s_i = s \ \forall i$) and there is a planning horizon of two days. Let #P1, #D1 and #D2 be the number of P1-customers, D1-customers and D2-customers, respectively. Since there is no travel time and all customers have equal service times, it is most efficient to serve each customer (with $p_i > 1$) exactly once in the two day planning horizon. If the problem is feasible, this gives the optimal objective value. To check feasibility, we just need to check two things. First, whether M vehicles suffice to serve all P1-customers and D1-customers on day 1: $\#\text{P1} + \#\text{D1} \leq M\lfloor L/s \rfloor$. And, if so, second, whether M vehicles suffice to satisfy all the customers' service requirements in the two days, taking into account that a D2-customer can be served on day 1: $2\#\text{P1} + \#\text{D1} + \#\text{D2} \leq 2M\lfloor L/s \rfloor$. Thus, the running time of the algorithm is linear in N.

2.3.1.2 Problem $M|half\text{-}line|s|1$

In this problem variant on the half-line, there are M vehicles, uniform service times ($s_i = s \ \forall i$) and a planning horizon of one day. Recall that there are N customers which are numbered in order of increasing distance to the depot on the half-line. Define a 'region' to be the interval in which a given set of customers is located on the half-line. Polynomial solvability follows from the following lemma, which is easily proved by a simple exchange argument, left to the reader.

Lemma 2.1. *In an optimal solution of the given problem, the regions of customer locations served by any two vehicles are disjoint.*

The optimal solution is established as follows. Let a vehicle serve the farthest customer N and include, on the way to the depot, as many customers as possible with the highest indices. Let the next vehicle serve the farthest customer that is not served by the first vehicle and again, let this vehicle serve as many customers as possible. Continue until all M vehicles are used or all customers are served. This results in a solution in which each vehicle serves consecutive customers on the half-line. In case all vehicles are used and there are some customers left, there is no feasible solution. Otherwise, a feasible solution is found, which is clearly optimal by construction. Hence, if the order of the customers is given, this is a linear time algorithm.

2.3.2 Hard Problems

The problem variants studied in this section are the problems having either arbitrary processing times s_i or a planning horizon of Z days. Both problem variants are minimally hard, i.e., restricting one of the characteristics makes a problem easy to solve.

2.3.2.1 Problem $M|point|s_i|1$ and problem $1|point|s_i|Z$

Consider the class of problems with customers having arbitrary service times s_i. For the problem on a point, any feasible solution has the same objective value, hence feasibility is the core problem. The problem variant that has one day and one vehicle ($1|point|s_i|1$) is trivially in P (which also holds for $1|half\text{-}line|s_i|1$). However, if there are multiple vehicles and/or multiple days in the planning horizon, the problems are equivalent to BPPs (cf. Das et al. [2011]). For example, in the problem $M|point|s_i|1$, the vehicles are equivalent to the bins in the BPP with bin capacity equal to duration limit L. Then the problem boils down to the feasibility question whether or not the number of available bins is sufficient to be able to assign each customer to a bin. Hence, this problem and more general variants are strongly NP-hard. The problem with one vehicle and a planning horizon of Z days ($1|point|s_i|Z$) is by similar arguments also strongly NP-hard. The problem with one vehicle and a planning horizon of two days ($1|point|s_i|2$) is a weakly NP-hard problem because of its equivalence to the *Partition Problem*. Concluding, any problem variant with arbitrary service times s_i and multiple vehicles and/or days is an NP-hard problem. In Section 2.4, but also in the next subsection, hardness is avoided through bin packing by restricting to uniform service times.

2.3.2.2 Problem $1|point|s|Z$

Consider the problem on a point with uniform service times ($s_i = s\ \forall i$), one vehicle, and an arbitrary long time horizon Z. A special case of this problem, in which a vehicle can replenish at most one customer per day, i.e., $L = s$, is equivalent to the Pinwheel Scheduling Problem. Hence, this problem variant is PSP-hard (cf. Bosman et al. [2018]). This does not imply that this variant is NP-hard, but that it is unlikely that it can be solved in polynomial time. Note that the recurrence of replenishments can occur because of the longer planning horizon. We avoid analyzing PSP-hard problems in Section 2.4 by restricting ourselves to a planning horizon of two days.

2.4 Polynomial time algorithm for $M|half\text{-}line|s|2$

In this section, we prove that problem variant $M|half\text{-}line|s|2$ is solvable in polynomial time. In this variant on the half-line there are M vehicles, uniform service times ($s_i = s\ \forall i$) and a planning horizon of two days. All previously mentioned hardness results are avoided as follows. By restricting the problem to the half-line, we avoid hardness through TSP; by restricting to uniform service times, we avoid hardness through bin packing; by restricting to a planning horizon of two days, we avoid hardness through PSP.

Recall that P1-customers have to be served on both days, D1-customers have to be served on day 1, whereas D2-customers can be served on either of the two days. To minimize total travel and service time, it must be determined how many vehicles to use every day and which customers to serve with each vehicle, such that the tour duration limit (of L time units) is not exceeded. Obviously, in any optimal solution each customer that does not need service every day is served only once. Hence total service time is always equal in any relevant solution and we disregard it from the objective from here onwards.

A dynamic programming algorithm (DP) is designed to solve this problem to optimality. To simplify explanations, in Section 2.4.1 we first assume that there are no customers that need service on both days (no P1-customers). In Section 2.4.2 this assumption will be relaxed. In Section 2.4.3 the running time of the DP will be analyzed, an observation to speed up the DP is made and the complexity of extensions of the problem is discussed. Finally, Section 2.4.4 discusses the variant of the respective IRP variant with a vehicle capacity constraint.

2.4.1 Dynamic Programming

Before discussing the DP in detail, we observe the following. Given the farthest customer that still needs to be assigned to a vehicle, there are only a limited number of options for the other customers that will be served by the same vehicle in an optimal solution. Consider the example in Figure 2.1. The figure shows two half-lines, the depot as black squares and the locations of the D1- and D2-customers. The D1-customers are indicated by circles and the D2-customers are indicated by squares.

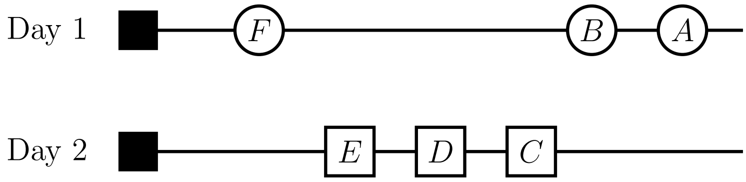


Figure 2.1 Example for observation, the depot is represented by black squares, D1-customers are represented by circles and D2-customers by squares.

Suppose customer A is the farthest customer that has not been assigned to a vehicle yet. Suppose that, because of the limitation on the time, at most three more customers can be served by the vehicle if customer A is the farthest served customer. For example, customers $\{A, B, C, D\}$ can be served by one vehicle. Now, suppose customer A is assigned to a vehicle v that will serve customers on day 1. Then, by Lemma 2.1, it is not optimal to have vehicle v serving customer F but not serving customer B. A similar argument holds for serving D2-customers with vehicle v. If customer D is served by vehicle v, but C is not served by vehicle v, this means that another vehicle w has to drive up to customer C. By interchanging customers C and D a better solution is constructed.

Hence, in general, in an optimal solution, any vehicle serves a combination of consecutive D1-customers and consecutive D2-customers, i.e., no vehicle skips a Dh-customer to serve another Dh-customer closer to the depot. This observation is used in the DP. The idea of the DP is to start at the customer farthest from the depot and work backwards to the depot. For every customer that needs service latest on day 2, a decision must be made whether this customer is served on day 1 or 2.

As a basis for the DP, Lemma 2.2 first generalizes Lemma 2.1 to prove the observation in the example above. Then, the DP is formulated.

Lemma 2.2. *Starting from the farthest customer and moving towards the depot, no vehicle skips a Dh-customer to serve another Dh-customer closer to the depot, for* $h \in \{1, 2\}$.

Proof. First, consider the case of assigning customers that need service latest on day h to a vehicle v serving these customers on day h (Case 1).

Case 1: the Dh-customers assigned to vehicle v will be consecutive on the half-line, otherwise the solution can be improved with the same interchange argument used in Lemma 2.1.

Second, consider the case of assigning customers that need service latest on any of the two days to a vehicle v on one of these days (such that the services are feasible). Define $S(v)$ to be the set of customers served by vehicle v. Again, distinguish two cases: the customer in $S(v)$ farthest from the depot is a D1-customer (Case 2a) and a D2-customer (Case 2b), respectively.

Case 2a: by Case 1 it has been shown that D1-customers served by vehicle v are consecutive on the half-line. It remains to show that the D2-customers served by vehicle v are also consecutive on the half-line. Let A be the farthest D1-customer, B the farthest D2-customer and C the second farthest D2-customer from the depot such that $d_C < d_B$. Suppose there is an optimal assignment in which customer B is not assigned to vehicle v on day 1, but customers A and C are: ($A, C \in S(v)$ and $B \notin S(v)$). Then, for customer B it has to be decided on which day it is served and by which vehicle. If customer B is served on day 1 with vehicle w, this gives overlap between the delivery regions of vehicles v and w on day 1, which cannot be optimal by Lemma 2.1. Hence, customer B must be served on day 2 by vehicle u ($A, C \in S(v)$ and $B \in S(u)$) and the cost of vehicles v and u is $d_A + d_B$. By interchanging customers B and C, the solution is still feasible, but the cost is $d_A + d_C$ which is less than $d_A + d_B$ since customer C is closer to the depot than customer B which contradicts the assumption of optimality.

Case 2b: by Case 1 it is known that D2-customers served by vehicle v on day 2 are consecutive on the half-line. Hence, if no D1-customers are served by vehicle v the lemma has been shown. If any D1-customer is served by v, the same arguments used in Case 2a prove the statement also for this case, and therefore complete the proof. □

It is easy to see that this lemma can be extended to $h \geq 3$. We will use Lemma 2.2 to find an optimal solution for problem $M|half\text{-}line|s|2$ in polynomial time. Again, assume that the customers on the half-line are numbered $1, 2, \ldots, N$ in increasing distance from the depot.

In the DP, in every state the customer farthest from the depot that has not been assigned to a vehicle yet is considered. The crucial observation is that given this farthest unassigned customer, n, and its latest service day h, the number of customers with latest service day 1 that is served by the same vehicle v as customer n defines the next state. Define $C(\ell)$ as the number of customers that can be served by a vehicle within its time limit L given that customer ℓ is the farthest customer served by the vehicle. Define k as the number of D1-customers that are served by a vehicle (including the farthest customer if it is also a D1-customer). The value of k ranges from 0 (if n is not a D1-customer) to $C(n)$. Given a value of k, the optimal route for vehicle v is easy to find: select the customers farthest from the depot including exactly k D1-customers. If $k \geq 1$ vehicle v has to ride on day 1. Note that servicing only D2-customers on day 1 is also an option. Therefore, if $k = 0$, it is yet to be decided in the DP on which of the two days vehicle v will ride, given that there are still vehicles left for both days. Concluding, the value of k, the availability of vehicles, and the decision on the delivery day, determine both the set of customers served by the same vehicle as customer n

and the day on which these customers are served. Hence, to find the optimal solution it is sufficient to consider for each customer n all possible values of k and the decision whether to serve a set of only D2-customers on day 1 or 2.

Let the indices of the D1-customers, in increasing order, be denoted by $1_1, 2_1, \ldots, I_1$, with I the total number of D1-customers. Similarly, D2-customers have indices $1_2, 2_2, \ldots, J_2$, with J the total number of D2-customers. Hence, customer 1 (which is closest to the depot) is either 1_1 or 1_2 and similarly, $N = I_1$ or $N = J_2$. Let n be the index of the customer farthest from the depot that still needs assignment to a vehicle, and let i_1 and j_2 be the indices of the D1- and D2-customer farthest from the depot that still need assignment. Hence, $n = i_1$ or $n = j_2$. Let M be the total number of vehicles available per day and let m_1 and m_2 be the number of remaining available vehicles for day 1 and 2, respectively. Define the value 0 for the indices i_1 and j_2 for the case that no D1- and D2-customer, respectively, exists or none is still to be assigned and let $x^+ = \max\{x, 0\}$.

A state of the DP is denoted by $<i_1, j_2, m_1, m_2>$. Let $f(i_1, j_2, m_1, m_2)$ be the minimal cost (time) of serving all customers in this state. The total minimal costs of an instance is given by $f(I_1, J_2, M, M)$. We present two recursion formulas: for $i_1 > j_2$, in which a D1-customer is the farthest unassigned customer:

$$f(i_1, j_2, m_1, m_2) = \min_{k=1,\ldots,\min\{C(i_1), i\}} \left\{2d_{i_1} + f\left((i-k)_1, ((j - C(i_1) + k)^+)_2, m_1 - 1, m_2\right)\right\}$$

and for the opposite case $j_2 > i_1$:

$$f(i_1, j_2, m_1, m_2) = \min \left\{ \begin{array}{l} \min_{k=0,\ldots,\min\{C(j_2), i\}} \\ \qquad \{2d_{j_2} + f\left((i-k)_1, ((j - C(j_2) + k)^+)_2, m_1 - 1, m_2\right)\}, \\ 2d_{j_2} + f\left(i_1, ((j - C(j_2))^+)_2, m_1, m_2 - 1\right) \end{array} \right\}$$

The second recursion (implicitly) compares three situations: at least one D1-customer included, only D2-customers with service on day 1, only D2-customers with service on day 2. By the restriction of the choice of k in the recursion it is avoided that $i - k$ could become less than 0. But $j - C(n) + k < 0$ may occur (though never optimal in combination with $i - k > 0$). Define the following starting conditions:

$$\begin{array}{ll} f(0, 0, m_1, m_2) = 0 & \forall m_1, m_2 \geq 0 \\ f(i_1, j_2, x, m_2) = \infty & \forall i_1 > 0,\ m_2 \geq 0,\ x \leq 0 \\ f(i_1, j_2, x, y) = \infty & \forall \max\{i_1, j_2\} > 0,\ x, y \leq 0. \end{array}$$

2.4.2 Including Customers with Period 1

It remains to extend the proof to the general case in which customers with period 1 (service is required on both days) can be present. First, we argue that the same reasoning as in Lemma 2.2 still provides an optimal solution if there are customers with period 1. Second, the DP is adjusted to cover for these customers.

First, interpret P1-customers as two customers, where the first is an additional D1-customer and the second customer must be served on day 2. This observation leads to three sets of customers. The first set of customers $\mathcal{A}$, with indices $1_1, 2_1, \ldots, I_1$, has to be served on day 1. Note that set $\mathcal{A}$ contains both the original D1-customers and the converted D1-customers. The second set $\mathcal{B}$, with indices $1_2, 2_2, \ldots, J_2$, contains the

D2-customers. Finally, the third set of customers $\mathcal{C}$, with indices $1_3, 2_3, \ldots, \Lambda_3$, must be served on day 2.

The need for having three sets is illustrated in Figure 2.2. Suppose that A_1 has already been assigned to a vehicle, so that A_2 is the farthest unassigned customer. Further, assume that $C(B) = 3$ and $C(A_2) = 2$. Now, it is optimal to serve A_2 and D_2 together on day 2 and to serve B, C and D_1 together on day 1. Observe that in the optimal solution there is a vehicle that does not serve consecutive customers of $\mathcal{B} \cup \mathcal{C}$. However, it does serve consecutive customers of every set as defined above, which holds for any optimal solution and is formalized in Lemma 2.3.

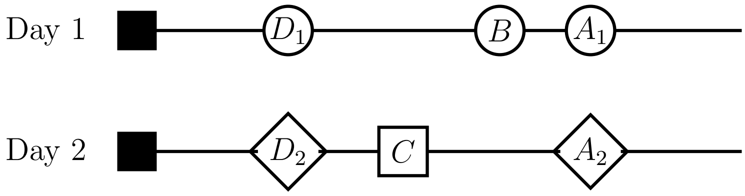


Figure 2.2 Example in which elements of $\mathcal{A}$ are represented by circles, elements of $\mathcal{B}$ by squares and elements of $\mathcal{C}$ by diamonds.

Lemma 2.3. *Starting from the farthest customer and moving towards the depot, no vehicle skips a customer in set $\mathcal{Z}$ to serve another customer in set $\mathcal{Z}$ closer to the depot, for $\mathcal{Z} \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$.*

The proof of Lemma 2.3 is similar the proof of Lemma 2.2 and is omitted for reasons of conciseness.

To describe the dynamic programming algorithm we define the following notation. Again, let n be the index of the customer farthest from the depot that still needs assignment to a vehicle, and let i_1, j_2 and λ_3 be the indices of the farthest customers in sets $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$, respectively, that still needs assignment. Hence, $n = \max\{i_1, j_2, \lambda_3\}$. Let M be the total number of vehicles available per day and let m_1 and m_2 be the number of remaining available vehicles for day 1 and 2, respectively. Redefine k to be the number of customers from set $\mathcal{A}$ that are served by a vehicle on day 1 and define ℓ to be the number of customers from set $\mathcal{C}$ that are served by a vehicle on day 2. Note that if $n = j_2$, then depending on the day j_2 is served, either customers from $\mathcal{A}$ or customers from $\mathcal{C}$ can be served by the same vehicle that serves j_2. Define the value 0 for the indices i_1, j_2 and λ_3 for the case that no customer exists in sets $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$, respectively, equivalent to the definition in Section 2.4.1.

Denote the current state of the DP by $<i_1, j_2, \lambda_3, m_1, m_2>$ and let $f(i_1, j_2, \lambda_3, m_1, m_2)$ be the minimal cost (time) of serving all customers in this state. The total minimal costs of an instance is given by $f(I_1, J_2, \Lambda_3, M, M)$. Three recursion formulas define the DP: the first one for $\max\{i_1, j_2, \lambda_3\} = i_1$, i.e., a customer in $\mathcal{A}$ is the farthest unassigned customer:

$$\begin{aligned} f(i_1, j_2, \lambda_3, m_1, m_2) = \min_{k=1,\ldots,\min\{C(i_1),i\}} \{2d_{i_1} + \\ f\left((i-k)_1, ((j - C(i_1) + k)^+)_2, \lambda_3, m_1 - 1, m_2\right)\} \end{aligned}$$

the second one for $\max\{i_1, j_2, \lambda_3\} = j_2$:

$$f(i_1, j_2, \lambda_3, m_1, m_2) = \min \left\{ \begin{array}{l} \min_{k=0,\ldots,\min\{C(j_2),i\}} \{2d_{j_2} + \\ \qquad f\left((i-k)_1, ((j - C(j_2) + k)^+)_2, \lambda_3, m_1 - 1, m_2\right)\}, \\ \min_{\ell=0,\ldots,\min\{C(j_2),\lambda\}} \{2d_{j_2} + \\ \qquad f\left(i_1, ((j - C(j_2) + \ell)^+)_2, (\lambda - \ell)_3, m_1, m_2 - 1\right)\} \end{array} \right\}$$

and the third one for $\max\{i_1, j_2, \lambda_3\} = \lambda_3$:

$$f(i_1, j_2, \lambda_3, m_1, m_2) = \min_{\ell=1,\ldots,\min\{C(\lambda_3),\lambda\}} \{2d_{\lambda_3} + \\ f\left(i_1, ((j - C(\lambda_3) + \ell)^+)_2, (\lambda - \ell)_3, m_1, m_2 - 1\right)\}.$$

The following starting conditions hold:

$$\begin{array}{ll} f(0, 0, 0, m_1, m_2) = 0 & \forall m_1, m_2 \geq 0 \\ f(i_1, j_2, \lambda_3, x, m_2) = \infty & \forall i_1 > 0, m_2 \geq 0, x \leq 0 \\ f(i_1, j_2, \lambda_3, m_1, x) = \infty & \forall \lambda_3 > 0, m_1 \geq 0, x \leq 0 \\ f(i_1, j_2, \lambda_3, x, y) = \infty & \forall \max\{i_1, j_2, \lambda_3\} > 0, x, y \leq 0. \end{array}$$

2.4.3 Running Time and Generalizations

The running time of the DP is polynomial. To see this, note that the number of states to be considered is $\mathcal{O}(N^3M^2)$ and each computation of the recursion takes $\mathcal{O}(N)$ time. Moreover, we may assume without loss of generality that $M \leq N$. Hence, our DP runs in $\mathcal{O}(N^6)$ time.

Theorem 2.4. *Problem $M|half\text{-}line|s|2$ can be solved in $\mathcal{O}(N^6)$ time.*

A first possible generalization of problem $M|half\text{-}line|s|2$ is extending the planning horizon to more than two days, for example three days. Then, immediately an essentially different ingredient is added to the problem: serving a D2-customer with period 2 on day 1 leads to the obligation of serving it again latest on day 3. This periodicity, asking for repetitive service, is the main issue in problems related to the PSP. As mentioned in Section 2.2.2, this is a badly-understood problem.

A second generalization concerns the underlying metric spaces. The DP can be adapted to yield a polynomial time algorithm for problem $M|\cdot|s|2$ on a line or a cycle. Furthermore, the problem on a tree is NP-hard, even if the planning horizon is a single day and the tree is a star with the depot at the center. This follows again through equivalence to BPP, because the travel time to each customer can be different. Section 2.5 considers the IRP as defined in Section 2.2 in the Euclidean plane.

2.4.4 Vehicle Capacity

Another aspect that can be incorporated in the studied IRP, is vehicle capacity in terms of load, i.e., the maximum number of units demand that can be delivered by a vehicle in one day. To facilitate the exposition of the impact of vehicle capacity constraints, consider the case which discards service times.

In case the demand of each customer can be different, similar argumentation as in Section 2.3.2.1 for problems with arbitrary service times can be used to establish NP-hardness for all problems with more than one vehicle or more than one day. In case all demands are identical, the same argumentation can be followed as for equal service times ($s_i = s \ \forall i$) and tour duration limit L in all problem variants. Also the DP still holds for the equivalent problem by only adjusting the definition of the function $C(\ell)$ that defines the number of customers that can be served by a vehicle given that customer ℓ is served. Note that if a vehicle only has a capacity constraint instead of a tour duration constraint, the maximal number of customers that can be served $C(\ell)$ is the same for any customer ℓ. By discarding service times and replacing them with uniform demands for the customers and a capacity constraint on the vehicle, solving the problem on a star is no longer hard since BPP in terms of service time is no longer an underlying hard problem.

2.5 Inventory Routing in the Euclidean Plane

Consider the IRP as defined in Section 2.2 in the Euclidean plane with a single vehicle, uniform service times at the customers ($s_i = s \ \forall i$) and a time horizon of Z days, denoted by $1|plane|s|Z$. Again, the travel time is equal to the total distance traveled. To avoid immediate NP-hardness from routing, we approximate the route length which provides an easy route length computation, instead of computing the exact optimal route length which requires solving TSPs [Papadimitriou, 1977]. In spite of trivializing the routing cost computation, we show that the resulting problem is NP-hard.

This variant of the IRP is interesting to investigate theoretically, given the discussion on the PSP in Section 2.2.2, but also has a practical application [Baller et al., 2019b]. For the tour length approximation, we use a result of Beardwood et al. [1959] who show that the tour length is asymptotically equal to $\phi\sqrt{A \cdot N}$ for large N, where ϕ is a constant and A is the surface of the area in which the N points can be placed uniformly at random. Chien [1992] considers approximation functions with a similar functional form, but considers several areas for A which take the actual depot and customer locations into account instead of the area in which the customers can be located as in Beardwood et al. [1959]. As an approximation to the route length we use the same functional form as Beardwood et al. [1959] and Chien [1992], and compute the area of the customers as the convex hull of the locations of the customers. The objective is to find a feasible solution, obeying customer periods and the tour duration constraint, that minimizes the total approximated route length. This section shows strong NP-hardness for the studied IRP with this tour length approximation as route length function.

Note that this IRP with the route length approximation is a generalization of the PSP in which the tasks are executed at different locations and more than one task can be scheduled per day. Hence, this section shows that we can prove NP-hardness of a generalization of the PSP in which the tasks are executed at different locations without using the hardness of TSP. Besides that, this IRP has features of the Joint Replenishment Problem (JRP), which is an NP-hard problem [Arkin et al., 1989]. The JRP is a multi-period replenishment problem in which a fixed fee is incurred per customer replenishment and per period in which at least one customer is replenished. Since in the JRP a fixed fee is paid per served customer, the JRP is not a special case of the IRP with approximated route length because of the different cost structure.

A reduction from *3-Partition* [Garey and Johnson, 1979] shows strong NP-hardness for the studied IRP in the plane with approximated route length. *3-Partition* is defined as follows: given $3m$ integers $a_1, \ldots, a_{3m}$ such that $\sum_{i=1}^{3m} a_i = mB$, the question is whether there exists a partition in sets $S_1, \ldots, S_m$ such that $|S_j| = 3$ and $\sum_{i \in S_j} a_i = B$ for $j = 1, \ldots, m$. The problem is hard even if $B/4 < a_i < B/2$ for $i = 1, \ldots, 3m$.

Theorem 2.5. *IRP in the plane with approximated route length is strongly NP-hard.*

Proof. Given an instance of *3-Partition*, create the following instance of IRP in the plane. Create customers with two different periods, period 1 and period m. Take $(c_1, \ldots, c_{3m})$ as the $3m$ extreme points of a regular polygon with area P. One of these points is chosen as the depot. Let all other points contain a customer with period 1.

Second, choose another $3m$ locations outside the polygon, each making a triangle of area Q with two neighboring extreme points of the regular polygon, as depicted in Figure 2.3. Let each of these locations have a set of customers with period m. At the first such location a number of a_1 customers is located, corresponding to the integer value a_1 from the 3-Partition instance, at the second location a_2 customers are located, etc. Moreover, P and Q are chosen such that an angle θ in Figure 2.3 is at most 180°. There is a single vehicle with tour duration limit $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$.

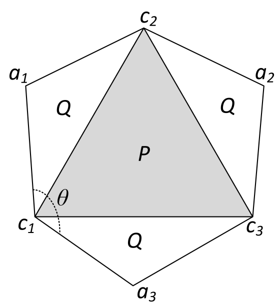


Figure 2.3 Positions customers for $m = 1$

Thus, there are $\sum_{i=1}^{3m} a_i = mB$ Pm-customers spread over $3m$ locations. We will show that there is a 3-Partition if and only if there is a feasible schedule in which no customer is out of stock and every day the approximated length of the tour is at most $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$.

If there is a 3-Partition, serve all customers at the location corresponding to integer a_i for each a_i in set S_j on day j. Moreover, all $3m$ P1-customers are served every day. Clearly, this is a feasible solution for the IRP instance. It remains to show that the bound on the route length holds. The total area per day to be covered in the route length function is P for the $3m$ customers with period 1 and Q per set a_i. Since there are exactly three such sets every day, the total area per day is $P + 3Q$. The number of customer services is $3m$ for the customers with period 1 plus $\sum_{k \in S_j} a_k$ for the selected sets on day j which is exactly equal to B by the 3-Partition. Hence, each day the approximated length of a tour is $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$.

Reversely, if there is a feasible schedule for the planning horizon of m days for which the approximated route length on each day is at most $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$, a feasible 3-Partition can be derived. First, since $\sum_i a_i = mB$, B Pm-customers and

$3m$ P1-customers should be served on average per day. Since $a_i < B/2$, serving more than $B+3m$ customers can only be done by adding at least three times Q to the area of the convex hull. Hence, exactly $B+3m$ customers will be served every day. This means that the set of Pm-customers can be partitioned into subsets with exactly B customers each. Then, because $B/4 < a_i < B/2$ for all i, on each day exactly three sets of customers a_i are served. Hence, the schedule corresponds to a feasible 3-Partition. $\square$

2.6 Conclusion

The main positive result in this paper is a polynomial time dynamic programming algorithm for the borderline problem variant $M|half\text{-}line|s|2$. In this problem on the half-line the planning horizon is two days, there are $M > 1$ vehicles available to serve the customers with uniform service times s.

If we extend the planning horizon from two days to any number of days, the problems, even on a point, are as least as hard as the Pinwheel Scheduling Problem (PSP), for which the complexity has not been determined. This is due to the fixed periods of the customers, which leads to a compact input description. The complexity of the considered IRP is open if the number of days is fixed but greater than 2, e.g., if $M|half\text{-}line|s|3$. These problem variants may very well be polynomially solvable, independent of the complexity of the PSP.

Less surprising is that allowing customers to have arbitrary service times introduces bin packing aspects into the problem, making the resulting IRP NP-hard, even when defined on a point. Our results show that not only the presence of a routing problem contributes to the complexity of the IRP, but also the service times and the periodicity of replenishments of the customers.

3

The Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs

3.1 Introduction

During the last decades, Vendor-Managed Inventory (VMI) systems have received a lot of attention in the literature [Andersson et al., 2010]. In such a system, a supplier manages the inventory of its customers and arranges the transportation of the replenishments. The supplier bears both the inventory holding and transportation costs and therefore strives to minimize these costs by optimizing inventory and shipping decisions. If the supplier would decide on the replenishments and the routes to deliver the replenishments, the supplier faces a problem known as the Inventory Routing Problem (IRP) [Coelho et al., 2014]. However, the transport of the replenishments is often outsourced to a Logistics Service Provider (LSP). As a consequence, the supplier pays a fixed transportation fee for a delivery that is specified in a long-term contract. The supplier therefore faces an optimization problem known as the Joint Replenishment Problem (JRP) [Khouja and Goyal, 2008]. When customer demand varies over time, this problem is known as the Dynamic-Demand Joint Replenishment Problem (DJRP). The DJRP decides which products to order or customers to serve in which periods of the planning horizon such that demand is satisfied at minimal inventory holding and servicing costs. More specifically, the cost of servicing a group of customers in a given period consists of two components, the first of which is a common set-up cost per period if at least one customer is served in that given period. The second component is a cost for each replenished customer. Because of the common set-up cost per period it can be beneficial for the supplier to have some customers replenished together with

This chapter is based on: A.C. Baller, S. Dabia, W.E.H. Dullaert, and D. Vigo, The Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs, European Journal of Operational Research, 276:1013-1033, 2019 [Baller et al., 2019b] other customers even before stock runs low. In that case the inventory holding cost is higher because more inventory is kept, but it allows the supplier to save the fixed fee for a period.

The DJRP encompasses a number of key problem features occurring in real-life applications, but also suffers from a number of drawbacks. First, because the actual transportation costs are not directly considered, the DJRP cannot identify closely situated customers that would be well-suited for joint replenishment. Hence, customers served in one period may be randomly located in a region, resulting in high actual transportation costs for the LSP. Also, the number of replenishments that the LSP is able to perform on behalf of the supplier can be lower due to large distances between deliveries. Therefore, consideration of proximity of customers could result in better utilization of transportation resources, decrease actual transportation costs for the LSP and eventually decrease transportation fees for the supplier. In short, based on the DJRP, the supplier generates requests that are expensive or hard to fulfill. Second, the DJRP ignores duration constraints. Vehicle capacity constraints have been considered in the JRP literature (see e.g. Anily and Tzur [2005]) and in many routing problems, including most studies on the IRP. However, tour duration constraints have proven to be more binding in several practical applications, such as online ordered package delivery, blood product distribution [Hemmelmayr et al., 2009] and replenishment of ATMs. Tour duration constraints are rarely found in the IRP literature. Finally, the DJRP does not take limited customer storage capacity into account, but in practice, storage capacity is often restricted. To address these shortcomings of the DJRP we propose an extension of the DJRP, the DJRP with Approximated Transportation Costs (DJRP-AT), that explicitly considers transportation costs. Furthermore, the DJRP-AT contains tour duration constraints and limits customer storage capacity. Because determining the optimal delivery tour is computationally expensive, we will approximate the transportation costs in the DJRP-AT by approximating the shortest traveling salesman tour. The only work that we are aware of that includes approximated transportation costs in a VMI setting is Larsen and Turkensteen [2014]. The authors consider a VMI setting with stochastic demand and order-up-to-levels at the customers which they solve with a Markov Chain simulation model.

Our research is motivated by ATM replenishment in the Netherlands. A single supplier (vendor) decides on the timing of ATM cash replenishment and on the delivery quantity. The actual ATM replenishment orders per day are outsourced to an LSP, in this application often referred to as Cash-in-Transit company (CIT), that schedules and performs the daily delivery routes. Currently, the supplier pays a fixed fee to the LSP for each ATM replenishment. Therefore, the current replenishment policy ignores the impact that ordering decisions have on distance traveled and vehicle utilization. The supplier is reconsidering the replenishment cost structure to better align decisions. To provide insight for future negotiations between the supplier and the LSP, we examine the benefit of adopting a DJRP and a DJRP-AT perspective. How a new ordering policy is to be incorporated in the contract between the supplier and the LSP, is beyond the scope of this paper. With this work we contribute to the recent stream of publications on ATM replenishment. For example, Van Anholt et al. [2016] develop a heuristic for a pickup and delivery IRP for an advanced type of ATMs. Larrain et al. [2017] focus on a local search based heuristic for an IRP in which stock-outs are allowed and cash is replenished by swapping cassettes.

To solve the DJRP-AT, this paper proposes a compact formulation in which transportation costs and inventory holding costs are minimized. Note that inventory holding costs in the application relate to the value of money or lost interest. The compact formulation is split by applying Dantzig-Wolfe decomposition [Desrosiers and Lübbecke, 2005]. The resulting Master Problem and Pricing Problem are solved in a Branch-and-Cut-and-Price framework [Nemhauser and Park, 1991, Lübbecke and Desrosiers, 2005]. The Master Problem selects customer subsets to be delivered and determines the corresponding delivery quantities. The Pricing Problem generates these customer subsets using a labeling algorithm with tailored dominance criteria to speed up the process. The solution method is tested on benchmark instances from the literature and on instances derived from a real-life case in ATM replenishment.

The contributions of this paper are threefold. First, the DJRP is extended to incorporate transportation costs, limitations on storage capacity at the customers and restricted tour duration. The results show that the proposed model leads to lower total costs compared with the DJRP. Second, we introduce novel dominance conditions for the labeling algorithm that is used to solve the Pricing Problem. Finally, existing valid inequalities originating from the inventory routing literature are tested, their impact on the integrality gap is demonstrated and it is shown that their effectiveness is different than for other models in which they have been applied.

The remainder of this paper is organized as follows: Section 3.2 discusses literature on the JRP and DJRP, together with their relation to the IRP. The DJRP-AT is described and modeled in Section 3.3. Section 3.4 proposes a decomposition of the model and specifies the Master and Pricing Problems. The algorithm to solve the Pricing Problem, including novel sufficient dominance conditions, the valid inequalities and the branching strategy are presented in Section 3.5. Section 3.6 presents the results of the experiments on benchmark instances from the literature, introduces the real-life case and reports on the results of instances derived from the real-life case. Finally, the conclusions and directions for further research are discussed in Section 3.7.

3.2 Literature review

The traditional JRP is the problem of minimizing holding and ordering costs, while ensuring that no customer runs out of stock in any period of the planning horizon. The ordering costs consist of a common set-up cost per period and a fixed fee per replenishment. An overview of the literature on the JRP from 1989 to 2005 distinguishes three types of models [Khouja and Goyal, 2008]: first, the traditional JRP, considers deterministic and static demand. This means that demand is known beforehand and remains the same for every period of the planning horizon. For this problem, analytical expressions have been derived for the minimal total costs and heuristics have been designed to determine the corresponding cyclic replenishment policy. Second, the extension to stationary stochastic demand in which the objective is to minimize the expected total cost. Solution methods mainly consist of using a periodic review policy or a can-order policy. Finally, the JRP with deterministic and dynamic demand (DJRP) in which the demand is known but can vary across periods is discussed. The solution for this type of problem is not necessarily a cyclic replenishment policy as for the traditional JRP. For the DJRP different formulations and heuristic solution methods have been proposed and studied [Webb et al., 1997, Boctor et al., 2004, Narayanan and Robinson, 2006, Robinson et al., 2007] and Robinson et al. [2009] have provided an overview of available solution methods. Webb et al. [1997] studied fixed replenishment cycle models for the problem and compared these to optimal solutions that do no constrain the replenishment cycle. Boctor et al. [2004] proposed several linear programming formulations, tested several heuristic solution methods and proposed an improvement procedure that can be used in combination with a heuristic method.

To increase practical relevance of the DJRP, several extensions have been proposed such as capacitated aggregate order size [Anily and Tzur, 2005, Federgruen et al., 2007, Narayanan and Robinson, 2010], supplier selection [Ventura et al., 2013], supplier selection with discounts [Kang et al., 2017], inventory decisions at the supplier [Solyalı and Süral, 2012, Cunha and Melo, 2016] as well as inventory decisions and capacitated production at the supplier [Senoussi et al., 2016]. A commonly occurring practical constraint is an inventory capacity limit at the customers. However, to our knowledge, this constraint has only been included in one paper on the DJRP [Senoussi et al., 2016]; two papers on the traditional JRP also include this constraint [Hoque, 2006, Hariga et al., 2013].

The IRP combines an inventory problem and a routing problem: it minimizes inventory holding and routings costs by optimizing replenishments for a set of customers and explicitly determining the delivery routes. The IRP is therefore related to the JRP, yet the IRP is structurally different from the JRP because the routing problem is explicitly solved. Various solution methods for the IRP have been proposed in the literature such as exact methods, matheuristics and metaheuristics (see Coelho et al. [2014] for an overview). Some of the exact solution methods for the IRP rely on the vehicle capacity constraint, for example in the Pricing Problem algorithms and valid inequalities. In our application a tour duration constraint is more appropriate. In some of the heuristic solution methods for the IRP, inventory and routing optimization are considered separately. In a first phase, decisions are made on the inventory policies, often incorporating a fixed replenishment cost per delivery, thus solving a variant of the JRP. In a second phase, routing is optimized given the replenishment decisions of the first phase. Iterative solution schemes have for example been proposed by Cordeau et al. [2015] and Absi et al. [2015].

Some attention has also been paid to the fact that charging a fixed fee for servicing a customer in the DJRP is not always representative for the actual costs involved. A fixed fee per customer replenishment assumes that the costs for replenishing customers are independent, but in practice, this is not always true. Olsen [2008] used the example of using a refrigerated truck for canned food delivery, which increases the marginal replenishment costs of the canned food. Olsen [2008] and Wang et al. [2012] proposed to model the marginal costs with additional fixed fees depending on the combination of items delivered. Senoussi et al. [2016] recognized that actual transportation cost are relevant, however, they assumed that the depot is located far away from a cluster of customers and that the transportation costs between the clustered customers are negligible, therefore the authors assumed that the costs of a tour are fixed. Rahmouni and Hennet [2015] took actual routing costs into account by combining the deterministic and static JRP with the Traveling Salesman Problem (TSP). For each possible subset of customers the actual tour length was computed beforehand by solving a TSP, then a linear programming model was used to select the optimal subsets and to determine the delivery quantities. However, this solution method can only be applied to instances of very limited size since for all combinations of customers the traveling salesman tour has to be computed.

3.3 Problem description

In the DJRP-AT, a single supplier supplies N customers. The customers face a certain demand per period and have a limited storage capacity, and therefore require replenishments to prevent them from running out of stock. The supplier arranges the customer replenishments in a VMI setting with the objective of minimizing transportation and customer inventory holding costs. The transportation costs in a period are represented by the approximated tour length visiting the replenished customers and a fixed set-up fee for a period if at least one customer is replenished in that period. In the DJRP-AT, constraints are incorporated on the composition of the set of customers served in one period, e.g., the number of customers served or the tour duration. Note that the inventory holding costs at the supplier are not considered since in our practical application there is an infinite supply (similar to Larrain et al. [2017]), but these costs could easily be added.

For the calculation of the transportation costs, consider that these costs must be estimated for a large number of customer subsets, which requires careful balance of approximation accuracy and calculation effort. Also, two sets of customers with the same cardinality, but with customers at different locations should result in different transportation costs. Therefore, based on the short literature review in Appendix A, we adopt the tour length approximation model of Chien [1992]:

$$D \approx 0.98\sqrt{RM'}, \tag{3.1}$$

in which R is the area of the smallest rectangle covering both the customers and the depot, and M' is the number of points in the tour (depot and customers). Note that this function underestimates the actual TSP tour length. The transportation costs also include a fixed cost B that is independent of the distance traveled, but is for example a setup cost related to vehicle use. Define binary vector $\boldsymbol{Y}$ to indicate which customers are served and binary variable $\hat{Y}$ to indicate whether any customer is served ($\sum_i Y_i \geq 1$). The following transportation cost function will be used to approximate the transportation costs for the customers in $\boldsymbol{Y}$ and the depot

$$f(\boldsymbol{Y}) = B\hat{Y} + 0.98\sqrt{R(\boldsymbol{Y})M'(\boldsymbol{Y})}. \tag{3.2}$$

To formulate the DJRP-AT, consider the following notation in Boctor et al. [2004]. A single depot and a set of $\mathcal{N} = \{1, 2, \ldots, N\}$ customers are positioned in Euclidean space and there is a finite time horizon $\mathcal{T} = \{1, 2, \ldots, T\}$. Let Y_{it} denote the binary decision variable that takes value 1 if and only if customer $i \in \mathcal{N}$ is visited in period $t \in \mathcal{T}$. Let $\boldsymbol{Y_t}$ denote the vector $\{Y_{1t}, Y_{2t}, \ldots, Y_{Nt}\}$. Define X_{it} as the quantity delivered to customer $i \in \mathcal{N}$ in period $t \in \mathcal{T}$ and let I_{it} be the quantity in stock at customer $i \in \mathcal{N}$ at the end of period $t \in \mathcal{T}$. I_{it} should be non-negative, because stock-outs are not allowed. The inventory level is measured at the end of the period assuming the following order of events: delivery of new stock, consumption, inventory calculation. This assumption coincides with JRP literature [Boctor et al., 2004] and with most literature on the IRP [Archetti et al., 2014a]. I_{i0} denotes the initial inventory level and

d_{it} is the dynamic and deterministic demand in period $t \in \mathcal{T}$ at customer $i \in \mathcal{N}$. For the items in stock at a customer $i \in \mathcal{N}$ an inventory holding rate of h_{it} is charged per period $t \in \mathcal{T}$.

We introduce the following additional notation for the DJRP-AT. Each customer $i \in \mathcal{N}$ has a storage capacity u_i. Define for each $i \in \mathcal{N}$ and $t \in \mathcal{T}$ big-M value $M_{it} = \min\left\{u_i, \sum_{s=t}^{T} d_{it}\right\}$. Furthermore, a single vehicle with unlimited load capacity performs at most one route in each time period, beginning and ending at the depot. Transportation costs in a period are represented by function $f(\cdot)$ as defined in equation (3.2). Finally, let function $g(\cdot)$ assess the composition of the tour in a given period. We consider two different functions for $g(\cdot)$. First, let $g(\cdot)$ be the approximated tour duration and define $k_D \in \mathbb{R}$ as the maximum tour duration. Second, we let $g(\cdot)$ be the number of customers in a tour and impose that at most $k_M \in \mathbb{N}$ customers can be served in a single tour. These constraints will be referred to as 'subset composition constraints' for the remainder of the paper. DJRP-AT models will only contain one of these two types of constraints to assess the tour composition.

The goal of the DJRP-AT is to minimize inventory holding and transportation costs by selecting, for each period, which customers to replenish, while avoiding stock-out at any customer, without violating the customer's storage capacity restrictions and the additional restrictions on the tour composition. This problem can be formulated as follows:

$$
\begin{aligned}
z = \min \quad & \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{N}} h_{it}I_{it} + \sum_{t\in\mathcal{T}} f(\boldsymbol{Y_t}) && && \text{(3.3a)}\\
\text{s.t.} \quad & I_{it} = I_{i,t-1} - d_{it} + X_{it} && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3b)}\\
& X_{it} \le u_i - I_{i,t-1} && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3c)}\\
& X_{it} \le M_{it}Y_{it} && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3d)}\\
& g(\boldsymbol{Y_t}) \le k && \forall t \in \mathcal{T} && \text{(3.3e)}\\
& I_{it} \ge 0 && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3f)}\\
& X_{it} \ge 0 && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3g)}\\
& Y_{it} \in \{0,1\} && \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} && \text{(3.3h)}
\end{aligned}
$$

in which $k = k_D$ or $k = k_M$, depending on the applied subset composition constraints. The objective function (3.3a) minimizes the costs for inventory holding and transportation $f(\cdot)$ defined in (3.2). The inventory balance for each customer in each period is maintained by constraints (3.3b). Constraints (3.3c) ensure that the customer's capacity is not exceeded when a delivery is made and constraints (3.3d) force the amount delivered to zero if a customer is not visited. Furthermore, constraints (3.3e) represent the additional constraints on the composition of the subset of customers replenished in a period. Constraints (3.3f), (3.3g) and (3.3h) impose binary and non-negativity constraints on the decision variables.

3.4 Column generation

When considering the complexity of the DJRP-AT with cost function (3.2), it is important to note that the traditional JRP is not a special case of the DJRP-AT, due to the different cost structure. Hence, although the traditional JRP is NP-complete [Arkin et al., 1989], this conclusion cannot be directly made for the DJRP-AT. Furthermore, analysis of the literature on the complexity of related problems shows that the so-called Pinwheel Scheduling Problem is a special case of the DJRP-AT with cost function (3.2). The Pinwheel Scheduling Problem is likely to be an NP-complete problem [Jacobs and Longo, 2014], but this has not yet been proven despite several attempts. A mapping between the Pinwheel Scheduling Problem and the DJRP-AT, including details on the complexity of the Pinwheel Scheduling Problem, are presented in Appendix B. Because the existence of a polynomial-time algorithm is unlikely and column generation has proven to be efficient for similar problem structures, this solution method will be used to solve the DJRP-AT.

Application of the Dantzig-Wolfe decomposition to problem (3.3a)-(3.3h) results in a Master Problem that selects for every period a subset of customers to replenish out of a collection of subsets to minimize the inventory holding and transportation costs. Moreover, the Master Problem optimizes the delivery quantities corresponding to constraints (3.3b)-(3.3d) in the compact formulation. The Pricing Problem generates subsets of customers, taking constraints (3.3e) into account, and is solved for each period separately.

To formulate the Master Problem, let $\mathcal{S}_t$ be the collection of subsets of customers that are generated by the Pricing Problem for period $t \in \mathcal{T}$. The binary decision variable Z_{st} equals 1 if subset $s \in \mathcal{S}_t$ is selected for period $t \in \mathcal{T}$. For a specific subset $s \in \mathcal{S}_t$ the transportation costs c_s for servicing its customers is given by the Pricing Problem. Furthermore, let a_{is} indicate whether customer $i \in \mathcal{N}$ is present in subset $s \in \mathcal{S}_t$. The decomposition gives the following Master Problem:

$$z = \min \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{N}} h_{it} I_{it} + \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}_t} c_s Z_{st} \tag{3.4a}$$

$$\text{s.t. } I_{it} = I_{i,t-1} - d_{it} + X_{it} \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.4b}$$

$$X_{it} \leq u_i - I_{i,t-1} \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.4c}$$

$$X_{it} \leq u_i \sum_{s\in\mathcal{S}_t} a_{is} Z_{st} \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.4d}$$

$$\sum_{s\in\mathcal{S}_t} Z_{st} \leq 1 \qquad \forall t \in \mathcal{T} \tag{3.4e}$$

$$I_{it} \geq 0 \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.4f}$$

$$X_{it} \geq 0 \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.4g}$$

$$Z_{st} \in \{0, 1\} \qquad \forall s \in \mathcal{S}_t,\ \forall t \in \mathcal{T} \tag{3.4h}$$

The objective function (3.4a) aims to minimize total costs. Constraints (3.4b)-(3.4d) are equivalent to constraints (3.3b)-(3.3d) of the compact formulation. Constraint (3.4e) ensures that at most one subset of customers is selected for each period. Finally, non-negativity and binary requirements on the decision variables are imposed by constraints (3.4f)-(3.4h).

We use column generation to solve the linear programming relaxation of (3.4a)-(3.4h) since the total number of variables Z_{st} is exponentially large. Starting with a small subset of all possible columns gives the Restricted Master Problem (RMP) and additional columns with negative reduced cost are generated by repeatedly solving the Pricing Problem. To formulate the Pricing Problem, let us associate the following dual variables with the Master Problem with respect to decision variables Z_{st}. Let π^1_{it} be a non-positive dual variable associated with constraints (3.4d) and let π^2_t be the non-positive dual variable of constraints (3.4e). Let us also reuse decision variables Y_{it} from the compact formulation: these variables indicate whether customer $i \in \mathcal{N}$ is replenished in period $t \in \mathcal{T}$ and remind that $\boldsymbol{Y_t} = \{Y_{1t}, Y_{2t}, \ldots, Y_{Nt}\}$. For a given time period $t \in \mathcal{T}$ the Pricing Problem can be formulated as follows:

$$\min \bar{c}_t(\boldsymbol{Y_t}) = f(\boldsymbol{Y_t}) + \sum_{i \in \mathcal{N}} u_i Y_{it} \pi^1_{it} - \pi^2_t \tag{3.5a}$$

$$\text{s.t. } g(\boldsymbol{Y_t}) \leq k \tag{3.5b}$$

$$\boldsymbol{Y_t} \in \{0,1\}^N \tag{3.5c}$$

The objective (3.5a) is to minimize the reduced cost $\bar{c}_t(\boldsymbol{Y_t})$ while the subset composition constraints are satisfied (3.5b). The reduced cost consists of the transportation costs of the subset $f(\boldsymbol{Y_t})$ and dual terms corresponding to the current solution of the RMP. The subset composition constraints (3.5b) can, in general, concern any function of the combination of customers in the subset. However, these constraints cannot contain the delivery quantities, since these quantities are determined in the Master Problem. Hence, in our model, a load capacity constraint cannot be in the Pricing Problem, but, for example, a tour duration constraint is possible.

3.5 Branch-and-Cut-and-Price

The Master Problem and Pricing Problem of Section 3.4 are solved in a Branch-and-Cut-and-Price framework. In Section 3.5.1, a tailored labeling algorithm to solve the Pricing Problem per period is described and novel sufficient conditions are presented that provide a dominance criterion to discard labels. Valid inequalities are presented in Section 3.5.2 and Section 3.5.3 provides a description of the branching strategy.

3.5.1 Labeling Algorithm for the Pricing Problem

To solve the Pricing Problem, we propose a tailored labeling algorithm that identifies subsets of customers that will improve the current solution of the RMP. Note that during the process of generating subsets of customers, only the customer combination is relevant, there is no sequential relationship between the customers as opposed to routing problems [Feillet et al., 2004].

Define label $L = <s(L),\ \bar{c}_t(L),\ g(L)>$ in which $s(L)$ is the subset of customers, $\bar{c}_t(L)$ is the corresponding reduced cost and $g(L)$ represents the value of the function $g(\cdot)$ in the subset composition constraint for subset $s(L)$. Hence, each label corresponds to a subset of customers that is a candidate to be added to the RMP.

The labeling algorithm starts for each customer $i \in \mathcal{N}$ separately and the labels are extended by adding the other customers one by one. The order of the customers in $s(L)$ is not important, since the subset of customers is considered for replenishment, but the order in which they are served is not determined. Hence, each possible subset has to be considered at most once. Therefore, when starting with a label containing customer i and extending with customer j, the inverse order of these customers, starting with j and adding i, does not have to be considered. The labeling algorithm terminates when all possible subsets of customers are considered.

Denote $L \oplus P$ as the resulting label from the extension of label L with the customers in set $P \subseteq \mathcal{N} \setminus s(L)$. The operation to extend a label L with the next customer j is to set $s(L \oplus \{j\}) = s(L) \cup \{j\}$ and to compute $\bar{c}_t(L \oplus \{j\})$ and $g(L \oplus \{j\})$:

$$\begin{aligned} \bar{c}_t(L \oplus \{j\}) &= \bar{c}_t(L) - f(\boldsymbol{Y_t}(s(L))) + f(\boldsymbol{Y_t}(s(L) \cup \{j\})) + u_j \pi^1_{jt} \\ &= f(\boldsymbol{Y_t}(s(L) \cup \{j\})) + \sum_{i \in s(L) \cup \{j\}} u_i \pi^1_{it} - \pi^2_t \end{aligned} \tag{3.6}$$

in which $\boldsymbol{Y_t}(s(L))$ is the vector in which the variables corresponding to $s(L)$ equal 1. If we consider the model with subset composition constraints that set a maximum on the tour duration, we have

$$g(L \oplus \{j\}) = 0.98\sqrt{R(s(L) \cup \{j\})(|s(L)| + 1)} \tag{3.7}$$

and if we consider the model with subset composition constraints that pose a maximum on the number of customers in the subset, we have

$$g(L \oplus \{j\}) = g(L) + 1 \tag{3.8}$$

The extended label is feasible if

$$s(L) \cap \{j\} = \emptyset \ \wedge \ g(L \oplus \{j\}) \leq k \tag{3.9}$$

When the number of customers N increases, the maximum number of labels becomes large $\left(2^N - 1\right)$. A dominance test will therefore be used to reduce the number of labels.

Denote the set of feasible extensions of label L by $E(L)$ which consists of all combinations of the customers that have not already been considered and for which the extension will satisfy the subset composition constraints. The following definition for dominance holds

Definition 3.1. *Label L dominates label L' if*

D.1 $E(L') \subseteq E(L)$

D.2 $\bar{c}_t(L \oplus P) \leq \bar{c}_t(L' \oplus P), \ \forall P \in E(L')$

The first condition, D.1, states that a feasible extension of L' must also be a feasible extension of L. The second condition, D.2, requires that all feasible extensions of L do not result in worse solutions than the same extensions of L'. These conditions are difficult to check in practice, since all feasible extensions would have to be computed. Therefore, Proposition 1 introduces sufficient conditions for dominance of L over L'.

Proposition 1. *Label L dominates label L' if the following conditions hold*

P.1 $s(L) \subseteq s(L')$

P.2 $g(L) \leq g(L')$

P.3 $\bar{c}_t(L) + \Delta(L, L') \leq \bar{c}_t(L')$

Conditions P.1 and P.2 combined imply condition D.1, such conditions are also used for shortest path problems [Feillet et al., 2004], and condition P.3 implies condition D.2.

Before a formal proof for Proposition 1 is presented, an intuitive reasoning for condition P.3 is given and an expression for $\Delta(L, L')$ is derived.

The cost function, and therefore the reduced costs, are dependent on the number of customers and the area in which these customers are located. Consider a comparison of the two labels L and L' for which it holds that $s(L) \subseteq s(L')$, $\bar{c}_t(L) < \bar{c}_t(L')$ and $g(L) \leq g(L')$. One would like to conclude that L dominates L'. However, if a set of customers P is added to both labels, the area that is used in the cost function can increase more for L than for L', i.e., the additional cost of the extension with P is not identical for both labels. This could result in $\bar{c}_t(L \oplus P) > \bar{c}_t(L' \oplus P)$, therefore, it cannot be concluded that L dominates L' since condition D.2 is violated. Hence, a sufficient dominance condition should be stricter than $\bar{c}_t(L) \leq \bar{c}_t(L')$, therefore sufficient condition P.3 is introduced. This will be illustrated in the following example. Consider an instance with four customers, indicated by white nodes and customer index in Figure 3.1. The depot is indicated by the black node with label D. The current terms for the reduced cost corresponding to each customer are indicated between brackets (suppose $\pi_t = 1000$).

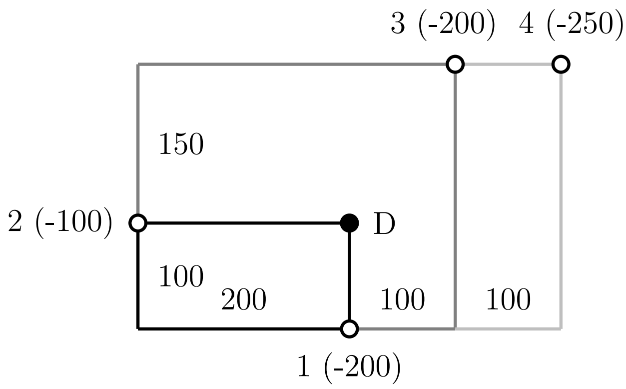


Figure 3.1 Example of dominance in labeling algorithm.

Now, consider subset $s_1 = \{1, 2\}$, the corresponding cost is $f(s_1) = 1000+ 0.98\sqrt{100 \times 200 \times 3} \approx 1240$ and the reduced cost of this subset is $\bar{c}_t(s_1) \approx 1240 - 200 - 100 - 1000 = -60$. Similarly, subset $s_2 = \{1, 2, 3\}$ has $f(s_2) \approx 1537$ and $\bar{c}_t(s_2) \approx 37$. In this case s_1 and s_2 are comparable, since $s_1 \subset s_2$. Note that $\bar{c}_t(s_1) < \bar{c}_t(s_2)$, hence, one would like to conclude that the label with subset s_1 dominates the label with s_2. However, suppose customer 4 is added to both subsets. This gives $s_3 = \{1, 2, 4\}$ with $f(s_3) \approx 1620$ and $\bar{c}_t(s_3) \approx 70$ and $s_4 = \{1, 2, 3, 4\}$ with $f(s_4) \approx 1693$ and $\bar{c}_t(s_4) \approx -57$. Note that the cost increase from subset s_1 to s_3 is larger than from subset s_2 to s_4. Because $\bar{c}_t(s_3) > \bar{c}_t(s_4)$, we cannot conclude from $\bar{c}_t(s_1) < \bar{c}_t(s_2)$ that the label with s_1 dominates the label with s_2.

A more strict condition is required and therefore $\Delta(L, L')$ is introduced, representing the maximum difference in costs between two labels. The value of $\Delta(L, L')$ must be sufficient to guarantee $\bar{c}_t(L \oplus P) \leq \bar{c}_t(L' \oplus P)$ $(\forall P \in E(L'))$ to conclude that L dominates L'. First, express $\bar{c}_t(L \oplus P)$ and $\bar{c}_t(L' \oplus P)$ in terms of $\bar{c}_t(L)$ and $\bar{c}_t(L')$, respectively. Combined with $\bar{c}_t(L) \leq \bar{c}_t(L')$, an upper bound can be derived for the difference between $\bar{c}_t(L \oplus P)$ and $\bar{c}_t(L' \oplus P)$. The derivation for the following sufficient value of $\Delta(L, L')$ is given in Appendix C:

$$\Delta(L, L') = \phi\sqrt{R(s(L'))|s(L') \cup P| - R(s(L))|s(L) \cup P|} \tag{3.10}$$

with $R(\cdot)$ as the area of the smallest rectangle and $P = \mathcal{N} \setminus s(L')$. A formal proof for Proposition 1 with $\Delta(L, L')$ as defined in (3.10) can now be presented.

Proof. Proof of Proposition 1 Assume two labels L and L' with corresponding subsets of customers $s = s(L)$ and $s' = s(L')$ satisfy the conditions in Proposition 1. Given $s \subseteq s'$ and therefore $s \cup P \subseteq s' \cup P$, it holds that if $g(L) \leq g(L')$, then $g(L \oplus P) \leq g(L' \oplus P)$ if $g(\cdot)$ is a monotone function. Hence, if $g(L' \oplus P) \leq k$, then $g(L \oplus P) \leq k$ and condition D.1 is satisfied. To show that condition D.2 holds, note that we have already shown in Appendix C that

$$\bar{c}_t(L \oplus P) - \bar{c}_t(L' \oplus P) \leq \bar{c}_t(L) - \bar{c}_t(L') + \Delta(L, L') \tag{3.11}$$

Hence, if $\bar{c}_t(L) + \Delta(L, L') \leq \bar{c}_t(L')$, then condition D.2 holds which concludes the proof of Proposition 1. □

An overview of the labeling algorithm is provided in Algorithm 1. A label is not extended any further in the labeling algorithm if the subset composition constraint is violated, since no feasible subsets of customers can be found by adding more customers because of the subset composition constraints. Moreover, after extending a label, it is tested whether adding another customer violates a subset composition constraint, in which case an extension to this customer from the current label is not considered in a later stage of the labeling algorithm.

Algorithm 1 Labeling algorithm

Initialize list of improving labels $\mathcal{I}$ and of labels to propagate $\mathcal{P}$
for $i = 0$ to N **do**
  Create label l containing i and add l to $\mathcal{P}$
  if Reduced cost of l is negative **then**
    Add l to $\mathcal{I}$
  while $\mathcal{P} \neq \emptyset$ **do**
    Consider a waiting label $p \in \mathcal{P}$
    for All customers j with higher index than the last added customer to p **do**
      Extend p with j to q
      if q is feasible and p does not dominate q **then**
        Add q to $\mathcal{P}$
      if Reduced cost of $u < 0$ **then**
        Add q to $\mathcal{I}$
    Remove p from $\mathcal{P}$

To accelerate the solution process, a heuristic variant of the labeling algorithm is applied. If the heuristic fails to find any improving subsets, the exact labeling algorithm is applied in which all combinations of customers are considered. In the heuristic pricing instrument, the process is identical to the exact pricing algorithm, but the number of customers to which a label can be extended is limited. Only the extensions to the b customers closest to the last added customer of a subset are evaluated. The initial value of b is small (2) and this value is doubled up to a certain limit (8) as long as no improving subsets of customers are found.

Preliminary experiments showed that adding all columns with negative reduced cost could be time consuming for the algorithm and moreover, many of these columns are not in the final solution. Therefore, at most 10,000 columns are added per call to the Pricing Problem for both the heuristic and the exact pricing instrument.

3.5.2 Valid inequalities

The formulation of the problem can be strengthened with valid inequalities. Two valid inequalities that were introduced by Archetti et al. [2007] for the IRP are also applicable to the DJRP-AT. The first inequality states that if a customer $i \in \mathcal{N}$ is not replenished in periods $t-r, t-r+1, \ldots, t$, then the inventory in period $t-r-1$ should be sufficient to cover demand of all periods up to t:

$$\text{(ILB)} \quad I_{i,t-r-1} \geq \left(\sum_{j=0}^{r} d_{i,t-j}\right)\left(1 - \sum_{j=0}^{r}\sum_{s\in\mathcal{S}_t} a_{is} Z_{s,t-j}\right) \qquad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T},\ \forall r = 0,1,\ldots,t-1 \tag{3.12}$$

The second inequality gives a lower bound for the number of required visits to a specific customer i up to a period t taking the customer's inventory capacity into account:

$$\text{(NrVis)} \quad \sum_{j=1}^{t}\sum_{s\in\mathcal{S}_t} a_{is} Z_{sj} \geq \left\lceil \frac{\sum_{j=1}^{t} d_{ij} - I_{i0}}{u_i} \right\rceil \quad \forall i \in \mathcal{N},\ \forall t \in \mathcal{T} \tag{3.13}$$

Preliminary experiments showed that dynamic management of the valid inequalities, i.e., adding them whenever violated, slowed down the execution. Therefore, for all experiments, all valid inequalities are added to the model in the root node of the Branch-and-Bound tree.

3.5.3 Branching

In the compact formulation (3.3a)-(3.3h), the variables indicating whether a customer is served in a certain time period, the assignment variables, are binary decision variables. The delivery quantity and inventory level decision variables are non-negative and continuous. To find binary results for the assignment variables a Branch-and-Bound tree is used, which is explored via a best bound strategy. First, the algorithm branches on the total number of replenishments of a customer $i \in \mathcal{N}$ over all periods $\sum_{t\in\mathcal{T}} Y_{it}$. In the Master Problem variables this can be expressed as $\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{S}_t} a_{is}Z_{st}$. If all customers have an integer number of replenishments, the second branching method branches on whether any customer is replenished in a period t or no customer is replenished ($\sum_{i\in\mathcal{N}} Y_{it}$). Expressed in the Master Problem variables, the corresponding constraints are $\sum_{s\in\mathcal{S}_t} z_{st} \leq 0$ and $\sum_{s\in\mathcal{S}_t} z_{st} \geq 1$. If no new branches can be identified for the first two branching methods, the algorithm branches on whether a customer $i \in \mathcal{N}$ is visited in a specific period $t \in \mathcal{T}$ or not ($Y_{it} = \sum_{s\in\mathcal{S}_t} a_{is}Z_{st}$). This branching method leads to binary solutions for the assignment variables in the compact formulation, hence, if no new branches are identified an integer solution is found. If there is more than one branch candidate, the following strategy to select a branch is followed for each type of branching. For each branch candidate the child nodes are quickly evaluated by solving the LP relaxation given the current set of columns. The branch that maximizes the lower bound is chosen. We consider at most 25 branch candidates in the first 20 nodes of the branch and bound tree and at most 15 candidates in the other nodes.

3.6 Computational results

The proposed model for the DJRP-AT is analyzed for two types of subset composition constraints: a tour duration constraint and a maximum number of customers served per period (i.e., in a subset). The effectiveness of the valid inequalities presented in Section 3.5.2 is evaluated for both types of subset composition constraints in Section 3.6.1. The model with a maximum tour duration most resembles practical cases; this model is tested with different values for the maximum tour duration. The model with a maximum number of customers per subset can be compared to the DJRP with fixed fees [Boctor et al., 2004] and to a variant of the IRP in which the actual routing costs are used. The first comparison provides insight in the potential improvements that can be achieved with the DJRP-AT compared to the DJRP. Section 3.6.2 explains how the DJRP-AT and the DJRP are compared and provides the results. The second comparison shows how well the DJRP-AT performs compared to an equivalent IRP in which actual routing costs are used, on which Section 3.6.3 reports the results. Note that results of the DJRP-AT are not comparable to the results of 'standard' IRP as often used in the literature [Coelho et al., 2014] because the constraints are different.

The instances for the IRP, created by Archetti et al. [2007], are used for the computational experiments. Although the IRP differs from the DJRP-AT regarding the cost structure, decision variables and constraints, the IRP instances are used as a base, since they contain most of the data required for the experiments on the DJRP-AT. The time horizon is equal to either 3 or 6 periods and instances with 5, 10, 15 and 20 customers are considered; there are five instances for each combination of number of customers and periods. For each customer, a location is given by two coordinates, both randomly chosen within the interval [0, 500]; demand is randomly selected between 10 and 100, and the customer holding rate is in the interval [0.1, 0.5]. The customer's inventory capacity is the demand of the customers multiplied by either 2 or 3, which is randomly selected. The initial inventory is the customer's capacity minus the demand of the first period. The instances are available online [Coelho, n.d.]. For the tests on the model including maximal tour duration the maximum (k_D) is set to 600, 800, 1000 and 1200 for all instances. If a maximum is set on the number of customers per subset (k_M), this maximum depends on the number of customers in the instance. For 5 customers, maxima of 3 and 4 are considered, for 10 customers 5, 6, 7 and 8 are considered as maxima, for 15 customers 7, 8, 9, 10 and 11 are considered, and for instances with 20 customers 10, 11, 12 and 13 are the maxima. The fixed major cost is set to $B = 1000$.

The RMP is initialized with dummy columns with very high costs to guarantee a feasible solution for the initial linear program. To improve computation times, two heuristics are designed to attempt quickly identifying columns that provide a feasible integer solution. The first heuristic assigns customers to periods in a greedy way, ensuring customers do not run out of stock and respecting the subset composition constraints. If the subset composition constraint is a maximum on the tour duration, a second heuristic is applied if the first one did not yield a feasible integer solution. In the second heuristic, before applying the steps of the first heuristic, the customers are sorted by decreasing distance from the depot. Since the sorting process indirectly takes the subset composition constraint into account, this provides a greater chance of finding a feasible solution. Limited computational experiments showed that the first heuristic provides a better upper bound if a solution is found, therefore this heuristic is applied first.

The algorithm to solve the DJRP-AT as described in the previous sections is implemented using Java and Gurobi 6.5. All tests are performed on a desktop computer running Windows 10, equipped with an eight core Intel(R) Core(TM) i7-6700K, CPU 4.00GHz processor with 24GB of RAM. A single core is used to generate the results and the maximum running time is two hours per instance.

3.6.1 Effectiveness of valid inequalities

Two valid inequalities are considered to strengthen the formulation for the DJRP-AT. These valid inequalities are tested for both types of subset composition constraints and the results are presented in Table 3.1. The results are aggregated per number of customers in the instance (N) and the length of the time horizon (T). The average solution time over all tested instances, the number of instances solved, and the average integrality gap for the solved instances are reported for the model without valid inequalities, the model with only the ILB inequalities (equation 3.12), the model with only NrVis inequalities (equation 3.13), and the model with both inequalities, respectively. Next to the number of instances solved, the total number of tested instances that have not been proved infeasible is indicated between brackets. The difference between the two numbers gives the number of instances that have not been solved in two hours. The integrality gap is the percentage difference between the optimal binary solution (LB_{best}) and the solution of the relaxation of the model in the root node of the Branch-and-Bound tree (LB_{root}), which is calculated by (LB_{best} - LB_{root})/LB_{best}. To test the effectiveness of the valid inequalities, for instances with 15 customers and 3 periods, duration 600 is not considered; for 15 customers and 6 periods, both 600 and 800 are not considered, since most instances would be infeasible. For instances with 15 customers, a maximum number of customers per period (k_M) of 8, 9, 10 and 11 are considered.

The observed integrality gaps, as reported in Table 3.1, are quite high and decrease if more valid inequalities are added. The high integrality gaps can be partially explained by the formulation of the Master Problem. The linear relaxation of the model allows for satisfying customer demand in a certain period using only a fractional value for the decision variable corresponding to a subset containing this customer. Therefore, the costs of the subset are only fractionally accounted for. Moreover, the linear relaxation allows infeasible combinations of customers to be served in one period in a fractional solution. Therefore, the costs of the fractional solution can be lower, even tough the solution is certainly not feasible. Both of these effects were observed in our experiments. For example, consider instance abs3n5 from Archetti et al. [2007] which contains 5 customers, 3 time periods and set $k_D = 600$ as the maximum on the duration. The optimal solution selects customer subset {3} in period 1, subset {1,2} in period 2 and subset {3,4,5} in period 3; the objective value is 4310. In the solution of the relaxed model, subsets {1,2} and {3,4,5} are selected with value 0.5 in both period 1 and period 2. The objective value of this fractional solution is 3397, resulting in an integrality gap of 21%. Note that in the fractional solution all customers can be replenished in both period 1 and 2, while a subset consisting of all customers {1,2,3,4,5} is not a feasible subset given the tour duration constraint. This demonstrates that the fractional model provides the opportunity to select suboptimal subsets of customers,

Table 3.1 Effectiveness of valid inequalities for DJRP-AT for two types of constraints.

		None			Only ILB			Only NrVis			ILB and NrVis		
N	T	Time (s)	# Solved	Gap (%)	Time (s)	# Solved	Gap (%)	Time (s)	# Solved	Gap (%)	Time (s)	# Solved	Gap (%)
Constraint on duration													
5	3	0	20 (20)	28	0	20 (20)	22	0	20 (20)	7	0	20 (20)	7
10	3	2	18 (18)	42	1	18 (18)	34	1	18 (18)	16	1	18 (18)	16
15	3	40	13 (13)	44	18	13 (13)	37	17	13 (13)	20	6	13 (13)	20
5	6	1	20 (20)	25	1	20 (20)	19	1	20 (20)	6	0	20 (20)	3
10	6	83	16 (16)	31	39	16 (16)	25	71	16 (16)	12	19	16 (16)	9
15	6	3801	6 (10)	31	3021	7 (10)	26	3316	7 (10)	13	3456	6 (10)	11
Sum			93 (97)			94 (97)			94 (97)			93 (97)	
Constraint on number of customers													
5	3	0	10 (10)	46	0	10 (10)	42	0	10 (10)	20	0	10 (10)	20
10	3	2	20 (20)	42	1	20 (20)	34	1	20 (20)	14	1	20 (20)	13
15	3	30	20 (20)	42	12	20 (20)	34	15	20 (20)	15	7	20 (20)	14
5	6	3	10 (10)	32	1	10 (10)	26	2	10 (10)	13	0	10 (10)	10
10	6	86	20 (20)	31	29	20 (20)	24	39	20 (20)	11	20	20 (20)	8
15	6	4269	12 (20)	30	3212	14 (20)	24	3264	15 (20)	11	2507	17 (20)	11
Sum			92 (100)			94 (100)			95 (100)			97 (100)	

causing high integrality gaps. Also note that the instances with the highest integrality gaps do not necessarily have the highest computation times for both types of subset composition constraints.

For the model with a tour duration constraint, the computational results show that valid inequalities decrease computation times and integrality gaps, but that adding both types of valid inequalities does not always improve computation times, compared with adding one type of inequality. The integrality gaps are best if both types of inequalities are used, but only adding the NrVis inequalities yields almost the same average integrality gap. For this subset composition constraint, the best performance in terms of computation times are found if only the ILB inequalities are added to the model.

For the model with the constraint on the number of customers replenished per period, the results show that both types of valid inequalities improve the efficiency of the model. The average computation time decreases strongly for all instance sizes. For the largest instances (15 customers, 6 time periods) the number of solved instances increases from 12 to 17 out of 20 by adding the valid inequalities and the average integrality gap is only a third of the average gap without inequalities. Furthermore, the integrality gap of the model including only the NrVis inequalities is almost identical to the model with both types of inequalities; however, computation times show that the model including both types of inequalities performs better.

Archetti et al. [2007] concluded that the NrVis inequalities (equation 3.13) are ineffective for solving their IRP model. It is therefore important to note that, for the DJRP-AT, these valid inequalities lower the integrality gaps substantially and reduce the computation time. Hence, for the DJRP-AT, the effectiveness of these inequalities has been demonstrated for both types of subset composition constraints. The results per instance with all valid inequalities in the model are available in Appendix D for both types of subset composition constraints.

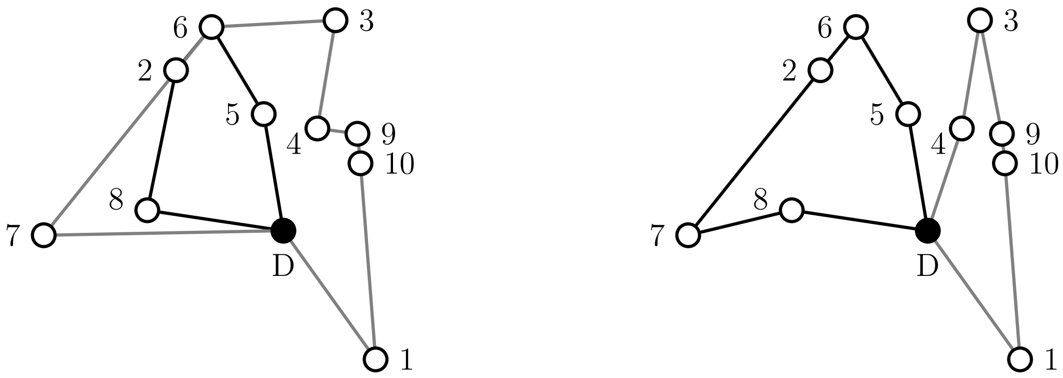
3.6.2 Comparison DJRP-AT and DJRP

The results of the DJRP-AT with a constraint on the number of customers per tour k_M can be compared with the existing DJRP with fixed fees [Boctor et al., 2004]. In this model a common cost is paid for serving at least one customer in a period ($B = 1000$) and an individual cost m_i is incurred for replenishing each customer. Note that the individual replenishment cost would in practice be given by the contract between the supplier and the LSP, and cannot be changed during the execution of the contract. To assess the impact of the individual replenishment costs, we test several values for $m_i,\ i \in \mathcal{N}$. For the experiments the individual replenishment cost m_i is either set to the same value for all customers (25 and 100) or set according to one of the following schemes. First, we set m_i to a value proportional to the distance to the depot, which we denote by 'prop'. Second, we define a zone around the depot in which at least one-third of the customers is located ('zones'). For the customers within the zone $m_i = 25$, and for the other customers $m_i = 100$. Third, we divide the total area in four quadrants. If a customer is in the same quadrant as the depot $m_i = 25$, and $m_i = 100$ otherwise ('quad').

To make a fair comparison between the DJRP and the DJRP-AT, the DJRP is extended with constraints on the customer's inventory capacity and a constraint on the number of customers served per period. Hence, the difference between this model and the DJRP-AT is the cost structure, i.e., fixed fees for individual replenishments versus transportation costs. Both models result in subsets of customers to be served in each period of the planning horizon and the corresponding delivery quantities. Importantly, it is not possible to directly compare the costs of both models. Therefore, the optimal traveling salesman tours for the resulting customer subsets are computed. The tour costs reflect the actual incurred routing costs. The total costs, for both models consisting of the inventory holding cost, the tour costs and the fixed costs per period, will be compared. First, the results of one instance are studied in more detail. Next, aggregated results over all tested instances are analyzed.

3.6.2.1 Illustrative result for one instance

In this section, one of the tested instances is studied in more detail and the results demonstrate the effect of the DJRP-AT compared with the DJRP. Consider an example containing 10 customers (instance abs3n10), 3 time periods, a maximum of $k_M = 7$ customers per period and an individual replenishment cost of $m_i = 25$. In Figure 3.2, the routes corresponding with the solution of the DJRP and the DJRP-AT are drawn. The depot is indicated by D and the customers are numbered according to the order of the customers in the instance. The black line represents the performed route in period two of the three periods and the grey line represents the route in period three; no customers are replenished in period one in both solutions. The DJRP solution shows that some customers are replenished multiple times in the planning horizon. Moreover, customers that are located in relatively close proximity, are not necessarily replenished in the same period, e.g., customers 7 and 8. The total costs of the DJRP solution are 4399 consisting of inventory holding cost of 123 and routing costs of 4276, including the fixed fee per period. The DJRP-AT solution is clearly more efficient from a routing point of view. The two routes cover distinct areas of the region in which the customers are located and all customers are replenished only once in the planning horizon. The holding cost, 159, is higher than in the solution of the DJRP. The routing costs, including the fixed fee per period, are 3914 which is lower than the DJRP routing costs. The total costs of DJRP-AT solution are 4073, which is 7.4% lower than the total costs of the DJRP solution.



(a) DJRP solution

(b) DJRP-AT solution

Figure 3.2 DJRP and DJRP-AT solutions for instance abs3n10 with $T = 3$, $k_M = 7$ and $m_i = 25$ (black route in period 2, gray route in period 3).

3.6.2.2 Aggregated results

Table 3.2 compares aggregated results of the DJRP-AT and the DJRP for the instances proposed by Archetti et al. [2007]. The first columns indicate the number of customers (N), the length of the time horizon (T), the maximum number of customers (k_M) and the number of instances solved out of the five instances (#). For each combination of N, T and k_M five different individual replenishment costs are considered for the DJRP ($m = 25, 100, \text{prop}, \text{zones}, \text{quad}$). The average percentage improvement in total costs, the maximum percentage improvement in total costs and the maximum percentage deterioration in any of the instances of the DJRP-AT, compared with the DJRP, are reported for each combination of parameter values. The percentage improvement in total cost of an instance is computed as (cost DJRP - cost DJRP-AT)/cost DJRP.

The results in Table 3.2 show that incorporating the approximated transportation costs in the DJRP reduces the total cost with 3.4% - 5.0% on average for different schemes for m and 4% overall. Individual savings up to 14.4% are achieved. The average improvement between instances with different numbers of customers or periods is similar.

For 494 out of 605 optimally solved instances the DJRP-AT outperforms the DJRP and matches its costs for 51 instances. For only 60 instances the DJRP resulted in a slightly better solution than the DJRP-AT with a maximum cost difference of 3.2%, but only 0.96% on average over all m schemes. The cases in which the DJRP resulted

Table 3.2 Average and maximum improvement and maximum deterioration of DJRP-AT compared with DJRP.

				Average cost improvement (%)					Maximum cost improvement (%)					Maximum cost deterioration (%)				
				m					m					m				
N	T	k_M	#*	25	100	prop	zones	quad	25	100	prop	zones	quad	25	100	prop	zones	quad
5	3	3	5	3.8	2.1	2.1	1.9	1.9	8.7	7.8	7.8	7.8	7.8	0.4	0.4	0.4	0.7	0.7
5	3	4	5	4.2	2.5	2.5	2.3	2.3	8.6	7.2	7.2	7.2	7.2	-	-	-	1.0	1.0
10	3	5	5	2.3	3.4	1.5	1.5	1.8	9.5	9.5	9.5	9.5	9.5	-	-	2.2	2.2	0.6
10	3	6	5	3.6	4.8	2.8	3.6	2.9	7.1	7.6	7.1	7.1	7.1	0.4	0.4	2.1	0.4	2.1
10	3	7	5	4.6	4.1	4.1	4.4	4.2	7.9	8.1	8.1	7.8	8.1	0.4	0.4	0.4	0.4	0.4
10	3	8	5	5.9	5.4	5.4	5.7	5.6	11.1	8.9	8.9	10.6	9.4	-	-	-	-	-
15	3	7	5	10.7	9.4	8.9	8.7	8.2	12.2	11.1	10.8	11.6	12.2	-	-	-	-	-
15	3	8	5	3.0	1.2	-0.5	0.0	1.1	6.7	3.4	0.4	2.7	5.3	0.5	-	3.0	2.6	2.6
15	3	9	5	3.6	0.7	0.6	1.7	2.5	5.6	4.6	4.1	4.7	5.7	-	1.5	1.5	0.2	0.2
15	3	10	5	6.1	3.4	3.3	4.3	5.1	11.0	9.6	9.1	9.6	11.1	-	1.0	1.0	0.1	0.1
15	3	11	5	8.2	5.6	5.5	6.2	7.0	14.4	14.3	13.9	14.4	13.9	-	2.7	2.7	3.2	3.2
20	3	10	5	6.8	4.9	1.9	4.3	2.8	8.7	8.7	8.7	8.7	8.7	-	-	0.8	-	0.4
20	3	11	4	2.2	2.4	2.4	2.7	2.7	6.6	7.4	7.4	8.2	8.2	0.5	-	-	-	-
20	3	12	5	2.6	1.5	1.5	2.2	2.2	8.5	5.2	5.2	8.1	8.1	0.9	-	-	-	-
20	3	13	5	4.4	3.4	3.4	3.6	3.6	10.1	8.8	8.8	8.8	8.8	0.9	-	-	-	-
5	6	3	5	5.8	2.6	4.4	4.5	4.5	12.3	5.9	6.6	6.6	6.6	-	-	-	-	-
5	6	4	5	3.2	1.8	1.2	1.1	1.4	6.2	4.2	4.1	4.2	4.2	-	0.2	1.5	1.5	1.5
10	6	5	5	5.3	4.8	4.8	4.8	5.3	10.4	7.9	7.9	8.2	10.4	-	-	-	-	-
10	6	6	5	5.7	5.7	5.9	5.7	5.8	9.5	8.4	8.1	8.9	8.0	-	-	-	-	-
10	6	5	5	3.0	2.5	2.2	2.1	2.9	6.9	5.3	5.1	4.2	5.8	0.6	-	-	-	-
10	6	8	5	4.3	2.6	3.2	3.0	4.5	7.6	4.9	5.4	5.0	7.7	-	-	-	-	-
15	6	8	3	8.6	5.4	6.2	7.7	7.7	11.3	6.1	7.4	11.3	11.3	-	-	-	-	-
15	6	9	4	8.1	7.1	6.6	7.1	7.9	11.7	9.5	9.3	9.3	11.6	-	-	-	-	-
15	6	10	5	3.7	2.4	2.4	3.2	3.4	6.1	6.1	6.3	5.2	6.2	-	-	-	-	-
15	6	11	5	5.1	3.8	3.1	4.3	4.2	7.6	5.3	4.8	5.9	5.5	-	-	-	-	-
Overall				5.0	3.7	3.4	3.9	4.1	14.4	14.3	13.9	14.4	13.9	0.9	2.7	3.0	3.2	3.2

*Number of instances solved out of 5 for each parameter combination.

in lower costs than the DJRP-AT can be explained by using the approximated transportation cost in the optimization, which does not always lead to the lowest routing and inventory holding cost. Moreover, the DJRP completely ignores customer location when determining replenishments and will mostly serve customers on the day their inventory is exhausted, provided all constraints are respected (except if one day's major cost can be saved). If not all customers can be served on the day their inventory is exhausted, the customers that have the lowest holding costs will be served a day earlier. In this case, the customer's holding costs exert substantial influence on the combination of customers served together in the DJRP solutions. This can, coincidentally, result in favorable combinations of customers regarding the actual routing cost, which can result in better DJRP solutions compared with the DJRP-AT. However, the results show that this scenario is unlikely, since this occurs in a limited number of the instances.

Table 3.2 suggests that the DJRP-AT performs better on problem instances with a longer planning horizon. The DJRP was only able to find a better solution than the DJRP-AT in five parameter settings of the instances with a six day planning horizon, and the savings were 0.2% - 1.5%. Therefore, we did additional experiments in which we increased the planning horizon from three to six days of the instances with a three day planning horizon. The results do not show clearly that the DJRP-AT performs relatively better on instances with a longer planning horizon.

The improvement in total costs of incorporating the transportation costs in the DJRP slightly decreases if the individual ordering cost m in the DJRP increases from 25 to 100. This can be explained by the fact that if the individual replenishment cost is lower, then the number of replenishments is higher in the DJRP outcomes, resulting in higher actual total routing costs. This effect can also be observed in the number of instances for which an improvement, deterioration, or equal costs are reported (Table 3.3). The number of instances showing an improvement decreases as the value of the individual replenishment cost m_i increases, compared with the DJRP. However, the percentage of instances for which deterioration must be reported does not increase as m_i increases; instead, the percentage of instances with equal costs for both models increases.

By using proportional costs, zones and quadrant based costs instead of the same individual replenishment costs for all customers, the routing costs are better reflected in the DJRP. Indeed, we can observe that on average the improvement of the DJRP-AT over the DJRP is lower than for the DJRP with $m_i = 25$. However, for $m_i = 100$ this is only the case for the proportional individual replenishment costs. The proportional costs best reflect the actual routing costs, in several works in the IRP literature the direct distance is used (as a starting point) to replace actual routing costs (see for example Cordeau et al. [2015] and Absi et al. [2015]). Still, from our results it shows that the DJRP-AT outperforms using proportional costs for replenishing a customer.

Table 3.3 Number and percentage of instances that report improvement, deterioration and equal costs.

	Number of instances						Percentage					
	m						m					
	25	100	prop	zones	quad	total	25	100	prop	zones	quad	total
Improvement	108	97	92	99	98	494	89%	80%	76%	82%	81%	82%
Detoriation	8	9	14	14	15	60	7%	7%	12%	12%	12%	10%
Equal	5	15	15	8	8	51	4%	12%	12%	7%	7%	8%
Total	121	121	121	121	121		100%	100%	100%	100%	100%	

In conclusion, these results show that in approximately 82% of all solved instances lower total costs can be achieved by using the DJRP-AT, instead of the DJRP. If the individual replenishment costs increase, the improvement that the DJRP-AT can achieve decreases, however, the number of instances with reported deteriorations does not increase. Also, for individual replenishment costs that better reflect the actual routing cost than the same cost for all customers, the DJRP-AT still outperforms the DJRP.

3.6.3 Comparison DJRP-AT and IRP

To gain further insight in the quality of the solutions of the DJRP-AT, we compare the results of the DJRP-AT to the results of a problem formulation that includes the actual routing problem. This formulation is a variant of the IRP with a constraint on the number of customers in a route, hence, this problem is different than the IRP often addressed in the literature [Coelho et al., 2014]. Therefore, to solve this IRP, the labeling algorithm that solves the Pricing Problem of the DJRP-AT is replaced by an Integer Linear Program (ILP) which solves a resource constrained elementary shortest path problem. The resource is the number of customers in the route and note that there are arcs with negative cost, hence, negative cost cycles need to be prevented by adding subtour elimination constraints. A solution of the ILP is a route with corresponding costs that consists of the arc costs and the fixed costs B. We use Gurobi to solve the ILP and apply the 'Solution Pool' option to generate multiple solutions in one iteration. Since the solution method is not especially designed for solving this IRP, only some of the very small instances, with low values for k_M, can be solved within four hours of running time. The differences in computation time are therefore only indicative.

Table 3.4 gives the aggregated results of the comparison between the DJRP-AT and the IRP, the results per instance can be found in Appendix D. As in Section 3.6.2, the optimal traveling salesman tours are computed to compare the costs. For each combination of parameters, the number of instances solved, the number of instances with equal results for the DJRP-AT and the IRP, the average difference between the DJRP-AT and the IRP, and the average computation times of both models are indicated. For 10 customers, 3 time periods and $k_M = 7$, none of the instances could be solved within four hours of running time. For 10 customers, 6 time periods and $k_M = 6$ only one instance was solved within four hours. In total 35 instances of the IRP are solved to optimality with computation times that are several orders of magnitude higher than those of the DJRP-AT. Out of the 35 instances, for 16 instances the DJRP-AT gives the same result as the IRP. On average, the costs of the solutions of the IRP are 0.77% lower than the costs of the DJRP-AT. Considering that solutions of the DJRP-AT are found by only using an approximation for the transportation costs, the results are quite close to the exact solutions.

Table 3.4 Average difference of DJRP-AT compared with IRP.

N	T	k_M	Number of instances solved	Number of instances equal result	Average difference (%)	Average time DJRP-AT (s)	Average time IRP (s)
5	3	3	5	3	-0.28	0.0	1.0
5	3	4	5	1	-1.30	0.2	1.6
10	3	5	5	4	-0.54	0.6	1155.8
10	3	6	4	1	-0.56	0.4	9348.8
5	6	3	5	4	-0.23	0.4	4.6
5	6	4	5	0	-1.47	0.4	13.4
10	6	5	5	3	-0.61	11.0	4944.0
10	6	6	1	0	-1.19	34.4	13 738.0

3.6.4 Case study ATM replenishment in Amsterdam

As described in the introduction, our research is motivated by a real-life case in ATM replenishment in the Netherlands, in which a supplier (vendor) decides on which ATMs to replenish per day and an LSP (CIT) designs the routes to perform the replenishments. Currently, the supplier only pays a 'minor' transportation cost to the LSP for each ATM replenishment and no 'major' cost. In this section we use company data to illustrate the benefit of alternative replenishment cost structures based on the DJRP and the proposed DJRP-AT.

Data on ATMs in Amsterdam and the depot location are provided by the supplier. The dataset contains per ATM, the address, storage capacity, dynamic daily demand, and initial inventory level. To use the existing solution framework, the ATM locations are mapped on the Euclidean plane and the demand data is expressed in thousands of Euros. Because the future cost structure parameters are not available and current numbers are not disclosed because of confidentiality reasons, the cost parameters are determined in consultation with the company to reflect the expected ratio between transportation and inventory holding costs. This includes minor (m_i) and major (B) transportation cost and inventory holding rates. A three-day planning horizon is considered appropriate and therefore we select the ATMs that need replenishment within the next three days, which results in 75 ATMs. Based on their geographical locations and postal codes, the set of ATMs is naturally split into four subsets of sizes 16, 19, and two of 20 customers. We let the holding rate vary from 0.1 to 0.3 to represent realistic cost ratios. Based on service times and travel times as observed by the company, we let the maximum number of customers served k_M range from 10 to 13. The same individual replenishment cost m_i is used for all customers, and tests are performed for values 25, 50 and 100. We only use the same values for m_i for all customers to stay close to the real-life case.

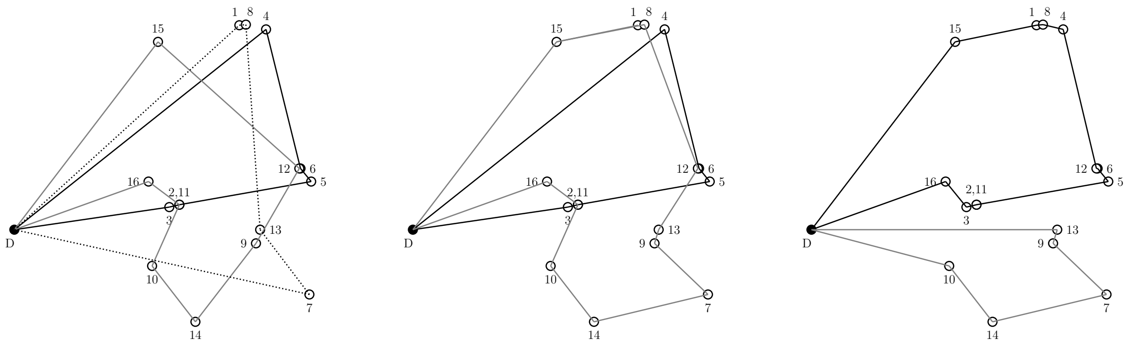
We estimate the costs of the current situation at the company by solving the DJRP with the major cost set to zero ($B = 0$, denoted by DJRP$_0$). To make a fair comparison with the DJRP with major cost (denoted by DJRP$_B$) and the DJRP-AT, we subsequently add the major cost $B = 1000$ for each period in which a replenishment takes place. Note that in practice, the current minor cost should be higher than the future minor cost to cover fixed costs. After computing the solutions of the three models, the optimal traveling salesman tours are computed to compare the costs.

Table 3.5 reports the percentage cost improvements of the DJRP$_B$ over the DJRP$_0$, and of the DJRP-AT over the DJRP$_B$. We only report the results for $m = 25$ and $m = 100$ since the results for $m = 50$ are similar to those for $m = 100$. Comparing DJRP$_0$ and DJRP$_B$ shows that a substantial cost improvement of 28.6% on average can be obtained, caused by having a route in every period in the DJRP$_0$ while in the DJRP$_B$ often only two routes are used. For the comparison between DJRP$_B$ and DJRP-AT, the results show that for 136 out of 144 cases, the DJRP-AT results in lower total costs than the DJRP$_B$, with a decrease in total costs up to 12.1%. For the remaining eight instances, the DJRP-AT results in slightly higher costs than the DJRP$_B$, with differences up to 1.7%. The DJRP-AT solutions are, on average, 6.4% better than the DJRP$_B$ solutions. Table 3.5 shows that the improvement of the DJRP-AT over the DJRP$_B$ decreases slightly for higher holding rates. For the DJRP-AT, it can be beneficial to serve customers earlier in the planning horizon than waiting until the customers run out of stock. However, if the holding rate is higher, serving customers earlier becomes more costly which results in higher costs for the DJRP-AT. Interestingly, varying the individual replenishment cost per ATM does not have a significant impact on the results. Furthermore, Table 3.5 shows that varying the maximum number of customers has a different impact per region. For Region 1 the results are similar for all values of k_M, while for Region 2, increasing the maximum number of customers to $k_M = 13$ leads to higher costs for the DJRP-AT, compared with the DJRP$_B$. For Region 2, the lower cost solution for $k_M = 13$ than for $k_M = 12$ of the DJRP$_B$ can be explained by the fact that many customers in the region are located in close proximity and a few customers are located further away from these clustered customers. In the DJRP-AT solutions, the customers not in the cluster are served on the same day, for all values of k_M. In the DJRP$_B$ solution, for $k_M = 12$, the customers outside the cluster are not all served on one day, which results in high transportation cost, but if $k_M = 13$, these customers are served on the same day, which lowers the total costs of the DJRP$_B$ substantially, as opposed to $k_M = 12$.

Table 3.5 Percentage improvement DJRP$_B$ over DJRP$_0$ (left), and percentage improvement DJRP-AT over DJRP$_B$ (right) in Amsterdam case.

	Region 1		Region 2		Region 3		Region 4		Region 1		Region 2		Region 3		Region 4	
$h = 0.1$	m		m		m		m		m		m		m		m	
k_M	25	100	25	100	25	100	25	100	25	100	25	100	25	100	25	100
10	30.1	30.1	30.2	30.2	31.1	31.1	28.2	28.2	11.4	11.4	4.0	4.0	1.1	1.1	8.3	8.3
11	30.1	30.1	30.3	30.3	29.3	29.3	27.8	27.8	11.4	11.4	4.4	4.4	4.3	4.3	8.1	8.1
12	30.1	30.1	30.1	30.1	29.2	29.2	24.6	24.6	11.4	11.4	3.1	3.1	4.3	4.3	12.1	12.1
13	30.1	30.1	32.4	32.4	29.3	29.3	25.0	25.0	11.4	11.4	-0.7	-0.7	4.6	4.6	11.5	11.5
$h = 0.2$	m		m		m		m		m		m		m		m	
k_M	25	100	25	100	25	100	25	100	25	100	25	100	25	100	25	100
10	29.6	29.6	29.1	29.4	30.3	30.3	27.7	27.7	10.8	10.8	3.6	3.6	0.9	0.9	7.7	7.7
11	29.6	29.6	29.2	29.4	28.6	28.6	27.3	27.3	10.8	10.8	2.7	2.7	3.9	3.9	7.3	7.3
12	29.6	29.6	29.0	29.3	28.5	28.5	24.1	24.1	10.8	10.8	2.6	2.6	4.0	4.0	11.3	11.3
13	29.6	29.6	31.4	31.6	28.6	28.6	24.6	24.6	10.8	10.8	-1.2	-1.2	4.0	4.0	10.6	10.6
$h = 0.3$	m		m		m		m		m		m		m		m	
k_M	25	100	25	100	25	100	25	100	25	100	25	100	25	100	25	100
10	30.1	29.1	26.9	28.6	29.5	29.6	26.9	27.2	9.8	10.3	3.8	3.3	0.7	0.7	7.2	7.2
11	29.8	29.1	27.3	28.7	29.5	27.9	26.5	26.8	10.3	10.3	2.4	2.4	1.4	3.6	6.8	6.8
12	30.0	29.1	27.9	28.5	27.8	27.9	23.4	23.7	10.0	10.3	1.0	2.0	3.7	3.6	11.0	11.0
13	30.0	29.1	27.8	30.8	27.8	28.0	23.8	24.2	10.0	10.3	0.4	-1.7	3.7	3.5	10.3	10.3

The results of DJRP$_0$, DJRP$_B$ and DJRP-AT are visualized for Region 1 in Figure 3.3, with $k_M = 11$, $m = 100$, and $h = 0.2$. The cost improvements for this instance are 29.6% and 10.8%, respectively. The results show that, as expected, the DJRP$_B$ saves cheaper solutions than the DJRP$_0$ and that the routes given by the DJRP-AT are much more efficient than the routes given by the DJRP$_B$. Overall, the results of the case study show that using a DJRP cost structure could provide significant savings, and the DJRP-AT results in similar improvements for the real-life case as for the artificial benchmark instances.



(a) DJRP_0 solution (b) DJRP_B solution (c) DJRP-AT solution

Figure 3.3 DJRP_0 , DJRP_B and DJRP-AT solutions for Region 1 ($k_M = 11,\ m = 100,\ h = 0.2$, black route in period 1, gray route in period 2, dotted route in period 3).

3.7 Conclusion

In practice it is regularly the case that a supplier outsources customer deliveries to a Logistics Service Provider (LSP); the supplier often pays a fixed transportation fee to the LSP for this service. Hence, when deciding on the timing of customer replenishments, the supplier often does not take efficiency of the delivery routes into account. To optimize costs, suppliers can use joint replenishment models, such as the Dynamic-Demand Joint Replenishment Problem (DJRP) [Boctor et al., 2004] in case of dynamic demand. The DJRP minimizes inventory holding and replenishment costs while ensuring that customers do exhaust their stock. The replenishment costs consist of fixed fees that are independent of the actual routing costs. As a result, we argue that the DJRP is incapable of proposing efficient solutions from a transportation point of view. Although the fee per delivery is fixed, the costs of the inefficient routes are indirectly paid by the supplier via the negotiated delivery fees in the following contract. Inspired by the practical relevance of the DJRP, this paper proposes the Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs (DJRP-AT) in which transportation costs are included by approximating the optimal tour length for given subsets of customers. Using the DJRP-AT will result in lower total costs and will increase resource utilization.

We propose a mathematical model for the DJRP-AT which is enriched with two different types of constraints. First, the tour duration is restricted to resemble the limitations encountered in practice. Second, the number of customers served per period is bounded, resulting in the opportunity to investigate the improvement of the proposed DJRP-AT compared with the existing DJRP. The Dantzig-Wolfe decomposition is applied to the proposed compact formulation and the resulting Master and Pricing Problems are solved in a Branch-and-Cut-and-Price framework. The Master Problem defines which subset of customers to replenish per period of the planning horizon and determines the delivery quantities of the customers served. The Pricing Problem generates, via a specially designed labeling algorithm, subsets of customers that can be served in one period and the transportation costs are then approximated for these customer subsets. To increase efficiency, labels need to be discarded during the labeling algorithm. However, existing sufficient dominance rules for shortest path problems are not adequate for discarding labels in the Pricing Problem of the DJRP-AT. Therefore, we introduce novel sufficient dominance conditions that make label discarding possible. Costs of the DJRP and DJRP-AT are compared by computing the optimal traveling salesman tours for the subsets of customers selected in the solutions of both models.

To assess the value of the DJRP-AT formulation and solution framework, existing problem instances from the Inventory Routing Problem (IRP) are adjusted for our experiments. The effectiveness of two types of valid inequalities that were proposed for the IRP [Archetti et al., 2007] is tested for both types of extra constraints. The results show that both inequalities are effective for the DJRP-AT, which differs from the results obtained from the IRP. Computational experiments show average improvements of total transportation and inventory holding costs of 3.4% to 5%, respectively. Depending on the individual fixed fee charged in the DJRP, maximum improvements around 14.4% are obtained. The DJRP outperformed the DJRP-AT for only a few instances, due to the approximation of the tour length. The DJRP-AT solutions are also compared to the equivalent IRP solutions for which a different Pricing Problem is implemented. The results show that for the solved instances, the costs of the IRP solutions are on average only 0.77% lower than the costs of the DJRP-AT solutions. The computation times for the IRP are orders of magnitude higher than for the DJRP-AT. Analysis of a real-life case in ATM replenishment shows that significant cost reductions can be achieved for both the LSP and the supplier when using the DJRP-AT.

Computational results with the DJRP-AT show that calculating transportation costs, instead of using fixed fees, in joint replenishment is worthwhile and that approximation of transportation costs works well. Future research could focus on developing novel formulations for this problem, possibly inspired by formulations discussed in Narayanan and Robinson [2006], that may improve the integrality gaps. Moreover, new valid inequalities can be proposed to strengthen the linear relaxation of DJRP-AT models.

4

The Inventory Routing Problem with Demand Moves

4.1 Introduction

The Inventory Routing Problem (IRP) combines the optimization of inventory management and routing of the replenishments for a set of customers. This problem is relevant in vendor-managed inventory settings, in which a supplier (the vendor) takes both the routing and replenishment decisions. The customers need to be served over a given time horizon and the vendor needs to decide when to replenish each customer, how much to deliver and how to combine the visits in feasible vehicle routes while minimizing total routing and inventory holding costs. Each customer faces a certain demand per period which must be satisfied from the customer's inventory, this demand is composed of demands of multiple end-users. If the customers are located relatively close to each other, one may have the opportunity to satisfy a part of the demand of a customer by another nearby customer by redirecting some of the end-users. This is for example the case when considering ATMs in urban areas where ATMs are often located in close proximity. This provides the opportunity to redirect ATM-users who want to withdraw money from one ATM to another in case of a cash shortage, hence, to move demand between ATMs. Our business partner considers this a realistic option to reduce their ATM replenishment costs. Moreover, in the future it might be possible to provide ATM-users with information upfront via a mobile application which can increase customer service by avoiding visits to out-of-service ATMs. Besides that, it is also an option to give the user a small discount if they withdraw cash from certain ATMs. Note that, for example in the Netherlands, a user can withdraw money at every ATM without transaction costs, even if an ATM is not owned by his own bank but by

This chapter is based on: A.C. Baller, S. Dabia, G. Desaulniers, and W.E.H. Dullaert, The Inventory Routing Problem with Demand Moves [Baller et al., 2019a] a competing bank. Hence, stimulating a user to withdraw at a certain ATM does not result in transaction costs for the user.

The possibility of redirecting end-users can be incorporated in the optimization of the replenishments to reduce total costs. We define the Inventory Routing Problem with Demand Moves (IRPDM) in which demand of a customer can (partially) be satisfied by another customer. We assume that a service cost is incurred by the vendor for each demand move. This cost can, for example, reflect a loss in service experienced by the end-user or the actual cost of the discount provided to the end-users. Hence, the IRPDM consists of deciding on the timing and quantity of deliveries to each customer, both to satisfy its own demand and potential demand moved from other customers to this customer, deciding on the vehicle routes to perform the replenishments and on demand moves between customers. The objective of the IRPDM is to minimize the total routing, inventory holding and service costs.

The contributions of this paper can be summarized as follows. We introduce the notion of demand moves and define the IRPDM. We solve the problem with a branch-price-and-cut solution method based on the approach by Desaulniers et al. [2016] for the IRP. Valid inequalities from the IRP literature are not directly applicable to the IRPDM. Non-trivial modifications of these inequalities are proposed to ensure that they capture the effect of the demand moves in the IRPDM. Experiments on IRP instances from the literature illustrate the performance of the proposed solution approach and offer insight in the benefits of allowing demand moves in inventory routing problems. Moreover, sensitivity analysis, for example on the service costs, is conducted to derive managerial insights.

The paper is organized as follows. Section 4.2 provides an overview of related literature. In Section 4.3 we formally describe the problem and we present a mathematical programming formulation for the IRPDM. Section 4.4 describes the solution method and contains an extensive description of the valid inequalities. The results of the computational experiments are detailed in Section 4.5. Finally, Section 4.6 discusses conclusions and directions for future research.

4.2 Literature

Inspired by a real-life case on ATM replenishment, this paper contributes to a recent stream of papers on cash supply chains. Van Anholt et al. [2016] propose a three-step heuristic to solve a combined inventory management and routing problem for so-called Recirculation ATMs (RATM). At an RATM, a user can both withdraw and deposit money, hence, the IRP solutions contain both delivery and pick-up activities. Money that is picked up at one ATM can be used for a replenishment of another ATM. Hence, transshipments performed by the private vehicle are included in the model. Instances are based on real-life data with up to 200 customers and one vehicle per depot for a planning horizon of 6 days. Larrain et al. [2017] considers an IRP that allows for stock-outs and the replenishment policy consists of swapping new cassettes of a chosen amount for the old cassettes that can still contain bank notes which are returned to the depot. The authors propose a matheuristic to solve the problem for instances with up to 60 locations, 3 vehicles and up to 18 periods (in at most 6 days). Geismar et al. [2017] provide a literature overview on currency supply chains by reviewing studies that look into the supply chain from the supply side (national banks), the demand side (commercial banks and ATM networks), and the private sector logistics providers' side. In their analysis on ATM replenishment related literature, Geismar et al. [2017] mention the study by Van Anholt et al. [2016] on RATMs and suggest as future research to investigate possible incentives to rebalance RATM inventories by steering users to a certain RATM (either withdraw from a full RATM or deposit at an empty RATM). As incentive they suggest a premium for making a deposit at a certain RATM and these premiums can be reviewed online by the user. In this paper we investigate the IRPDM to examine possible supply chain savings when implementing a similar system for ATMs.

The IRPDM is related to the IRP with Transshipment (IRPT) introduced by Coelho et al. [2012]. In the IRPT one can move goods from an origin customer or depot to a destination customer in order to redistribute merchandise between stores of the same chain to cover unexpected demand variations, redistribute inventory to reduce handling costs, and in case storage capacity is limited at certain locations. It is assumed that these transshipments are performed by an outsourced carrier. Coelho et al. [2012] propose an ILP formulation for this problem and develop an adaptive large neighborhood search heuristic for the single vehicle case. Instances from the literature with one vehicle, with up to 30 customers and 6 periods and with up to 50 customers and 3 periods are used to test the heuristic. The heuristic's stopping criterion is 25,000 iterations or one hour of computation time (which was reached for some of the largest instances). Coelho and Laporte [2013] use a branch-and-cut algorithm without problem specific valid inequalities for the IRPT, solving instances up to 30 customers with 6 periods and 50 customers with 3 periods with a maximum running time of 12 hours per instance. Lefever et al. [2018] propose an improved formulation for the IRPT which is solved by branch-and-cut and uses two problem specific valid inequalities. The computation times are lower than those achieved by Coelho and Laporte [2013] and Lefever et al. [2018] solves two more instances with 6 periods.

On the one hand, for certain features, the IRPDM can be seen as a special case of the IRPT. First, in the IRPT it is possible to transship goods and store the goods at the destination customer to be used during multiple periods. In the IRPDM, the goods are not transshipped, but the demand is moved. Therefore, a demand move in one direction, is equivalent to a transshipment of goods in the other direction if these transshipped goods are immediately consumed. Second, in the IRPDM demand moves to the depot are not possible while in the IRPT goods can be moved directly from the depot to a customer.

On the other hand, the IRPDM contains some features that generalize the IRPT. First, we handle the multiple vehicle case, while in Coelho et al. [2012] and Coelho and Laporte [2013] only the single vehicle case is considered. Second, we restrict for each customer the set of other customers to which demand can be moved. In the IRPDM, a large distance between customers would make a demand move impractical. Therefore, we restrict demand moves for each customer to a small subset of neighboring customers in close proximity and this subset can be different for each customer. In contrast, both in Coelho et al. [2012] and Coelho and Laporte [2013] the set of customers from which goods can be transshipped is limited to a subset of the customers and the depot, and this set is fixed for the instance. Hence, there is a set of 'source' locations from which goods can be transshipped to any other customer. This can be modeled as a special case in the framework of neighbors that we define for the IRPDM. Although the limitation to a general set of source locations is modeled in Coelho et al. [2012], this feature does not seem to be used in the computational experiments. Concluding, in the IRPDM any customer can be a 'source' location, not only a predetermined subset of the customers, and in the IRPDM the 'source' customers can be different for each customer.

For extensive surveys on variants and solution methods for the IRP we refer to Coelho et al. [2014] and Andersson et al. [2010]. Next to solution methods mentioned in these surveys, Desaulniers et al. [2016] presents a new formulation for the IRP that performs better for instances with multiple vehicles. In Desaulniers et al. [2016] a Branch-Price-and-Cut algorithm is proposed to solve the IRP. In the model, columns represent a combination of a route and a so-called Route Delivery Pattern (RDP) specifying the quantity delivered to each customer along the route. In the master problem, the optimal combinations of routes and RDPs are selected to minimize total routing and inventory holding costs. In the pricing problem, routes and associated RDPs are generated based on the dual variables retrieved from the master problem. To model demand moves in the IRP, we extend the IRP formulation as introduced by Desaulniers et al. [2016]. The main differences are the handling of initial inventory at the customers at the beginning of the planning horizon, the introduction of the neighboring customers and the non-trivial adjustments to the valid inequalities. Section 4.3 provides more details on the IRP formulation by Desaulniers et al. [2016] and the extension for the IRPDM.

4.3 Problem description and formulation

In the IRPDM, a supplier replenishes inventory for a set of customers over a certain planning horizon. The supplier has an initial inventory at the beginning of the planning horizon and a known quantity becomes available in each period. A given number of vehicles with a load capacity restriction is available to perform the replenishments. A customer can be serviced at most once per period, i.e., split deliveries are not allowed. Each customer has an initial inventory at the beginning of the planning horizon, faces a given demand per period, and has an inventory capacity that must be respected. Via demand moves, part of a customer's demand can be satisfied by another, nearby customer. Note that we consider moving a part of the demand of the customer (the ATM) which implies that in practice the demand of several end-users of the customer is moved. For each demand move a cost is incurred that depends on the quantity moved and the distance between the customers involved. The other costs consist of routing costs for the distance traveled by the vehicles and inventory holding costs at the supplier and the customers. The objective of the IRPDM is to minimize the routing, inventory holding and demand move costs. The inventory holding costs are charged on the quantity in stock at the end of each period assuming the following order of events in a period: increase inventory at the supplier, delivery of goods to the customers, consumption, inventory calculation. This order of events coincides with most literature on the IRP [Archetti et al., 2014a].

More formally, a single supplier, denoted by 0, needs to serve a set of customers N over a time horizon $P = \{1, 2, \dots, \rho\}$. A fictitious period $\rho + 1$ is considered to handle end inventories. At each discrete time moment $p \in P$ a quantity d_0^p becomes available at the supplier 0 and each customer $i \in N$ faces a demand d_i^p. A homogeneous fleet of K vehicles with capacity Q is available to deliver the goods to the customers. For each customer $i \in N$ a holding capacity C_i needs to be respected and backlogging is not allowed. A customer $i \in N$ (the supplier 0) has an initial inventory I_i^0 (I_0^0) and a unit holding cost h_i (h_0). A Maximum Level inventory policy is adopted for the customers which means the delivery quantity can be chosen freely as long as inventory capacity is respected. The distance between the depot and each customer, and between all customers is given and denoted by c_{ij}. Each customer can be served by at most one vehicle per period. Each customer can redirect (part of) its demand to another customer. Therefore, for each customer $i \in N$ a set of neighboring customers $\mathcal{N}_i$ is established; the demand of the customers in $\mathcal{N}_i$ can be satisfied by i via demand moves. So, if a demand move from i to j is possible, this means that $i \in \mathcal{N}_j$, i.e., j can satisfy demand of i. If a demand move takes place, a cost m_{ij} is charged per unit moved and per unit distance between i and j. The costs related to demand moves are in the objective function added to the routing and inventory holding costs, which is similar to incorporating costs of, for example, backlogging [Abdelmaguid et al., 2009] and lost sales [Larrain et al., 2017] in IRP settings.

In most IRP formulations the model involves variables indicating the quantity delivered in a period to a certain customer. Inventory balance constraints keep track of the use of the inventory to satisfy the demand in the different periods. In contrast, in the formulation for the IRP proposed by Desaulniers et al. [2016], the model uses variables indicating for which periods the delivered quantities are dedicated. The authors use the fact that there always exists an optimal solution that respects the first-in, first-out (FIFO) principle. Hence, it is possible, given the period of delivery, to determine the periods to which a delivered quantity is assigned. Moreover, if there is initial inventory, FIFO implies that this inventory is used to cover the demand in the first periods of the planning horizon. Therefore, so-called residual demands $\bar{d}_i^p$ can be calculated. In periods for which the initial inventory cannot cover the demand, customers have a positive residual demand. Constraints that make sure all demand is covered are only needed for periods with a positive residual demand. In the IRPDM, initial inventory can also be used to satisfy moved demand of another customer. Therefore, we cannot use residual demands as in Desaulniers et al. [2016], but we have to model the use of the initial demand explicitly. This also implies that these constraints are needed for all customers and all periods.

Given the FIFO rule, let $I_i^{0,s} = \max\{0, I_i^0 - \sum_{\ell=1}^{s} d_i^\ell\}$ be the quantity remaining from initial inventory at customer i at the end of period s if initial inventory is only used to satisfy demand of customer i itself. Let P_{ijp}^+ denote the subset of periods to which a replenishment can be dedicated which is delivered in period $p \in P$ to customer $i \in N$ dedicated to customer $j \in \mathcal{N}_i \cup \{i\}$. The deliveries for periods P_{ijp}^+ and customers $j \in \mathcal{N}_i \cup \{i\}$ are so-called subdeliveries. Note that a subdelivery can be zero, then no delivery is made for the corresponding period and customer. The set P_{ijp}^+ can be determined as follows:

$$P^{+}_{ijp} = \begin{cases} \left\{ s \in \{p, p+1, \ldots, \rho+1\} \middle| \sum_{l=p}^{s-1} d_i^l < C_i \right\} & \text{if } i = j \text{ and } \mathcal{N}_i \neq \emptyset \\ \Big\{ s \in \{p, p+1, \ldots, \rho+1\} \Big| & \\ \qquad \left(s \in P, \bar{d}_i^s > 0 \text{ and } (s = p \text{ or } \sum_{l=p}^{s-1} d_i^l + I_i^{0,s-1} < C_i) \right) & \\ \qquad \text{or } \left(s = \rho+1 \text{ and } \sum_{l=p}^{s-1} d_i^l + I_i^{0,s-1} < C_i \right) \Big\} & \text{if } i = j \text{ and } \mathcal{N}_i = \emptyset \\ \left\{ s \in \{p, p+1, \ldots, \rho\} \middle| \sum_{l=p}^{s} d_i^l < C_i \right\} & i \neq j. \end{cases}$$

The set P^{+}_{ijp} should be large enough such that possible solutions for the IRP are not excluded. If $i = j$ and $\mathcal{N}_i \neq \emptyset$, there are neighbors for which the initial inventory can satisfy the demand. Set P^{+}_{ijp} is then largest if all periods are included such that inventory capacity of the customer is not exceeded by the total delivery made. If $i = j$ and $\mathcal{N}_i = \emptyset$, all initial inventory is used to satisfy demand of customer i itself, and subdeliveries are only needed for periods with a positive residual demand, and such that inventory capacity is not exceeded. If $i \neq j$, because of the FIFO principle, demand of customer i needs to be satisfied from inventory before satisfying (part of) the demand of a neighbor j. This means that a subdelivery for a customer j in a period s is possible if the total demand of customer i up to and including period s does not exceed the inventory capacity. For ease of notation, denote $P^{+}_{ip} = P^{+}_{iip}$, and introduce $P^{+\ell}_{ip}$ denoting the latest period in set P^{+}_{ip}.

Let u^s_{ijp} denote the upper bound on the quantity of a subdelivery in period $s \in P^{+}_{ijp}$. For the visited customer i $u^s_{ip} := u^s_{iip}$ is computed as follows

$$u^s_{ip} = u^s_{iip} = \begin{cases} \min\left\{d_i^s, C_i - I_i^0\right\} & \text{if } s = p = 0 \\ \min\left\{d_i^s, C_i\right\} & \text{if } s = p \neq 0 \\ C_i - \sum_{\ell=p}^{s-1} d_i^\ell & \text{if } s = \rho+1 \\ \min\left\{d_i^s, C_i - \sum_{\ell=p}^{s-1} d_i^\ell\right\} & \text{otherwise} \end{cases}$$

and the upper bound u^s_{ijp} for a neighboring customer $j \in \mathcal{N}_i$ is given by

$$u^s_{ijp} = \begin{cases} \min\left\{d_j^s, C_i - I_i^0 - \bar{d}_i^s\right\} & \text{if } s = p = 0 \\ \min\left\{d_j^s, C_i - d_i^s\right\} & \text{if } s = p \neq 0 \\ 0 & \text{if } s = \rho+1 \\ \min\left\{d_j^s, C_i - \sum_{\ell=p}^{s} d_i^\ell\right\} & \text{otherwise.} \end{cases}$$

Note that delivering goods for the fictitious ending period will be in inventory at the same customer $i \in N$, no matter whether these goods are dedicated to the customer itself or one of its neighbors. Therefore, without changing the solutions, we set the upper bound for the quantity dedicated to each neighbor to 0 for this fictitious period. Also note that it is never optimal to have an incoming demand move and outgoing demand move in the same period. Therefore, first the customer's own demand needs to be satisfied before using goods in inventory to satisfy the demand of a neighbor. This influences the upper bound on the delivered quantity that is dedicated to a neighbor. The set of feasible routes is denoted by R, with for each route $r \in R$, N_r indicating the set of visited customers and A_r the set of arcs used in the route. Let a_{ri} be equal to 1 if customer $i \in N$ is visited in route $r \in R$ and 0 otherwise. A Route Delivery Pattern (RDP) w corresponding to period p details the quantities $q^s_{wij} \in [0, u^s_{ijp}]$ delivered to customer $i \in N_r$ dedicated to satisfy the demand of customer $j \in \mathcal{N}_i \cup \{i\}$ in period $s \in P^+_{ijp}$. Equivalent to Desaulniers et al. [2016], $q^s_{wij} = 0$ corresponds to a *zero subdelivery*, $q^s_{wij} = u^s_{ijp}$ to a *full subdelivery*, and a *partial subdelivery* if $0 < q^s_{wij} < u^s_{ijp}$. An extreme RDP contains at most one partial subdelivery. A set of extreme RDPs W^p_r is associated with each route $r \in R$ in period $p \in P$. Note that with a convex combination of multiple extreme RDPs any combination of delivered quantities can be constructed. The total quantity delivered in RDP $w \in W^p_r$ is the quantity that is loaded at the supplier, which is denoted by $q_w = \sum_{i \in N_r} \sum_{j \in \mathcal{N}_i \cup \{i\}} \sum_{s \in P^+_{ijp}} q^s_{wij}$.

Given a route $r \in R$ and an extreme RDP $w \in W^p_r$, we can identify the quantity $\hat{b}^s_{wi}$ delivered to customer $i \in N$ that will be in inventory at the end of period $p \leq s \leq P^{+\ell}_{ip}$. Compared with Desaulniers et al. [2016], we use $\hat{b}^s_{wi}$ instead of b^s_{wi} to indicate that in our case deliveries dedicated to neighboring customers are also included. Let $c_{rw} = \sum_{(i,j) \in A_r} c_{ij} + \sum_{i \in N_r} \sum_{s \in P^+_{ip}} h_i \hat{b}^s_{wi}$ be the costs associated with route r and RDP w in which the first term is the routing costs and the second term is the inventory holding costs of all units delivered to the visited customers. Note that a unit dedicated to satisfy the demand of a neighboring customer, stays in inventory at the customer until consumption. Denote by $P^-_{ijs} = \{p \in P | s \in P^+_{ijp}\}$ the set of periods in which a subdelivery can be made to customer $i \in N$ to satisfy the demand of customer $j \in \mathcal{N}_i \cup \{i\}$ in period $s \in P$. Use P^-_{is} to represent the union of sets P^-_{ijs} over all $j \in \mathcal{N}_i \cup \{i\}$.

To model the IRPDM we use the following decision variables. Continuous variables $y^p_{rw} \in [0, 1]$ are indicating the proportion of route $r \in R$ with RDP $w \in W^p_r$ in period $p \in P$. Nonnegative variables I^p_0 indicate the inventory level at the supplier at the end of period $p \in P$. Nonnegative, integer variables i^p_{ij} indicate the quantity out of initial inventory at customer $i \in N$ used to satisfy demand of customer $j \in \mathcal{N}_i \cup \{i\}$ in period $p \in P$.

To comply with the FIFO principle, we prevent a demand move from i to j if customer i still has inventory left. Therefore, we introduce binary decision variables v^s_i. This variable will be equal to 1 if there is a positive inventory level at customer $i \in N$ at the end of period $s \in P$ and can be 0 if there is no inventory left. Hence, if $v^s_i = 0$, a demand move from customer i to j ($i \in \mathcal{N}_j$) in period s is possible. If $v^s_i = 1$, customer i still has inventory left that should be used first and a demand move is definitely not possible.

We can now formulate the IRPDM as follows:

$$\min \sum_{p\in P}\sum_{r\in R}\sum_{w\in W_r^p} c_{rw}y_{rw}^p + \sum_{p\in P} h_0 I_0^p + \sum_{p\in P}\sum_{i\in N}\left(I_i^0 - \sum_{s\leq p}\sum_{j\in\mathcal{N}_i\cup\{i\}} i_{ij}^s\right) h_i$$

$$+ \sum_{p\in P}\sum_{r\in R}\sum_{w\in W_r^p}\sum_{i\in N_r}\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} m_{ij}q_{wij}^s y_{rw}^p + \sum_{p\in P}\sum_{i\in N}\sum_{j\in\mathcal{N}_i} m_{ij} i_{ij}^p \tag{4.1a}$$

$$\text{s.t. } I_0^{p-1} + d_0^p - \sum_{r\in R}\sum_{w\in W_r^p} q_w y_{rw}^p = I_0^p, \qquad \forall p\in P, \tag{4.1b}$$

$$\sum_{i:j\in\mathcal{N}_i\cup\{j\}}\sum_{p\in P_{ijs}^-}\sum_{r\in R}\sum_{w\in W_r^p} q_{wij}^s y_{rw}^p + \sum_{i:j\in\mathcal{N}_i\cup\{j\}} i_{ij}^s = d_j^s, \qquad \forall j\in N,\ s\in P, \tag{4.1c}$$

$$I_i^0 - \sum_{p\leq s}\sum_{j\in\mathcal{N}_i\cup\{i\}} i_{ij}^p + \sum_{p\in P_{is}^-}\sum_{r\in R}\sum_{w\in W_r^p} \hat{b}_{wi}^s y_{rw}^p$$

$$+ \sum_{j\in\mathcal{N}_i}\sum_{p\in P_{ijs}^-}\sum_{r\in R}\sum_{w\in W_r^p} q_{wij}^s y_{rw}^p + d_i^s \leq C_i, \qquad \forall i\in N,\ s\in P, \tag{4.1d}$$

$$\sum_{r\in R}\sum_{w\in W_r^p} a_{ri} y_{rw}^p \leq 1, \qquad \forall i\in N,\ p\in P, \tag{4.1e}$$

$$\sum_{r\in R}\sum_{w\in W_r^p} y_{rw}^p \leq K, \qquad \forall p\in P, \tag{4.1f}$$

$$\sum_{p\in P}\sum_{j\in\mathcal{N}_i\cup\{i\}} i_{ij}^p \leq I_i^0, \qquad \forall i\in N, \tag{4.1g}$$

$$I_i^0 - \sum_{p\leq s}\sum_{j\in\mathcal{N}_i\cup\{i\}} i_{ij}^p + \sum_{p\in P_{is}^-}\sum_{r\in R}\sum_{w\in W_r^p} \hat{b}_{wi}^s y_{rw}^p \leq \left(C_i - d_i^s\right) v_i^s, \quad \forall i\in N,\ s\in P, \tag{4.1h}$$

$$\sum_{i:j\in\mathcal{N}_i}\sum_{p\in P_{ijs}^-}\sum_{r\in R}\sum_{w\in W_r^p} q_{wij}^s y_{rw}^p + \sum_{i:j\in\mathcal{N}_i} i_{ij}^s \leq d_j^s\left(1 - v_j^s\right), \qquad \forall j\in N,\ s\in P, \tag{4.1i}$$

$$0 \leq I_0^p \leq C_0, \qquad \forall p\in P, \tag{4.1j}$$

$$i_{ij}^p \in \mathbb{N}, \qquad \forall i,j\in N,\ p\in P, \tag{4.1k}$$

$$y_{rw}^p \geq 0, \qquad \forall p\in P,\ r\in R,\ w\in W_r^p, \tag{4.1l}$$

$$\sum_{w\in W_r^p} y_{rw}^p \in \{0,1\}, \qquad \forall p\in P,\ r\in R, \tag{4.1m}$$

$$v_i^s \in \{0,1\}, \qquad \forall i\in N,\ s\in P. \tag{4.1n}$$

The objective function (4.1a) minimizes the total routing, inventory holding and demand move costs. Note that the demand move costs consist of the moves satisfied by initial inventory and by deliveries made during the planning period. Constraints (4.1b) balance the inventory level at the supplier between periods. Constraints (4.1c) make sure that, for each customer $j \in N$, all demand is satisfied by deliveries to customer j itself, to one of the customers i for which $j \in \mathcal{N}_i$, or from initial inventory. Constraints (4.1d) are the capacity constraints at the customers. Note that the inventory level at customer i in a period is highest after the deliveries and before consumption. The highest inventory level is thus equal to the remaining initial inventory, plus past deliveries (dedicated to i or to $j \in \mathcal{N}_i$) that are not consumed yet at the end of the period, plus the demand satisfied for other customers $j \in \mathcal{N}_i$ in this period and the demand at i in this period. Split deliveries are prevented by constraints (4.1e) and the number of used vehicles per period is limited by constraints (4.1f). Constraints (4.1g) make sure that the amount used from initial inventory does not exceed the actual initial inventory. In (4.1h), the left hand side is equal to the ending inventory at customer i in period s. If the ending inventory is positive, variable v_i^s must be equal to 1. Note that in this case constraints (4.1d) are more restrictive than (4.1h). If $v_i^s = 0$ then there cannot be any ending inventory. By constraints (4.1i), a demand move from j to i can only occur if $v_j^s = 0$ which implies that there cannot be any ending inventory in the same period.

Note that the maximum number of periods in which demand can be satisfied from initial inventory is limited, for example, if the demand is the same every period, the number of periods is $\lceil I_i^0 / d_i \rceil$. Hence, the variables i_{ij}^p only need to be introduced for those periods.

4.3.1 Limiting the moved demand

In the current formulation of the problem, it is possible that one customer is never replenished by a vehicle, but that all of its demand is satisfied from the customer's initial inventory and via demand moves. In practice, this might not be desirable. Therefore, we can limit the quantity that is satisfied by another customer via demand moves to a certain percentage θ of the demand. The left hand side of the constraint should be the quantity of the demand of customer $j \in N$ in period $s \in P$ satisfied via demand moves, either via a delivery to another customer i such that $j \in \mathcal{N}_i$ or from initial inventory. The right hand side should limit the quantity to θd_j^s. The left hand side of this constraint is identical to the left hand side of constraint (4.1i), hence, we can merge the two types of constraints as follows:

$$\sum_{i:j \in \mathcal{N}_i} \sum_{p \in P_{ijs}^-} \sum_{r \in R} \sum_{w \in W_r^p} q_{wij}^s y_{rw}^p + \sum_{i:j \in \mathcal{N}_i} i_{ij}^s \leq \theta d_j^s \left(1 - v_j^s\right) \quad \forall j \in N, \ \forall s \in P \tag{4.2}$$

4.3.2 Using initial inventory for demand moves

In Section 4.4 we will present a branch-price-and-cut algorithm to solve the IRPDM. Existing cuts for the IRP are no longer valid and need to be adjusted to handle demand moves. Because initial inventory can be used to satisfy moved demand, it is not straightforward how the cuts can be properly adjusted. Therefore, we first study a simplified variant of the problem, in which initial inventory can only be used to satisfy demand of the customer itself. In the remainder of the paper we consider this variant of the problem, unless indicated otherwise. In formulation (4.1a)-(4.1n) this variant can be modeled by setting $i_{ij}^p = 0$ for all $i \neq j$ or by considering a formulation with residual demands as in Desaulniers et al. [2016]. Moreover, we need to adjust the calculations of P_{ijp}^+ and u_{ijp}^s since goods only need to be delivered for periods with a positive residual demand. The set of periods to which a delivery in period p to customer i can be dedicated is given by

$$P^{+}_{ijp} = \begin{cases} \Big\{ s \in \{p, p+1, \ldots, \rho+1\} | \\ \qquad \Big(s \in P, \bar{d}^s_i > 0 \text{ and } (s = p \text{ or } \sum_{l=p}^{s-1} d^l_i + I^{0,s-1}_i < C_i) \Big) \\ \qquad \text{or } \Big(s = \rho + 1 \text{ and } \sum_{l=p}^{s-1} d^l_i + I^{0,s-1}_i < C_i \Big) \Big\} & \text{if } i = j \\ \Big\{ s \in \{p, p+1, \ldots, \rho\} | \bar{d}^s_j > 0 \text{ and } \sum_{l=p}^{s} d^l_i + I^{0,s}_i < C_i \Big\} & \text{otherwise} \end{cases}$$

An upper bound u^s_{ip} on the quantity dedicated to each period $s \in P^{+}_{ijp}$ is computed as follows for the visited customer

$$u^s_{ip} = u^s_{iip} = \begin{cases} \min \left\{ \bar{d}^s_i, C_i - I^{0,s-1}_i \right\} & \text{if } s = p \\ C_i - \sum_{\ell=p}^{s-1} d^\ell_i - I^{0,s-1}_i & \text{if } s = \rho + 1 \\ \min \left\{ \bar{d}^s_i, C_i - \sum_{\ell=p}^{s-1} d^\ell_i - I^{0,s-1}_i \right\} & \text{otherwise} \end{cases}$$

and the upper bound u^s_{ijp} for a neighboring customer $j \in \mathcal{N}_i$ is given by

$$u^s_{ijp} = \begin{cases} \min \left\{ \bar{d}^s_j, C_i - I^{0,s-1}_i - \bar{d}^s_i \right\} & \text{if } s = p = 1 \\ 0 & \text{if } s = \rho + 1 \\ \min \left\{ \bar{d}^s_j, C_i - \sum_{\ell=p}^{s-1} d^\ell_i - I^{0,s-1}_i - \bar{d}^s_i \right\} & \text{otherwise} \end{cases}$$

4.4 Solution method

A branch-price-and-cut method is used to solve model (4.1a)-(4.1n). Column generation is applied to the master problem consisting of the linear relaxation of (4.1a)-(4.1l) and (4.1n) to compute lower bounds within a Branch-and-Bound algorithm. Columns represent a route and an extreme delivery pattern, and these are generated by the pricing problem. This solution method can be applied to both the case in which the initial inventory can be used to satisfy moved demand and the case in which initial inventory cannot be used for this purpose. Valid inequalities are added dynamically to the master problem to tighten the linear relaxations. The valid inequalities are based on inequalities proposed in the literature for the IRP, and we adjust these for the case in which initial inventory cannot be used to satisfy moved demand. An integer feasible solution is found by branching on the appropriate variables. Below, we describe the column generation process, the pricing problem, the valid inequalities, and the branching procedure.

4.4.1 Column generation

Column generation is an iterative procedure that solves a linear program (LP). The procedure to solve the linear relaxation of (4.1a)-(4.1l) and (4.1n) starts with an LP with a limited set of variables y^p_{rw}, which is called the restricted master problem (RMP). Then, new variables are added which are found by solving one or more pricing problems and with these new variables the RMP is resolved. The pricing problems generate negative reduced cost variables y^p_{rw} (also called columns) with respect to the dual values of the current RMP. This process continues until the pricing problems do not generate new variables.

Initially, artificial columns with very high costs are added to guarantee a feasible solution for the RMP, such that dual values can be retrieved to be used in the pricing problem. To obtain a better initial solution, an additional set of columns is computed in the following greedy way. Consider, for each period p, the customers $\mathcal{S}$ that have residual demand in this period, and if there are none, consider the customers with residual demand in period $p+1$. For each customer, consider the delivery pattern with a full delivery in period p (or $p+1$) and zero deliveries for other periods. Create a route to visit the customers in $\mathcal{S}$ by applying the nearest neighbor heuristic starting at the depot and adding customers as long as vehicle capacity is not violated. Each customer that is added to the route is marked as visited. If no customers in $\mathcal{S}$ can be added anymore without violating vehicle capacity, the route is finished. If there are still unvisited customers in $\mathcal{S}$, create another route.

4.4.2 Pricing Problem

For the IRPDM there is a pricing problem for each period in the planning horizon. A column generated by the pricing problem for period $p \in P$ corresponds to a delivery route $r \in R$ and an extreme RDP $w \in W_r^p$ that are feasible with respect to the constraints of the problem. Hence, the pricing problem consists of a routing part and a delivery part which results in solving an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) combined with the linear relaxation of a knapsack problem. After providing more explanation on the pricing problem, the details on solving the ESPPRC will be discussed in Section 4.4.2.1. The pricing problem for the IRPDM is an extension of the one for the IRP in Desaulniers et al. [2016].

Associate dual variables $\pi_p^{4.1b}$, $\pi_{is}^{4.1c}$, $\pi_{is}^{4.1d}$, $\pi_{ip}^{4.1e}$, $\pi_p^{4.1f}$, $\pi_{is}^{4.1h}$ and $\pi_{js}^{4.1i}$ with constraints (4.1b)-(4.1f) and (4.1h)-(4.1i) respectively. The reduced cost of a variable y_{rw}^p is given by

$$\begin{aligned}\bar{c}_{rw}^p =& c_{rw} + \sum_{i\in N_r}\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} m_{ij}q_{wij}^s + q_w\pi_p^{4.1b} - \sum_{i\in N_r}\sum_{j\in\mathcal{N}_i\cup\{i\}}\sum_{s\in P_{ijp}^+} q_{wij}^s\pi_{js}^{4.1c} \\ &- \sum_{i\in N_r}\sum_{s=p}^{P_{ip}^{+\ell}} \hat{b}_{wi}^s\pi_{is}^{4.1d} - \sum_{i\in N_r}\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} q_{wij}^s\pi_{is}^{4.1d} - \sum_{i\in N_r}\pi_{ip}^{4.1e} - \pi_p^{4.1f} \\ &- \sum_{i\in N_r}\sum_{s=p}^{P_{ip}^{+\ell}} \hat{b}_{wi}^s\pi_{is}^{4.1h} - \sum_{i\in N_r}\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} q_{wij}^s\pi_{js}^{4.1i}\end{aligned} \tag{4.3}$$

in which $c_{rw} = \sum_{(i,j)\in A_r} c_{ij} + \sum_{i\in N_r}\sum_{s=p}^{P_{ip}^{+\ell}} h_i\hat{b}_{wi}^s$ which are the routing and inventory holding costs for a route r and RDP w.

For the routing part of the problem, define a graph $G^p = (V^p, A^p)$ in which V^p is the set of nodes, and A^p is a set of arcs with arc travel costs $c_{ij},\ i,j \in A^p$. The set of nodes includes nodes corresponding to the customers v^i, and to a depot source node v^S and sink node v^E. Set A^p contains all arcs between the customers $(i,j) \in N \times N,\ i \neq j$, all arcs from the source node $(v^S, i),\ i \in N$ and all arcs entering the sink node

$(i, v^E),\ i \in N$. In the ESPPRC, define the cost of an arc to be

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_p^{4.1f} & \text{if } i = v^S \\ c_{ij} - \pi_{ip}^{4.1e} & \text{otherwise} \end{cases} \quad \forall (i,j) \in A^p. \tag{4.4}$$

For the delivery part of the problem a linear relaxation of a knapsack problem needs to be solved with the extra feature that the delivery quantity for a customer consists of goods to satisfy the demand of the customer itself and of its neighbors. Therefore, introduce two sets of variables. First, associate with each customer $i \in N$ and period $s \in P_{is}^+$ the variable $\xi_i^s \in [0, u_{ip}^s]$ specifying the quantity delivered to customer i that is dedicated to satisfy the demand of customer i in period s if $s \in P$ or to the end inventory if $s = \rho+1$. Second, associate with each customer $i \in N$, each of its neighbors $j \in \mathcal{N}_i$ and each period $s \in P_{ijp}^+$ variable $\psi_{ij}^s \in [0, u_{ijp}^s]$ specifying the quantity delivered to customer i dedicated to satisfy the demand of customer j in period s. As indicated before, $\rho + 1 \notin P_{ijp}^+$ for $j \in \mathcal{N}_i$. Given a route $r \in R$ and its visited customers N_r, variables ξ_i^s, $s \in P_{is}^+$, and ψ_{ij}^s, $s \in P_{ijp}^+$, must be 0 for customers $i \in N \setminus N_r$. Moreover, it must hold that $\sum_{i \in N_r} \left(\sum_{s \in P_{ip}^+} \xi_i^s + \sum_{j \in \mathcal{N}_i} \sum_{s \in P_{ijp}^+} \psi_{ij}^s \right) \leq Q$ to respect vehicle capacity. Given the conditions above, $(\xi_i^s)_{i \in N, s \in P_{ip}^+}$ and $(\psi_{ij}^s)_{i \in N, j \in \mathcal{N}_i, s \in P_{ijp}^+}$ define an RDP w associated with route r. The reduced cost can be rewritten as follows

$$\begin{aligned} \bar{c}_{rw}^p = & \sum_{(i,j) \in A_r} \bar{c}_{ij} + \sum_{i \in N_r} \sum_{s \in P_{ip}^+} \xi_i^s \left(\pi_p^{4.1b} - \pi_{is}^{4.1c} + \sum_{p \leq t < s} \left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h} \right) \right) \\ & + \sum_{i \in N_r} \sum_{j \in \mathcal{N}_i} \sum_{s \in P_{ijp}^+} \psi_{ij}^s \left(m_{ij} + \pi_p^{4.1b} - \pi_{js}^{4.1c} - \pi_{is}^{4.1d} - \pi_{js}^{4.1i} + \sum_{p \leq t < s} \left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h} \right) \right) \end{aligned} \tag{4.5}$$

in which $A_r \subset A^p$ is the set of arcs visited in route $r \in R$. An extreme RDP has at most one partial subdelivery, hence, in an extreme RDP at most one variable ξ_i^s or ψ_{ij}^s can take a value in the open interval $]0, u_{ip}^s[$ and $]0, u_{ijp}^s[$, respectively.

4.4.2.1 Labeling algorithm

Labeling algorithms have been proposed in the literature to solve the pricing problems of a wide variety of routing problems [Irnich and Desaulniers, 2005]. To solve the pricing problem of the IRPDM, we propose a labeling algorithm in which a label represents both a partial route (path) and an associated extreme RDP. The labeling algorithm starts with a label at the source node v^S in the graph G^p, and the label is extended to subsequent nodes if such extensions are feasible. An extension to the sink node v^E results in a route with corresponding extreme RDP. During the algorithm, a dominance rule can be used to discard labels that will not result in the optimal solution of the pricing problem.

An extreme RDP consists of full subdeliveries, zero subdeliveries and at most one partial subdelivery. During the execution of the labeling algorithm, the quantity delivered in the partial subdelivery is unknown, because this quantity can depend on the other deliveries made. When a label is extended to the sink node v^E, the size of the partial subdelivery is determined. Following Desaulniers et al. [2016], we keep track of the possible contribution of the partial subdelivery to the reduced costs.

A label L_i corresponding to a partial route ending in node i with associated RDP w contains the following elements

- T_i^{cost} : Reduced cost of the route/RDP (r, w), excluding the dual contribution of the partial subdelivery if $i \neq v^E$.
- T_i^{loadF} : Total quantity delivered along (r, w), the quantity of full subdeliveries only if $i \neq v^E$.
- $T_i^{cust_k}$: Binary value indicating whether or not customer $k \in N$ has been visited in the route.
- T_i^{part} : Binary value indicating whether or not RDP w contains a partial subdelivery.
- T_i^{rateP} : Unit rate of contribution of the partial subdelivery to the reduced costs, if applicable.
- T_i^{maxP} : Maximum quantity that can be delivered in the partial subdelivery, if applicable.

Therefore, the label is denoted by $L_i = \left(T_i^{cost}, T_i^{loadF}, \left(T_i^{cust_k}\right)_{k \in N}, T_i^{part}, T_i^{rateP}, T_i^{maxP}\right)$. There are three subdelivery types: a full (F), partial (P) and zero (Z) subdelivery. An extreme RDP consists of the subdeliveries types for the visited customers and their neighbors for the periods in P_{ip}^+ and P_{ijp}^+, respectively, which we call a customer delivery pattern (CDP). For example, for a visit to customer i with one neighbor j, the CDP FF–P means that full subdeliveries are made for the two periods in P_{ip}^+ for customer i and that a partial subdelivery is made to satisfy a demand move from customer j in the single period in P_{ijp}^+. A CDP can contain at most one partial subdelivery, since an RDP can contain at most one, and the full subdeliveries cannot exceed vehicle capacity Q. For each customer $i \in N$ and period $p \in P$, we determine a list of feasible CDPs Γ_{ip} that we consider in the labeling algorithm at a node corresponding to customer i in period p. To make this list as short as possible, which will speed up the labeling algorithm, the list can be filtered to exclude CDPs that do not comply with the FIFO rule. For example, for customer i without neighbors a CDP FPF cannot be optimal, and hence, this CDP is excluded from the list. Note that the FIFO rule can only be applied to the part of the CDP that indicate the subdeliveries for the visited customer itself and cannot be applied to the part of the CDP indicating the deliveries for the neighboring customers. For example, a delivery pattern FFF-FPF is feasible with respect to the FIFO rule and can be in the optimal solution if the neighboring customer receives a delivery in the second period of P_{ijp}^+.

To express the resource extension functions, define binary parameters f_γ^s (respectively, $f_{\gamma j}^s$) which is equal to 1 if CDP $\gamma \in \Gamma_{ip}$ contains a full subdelivery for period s for the visited customer (respectively, neighbor j). Similarly, define g_γ^s (respectively, $g_{\gamma j}^s$) which is equal to 1 for a partial subdelivery in period s for the visited customer (respectively, neighbor j). Now, we can define for each CDP $\gamma \in \Gamma_{ip}$ the cost τ_γ^{cost}, the load of the full deliveries τ_γ^{loadF}, an indicator whether there is a partial subdelivery in the CDP τ_γ^{part}, the rate of the partial delivery τ_γ^{rateP} and the maximum size of the partial delivery τ_γ^{maxP}, which are defined as follows

$$\begin{aligned}
\tau_\gamma^{cost} &= \sum_{s\in P_{ip}^+} f_\gamma^s u_{ip}^s \left(\pi_p^{4.1b} - \pi_{is}^{4.1c} + \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\right) + \\
&\quad \sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} f_{\gamma j}^s u_{ijp}^s \Bigg(m_{ij} + \pi_p^{4.1b} - \pi_{js}^{4.1c} - \pi_{is}^{4.1d} - \pi_{js}^{4.1i} + \\
&\quad \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\Bigg) \\
\tau_\gamma^{loadF} &= \sum_{s\in P_{ip}^+} f_\gamma^s u_{ip}^s + \sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} f_{\gamma j}^s u_{ijp}^s \\
\tau_\gamma^{part} &= \sum_{s\in P_{ip}^+} g_\gamma^s + \sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} g_{\gamma j}^s \\
\tau_\gamma^{rateP} &= \sum_{s\in P_{ip}^+} g_\gamma^s \left(\pi_p^{4.1b} - \pi_{is}^{4.1c} + \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\right) + \\
&\quad \sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} g_{\gamma j}^s \left(m_{ij} + \pi_p^{4.1b} - \pi_{js}^{4.1c} - \pi_{is}^{4.1d} - \pi_{js}^{4.1i} + \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\right) \\
\tau_\gamma^{maxP} &= \sum_{s\in P_{ip}^+} g_\gamma^s (u_{ip}^s - 1) + \sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} g_{\gamma j}^s (u_{ijp}^s - 1)
\end{aligned}$$

Any CDP with a partial subdelivery for which $\tau_\gamma^{rateP} \ge 0$ can be discarded, since replacing the partial subdelivery with a zero subdelivery provides a solution with at most the same reduced cost.

The resource extension functions are defined as follows. Assume we have a label $L_i = \left(T_i^{cost}, T_i^{loadF}, \left(T_i^{cust_k}\right)_{k\in N}, T_i^{part}, T_i^{rateP}, T_i^{maxP}\right)$ corresponding to a node $i \neq v^E$ and the label is extended along an arc $(i,j) \in A^p$ ($j \neq v^E$), for every CDP in Γ_{jp}. Let $\gamma \in \Gamma_{jp}$ be one of those CDPs. The extended label is given by $L_j = \left(T_j^{cost}, T_j^{loadF}, \left(T_j^{cust_k}\right)_{k\in N}, T_j^{part}, T_j^{rateP}, T_j^{maxP}\right)$ with

$$T_j^{cost} = T_i^{cost} + \bar{c}_{ij} + \tau_\gamma^{cost} \tag{4.6}$$

$$T_j^{loadF} = T_i^{loadF} + \tau_j^{loadF} \tag{4.7}$$

$$T_j^{cust_k} = \begin{cases} T_i^{cust_k} + 1 & \text{if } j = k \\ T_i^{cust_k} & \text{otherwise,} \end{cases} \quad \forall k \in N \tag{4.8}$$

$$T_j^{part} = T_i^{part} + \tau_\gamma^{part} \tag{4.9}$$

$$T_j^{rateP} = T_i^{rateP} + \tau_\gamma^{rateP} \tag{4.10}$$

$$T_j^{maxP} = \begin{cases} \min\{\tau_\gamma^{maxP},\ Q - T_i^{loadF} - \tau_\gamma^{loadF}\} & \text{if } \tau_\gamma^{part} = 1 \\ \min\{T_i^{maxP},\ Q - T_i^{loadF} - \tau_\gamma^{loadF}\} & \text{otherwise.} \end{cases} \tag{4.11}$$

The resulting label is feasible if $T_j^{loadF} \le Q$, $T_j^{cust_k} \le 1$ for all $k \in N$, and $T_j^{part} \le 1$.

When extending to the sink node $j = v^E$ the cost computation differs to account for the partial subdelivery:

$$T_j^{cost} = \begin{cases} T_i^{cost} + \bar{c}_{ij} + T_i^{maxP} T_i^{rateP} & \text{if } T_i^{rateP} < 0 \\ T_i^{cost} + \bar{c}_{ij} & \text{otherwise.} \end{cases} \tag{4.12}$$

The number of labels can become very large, therefore, a dominance rule is used to reduce the number of labels. The dominance rule introduced for the IRP by Desaulniers et al. [2016] still holds for the IRPDM:

Definition 4.1. *A label* $L_1 = \left(T_1^{cost}, T_1^{loadF}, \left(T_1^{cust_k}\right)_{k \in N}, T_1^{part}, T_1^{rateP}, T_1^{maxP}\right)$ *is said to dominate a label* $L_2 = \left(T_2^{cost}, T_2^{loadF}, \left(T_2^{cust_k}\right)_{k \in N}, T_2^{part}, T_2^{rateP}, T_2^{maxP}\right)$ *if both labels* L_1 *and* L_2 *are associated with the same vertex and the following conditions are satisfied:*

(a) $T_1^{loadF} \leq T_2^{loadF}$;
(b) $T_1^{cust_k} \leq T_2^{cust_k}$;
(c) $T_1^{part} \leq T_2^{part}$;
(d) $T_1^{cost} - T_1^{maxP} T_1^{rateP} \leq T_2^{cost} - T_2^{maxP} T_2^{rateP}$;
(e) $T_1^{cost} - \left(T_2^{loadF} - T_1^{loadF}\right) T_1^{rateP} \leq T_2^{cost}$;
(f) $T_1^{cost} - \left(T_2^{loadF} + T_2^{maxP} - T_1^{loadF}\right) T_1^{rateP} \leq T_2^{cost} - T_2^{maxP} T_2^{rateP}$.

4.4.2.2 Heuristic labeling algorithms

Before applying the exact labeling algorithm described above, two heuristic labeling algorithms are applied. First, for each route/RDP combination in the current RMP solution, optimize the CDPs for the given route with respect to the current dual variables. To optimize the CDPs, the labeling algorithm is solved with only the arcs in the given route. Second, the labeling algorithm is performed on a graph that contains only a subset $\hat{A}^p$ of the arcs A^p for each period $p \in P$. The arcs are selected by the procedure proposed in Desaulniers et al. [2008]. Arcs that do not belong to the κ least reduced cost out of an origin node, or into a destination node, are removed. To compute the reduced cost of an arc A, the average cost over all possible CDPs for the destination node is computed (or similarly, the average cost over the CDPs of the origin node of an arc) and added to the reduced cost. In this calculation we assume that no quantity is delivered in the partial deliveries. Then, for each node the κ arcs with the lowest reduced cost, both incoming and outgoing, are kept in the graph. We set a dynamic value for κ, which starts at 1 and is incremented by 2 if no columns are generated in every subproblem for $\kappa = 1$.

4.4.2.3 Acceleration techniques

Next to the heuristic labeling procedures, the following acceleration techniques are implemented to speed up the column generation procedure.

First, the list of CDPs Γ_{ip} associated with a customer $i \in N$ in period $p \in P$ can be established once before the solution procedure starts. The costs and values associated with each CDP $\gamma \in \Gamma_{ip}$ need to be updated at each iteration with the corresponding dual variables. Before the (heuristic) labeling algorithm solves the pricing problem, we filter the list of CDPs by applying the dominance conditions as in Definition 4.1, except for condition (b), in which all T values are replaced by the current τ values of the CDPs.

Second, ng-path relaxation is applied as defined in Baldacci et al. [2011]. This relaxation of the pricing problem allows for cycles in the paths. To apply ng-paths, define for each node $v \in V^p$ in network $G^p = (V^p, A^p)$ a subset of customers NG_v. Let NG_v contain v itself and a subset of vertices that are closest to v such that $|NG_v| = b$. Here, b is a predefined parameter (which is set to 7 in our experiments). An ng-path can contain a sequence of visits $v - v_1 - v_2 - \ldots - v$ only if at least one node $w \notin NG_v$ is visited in between two visits of v. The labeling algorithm is adjusted to accommodate ng-paths as explained in Desaulniers et al. [2014].

Finally, since constraints (4.1e), which limit the number of visits to a customer in one period to one, are numerous and often not binding in the optimal solution, these constraints are relaxed first and added only if violated in a branch-and-cut form. Moreover, Desaulniers et al. [2016] showed that for the IRP some holding capacity constraints (equivalent to 4.1d) are redundant with the constraints equivalent to 4.1c and 4.1e. However, for the IRPDM it is not possible to establish a comparable statement. Hence, all capacity constraints (4.1d) are now present in the master problem. Yet, it is likely that for each customer this constraint is only binding in one or two periods. Therefore, we add the holding capacity constraints (4.1d) also in a dynamic way similar to constraints 4.1e.

4.4.3 Valid Inequalities

Next to the heuristic labeling described in Section 4.4.2.2 and the acceleration techniques described in Section 4.4.2.3, valid inequalities are implemented to strengthen linear relaxations of the problem and hence, to speed up the solution method. Only one family of valid inequalities that was proposed for the IRP can immediately be applied to the IRPDM. For the variant of the IRPDM in which initial inventory cannot be used to satisfy moved demand, existing valid inequalities can be adjusted, although the adjustments are not trivial. For the variant of the IRPDM in which initial inventory can be used to satisfy moved demand, it is not clear whether or how some of the existing IRP valid inequalities can be adjusted. Therefore, we restrict ourselves to developing valid inequalities for the variant in which initial inventory cannot be used to satisfy moved demand.

First, in Section 4.4.3.1 we describe a family of valid inequalities proposed in Desaulniers et al. [2016] for the IRP and we argue why this family of inequalities is also valid for the IRPDM. Second, we propose a generalization of the first family of inequalities in Section 4.4.3.2. Third, Sections 4.4.3.3 to 4.4.3.6 propose valid inequalities for the IRPDM that are derived from valid inequalities for the IRP. Finally, in Section 4.4.3.7 we elaborate why the valid inequalities in Sections 4.4.3.3 to 4.4.3.6 need structural changes for the variant of the IRPDM in which initial inventory can be used to satisfy moved demand.

4.4.3.1 Valid inequalities on the minimum number of routes per time interval

In the IRP, given the total quantity that must be delivered and the vehicle capacity, one can compute the minimum number of vehicle routes needed to deliver all goods. So, if one adds up the residual demand of all customers up to period $\rho \in P$, a lower bound can be established on the number of routes to fulfill the total residual demand. This also holds for the number of routes needed up to a certain period $\ell \in P$. We denote these inequalities as Route Inequalities (RI). In the IRPDM, both in case initial inventory can and cannot be used to satisfy moved demand, the total quantity that needs to be delivered by vehicles remains the same as in the IRP. Therefore, these inequalities can be applied to the IRP and both variants of the IRPDM without adjustments. A lower bound on the number of routes is given by $lb_\ell^R = \left\lceil \sum_{i \in N} \sum_{s=1}^{\ell} \bar{d}_i^s / Q \right\rceil$ and the following valid inequalities hold

$$\sum_{p=1}^{\ell} \sum_{r \in R} \sum_{w \in W_r^p} y_{rw}^p \geq lb_\ell^R, \quad \forall \ell \in P \tag{4.13}$$

Let $\pi_\ell^{4.13}$, $\ell \in P$ be the dual variables associated with valid inequalities (4.13). The reduced cost is adjusted the same way as in Desaulniers et al. [2016]:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_p^{4.1f} - \sum_{\ell=p}^{\rho} \pi_\ell^{4.13} & \text{if } i = v^S \\ c_{ij} - \pi_{ip}^{4.1e} & \text{otherwise,} \end{cases} \quad \forall (i,j) \in A^p. \tag{4.14}$$

4.4.3.2 Generalized valid inequalities on the minimum number of routes per time interval

The cuts in Section 4.4.3.1 can be generalized to time intervals $[\ell, \ell']$ where $\ell, \ell' \in P$ are such that $\ell' > \ell$. For example, suppose there is one customer and it has the following residual demands: 25, 40, 40, 40, 40, 40 for the six periods in the planning horizon, the vehicle capacity is $Q = 100$, and inventory capacity at the customer is $C = 80$. Then, at the end of period 4 there can be at most an inventory of 40, hence, at least one vehicle must visit this customer in the interval $[5, 6]$. All residual demands for this customer can be covered with 3 vehicles, but if, in a fractional solution, this customer receives a visit of one vehicle in periods 1, 3 and 4, and of 0.4 vehicle in period 5, the inequality for interval $[5, 6]$ will be violated. We denote these inequalities as Generalized Route Inequalities (GRI). If $\ell = 1$, the inequalities are the same as in Section 4.4.3.1.

Define a new lower bound $lb_{ll'}^{\bar{R}} = \left\lceil \frac{\sum_{i \in N} \left(\sum_{p=\ell}^{\ell'} d_i^p - \left(C_i - d_i^{\ell-1} \right) \right)}{Q} \right\rceil$. The numerator now accounts for the maximum possible inventory level at the end of period $\ell - 1$ at each customer. Note that the fraction can be rounded up because all terms are known at the beginning of the planning horizon. We propose the following generalized valid inequalities:

$$\sum_{p=\ell}^{\ell'} \sum_{r \in R} \sum_{w \in W_r^p} y_{rw}^p \geq lb_{ll'}^{\bar{R}}, \quad \forall \ell, \ell' \in P \tag{4.15}$$

Associating dual variables $\pi_{\ell\ell'}^{4.15}$, $\ell, \ell' \in P$ with these inequalities, then the modified arc reduced costs $\bar{c}_{ij}$ become:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_p^{4.1f} - \sum_{\ell=1}^{p} \sum_{\ell'=p}^{\rho} \pi_{\ell\ell'}^{4.15} & \text{if } i = v^S \\ c_{ij} - \pi_{ip}^{4.1e} & \text{otherwise,} \end{cases} \quad \forall (i,j) \in A^p. \tag{4.16}$$

4.4.3.3 Valid Inequalities on the minimum number of visits per customer

For the IRP, given the residual demand at a customer over periods 1 to $\ell \in P$, the inventory capacity at the customer and the vehicle capacity, one can compute how many visits are at least needed to satisfy a customer's demand [Archetti et al., 2007, Coelho and Laporte, 2014]. In the IRPDM, demand at a customer i cannot only be satisfied via deliveries by a vehicle, but also via other customers $j : i \in \mathcal{N}_j$. Hence, a delivery to such a customer j should also be counted as a 'visit' to customer i. Note that the inventory capacity at customer j can also decrease the number of visits needed for customer i, if residual demand is satisfied via a customer j. Therefore, define $C_i^{\max} = \max_{j:i\in\mathcal{N}_j\cup\{i\}}\{C_j\}$. Then the minimum number of visits needed to a customer between periods 1 and ℓ is given by $lb^V_{i\ell} = \left\lceil \frac{\sum_{p=1}^{\ell} \bar{d}_i^p}{\min\{Q, C_i^{\max}\}} \right\rceil$. The following valid inequalities hold

$$\sum_{p=1}^{\ell} \sum_{r\in R} \sum_{w\in W_r^p} \left(a_{ri} + \sum_{j:i\in\mathcal{N}_j} a_{rj} \right) y^p_{rw} \geq lb^V_{i\ell} \quad \forall i \in N, \ \forall \ell \in P. \tag{4.17}$$

Associate dual variables $\pi^{4.17}_{i\ell}$, $i \in N$, $\ell \in P$ with the valid inequalities. In the pricing problem for period p, the arc reduced costs are adjusted as follows

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_p^{4.1f} & \text{if } i = v^S \\ c_{ij} - \pi_{ip}^{4.1e} - \sum_{\ell=p}^{\rho} \pi^{4.17}_{i\ell} - \sum_{j\in\mathcal{N}_i} \sum_{\ell=p}^{\rho} \pi^{4.17}_{j\ell} & \text{otherwise} \end{cases} \quad \forall (i,j) \in A^p. \tag{4.18}$$

4.4.3.4 Valid Inequalities on the minimum number of subdeliveries per demand

Inequalities on the minimum number of subdeliveries per demand (MNSD) for the IRP were proposed by Desaulniers et al. [2016] based on the idea of Desaulniers [2010] on similar inequalities for the Split Delivery Vehicle Routing Problem. The idea is that the residual demand $\bar{d}_i^s$ of customer $i \in N$ in period $s \in P$ can be fulfilled via one subdelivery of size $\bar{d}_i^s$, or via at least two smaller subdeliveries in different periods. We extend the inequalities to incorporate demand moves.

A given residual demand $\bar{d}_i^s > 0$ can, in the IRPDM, be fulfilled in four different ways: (i) either by performing one subdelivery to customer i in a period $p \in P^-_{is}$, (ii) at least two subdeliveries to customer i in different periods $p \in P^-_{is}$, (iii) one subdelivery to a customer $j : i \in \mathcal{N}_j$ in a period $p \in P^-_{jis}$ or (iv) at least two subdeliveries to a customer j in different periods $p \in P^-_{jis}$. Define a^S_{ijw} (respectively, a^M_{ijw}) as a binary parameter equal to 1 if $a_{ri} = 1$ and $\bar{d}_j^s$ (respectively, less than $\bar{d}_j^s$) units are delivered in the subdelivery to customer i dedicated to customer $j \in \mathcal{N}_i \cup \{i\}$ and period $s \in P^+_{ijp}$ in RDP w, 0 otherwise. Define $a^S_{wi} = a^S_{wij}$ if $i = j$, and similarly for a^M_{wi}. The MNSD inequalities can be stated as follows:

$$\sum_{j:i\in\mathcal{N}_j\cup\{i\}} \sum_{p\in P^-_{jis}} \sum_{r\in R} \sum_{w\in W_r^p} \left(2a^S_{wji} + a^M_{wji}\right) y^p_{rw} \geq 2, \quad \forall i \in N, \ \forall s \in P : \ \bar{d}_i^s > 0. \tag{4.19}$$

Define $\pi^{4.19}_{is}$, $i \in N$, $s \in P$ as the dual variables of valid inequalities (4.19). To take the dual variables into account in the pricing problem for period $p \in P$, modify parameters τ_γ^{cost} as follows:

$$\tau_\gamma^{cost} = \sum_{s\in P_{ip}^+}\left[f_\gamma^s u_{ip}^s\left(\pi_p^{4.1b} - \pi_{is}^{4.1c} + \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\right) - (1+f_\gamma^s)\pi_{is}^{4.19}\right] +$$
$$\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+} f_{\gamma j}^s u_{ijp}^s\left(m_{ij} + \pi_p^{4.1b} - \pi_{js}^{4.1c} - \pi_{is}^{4.1d} - \pi_{js}^{4.1i} + \sum_{p\le t<s}\left(h_i - \pi_{it}^{4.1d} - \pi_{it}^{4.1h}\right)\right) -$$
$$\sum_{j\in\mathcal{N}_i}\sum_{s\in P_{ijp}^+}(1+f_{\gamma j}^s)\pi_{js}^{4.19} \tag{4.20}$$

4.4.3.5 Multiperiod Capacitated Subtour Inequalities

Avella et al. [2018] formulate Multiperiod Capacitated Subtour Inequalities (MCS) for the IRP. The MCS inequalities exploit that over a given set of subsequent periods p_1 to p_2, one can determine the minimum vehicle flow needed to satisfy the demand of a subset of customers $S \subseteq N$. Before deriving the MCS inequalities for the IRPDM, we will rewrite the inequalities of Avella et al. [2018] for the IRP in the terminology of our paper.

Let $(E : F)$ denote the set of arcs $(i, j) \in A$ for which $i \in E$ and $j \in F$ with A the complete set of arcs. Suppose there is a subset of customers $S \subseteq N$ and a time interval $[p_1, p_2]$ in which $p_1 \le p_2,\ p_1, p_2 \in P$. Define a_{rij} to be a binary parameter indicating whether arc (i, j) is traversed in route $r \in R$. The following inequalities hold for the IRP:

$$\sum_{t=p_1}^{p_2}\sum_{r\in R}\sum_{w\in W_r^t}\sum_{\substack{i,j:\\ i\in N\cup\{v^S\}\setminus S\\ \text{and } j\in S}} a_{rij}y_{rw}^t \ge \left\lceil \frac{\sum_{i\in S}\left(\sum_{t=p_1}^{p_2} d_i^t - \left(C_i - d_i^{p_1-1}\right)\right)}{Q}\right\rceil,$$
$$\forall S\subseteq N,\ \forall p_1, p_2 \in P. \tag{4.21}$$

The left hand side computes the vehicle flow into $S \subseteq N$ during the periods p_1 to p_2. In the nominator of the right hand side, for each customer in S, we add up the demand over the periods in the time interval, minus the largest possible inventory at the end of period $p_1 - 1$. The largest possible ending inventory at a customer i is equal to the holding capacity C_i minus the demand in period $p_1 - 1$. Note that for $p_1 = 1$ the right hand side can be improved to $\left\lceil \frac{\sum_{i\in S}\sum_{t=1}^{p_2} d_i^t}{Q}\right\rceil$ since there is no delivery possible before this time period and the remaining (residual) demand is known.

Avella et al. [2018] introduce a quadratic program to solve the separation problem for this family of inequalities, which is rewritten as follows for the IRP:

$$\min \sum_{t=p_1}^{p_2} \sum_{r\in R} \sum_{w\in W_r^t} \sum_{(i,j)\in A} a_{rij}\bar{y}_{rw}^t(1-\alpha_i)\alpha_j - \gamma \tag{4.22}$$

$$\text{s.t. } Q\gamma \le \sum_{i\in N}\left(\sum_{t=p_1}^{p_2} d_i^t - \left(C_i - d_i^{p_1-1}\right)\right)\alpha_i + Q - \epsilon \tag{4.23}$$

$$\alpha_i \in \{0,1\} \qquad \forall i \in N \tag{4.24}$$

$$\gamma \in \mathbb{Z} \tag{4.25}$$

in which $\bar{y}_{rw}^t$ are the values of the current fractional solution, $\alpha_i = 1$ if $i \in S$ and 0 otherwise for $i \in N$, and ϵ is a very small positive constant. γ represents the value of the right hand side of (4.21). Solutions with a negative objective correspond to violated cuts.

In the IRPDM, the demand of a customer $j \in S$ cannot only be satisfied by vehicles going into the set S (first term of (4.26)), but also by deliveries to a customer $i \in N \setminus S$ for which $j \in \mathcal{N}_i$ (second term of (4.26)). Note that if there are customers $j, k \in S$ and $k \in \mathcal{N}_j$, flow into j does not have to contribute twice to the flow into S to account for a possible demand move. We therefore adjust the MCS inequalities as follows for the IRPDM:

$$\sum_{t=p_1}^{p_2} \sum_{r\in R} \sum_{w\in W_r^t} \sum_{\substack{i,j:\\ i\in N\cup\{v^S\}\setminus S\\ \text{and } j\in S}} a_{rij} y_{rw}^t + \sum_{i\in N\setminus S} \sum_{j\in S\cap\mathcal{N}_i} \sum_{\substack{1\le t\le p_2:\\ P_{ijt}^+\cap[p_1,p_2]\neq\emptyset}} \sum_{r\in R} \sum_{w\in W_r^t} a_{ri} y_{rw}^t$$
$$\ge \left\lceil \frac{\sum_{i\in S}\left(\sum_{t=p_1}^{p_2} d_i^t - \left(C_i - d_i^{p_1-1}\right)\right)}{Q} \right\rceil, \quad \forall S \subseteq N,\ \forall p_1, p_2 \in P. \tag{4.26}$$

The second term represents deliveries to customers $i \in N \setminus S$ for which $j \in S \cap \mathcal{N}_i$, but only for the periods $1 \le t \le p_2$ in which the subdelivery periods P_{ijt}^+ have any overlap with the interval $[p_1, p_2]$ under consideration. Again, for $p_1 = 1$ the right hand side can be improved to $\left\lceil \frac{\sum_{i\in S}\sum_{t=1}^{p_2} \bar{d}_i^t}{Q} \right\rceil$.

To separate the inequalities for the IRPDM, the program (4.22)-(4.25) can still be used, but with the following extended objective function

$$\min \sum_{t=p_1}^{p_2} \sum_{r\in R} \sum_{w\in W_r^t} \sum_{(i,j)\in A} a_{rij}\bar{y}_{rw}^t(1-\alpha_i)\alpha_j +$$
$$\sum_{i\in N} \sum_{j\in\mathcal{N}_i} \sum_{\substack{1\le t\le p_2:\\ P_{ijt}^+\cap[p_1,p_2]\neq\emptyset}} \sum_{r\in R} \sum_{w\in W_r^t} a_{ri}\bar{y}_{rw}^t(1-\alpha_i)\alpha_j - \gamma \tag{4.27}$$

Associate dual variables $\pi_{S\ell\ell'}^{4.26}$, $S \subseteq N$, $\ell, \ell' \in P$ with inequalities (4.26). In the subproblem for period $p \in P$ the reduced costs are adjusted as follows

$$\bar{c}_{ij} = \begin{cases} c_{ij} - \pi_p^{4.1f} - \sum_{S \subseteq N} \sum_{\ell=1}^{p} \sum_{\ell'=p}^{\rho} \bar{z}_{ij} \pi_{S\ell\ell'}^{4.26} & \text{if } i = v^S \\ c_{ij} - \pi_{ip}^{4.1e} - \sum_{S \subseteq N} \sum_{\ell=1}^{p} \sum_{\ell'=p}^{\rho} \bar{z}_{ij} \pi_{S\ell\ell'}^{4.26} & \\ \quad - \sum_{S \subseteq N} \sum_{k \in \mathcal{N}_i} \sum_{\ell=1}^{p} \sum_{\ell'=p}^{\rho} \bar{z}_{ik} \hat{z}_{p\ell\ell'} \pi_{S\ell\ell'}^{4.26} & \text{otherwise,} \end{cases} \quad \forall (i,j) \in A^p. \tag{4.28}$$

with $\bar{z}_{ij}$ a parameter equal to 1 if customer $i \in N \cup \{v^S\} \setminus S$ and $j \in S$, and $\hat{z}_{p\ell\ell'}$ equal to 1 if $P_{ijp}^+ \cap [\ell, \ell'] \neq \emptyset$.

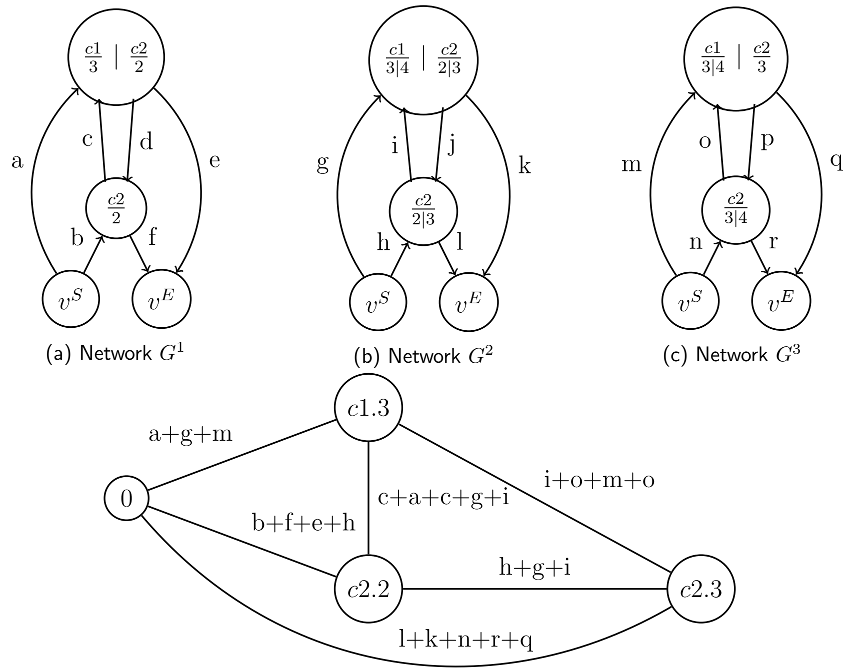
4.4.3.6 Capacity inequalities

The capacity inequalities were introduced for the Capacitated Vehicle Routing Problem (CVRP) by Laporte et al. [1985]. For the CVRP, these inequalities can be described as follows. Given a subset of customers $U \subseteq N$ and a lower bound $\kappa(U)$ on the number of vehicles required to service these customers given the vehicle capacity, the total flow of vehicles incident to subset U must be at least $2\kappa(U)$. Desaulniers et al. [2016] propose an adaptation of these inequalities for the IRP. Instead of a subset of customers, the authors use subsets of positive residual demands to define the capacity inequalities. For the CVRP, a graph depicting the flow between customers is used for separating the valid inequalities. For the IRP, an auxiliary graph is used which depicts the flow between consecutive residual demands assuming the FIFO principle.

For the IRPDM in which initial inventory cannot be used to satisfy moved demand, we extend the notion of the flow between residual demands. We will use a similar auxiliary graph as Desaulniers et al. [2016], however, the underlying structure of the graph per period changes. Define the set of residual demands $RD = \{(i,s) \in N \times P \mid \bar{d}_i^s > 0\}$ and the auxiliary graph $G^* = (V^*, E^*)$. Node set V^* contains a depot node 0 and a node for each residual demand in RD. The edge set E^* contains the following types of edges. First, an edge is present between the depot node and each residual demand node. Secondly, edges are present between consecutive nodes that correspond to the same customer, i.e., an edge exists between (i,s) and $(i,s+1)$. Third, there is an edge between nodes (i,s) and (i',s'), $i \neq i'$, if there exists a period $p \in P$ such that s is the latest period in $P_{ip}^+ \cap P$ and s' is the earliest period in $P_{i'p}^+ \cap P$. Until now, this definition is the same as in Desaulniers et al. [2016]. Additionally, for the IRPDM, an edge exists between (i,s) and (i',s'), $i \neq i'$, if i is in $\mathcal{N}_k$ for some $k \neq i, i'$ and there is a period $p \in P$ such that s is the latest period in P_{kip}^+ and s' is the earliest period in $P_{i'p} \cap P$. Edges that do not have any weight can be discarded.

For the weights on the edges, for a given (fractional) solution, we look into a network per period $G^p = (V^p, A^p)$. For the IRP, a node in this network for a given period p represents a customer and the periods for which a subdelivery can be made P_{ip}^+. For the IRPDM, a node represents both a customer and its neighbors, and the periods for which a subdelivery can be made for these customers in period p, i.e., the periods in P_{ip}^+ and P_{ijp}^+ for all $j \in \mathcal{N}_i$. To illustrate the structure of the auxiliary graph for the IRPDM, consider the example in Figure 4.1.

Consider the example with $N = \{c1, c2\}$ and $P = \{1, 2, 3\}$. Customer $c1$ has a positive residual demand in period 3 and customer $c2$ has a positive residual demand in periods 2 and 3. Moreover, customer $c2$ is a neighbor of customer $c1$, i.e., $\mathcal{N}_{c1} = \{c2\}$. The nodes in the networks in Figures 4.1a-4.1c represent both the customer and their



(a) Network G^1 (b) Network G^2 (c) Network G^3

(d) Auxiliary graph

Figure 4.1 Example for capacity inequalities

neighbor, if applicable. Period $4 = \rho + 1$ is the fictitious period and recall that this period is never included in the delivery periods for a neighbor. Figure 4.1d gives the corresponding auxiliary graph G^* in which customer i and period s are indicated by $ci.s$.

To illustrate the association of the arcs with the edges, consider as an example the edge between $c1.3$ and $c2.2$. This edge is present because the latest period in $P^+_{c2,1}$ is 2 and the first period in $P^+_{c1,1}$ is 3. The edge represents the flow between the residual demands and can be computed by summing the flow on arc $(c2, c1) \in A^1$ (arc c), the incoming arcs of node $c1 \in V^1$ (arcs a and c) and the incoming arcs of node $c1 \in V^2$ (arcs g and i). The last two sets are added since customer $c2$ is a neighbor of customer $c1$, the last period of $P^+_{c1,c1,1} \cap P$ and $P^+_{c1,c1,2} \cap P$ is period 3, and the first period of $P^+_{c1,c2,1}$ and $P^+_{c1,c2,2}$ is period 2. Note that arc c is added twice to this edge flow; counting arcs twice for one edge was not possible in the auxiliary graph for the IRP, but is now necessary to account for the demand moves.

Since we only change the underlying auxiliary graph for the IRPDM, the valid inequalities as defined in Desaulniers et al. [2016] and the impact on the reduced cost remain the same and are not repeated here for conciseness. Desaulniers et al. [2016] uses three separation heuristics for the capacity cuts for the IRP. The first one is the separation routine of the CVRPSEP package developed by Lysgaard et al. [2004] which is followed by a filter to incorporate that, for the IRP, the flow through a node can be higher than one. Second, current routes/RDPs with exactly one partial subdelivery in a current solution are considered to construct subsets $U \subseteq N$ on which the valid inequality is evaluated. Finally, a route-based connected component heuristic is applied which was proposed by Archetti et al. [2011] for the Split Delivery Vehicle Routing

Problem with Time Windows. For details on the separation heuristics, which are also used for the IRPDM, we refer to Desaulniers et al. [2016].

4.4.3.7 IRPDM in which initial inventory can satisfy moved demand

The inequalities presented in Sections 4.4.3.3 to 4.4.3.6 cannot be adjusted without changing their structure and effectiveness for the variant of the IRPDM in which initial inventory at a customer can be used to satisfy the demand of another customer via a demand move. For the inequalities in Sections 4.4.3.3 to 4.4.3.5, two main reasons preventing effective adjustments are (1) the ‘flow’ resulting from the use of initial inventory should be accounted for in the left hand side and (2) residual demand can no longer be used in the right hand side of the inequalities. For the capacity inequalities in Section 4.4.3.6, similar to the other inequalities, the residual demands can no longer be used to construct the auxiliary graph. Hereafter, we discuss the two main reasons in more detail by reflecting on the valid inequalities on the minimum number of visits per customer (Section 4.4.3.3).

(1) If it is possible to satisfy a demand move from initial inventory, potentially all demand at a customer j is satisfied from the initial inventory of customers j and $i : j \in \mathcal{N}_i$. In that case, no visits by a vehicle (to j or i) are needed to satisfy the demand of j. The idea of considering the use of initial inventory of i to satisfy demand of j as a ‘visit’, could be applied to adjust the valid inequality in two ways. A first approach could be to add the initial inventory variables to the left hand side, however, it would not be counting visits, but units of goods. One could divide by the demand and round up, but this would be non-linear. A second approach could be to add supplementary binary variables, which are equal to 1 if initial inventory is used to satisfy a demand move. However, these binary variables must be added to the master problem and moreover, a set of big-M constraints is needed to make sure the binary variables have the correct value.

(2) In the right hand side of the inequality, residual demands $\bar{d}_j$ can no longer be used since initial inventory of customer j can be used to satisfy demand of a customer $k \in \mathcal{N}_j$. Hence, not all initial inventory is necessarily available to satisfy demand of customer j itself. In the right hand side of inequality (4.17) we could therefore incorporate the variables that represent the use of initial inventory to satisfy moved demand. The disadvantage of that is that decision variables are included in the fraction and rounding the fraction would make the inequality non-linear.

Concluding, both on the left and the right hand side of the valid inequality the structure has to be changed to handle the possibility of using initial inventory to satisfy moved demand, making the inequalities weaker. A similar reasoning can be followed for the valid inequalities in Sections 4.4.3.4 and 4.4.3.5. Therefore, the inequalities cannot be adjusted for this variant of the IRPDM without changing their structure and effectiveness.

4.4.4 Branching

To find a feasible solution to the problem, seven types of branching decisions are evaluated if a fractional solution of the linear relaxation is computed. The branching decisions are defined on the following variables:

1. The total number of routes over all periods $\left(\sum_{p\in P}\sum_{r\in R}\sum_{w\in W_r^p} y_{rw}^p\right)$.
2. The number of routes per period $p \in P$ $\left(\sum_{r\in R}\sum_{w\in W_r^p} y_{rw}^p\right)$.
3. The number of visits per customer $i \in N$ $\left(\sum_{p\in P}\sum_{r\in R}\sum_{w\in W_r^p} a_{ri} y_{rw}^p\right)$.
4. v_i^s variables.
5. i_{ij}^p variables.
6. The flow through each customer vertex $i \in N$ in each period $p \in P$ $\left(\sum_{r\in R}\sum_{w\in W_r^p} a_{ri} y_{rw}^p\right)$.
7. The flow on each edge $<i,j>$ in each period $p \in P$ which is equal to the sum of the flows on the corresponding arcs (i,j) and (j,i) in A^p $\left(\sum_{r\in R}\sum_{w\in W_r^p} (a_{rij} + a_{rji}) y_{rw}^p\right)$.

Compared with the solution method proposed by Desaulniers et al. [2016] for the IRP, we added three types of variables to branch on: 3, 4 and 5. Types 4, 5 and 7 are sufficient to guarantee an optimal integer solution. For a discussion on the arc and edge flows we refer to Desaulniers et al. [2016]. Branching decisions are imposed in the model by adding a constraint, except for setting the flow on an edge to zero for which both corresponding arcs are removed from the arc set A^p. Adding an extra constraint to the master problem implies an adjustment in the reduced costs, specifics are omitted here for conciseness.

The next steps are followed to decide which branching decision is imposed. Compute the values of the variables for each type of decisions 1 to 7 and select for each type the candidate variable with a value closest to 0.5. If the candidates for types 6 and 7 have fractional values between 0.25 and 0.75, then branch on the variable with the value closest to 0.5 out of these two (at equality, select the type 3 decision). If there are no type 6 or 7 variables to branch on, if there is a candidate variable of type 1 or 2, select the candidate with the value closest to 0.5. If no candidate exists in the previous types, branch on the candidate variable of type 3 if one exists. Otherwise, choose the candidate variable of type 4 or 5 of which the value is closest to 0.5 to branch on.

A local-depth first search approach as described in Desaulniers et al. [2016] is applied to select the next node in the branch-and-bound tree to explore.

4.5 Computational experiments

To assess the impact of including the demand moves in the IRP, we performed computational experiments using the described branch-price-and-cut algorithm that was implemented using C++ and the Gencol library. CPLEX 12.6 is used to solve all restricted master problems during the solution procedure. These experiments are run on an Intel Core i7-4770 processor at 3.40 GHz, with 8 cores and 16 GB RAM. For all tests, only a single core is used and a time limit of two hours is imposed for each instance. To evaluate the benefits of demand moves, the IRPDM results are compared to the IRP by using the solution values obtained by Coelho and Laporte [2014] and Desaulniers et al. [2016] (see the results on Coelho [n.d.]).

To design our test set, we use the benchmark instances proposed by Archetti et al. [2007] for the IRP. The time horizon in these instances is either 3 or 6 periods, instances have a multiple of five customers, and there is one vehicle with a given capacity. Moreover, an instance contains for each customer the location, the initial inventory level, the maximum inventory capacity, the demand and the inventory holding rate. For the depot, the quantity becoming available is given instead of the demand and there is no maximum on the inventory. There are two levels for the inventory holding rate. Based on that, there are four classes of instances denoted by their inventory holding rate level (High or Low) and planning horizon (3 or 6), resulting in classes H3, H6, L3 and L6. The instances originally include a single vehicle, but the instances have been used for the multi-vehicle case by dividing the vehicle capacity by the chosen number vehicles. Details on the instances are available in Archetti et al. [2007]. Both the instances and the detailed results for the IRP by Coelho and Laporte [2014] and Desaulniers et al. [2016] are available at Coelho [n.d.].

To determine the set of neighbors to incorporate demand moves, we find for each customer $i \in N$ the closest customer $j \in N \setminus \{i\}$ that is within 150 units of distance and set $\mathcal{N}_i = \{j\}$. All valid inequalities are added in a dynamic way in each node of the branch-and-bound tree. The capacity cuts and MCS cuts are only added in nodes that are at most at depth two in the tree. The costs for demand moves is set to $m_{ij} = 0.01$ per unit of goods and per unit of distance between locations i and j (following Coelho et al. [2012]) and there is no limit on the amount of demand that can be moved, unless indicated otherwise.

In Section 4.5.1, we present results to assess the effectiveness of the new GRI (Section 4.4.3.2), the MCS inequalities (Section 4.4.3.5), and the capacity inequalities (Section 4.4.3.6). Thereafter, generating results for the IRPDM with the most efficient settings, Section 4.5.2 compares the solution values of the IRPDM with the solution values of the IRP. In Section 4.5.2.1 the cost of a demand move is set to $m_{ij} = 0.05$ and $m_{ij} = 0.1$ to assess the impact of changing this cost. In practice it might not be desirable that all demand of a customer is satisfied by another customer as described in Section 4.3.1. Hence, Section 4.5.2.2 reports the effect of limiting the percentage of demand that can be moved to 25%, 50% and 75% of the demand of one customer in each period.

4.5.1 Effectiveness of valid inequalities

We assess the effectiveness of the GRI, MCS and capacity inequalities by solving the IRPDM with different combinations of valid inequalities. The first setting includes the capacity inequalities and the GRI, while the second setting includes the MCS inequalities and the GRI. For the third and fourth settings, both the capacity and MCS inequalities are included. Additionally, setting three includes the GRI, while setting four uses the original route inequalities (RI). The remaining valid inequalities are used in all settings. For each setting, the IRPDM is solved for a subset of the instances. The algorithm is tested on the instances with 3 and 4 vehicles ('K'), with 5 and 10 customers for horizon 3, and with 5 customers for horizon 6, for both high and low inventory holding costs. Hence, for each H3/L3 class ('Class') there are 10 instances, and for each H6/L6 class there are 5 instances in this subset.

Table 4.1 reports for each class and number of vehicles the average integrality gap at the root node before adding valid inequalities ('Gap$_r$'), which is the same for all settings. Thereafter, the table compares for each setting the number of instances solved to optimality ('Opt'), the average running time of instances solved to optimality ('T(s)'), the average integrality gap at the root node after adding valid inequalities ('Gap'), and the average number of capacity ('CI') and MCS ('MCS') inequalities added during the execution of the algorithm, respectively. Only the instances solved to optimality are considered when computing the averages. The integrality gap is computed as $(\overline{z} - \underline{z})/\overline{z}$ with $\underline{z}$ the lower bound computed at the root node of the branch-and-bound tree and $\overline{z}$ the optimal value. The total averages over all instances are also given in the table.

Table 4.1 Effectiveness of GRI, MCS and Capacity inequalities

			1. Cap ineq - GRI				2. MCS ineq - GRI				3. Cap & MCS ineq - GRI					4. Cap & MCS ineq - RI				
Class	K	Gap_r	Opt.	T(s)	Gap	CI	Opt.	T(s)	Gap	MCS	Opt.	T(s)	Gap	CI	MCS	Opt.	T(s)	Gap	CI	MCS
H3	3	2.9	10	19	1.3	6.7	10	12	1.0	4.4	10	14	1.0	1.5	4.2	10	14	1.0	1.5	4.2
H3	4	3.9	10	16	1.6	8.0	10	12	1.4	4.9	10	11	1.4	1.3	4.9	10	11	1.4	1.3	4.9
H6	3	2.7	4	1006	2.5	1.3	4	669	2.3	4.8	4	711	2.3	0.3	4.3	4	698	2.3	0.5	4.3
H6	4	2.4	5	1821	2.0	6.8	5	1475	1.7	3.0	5	1348	1.7	2.6	3.6	5	1333	1.7	2.6	3.6
L3	3	5.2	10	20	2.3	5.5	10	20	1.8	4.6	10	19	1.8	1.8	4.5	10	19	1.8	1.8	4.5
L3	4	6.7	10	19	2.6	7.7	10	16	2.4	4.9	10	15	2.4	1.8	4.8	10	15	2.4	1.8	4.8
L6	3	4.1	4	2414	3.7	1.8	4	1894	3.5	4.0	4	1885	3.5	0.3	4.0	4	1834	3.5	0.3	4.0
L6	4	3.4	4	1238	2.4	7.8	4	1299	2.4	2.5	4	1174	2.4	5.8	3.0	4	1168	2.4	4.0	3.3
Total		4.2	57	500	2.1	6.2	57	411	1.9	4.4	57	393	1.9	1.8	4.3	57	387	1.9	1.7	4.4

Table 4.1 shows that with all four settings the same number of instances can be solved. Moreover, the capacity and MCS inequalities are effective since the integrality gap is approximately halved by adding these valid inequalities. This can be observed from the overall integrality gaps of 2.1% and 1.9% for settings 1 and 2 respectively, compared to the gap of 4.2% in the root node before adding valid inequalities. Tests in which the capacity and MCS inequalities are not included but the RI are, show that still 55 instances can be solved to optimality, but that the average running time is more than 2.5 times as high and the integrality gap is more than twice as high as in setting 4. Details on these tests are not reported, but are available on request.

The results indicate that the MCS inequalities are slightly more effective than the capacity inequalities since setting 2 gives both a lower average computation time and lower integrality gap after adding the valid inequalities than setting 1. Combining these types of inequalities in settings 3 and 4 increases the efficiency since the running time goes down, even tough the integrality gap is the same as for settings 2 and 3. In settings 3 and 4, the average number of identified capacity inequalities is much lower than in setting 1, but since the MCS inequalities are slightly more effective, the average computation time is still lower in settings 3 and 4 than in setting 1. Comparing settings 3 and 4 shows that the integrality gaps are the same, which implies that using the generalized route inequalities does not seem to improve the solution method for the IRPDM. Note that the average number of capacity and MCS inequalities differs slightly between settings 3 and 4. Although the generalization of the route inequalities is not effective for the IRPDM, this cannot immediately be concluded for other problems. Based on these observations, setting 4 will be used for the remainder of the experiments.

4.5.2 Comparing IRP and IRPDM

To evaluate the benefit of exploiting demand moves in the IRP, the solutions of the IRPDM are compared to the solutions of the IRP. We look into the (percentage) cost improvement that is achieved for the test instances, and examine the number moves and their size that actually take place in the IRPDM solutions. The solutions for the IRP are collected from Coelho [n.d.].

Table 4.2 shows the obtained results. For each class of instances ('Class'), fleet size ('K') and number of customers ('N'), Table 4.2 first shows the number of instances that are solved to optimality for both the IRP and IRPDM ('Opt.'). Secondly, the average computation time for the branch-and-cut IRP algorithm by Coelho and Laporte [2014] ('T(s)-BC'), for the branch-price-and-cut IRP algorithm by Desaulniers et al. [2016] ('T(s)-BPC'), and for the IRPDM ('T(s)') are reported. Thereafter, the average ('Av. Impr'), maximum ('Max. Impr') and minimum ('Min. Impr') percentage cost improvement of the IRPDM over the IRP are stated. Finally, the average number of demand moves ('Av. Nr. of DMs') and the average size of a demand move are reported ('Av. Sz. of DM').

We run the algorithm on instances with up to 25 customers for a three period horizon and up to 10 customers for a six period horizon. All results are reported in Appendix E. Two instances can be solved for the IRPDM while no feasible solution for the IRP exists. These instances are not included in the averages in the table, since there are no IRP results to compare with. For instances with a horizon of six periods, we can only solve instance with five customers to optimality. This is limited, however, note that for the IRP not all instances with six periods and ten customers have been solved to optimality with branch-price-and-cut in state-of-the-art literature [Desaulniers et al., 2016] and the proposed IRPDM is a more complicated problem.

The average computation times show that the IRPDM instances require more time to solve than the IRP with the branch-price-and-cut solution method. This can be expected since the branch-price-and-cut solution method for the IRPDM is an extension of the one for the IRP. Compared to the branch-and-cut method for the IRP, solving the IRP is in general easier. The higher computation times are caused by having a more extensive master problem which includes additional binary variables and new constraints. Also, the capacity constraints are not redundant with the other constraints in the master problem which was the case for the IRP [Desaulniers et al., 2016]. Moreover, the number of delivery patterns per customer increases substantially since the patterns include the deliveries dedicated to the neighboring customers. Consequently, the number of labels is greater which slows down the pricing problem. However, for some instances the branch-price-and-cut method for the IRPDM solves them more quickly, for example class H3 with 4 vehicles and 10 customers.

The average cost improvement is around 2.35% for instance classes with high holding costs and above 3.5% for instance classes with low inventory holding costs, respectively. In general, it can be observed that the average improvements are higher for low inventory holding costs while the average number and the average size of the demand moves do not differ much. This can be explained by the fact that for high holding costs the routing costs are a smaller part of the solution value than for low holding costs. Since the demand moves decrease the routing costs and increase the holding costs, the savings yielded by including demand moves are larger when holding costs are lower.

The maximum improvements go up to 10% and for only three classes of instances the minimum cost improvement equals zero. A zero cost improvement means that solving the IRPDM results in the same solution as the IRP, i.e., exploiting demand moves does not result in a cost saving. Looking into the detailed results in Appendix E shows that out of 96 instances only three instances do not result in a cost improvement. The number of demand moves per instance is 2.6 and 2.8 for short time horizons, and 5.3 and 5.5 for long horizons, respectively. This implies that there is on average one

Table 4.2 Comparison of solution values IRP and IRPDM

				IRP		IRPDM					
Class	K	N	Opt.	T(s)-BC	T(s)-BPC	T(s)	Av. Impr. (%)	Max. Impr. (%)	Min. Impr. (%)	Av. Nr. of DMs	Av. Sz. of DM
H3	3	5	5/5	0.6	0.0	0.2	3.35	4.10	1.49	2.2	39.9
H3	3	10	5/5	8.2	4.9	26.9	1.43	3.56	0.16	2.6	37.4
H3	3	15	3/5	22.0	8.2	279.9	0.58	0.93	0.26	2.0	12.3
H3	3	20	1/5	26.0	1.1	2402.9	0.08	0.08	0.08	2.0	16.5
H3	4	5	5/5	0.8	0.0	3.8	3.43	5.55	0.86	2.8	28.8
H3	4	10	5/5	110.4	4.3	18.4	1.22	2.99	0.00	3.2	39.5
H3	4	15	1/5	84.0	33.5	7155.5	0.78	0.78	0.78	3.0	26.0
H3	5	5	5/5	0.6	0.0	0.4	5.00	6.07	3.75	2.0	30.0
H3	5	10	5/5	264.8	1.5	778.8	2.22	4.01	1.49	3.6	33.9
H3	5	15	1/5	2252.0	7.6	5860.0	0.55	0.55	0.55	2.0	36.0
H3	5	20	1/5	2918.0	3.4	4858.2	0.92	0.92	0.92	2.0	15.0
Total	**H3**		**37/60**	**196.6**	**3.3**	**682.7**	**2.36**	**6.07**	**0.00**	**2.6**	**31.6**
H6	3	5	4/5	34.8	490.7	697.6	1.83	3.04	0.00	4.8	54.2
H6	4	5	5/5	37.6	110.2	1332.7	2.47	4.29	0.46	5.6	41.7
H6	5	5	3/5	110.0	102.6	1188.9	2.99	4.17	1.04	5.3	37.1
Total	**H6**		**12/15**	**54.8**	**235.1**	**1085.0**	**2.38**	**4.29**	**0.00**	**5.3**	**43.8**
L3	3	5	5/5	0.6	0.0	0.3	5.25	6.98	2.12	2.2	39.9
L3	3	10	5/5	8.4	4.9	37.8	2.66	6.47	0.24	2.4	42.0
L3	3	15	3/5	18.3	9.8	379.3	1.24	1.97	0.57	1.7	13.2
L3	3	20	1/5	11.0	0.9	737.5	0.50	0.50	0.50	2.0	16.5
L3	4	5	5/5	0.8	0.0	4.1	5.09	7.22	1.18	3.2	26.4
L3	4	10	5/5	111.2	6.2	26.7	2.13	5.25	0.00	3.8	34.8
L3	4	15	1/5	91.0	45.8	3454.7	1.80	1.80	1.80	7.0	14.3
L3	5	5	5/5	0.8	0.0	0.5	7.20	10.02	5.50	2.0	29.6
L3	5	10	5/5	262.2	3.9	713.8	3.80	6.87	2.33	3.4	35.0
L3	5	15	1/5	2037.0	14.1	1821.5	1.76	1.76	1.76	2.0	36.0
L3	5	20	1/5	3975.0	3.0	5899.1	2.14	2.14	2.14	3.0	30.7
Total	**L3**		**37/60**	**218.6**	**4.5**	**458.6**	**3.80**	**10.02**	**0.00**	**2.8**	**31.7**
L6	3	5	4/5	46.3	1169.5	1833.9	2.96	5.06	0.14	5.5	39.2
L6	4	5	4/5	127.3	549.6	1168.3	4.11	6.37	0.89	6.3	34.9
L6	5	5	2/5	110.5	0.6	216.2	3.55	5.26	1.83	4.0	41.0
Total	**L6**		**10/15**	**91.5**	**687.8**	**1244.1**	**3.54**	**6.37**	**0.14**	**5.5**	**37.8**

demand move per period of the planning horizon. The higher number of moves for longer horizons can be explained by the fact that in a longer planning horizon there are more opportunities to incorporate a demand move for multiple periods. Note that the percentage cost improvement is not higher for a longer planning horizon than for a shorter planning horizon. If a demand move takes place, the number of units moved is quite substantial with averages between 30 and 45 units, which is approximately between half and three-quarters of the average demand (the demand is between 10 and 100).

4.5.2.1 Impact of the demand move costs

In the previous experiments, the service fee incurred for a demand move (per unit of goods and per unit of distance) is set to $m = 0.01$. This section examines the impact of this parameter on the IRPDM solutions by solving the IRPDM for different values of m. For a subset of instances (instance sizes 5, 10 and 15 for horizon 3, and sizes 5 and 10 for horizon 6), the IRPDM is solved for $m = 0.005$, $m = 0.05$ and $m = 0.1$ as well. The latter two values were also tested by Coelho et al. [2012] for the IRPT. Table

4.3 reports the average cost improvement over the IRP ('Av. (%)'), the maximum cost improvement ('Max. (%)'), the average number of demand moves ('Av. Nr.') and the average size of the demand move ('Av. Sz.') per instance class and fleet size. Only instances that are solved to optimality for all parameter values of m are considered (the number of instances solved is indicated in column 'Opt.'), therefore, averages can differ marginally from the reported results in Table 4.2 for $m = 0.01$. Detailed results can be found in Appendix F.

Table 4.3 Impact of move cost m

			m=0.005				m=0.01				m=0.05				m=0.1			
Class	K	Opt.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.
H3	3	12	3.0	6.2	2.5	32.2	2.1	4.1	2.3	34.4	0.4	1.7	0.3	6.0	0.2	0.7	0.3	3.7
H3	4	10	3.5	6.6	3.4	40.8	2.3	5.5	3.0	33.6	0.4	2.0	0.4	4.8	0.2	1.6	0.3	2.0
H3	5	10	4.5	7.9	3.1	33.4	3.6	6.1	2.8	31.9	0.6	1.6	0.8	12.2	0.0	0.4	0.2	6.0
Total	**H3**	**32**	**3.6**	**7.9**	**3.0**	**35.3**	**2.6**	**6.1**	**2.7**	**33.4**	**0.5**	**2.0**	**0.5**	**8.3**	**0.1**	**1.6**	**0.3**	**3.6**
H6	3	4	3.9	5.8	7.3	50.8	1.8	3.0	4.8	54.2	0.4	1.5	0.3	6.0	0.3	1.1	0.3	6.0
H6	4	4	3.3	6.2	5.8	39.1	2.2	4.3	5.0	44.2	0.2	0.3	0.5	16.0	0.0	0.0	0.3	2.0
H6	5	1	7.6	7.6	12.0	45.3	3.8	3.8	5.0	34.0	0.0	0.0	0.0	-	0.0	0.0	0.0	-
Total	**H6**	**9**	**4.1**	**7.6**	**7.1**	**45.0**	**2.2**	**4.3**	**4.9**	**46.7**	**0.2**	**1.5**	**0.3**	**12.7**	**0.1**	**1.1**	**0.2**	**4.0**
L3	3	12	5.0	10.5	2.3	35.8	3.5	7.0	2.2	36.6	0.7	3.1	0.3	6.0	0.3	1.4	0.3	3.7
L3	4	11	5.3	8.6	4.3	34.4	3.4	7.2	3.8	28.5	0.5	3.1	0.4	4.8	0.3	2.5	0.3	2.0
L3	5	10	6.8	12.7	3.1	33.6	5.5	10.0	2.7	32.3	1.0	2.5	1.0	11.7	0.1	0.7	0.1	6.0
Total	**L3**	**33**	**5.7**	**12.7**	**3.2**	**34.7**	**4.1**	**10.0**	**2.9**	**32.7**	**0.7**	**3.1**	**0.5**	**8.3**	**0.2**	**2.5**	**0.2**	**3.3**
L6	3	4	6.1	9.5	7.0	51.4	3.0	5.1	5.5	39.2	0.6	2.4	0.3	6.0	0.4	1.7	0.3	6.0
L6	4	3	5.3	9.0	6.3	34.0	3.5	6.4	6.0	34.6	0.4	0.7	0.7	16.0	0.0	0.0	0.0	-
L6	5	1	10.5	10.5	11.0	49.4	5.3	5.3	5.0	33.4	0.0	0.0	0.0	-	0.0	0.0	0.0	-
Total	**L6**	**8**	**6.4**	**10.5**	**7.3**	**44.6**	**3.4**	**6.4**	**5.6**	**36.7**	**0.4**	**2.4**	**0.4**	**12.7**	**0.2**	**1.7**	**0.1**	**6.0**

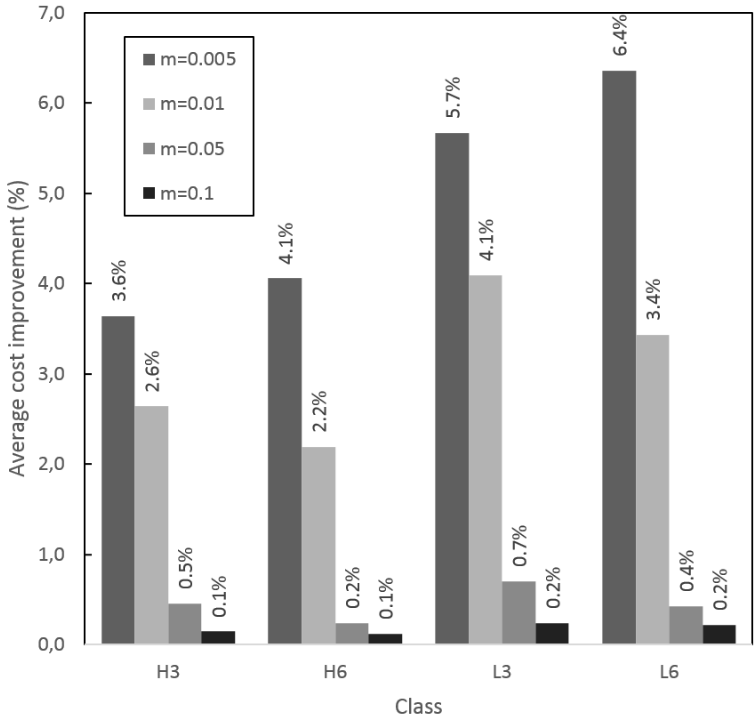


Figure 4.2 Average cost improvement for m-values by class

The results in Table 4.3 show that the improvement over the IRP by using demand moves diminishes if the cost of demand move increases, which can be expected. Increasing the value of m from $m = 0.01$ to $m = 0.05$ results in average improvements that are approximately a factor five lower, as illustrated in Figure 4.2. The average number of demand moves decreases more rapidly as planning horizons become longer. Increasing the value of m to 0.1 results in very few demand moves, and hence, a very minor cost improvement of only 0.1% and 0.2% on average for high and low inventory holding costs, respectively. Moreover, if a demand move takes place, the number of units moved is very limited and approximately half of the size for $m = 0.05$.

Lowering m from $m = 0.01$ to $m = 0.005$ leads to higher improvements, as can be expected. Note that mainly for long horizon instances, the number of demand moves increases which results in an average cost improvement twice the improvement for $m = 0.01$. The average size of the demand moves does not change substantially for this change in demand move cost m.

Overall, it can be observed that the value of m has a larger impact for instances with a longer planning horizon. Figure 4.2 shows that when m is increased, the average cost improvement decreases faster for a longer than for a shorter planning horizon. Also the number of demand moves declines faster for a longer planning horizon, starting at averages of well above one move per period for $m = 0.005$, but reducing to almost zero for $m = 0.05$ and $m = 0.1$.

4.5.2.2 Impact of limit on moved demand

As discussed in Section 4.3.1, the IRPDM allows that all demand of one customer is moved to another customer (after using the initial inventory). This could imply that some customers are never replenished by a vehicle. From a service point of view, this might be unacceptable. In this section we therefore analyze the impact on the solutions when the moved demand per customer per period is limited to a given percentage, as discussed in Section 4.3.1. We solve the IRPDM for a maximum of 25%, 50% and 75%, additionally to the results already obtained for 100% (which allows for moving all demand). The same instances are used as in Section 4.5.2.1 and Table 4.4 reports similar information as Table 4.3. Only instances solved to optimality for all settings are taken into account, therefore, the averages for 100% can deviate slightly from the reported results in Table 4.2. Appendix F reports the detailed results.

Table 4.4 shows that the average number of units moved decreases if the maximum demand moved becomes smaller. For example, for class H3, the average size is only 6 units if the limit is 25% compared to 32.3 units if there is no maximum. It is interesting to observe that the average number of units declines stronger for a shorter planning horizon than for a longer planning horizon. For instance, for class H6 the average number of units is 15.6 for a 25% limit, which is much larger than the 6 units for class H3 while the differences between the classes is small if there is no limit imposed (32.3 vs. 37.8 units). The same observation holds for classes with low holding costs.

Furthermore, the results show that the difference in cost improvement is small between a limit of 25% and 50%. Figure 4.3 shows that the largest difference can be observed between maxima of 75% and 100% (i.e., no limit). Restricting the demand moved to 75% of the demand per customer per period approximately halves the percentage cost improvement over the IRP. As an example, consider instance class L3

Table 4.4 Impact of maximum on moved demand

			Max. 25%				Max. 50%				Max. 75%				Max. 100%			
Class	K	Opt.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.	Av. (%)	Max. (%)	Av. Nr.	Av. Sz.
H3	3	13	0.3	2.6	0.3	4.3	0.5	2.6	0.5	10.6	0.8	3.2	1.0	23.5	2.0	4.1	2.3	32.6
H3	4	10	0.4	2.4	0.4	2.5	0.7	2.9	0.7	21.0	0.8	2.9	0.8	25.5	2.3	5.5	3.0	33.6
H3	5	9	1.0	3.2	1.1	9.4	1.2	3.8	1.4	12.4	2.1	4.7	1.7	22.1	3.8	6.1	2.9	30.6
Total	**H3**	**32**	**0.5**	**3.2**	**0.6**	**6.0**	**0.8**	**3.8**	**0.8**	**14.0**	**1.2**	**4.7**	**1.1**	**23.6**	**2.6**	**6.1**	**2.7**	**32.3**
H6	3	3	0.8	2.0	1.3	16.5	0.8	2.0	1.0	23.5	0.9	2.0	2.0	21.1	1.9	3.0	4.7	49.0
H6	4	3	0.8	1.4	0.7	16.0	0.9	1.4	0.7	18.5	1.0	1.7	1.3	23.3	2.4	4.3	6.0	31.6
H6	5	1	1.1	1.1	4.0	13.0	1.1	1.1	2.0	27.5	2.5	2.5	4.0	36.5	3.8	3.8	5.0	34.0
Total	**H6**	**7**	**0.9**	**2.0**	**1.4**	**15.6**	**0.9**	**2.0**	**1.0**	**22.3**	**1.2**	**2.5**	**2.0**	**25.1**	**2.4**	**4.3**	**5.3**	**37.8**
L3	3	12	0.6	4.6	0.3	3.7	0.9	4.6	0.4	9.0	1.3	5.6	0.9	27.6	3.1	7.0	1.8	34.7
L3	4	11	0.6	3.6	0.5	4.2	1.0	3.6	0.8	18.5	1.1	3.6	0.9	22.7	3.4	7.2	3.8	28.5
L3	5	9	1.7	4.8	1.0	9.4	2.1	6.1	1.2	12.6	3.4	6.6	1.4	21.8	5.8	10.0	2.8	31.0
Total	**L3**	**32**	**0.9**	**4.8**	**0.6**	**6.5**	**1.3**	**6.1**	**0.8**	**13.6**	**1.8**	**6.6**	**1.1**	**24.4**	**4.0**	**10.0**	**2.8**	**31.6**
L6	3	2	0.5	0.9	1.5	14.0	0.6	1.0	1.0	21.0	0.7	1.3	2.5	18.6	2.3	4.4	3.5	34.9
L6	4	3	1.2	2.1	0.7	16.0	1.3	2.1	0.7	18.5	1.5	2.8	1.3	23.3	3.5	6.4	6.0	34.6
L6	5	1	1.8	1.8	4.0	10.7	1.9	1.9	2.0	21.5	3.7	3.7	4.0	35.8	5.3	5.3	5.0	33.4
Total	**L6**	**6**	**1.1**	**2.1**	**1.5**	**14.2**	**1.1**	**2.1**	**1.0**	**20.1**	**1.6**	**3.7**	**2.2**	**23.9**	**3.4**	**6.4**	**5.0**	**34.5**

which has an average cost improvement of 4.0% if there is no limit, and only 1.5% in case of a maximum of 75%. This shows that if there is any restriction on the amount of demand that can be moved, a large share of the potential cost improvement is lost. This can be explained by the fact that in case of a limit, some customers must be served by a vehicle while their demand would have been moved and no visit would be required if there was no limitation on the moved demand. Therefore, routing costs increase and the improvement over the IRP is lower. The number of required replenishments is also enhanced by the limitation that demand can only be moved if there is no inventory left. Hence, replenishing a customer once and spreading this inventory over multiple periods combined with moving some demand every period is not possible, instead, replenishments with a vehicle are necessary.

Although limiting the demand that can be moved, to 25% for example, clearly results in lower cost improvements over the IRP than imposing no limit. From a service perspective, this can still be preferable. Even with moving a very limited amount of goods, we still find average cost improvements between 0.5% and 1.1%, and up to 4.8% maximally, which can be substantial in practice.

4.6 Conclusion

In this paper, we introduced the Inventory Routing Problem with Demand Moves (IRPDM). This problem is an extension of the Inventory Routing Problem (IRP) with the addition that a customer can satisfy (part of) the demand of another customer. Although originally inspired by redirecting ATM-users to nearby ATMs, the IRPDM can prove useful to a variety of settings. We formulate a mathematical model for the IRPDM as an extension of the IRP formulation of Desaulniers et al. [2016] and we develop a branch-price-and-cut solution method including non-trivially adjusted valid inequalities stemming from the IRP.

The IRPDM is solved on IRP benchmark instances from the literature [Archetti et al., 2007] and the performance of three types of valid inequalities is analyzed. The

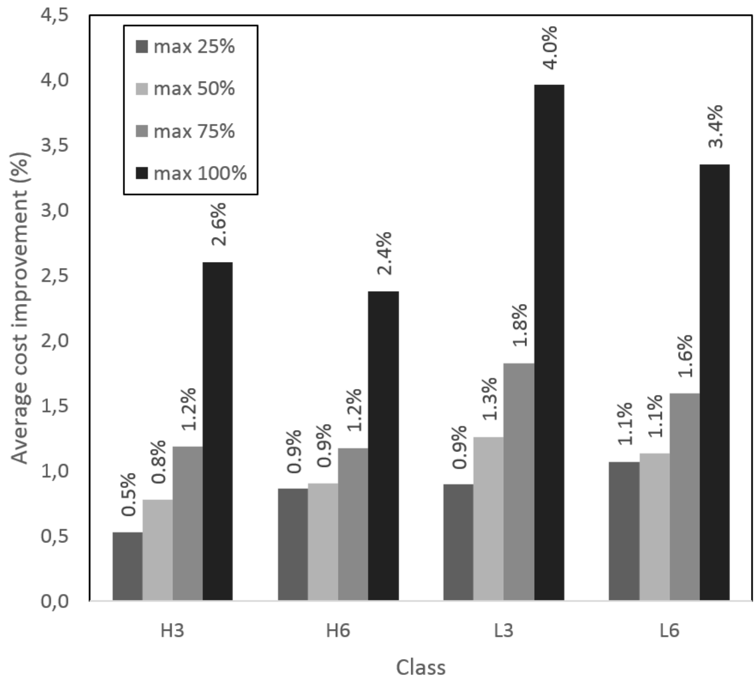


Figure 4.3 Average cost improvement for maximum demand move by class

tests show that MCS inequalities (by Avella et al. [2018] for the IRP) adjusted for the IRPDM are more effective than adjusted capacity cuts (by Desaulniers et al. [2016] for the IRP), and that using both these types of inequalities results in the best performance of the algorithm. To assess the impact of allowing for demand moves in the IRP, we compare the solutions of the IRPDM to those of the IRP. Moreover, we analyze the average number and size of demand moves to develop management insights.

Cost improvements of up to 10% are achieved for a demand move cost of $m = 0.01$ per unit of demand and unit of distance and if there is no limit on the moved demand. Moreover, it is observed that there is on average approximately one demand move per day, which implies that these improvements are achieved without a large change in the solutions compared to the IRP. The designed algorithm can solve instances with up to twenty customers, three periods and five vehicles to optimality, which is limited. It must be noted that the IRPDM is much more difficult than the IRP, for which instances up to fifty customers can be solved to optimality with a state-of-the-art branch-price-and-cut method [Desaulniers et al., 2016]. Sensitivity analysis on both the demand move costs and the maximum on the moved demand per customer per period is performed. Varying the demand move costs shows that the impact of increasing the costs is larger for a longer planning horizon than for a shorter planning horizon on both the percentage cost improvement over the IRP and the number of demand moves performed. Limiting the demand that can be moved per period of one customer to 75% of its demand, already has a considerable impact on the cost improvement over the IRP compared with the cost improvement if there is no limit. The percentage cost improvement is approximately halved in case of 75% compared to 100%. Even by allowing only 25% of the demand to be moved, we observe cost improvements up to 4.8% and around 1% on average compared to the classical IRP.

In this paper, we limit ourselves to the case in which initial inventory can only be used to satisfy demand of the customer itself for algorithmic reasons. An extension would be to develop an exact solution method that does accommodate satisfying moved demand with the initial inventory. A challenge can especially be found in the design of valid inequalities for this problem as discussed in Section 4.4.3.7. Moreover, the results show that allowing for demand moves can lead to significant cost savings. Therefore, the design of a heuristic solution method for the IRPDM capable of solving larger scale IRPDM instances is an interesting future research direction. A helpful insight obtained in this paper which can be used in the development of heuristics, is that the number of demand moves taking place in optimal solutions is rather limited. Finally, in our model we only consider a load capacity constraint on the vehicles. In practice, the number of ATMs that can be served by one vehicle in one period is often limited by time, which is now not considered in the IRPDM. Hence, demand moves can also be useful if there is insufficient vehicle *time* capacity to replenish all ATMs in a certain area. It would be interesting to investigate the impact of allowing demand moves if the number of customers that can be replenished is further limited, for example, by limiting the number of customers served per vehicle.

5

The Vehicle Routing Problem with Partial Outsourcing

5.1 Introduction

To improve the efficiency of distribution logistics, academic literature has developed efficient solution algorithms and has explored novel distribution strategies. Fundamental optimization problems such as the Capacitated Vehicle Routing Problem and the Vehicle Routing Problem with Time Windows (VRPTW), which underpin several distribution problems occurring in practice, have been the subject of intensive research efforts in search of more (cost) efficient solutions [Toth and Vigo, 2014]. Novel distribution strategies such as outsourcing and split deliveries contribute in a different but no less effective way to lowering distribution costs.

Outsourcing refers to the fact that the service of some customers can be entrusted to a third party logistics service provider. In routing problems, outsourcing can be applied in case demand exceeds available transportation capacity or if it is more economical to do so. The two options for servicing customers - by a private vehicle fleet or a common carrier - open up additional opportunities for reducing distribution costs. Applications in the literature date back to Chu [2005] and a recent overview of the literature on the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC) can be found in Dabia et al. [2019]. Split deliveries imply that the demand of a customer is not necessarily delivered by a single vehicle. Split deliveries are a necessity if demand exceeds vehicle capacity but can also prove cost efficient if e.g., demand of several customers is slightly higher than half of the vehicle capacity [Archetti et al., 2008]. Chen et al. [2007] review several applications of split deliveries in VRPs.

Combining outsourcing and split delivery features has received increasing attention

This chapter is based on: A.C. Baller, S. Dabia, W.E.H. Dullaert, and D. Vigo, The Vehicle Routing Problem with Partial Outsourcing, Transportation Science, 2019, to appear [Baller et al., 2019c] in the literature in recent years. To the best of our knowledge, only application-based studies have been reported in which heuristic methods are proposed, as discussed in Section 5.2.3. Our study wants to contribute to the literature by formally describing the vehicle routing problem in which a customer can either be served by a single private vehicle, by a common carrier, or by both a single private vehicle and a common carrier. We do not allow for multiple private vehicles to serve the same customer, as serving customers by multiple private vehicles and a common carrier may lead to customer inconvenience (see for example Bianchessi et al. [2019]). It should also be noted that allowing multiple private vehicle visits to a customer would lead to a more complicated optimization problem. We refer to the defined problem as the Vehicle Routing Problem with Partial Outsourcing (VRPPO). We assume that customers impose time windows, there is a heterogeneous limited fleet and the outsourcing cost is a fixed fee per unit.

In this paper, we propose two path-based formulations for the VRPPO and solve these with a branch-and-price-and-cut approach since this method has proven to be an effective method for various (related) routing problems. The first formulation explicitly models the quantity delivered by a private vehicle to each customer, while the second formulation defers modeling this quantity and the resulting outsourced units to the pricing problem. For both problem formulations, we design two specialized pricing procedures to generate additional columns during the solution process. The first pricing procedure remains close to current literature and is closely related to the pricing algorithm proposed for the SDVRP by Desaulniers [2010]. The second pricing procedure is closer to the algorithm proposed by Luo et al. [2016] and exploits some specific properties of the VRPPO.

A solution to the VRPPO consists of a set of routes for the private vehicles and a set of customers for which (part of) the demand's delivery is outsourced, and for both the corresponding quantities. The objective is to minimize the total routing and outsourcing cost. The purpose of this paper is to investigate the effectiveness of the two formulations and corresponding solution methods and to compare the costs with the VRPPC to assess the gain that can be achieved by embracing both distribution strategies.

The paper is organized as follows: in Section 5.2 we review the literature related to the VRPPO. In Section 5.3, we describe the problem more formally and present the problem formulations and solution methods. Section 5.4 presents some implementation features and Section 5.5 describes the computational results. Conclusions and suggestions for future research follow in Section 5.6.

5.2 Literature Review

In this section the main literature related to the VRPPO is reviewed. First, in Section 5.2.1, we discuss literature on the SDVRP because the solution methods are relevant for solving the VRPPO. Secondly, the literature on the VRPPC is examined in Section 5.2.2 since a generalization of the VRPPC is studied. Finally, in Section 5.2.3 we review some work on combinations of the SDVRP and VRPPC.

5.2.1 Literature on the SDVRP

The vehicle routing problem with split deliveries and a private fleet is extensively studied. Archetti and Speranza [2012] provide a thorough overview of properties and solution methods, both exact and heuristic, for this problem and its variants, including the variant with time windows (SDVRPTW).

Desaulniers [2010] studies the SDVRPTW and proposes a branch-and-price-and-cut solution method. The columns in the master problem represent a route and corresponding 'route delivery pattern' which indicates the quantities delivered to each customer. The pricing problem is a variant of the Elementary Shortest Path Problem with Resource Constraints (ESSPRC) which is solved with a labeling algorithm. In order to generate relevant delivery patterns, a label is extended to the next customer at most three times, for a delivery of zero units, a delivery equal to the demand, and a delivery quantity strictly between zero and the demand. Archetti et al. [2011] propose an improved version of the algorithm by Desaulniers [2010] mainly by introducing a tabu search heuristic for the pricing problem and by applying new valid inequalities and separation procedures. Luo et al. [2016] propose another improvement over the algorithm by Archetti et al. [2011] and also include a linear weight-related cost function. Instead of extending each label up to three times, they observe that the delivery quantities can be determined in a greedy way by fully serving the customers with the highest delivery quantity-related dual values. Therefore, during the labeling algorithm, Luo et al. [2016] do not keep track of the load of the vehicle and determine the delivery quantities afterwards.

Archetti et al. [2011] propose a different solution method for the SDVRP with both limited and unlimited fleets. The master problem is comparable to the one of Desaulniers [2010] but the pricing problem is approached completely different. The pricing problem is also modeled as an ESPPRC, but each customer is represented by multiple nodes; one for each possible delivery quantity. As a result, the number of nodes in the expanded network increases rapidly but the approach requires less complex dominance rules than Desaulniers [2010]. Similarly, Salani and Vacca [2011] and Archetti et al. [2015] use expanded networks to solve variants of the SDVRP.

Several variants of the SDVRP have been proposed in the literature and some of these variants attempt to prevent inconvenient situations for the customers for the SDVRP. Gulczynski et al. [2010] and Han and Chu [2016] consider the case with minimum delivery amounts, which reflect the fact that each delivery can be costly to both the distributor and the customer, hence, a delivery should be significant in terms of goods or value delivered. Ozbaygin et al. [2018] introduce for the SDVRP customer inconvenience constraints that set a maximum on the number of vehicles serving each customer with the reasoning that handling multiple deliveries is not desirable in practice. Bianchessi et al. [2019] apply these constraints to the SDVRPTW and moreover, consider two other types of customer inconvenience constraints. They consider a maximum total number of visits to all customers together and also temporal synchronization in which multiple visits can only be a certain number of time units apart from each other.

5.2.2 Literature on the VRPPC

The VRPPO is an extension of the VRPPC in which the service of a customer may be split between the two delivery types. The VRPPC was first proposed by Chu [2005] as a single depot routing problem with outsourcing options for which a simple heuristic based on a modified savings algorithm was developed. Later, some metaheuristics were introduced and these achieved very good results on a large set of test instances considering both homogeneous and heterogeneous private fleet. In particular, Bolduc et al. [2008] proposed a perturbation-based procedure, Côté and Potvin [2009] defined a tabu search approach, and Potvin and Naud [2011] used an ejection-chain neighborhood within another tabu search algorithm. More recently, Stenger et al. [2013] introduced a multiple-depot version of the problem, called MDVRPPC, for which they developed a variable neighborhood search algorithm incorporating an innovative adaptive shaking mechanism to select routes and customers involved in the shaking step.

To the best of our knowledge, just two exact methods currently exist for this problem. Dabia et al. [2019] developed a branch-cut-and-price algorithm for a variant of the VRPPC with heterogeneous fleet which included time windows and quantity discount on the outsourced deliveries. Goeke et al. [2018] propose a similar solution method for the VRPPC with customer-dependent, fixed fees as outsourcing cost. For more recent literature on the VRPPC we refer to the literature reviews in Gahm et al. [2017] and Dabia et al. [2019].

5.2.3 Literature on combined split delivery and outsourcing

Several studies consider the option to split deliveries over several shipment types, including private and common vehicles. Bolduc et al. [2010] study the so-called SDVRP with Production and Demand Calendars which is a multi-period, inventory routing-like problem in which a delivery to a customer can be split over multiple, both private and common, vehicles. The authors propose a tabu search heuristic to solve the problem. A compact formulation for the studied problem, which is discarded in this paper because of conciseness, would be very similar to a single-period formulation of the problem in Bolduc et al. [2010] except for the splitting over private vehicles.

A ship routing problem with pickup and deliveries is studied by Lee and Kim [2015]. It is possible to split the deliveries over multiple private vehicles and outsourcing (part of) the delivery to a so-called tramp ship is also possible. The problem is formulated as a Mixed Integer Linear Program in which the outsourcing costs are modeled customer dependent and proportional to the outsourced quantity. Having pickup and deliveries is the main difference with the VRPPO. An Adaptive Large Neighborhood Search heuristic is proposed to solve the problem.

Keskin et al. [2014] consider a practical application in which outbound shipment of products needs to be optimized. Three transportation modes are considered simultaneously with all vehicles belonging to outsourced carriers. Split deliveries between the different transportation modes is possible. For the so-called truckload mode, the routes also need to be determined since these are fully controlled by the company. The problem is split into an assignment and a routing problem, which are both solved using CPLEX.

Yan et al. [2015] study a multi-trip SDVRP problem with soft time windows in which not all customers necessarily have to be fully served. For each undelivered unit of demand a large penalty is incurred. This can also be seen as if delivery of these units is outsourced to a common carrier. The tests performed by Yan et al. [2015] are very limited; the authors speak about numerous test instances to assess the quality of the proposed heuristic but they only report detailed results on one instance based on real-life data. In the solution of the real-life instance all units of demand are delivered and the 'outsourcing' option is not used, probably caused by the huge penalty on not satisfying some demand (10,000 Taiwan New Dollar (TWD) per unit compared to costs of 12.53TWD per unit distance traveling cost). The two-step solution approach based on time-space networks could however be used to model (a variant of) the VRPPO by choosing the appropriate values for the parameters.

In summary, for problems that contain both split delivery and outsourcing features, to our knowledge, only heuristic solution methods have been proposed in the literature. The problem formulation of Bolduc et al. [2010] is closest to a formulation for our problem. For the SDVRP, problem variants have been proposed that, for example, limit the number of visits to a customer, with the motivation to limit customer inconvenience. In the VRPPO, we do not allow multiple private vehicles to service the same customer. This restriction limits customer inconvenience by preventing a customer being served by, e.g., two private vehicles and a common carrier. Moreover, this also substantially reduces the solution space leading to more efficient solution methods. Some solution methods developed for the SDVRP provide a starting point for our solution method for the VRPPO.

5.3 Problem Description and Formulation

The VRPPO is defined on a graph $G = (V, A)$ in which V is the set of nodes containing a depot 0 and a set of customers $V_0 = V \setminus \{0\}$ and A is the set of arcs. Each customer has a certain demand d_i that must be fulfilled and a time window $[e_i, l_i]$ in which service must take place. There is a set of private, heterogeneous vehicles available to serve the customers, each with a capacity $Q_k,\ k \in K$, with K the set of vehicle types. There are m_k vehicles available of vehicle type $k \in K$. Next to the private vehicles there is an option to outsource a delivery to a common carrier. In the VRPPO it is possible to have a customer both served by a private vehicle and by the common carrier. To limit the customer inconvenience it is not possible to have multiple private vehicles serve the same customer. This results in three service options for each customer: full private delivery, full outsourcing or a split delivery between one private vehicle and the common carrier.

Each arc (i, j) in graph G has an associated cost c_{ij} and a travel time t_{ij}. We assume that both the arc costs and travel times satisfy the triangle inequality. For each vehicle there is a set up cost f_k depending on the type of vehicle $k \in K$. For outsourcing, a fixed fee of v is charged per unit of outsourced demand. Using a fixed fee per unit implies that there is no benefit from outsourcing more units than strictly necessary because of flat rate or discount structures. Hence, for a split customer, as many units as possible are delivered by the private vehicle. Moreover, since the fee is customer independent, there is no difference between outsourcing the delivery to different customers. The objective is to minimize total routing and outsourcing costs whilst respecting time windows, vehicle capacity, and vehicle fleet limitations.

We propose two path-based master problem formulations for this problem in Sections 5.3.2 and Section 5.3.3 respectively, which are solved with a branch-and-price-and-cut solution approach. Both formulations potentially have an exponential number of columns, therefore, we start with a limited subset of initial columns and iteratively generate more columns by solving a pricing problem. The solution of the pricing problem is a set of columns with negative reduced costs. If this set is empty, either branching is necessary or an integer solution is found. The procedure results in the optimal solution if all branch-and-bound nodes have been explored. To strengthen the linear relaxation of the master problem Subset-Row (SR) inequalities are applied. These inequalities are introduced by Jepsen et al. [2008] for the VRPTW. The SR inequalities can immediately be applied to the first master problem; for the second master problem we use the generalized version as introduced by Dabia et al. [2019] for a Rich Vehicle Routing Problem with Private Fleet and Common Carrier.

5.3.1 Properties

For the SDVRP some properties have been established in the literature [Archetti and Speranza, 2012], however, these mainly have to do with interactions among routes that visit the same customers and are not applicable here since in the VRPPO the routes of private vehicles do not visit the same customers. In the pricing problem we do make use of the property in Lemma 5.1.

Lemma 5.1. *If the arc costs satisfy the triangle inequality, a route from the starting to the end depot contains at most one customer whose demand is split between the private vehicle and common carrier.*

The lemma can be proved by a simple exchange argument. Suppose that a route of a private vehicle contains a split delivery for two different customers i and j and that the vehicle's total load is equal to the vehicle capacity. By delivering more units to customer i with the private vehicle and outsourcing less units for customer i, and vice versa for customer j, and continuing this exchange until no units are delivered to customer j by the private vehicle, this exchange process results in a zero delivery by the private vehicle to customer j. Because of the triangle inequality, it is more expensive to visit customer j with the private vehicle (with a zero delivery) than not to visit customer j. Hence, such a route, with one split delivery, is never more expensive than the same route that contains two split deliveries. Therefore, this lemma implies that during execution of the labeling algorithm, in each partial path, there has to be at most one customer that does not receive its full demand by the private vehicle.

5.3.2 Master problem 1: MP1

In the first master problem (MP1) a column represents a route visiting a set of customers and the corresponding delivery quantities delivered by the private vehicle. Let Ω_k be the set of routes p and associated delivery quantities for vehicle type $k \in K$ and let $\Omega = \bigcup_{k \in K} \Omega_k$. Let c_p be the routing cost of route $p \in \Omega$. Associate δ_i^p and a_{ip} with a route $p \in \Omega$ representing the delivery quantity for customer $i \in V$ and the number of times customer $i \in V$ is in the route respectively. Define binary decision variables y_p which indicate whether route $p \in \Omega$ is in the solution of MP1 and continuous decision variables β_i being the demand of customer $i \in V$ that is outsourced. The VRPPO can be formulated as follows:

$$\min \sum_{p\in\Omega} c_p y_p + \sum_{i\in V} v\beta_i \tag{5.1a}$$

$$\text{s.t.} \sum_{p\in\Omega} \delta_i^p y_p + \beta_i \geq d_i, \forall i \in V, \tag{5.1b}$$

$$\sum_{p\in\Omega} a_{ip} y_p \leq 1, \qquad \forall i \in V, \tag{5.1c}$$

$$\sum_{p\in\Omega_k} y_p \leq m_k, \qquad \forall k \in K, \tag{5.1d}$$

$$\beta_i \geq 0, \qquad \forall i \in V, \tag{5.1e}$$

$$y_p \in \{0,1\}, \qquad \forall p \in \Omega. \tag{5.1f}$$

The objective function (5.1a) minimizes the routing and outsourcing costs. Constraints (5.1b) make sure that the demand of each customer is satisfied, by a delivery of a private vehicle, via outsourcing or a combination of these. We do not allow for multiple visits by a private vehicle to a customer, i.e., no private split delivery, which is enforced by constraints (5.1c). Constraints (5.1d) limit the number of vehicles used per type and the domains of the decision variables are defined in constraints (5.1e) and (5.1f).

The pricing problem should generate columns defining a route and delivery quantities, in which at most one customer does not receive its full demand. Associate dual variables $\pi_i^{5.1b} \geq 0$, $\pi_i^{5.1c} \leq 0$, $\pi_k^{5.1d} \leq 0$ with constraints (5.1b)-(5.1d) respectively. Define $\pi_0^{5.1b} = 0$ and $\pi_0^{5.1c} = 0$ for the depot. Let $\boldsymbol{\delta}^p$ be the vector of delivery quantities for the customers in route p. The reduced cost of a column for route p associated with vehicle type $k \in K$ can be expressed as follows:

$$\bar{c}_p(\boldsymbol{\delta}^p) = f_k + \sum_{(i,j)\in A} \left(c_{ij} - \pi_i^{5.1c}\right) x_{ijp} - \sum_{i\in V^P} \delta_i^p \pi_i^{5.1b} - \pi_k^{5.1d}, \tag{5.2}$$

in which x_{ijp} is an integer variable counting the number of times arc (i,j) is traversed in route p and V^P is the set of nodes in the route. Note that the reduced cost of a column depends on the delivery quantity to each customer in the route. For ease of notation, define the indicator function $\mathbb{1}\{event\} = 1$ if *event* is true, 0 otherwise, and define

$$\bar{c}_{ij} = c_{ij} - \pi_i^{5.1c} + \mathbb{1}\{i = 0\} \left(f_k - \pi_k^{5.1d}\right), \tag{5.3}$$

which gives reduced arc costs in which the vehicle set up and dual costs are accounted for in the outgoing arcs of the depot.

5.3.2.1 Pricing algorithm 1 for MP1: MP1-PA1

The pricing problem is a variant of the Elementary Shortest Path Problem with Resource Constraints (ESPPRC) which we solve with a labeling algorithm (see e.g. Feillet et al. [2004], Righini and Salani [2006], Tilk et al. [2017]). Since the pricing problem for the VRPPO is similar to the one for the SDVRPTW, we first adjust the labeling algorithm proposed for the SDVRPTW by Desaulniers [2010] to our problem (MP1-PA1). Desaulniers [2010] proposes to create up to three labels for each extension of a partial path in which the delivery quantity to the next customer is either zero, full (equal to the demand), or partial (strictly between zero and the demand). Subsequently, in the master problem, any delivery quantity pattern for a route is created by taking convex combinations of the columns. To create any combination of delivery quantities, the zero deliveries are necessary. Desaulniers [2010] observes that only so-called extreme delivery patterns, which contain at most one partial delivery, are needed. In the SD-VRPTW multiple private vehicles can visit the same customer, therefore, routes are interdependent and the actual delivery quantity cannot be determined in the pricing problem. Contrary, in the VRPPO this interdependency is not present; the delivery quantities can, therefore, be determined in the pricing problem. Consequently, we do not take convex combinations of columns in MP1 and hence, in the pricing algorithm for the VRPPO the zero delivery option is not needed. By Lemma 5.1, only routes with at most one partial delivery need to be created.

Therefore, for the VRPPO, when extending a partial path to a customer j, up to two labels can be created; one label in which customer j is fully delivered by the private vehicle and one label in which the demand of customer j will be partially delivered by the private vehicle and partially outsourced. Note that it is never optimal to fully outsource the demand of a visited customer, since in that case it is more efficient to not visit the customer at all. The outsourced part of the demand of the split customer is determined by the vehicle capacity and the demand of the other customers in the route which means that we can only determine the delivery quantity when the route is complete. Since the delivery quantity δ_i^p for the split customer i is not known during the labeling algorithm, the reduced cost of a partial path p as in function (5.3) cannot contain the contribution of this split delivery until reaching the end node. Therefore, during the labeling algorithm we keep track of the maximum reduced cost of the partial path. This is equal to the case in which no units are delivered to the split customer by the private vehicle.

Let a label L corresponds to a partial path $p(L)$ in the graph G starting at the depot. For a type-k vehicle, associate the following attributes with a label L:

$i(L)$	Last node visited in partial path $p(L)$,
$c(L)$	Maximum reduced cost of partial path $p(L)$ (i.e., no units to split customer),
$q(L)$	Load of full deliveries in partial path $p(L)$,
$t(L)$	Ready time at node $i(L)$ when reached through partial path $p(L)$,
$r(L)$	Customer with split in path $p(L)$, -1 if no split in $p(L)$,
$\phi(L)$	The maximum quantity delivered in the split delivery by the private vehicle in partial path $p(L)$, if any $(\phi(L) = 0$ if $r(L) = -1)$,
$V(L)$	Set of visited nodes along path $p(L)$.

Furthermore, let $\overline{V}(L)$ denote the set of visited and unreachable nodes. Nodes are unreachable if they cannot be visited by extending the path $p(L)$ because of time windows. Also, if a split delivery is already in the path and visiting a customer $j \in V_0$ would violate vehicle capacity, even by setting the split delivery to zero, then customer j is unreachable. Note that this differs from what can be assumed for the SDVRP (e.g. Desaulniers [2010]), since for the VRPPO we do not have to consider extreme delivery patterns that contain zero deliveries.

Suppose we extend a label L' along arc $(i(L'), j)$ to node $j \in V \setminus \overline{V}(L')$ to generate a new label L. The resources for the new label L are established as follows:

$$\begin{aligned}
i(L) &= j,\\
c(L) &= \begin{cases} c(L') + \bar{c}_{ij} - d_j\pi_j^{5.1b} & \text{if no split delivery at } j,\\ c(L') + \bar{c}_{ij} & \text{if a split delivery at } j, \end{cases}\\
q(L) &= \begin{cases} q(L') + d_j & \text{if no split delivery at } j,\\ q(L') & \text{if a split delivery at } j, \end{cases}\\
t(L) &= \max\left\{t(L') + t_{ij}, e_j\right\},\\
r(L) &= \begin{cases} r(L') & \text{if no split delivery at } j,\\ j & \text{if a split delivery at } j, \end{cases}\\
\phi(L) &= \begin{cases} \min\{\phi(L'),\ Q - q(L)\} & \text{if no split delivery at } j,\\ \min\{d_j,\ Q - q(L)\} & \text{if a split delivery at } j, \end{cases}\\
V(L) &= V(L') \cup \{j\}.
\end{aligned}$$

An extension from label L' to label L with $i(L) = j$ is feasible if $j \notin \overline{V}(L')$, $q(L) \leq Q$, $e_j \leq t(L) \leq l_i$, $\phi(L) \geq 0$. Note that a label with a split delivery for customer j cannot be created if $r(L') > -1$ since a split is already in the path.

The potential number of labels is huge. Therefore, to discard labels during the algorithm sufficient dominance conditions are formulated. If $i \in V$ is the split customer, then the reduced cost of a label is a linear function in δ_i. Therefore, to compare two labels, the dominance criteria need to compare two linear functions. Since the linear functions have a limited domain, two line segments have to be considered to compare the reduced costs of two labels. Specifically, the dominance conditions must be able to handle the comparison of line segments which are restricted in domain by the quantity delivered to the split customer (between zero and $\phi(L)$), and in range by $c(L)$ and the minimum reduced cost that can be reached given $\phi(L)$ which is $c(L) - \phi(L)\pi^{5.1b}$. In Desaulniers [2010] the same issue is encountered for the SDVRPTW and the authors propose the following sufficient dominance conditions to establish whether label L_1 dominates label L_2 associated with the same node:

(A1) $t(L_1) \leq t(L_2)$;
(A2) $q(L_1) \leq q(L_2)$;
(A3) $\mathbb{1}\{r(L_1) > -1\} \leq \mathbb{1}\{r(L_2) > -1\}$;
(A4) $\overline{V}(L_1) \subseteq \overline{V}(L_2)$;
(A5) $c(L_1) - \phi(L_1)\pi^{5.1b}_{r(L_1)} \leq c(L_2) - \phi(L_2)\pi^{5.1b}_{r(L_2)}$;
(A6) $c(L_1) - (q(L_2) - q(L_1))\,\pi^{5.1b}_{r(L_1)} \leq c(L_2)$;
(A7) $c(L_1) - (q(L_2) + \phi(L_2) - q(L_1))\,\pi^{5.1b}_{r(L_1)} \leq c(L_2) - \phi(L_2)\pi^{5.1b}_{r(L_2)}$;

in which condition (A5) compares the minimums of both segments, and conditions (A6) and (A7) compare the costs of both paths at the lowest and highest load of the path in label L_2 respectively. These conditions prevent comparing crossing line segments; for details we refer to Desaulniers [2010].

5.3.2.2 Pricing algorithm 2 for MP1: MP1-PA2

In MP1-PA1 (Section 5.3.2.1) in each label extension both a label with and without a split are explicitly created with at most one split per path. Hence, when a label with a split is created, the split customer is immediately determined for the resulting route. However, knowing the split customer is not necessary, it is possible instead to decide which customer to split when extending a path to the end node. Then, only a path with a total delivery quantity larger than the vehicle capacity will have a split and any customer with a sufficiently high demand can be the split customer. The demand of a customer is sufficiently high if the demand is higher than the vehicle capacity shortage given the total demand in the path. For example, if $Q = 25$ and the total demand is 30, the capacity shortage is 5 units and splitting a customer with demand 3 does not give a feasible path. Luo et al. [2016] come to a similar insight for the SDVRPTW. However, as explained in Section 5.3.2.1, also zero deliveries are necessary and therefore, Luo et al. [2016] do not keep track of the vehicle capacity and hence, are not able to exclude paths because vehicle capacity is violated. This leads to many very long routes, while by keeping track of the load we can prevent many inefficiently long paths that would have many zero deliveries or, in our case, many outsourced units.

This leads us to a different solution method for the pricing problem (MP1-PA2). Instead of creating the split explicitly at a node during the label extension, a partial path is extended until it exceeds the vehicle capacity, after which a split is definitely needed. After exceeding the vehicle capacity, only customers with a small demand that respect an additional constraint can still be added to the path. To explain the additional constraint, consider Figure 5.1. Suppose the vehicle capacity is $Q = 25$ and there is a partial path depot-1-2-3 which exceeds the vehicle capacity with the extension to customer 3. The demand of each customer is indicated in the figure. The vehicle capacity is already exceeded by 5 units which have to be outsourced. Consider three possible extensions to customers 4, 5 and 6, respectively, with the indicated corresponding demands.

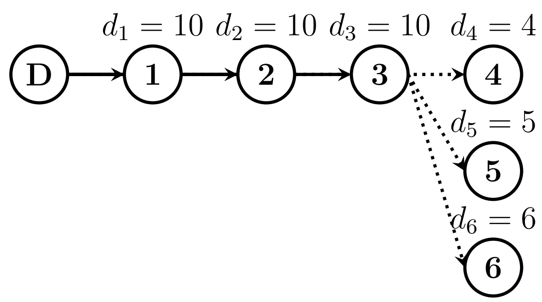


Figure 5.1 Example possible label extensions.

Suppose an extension is made to customer 6, then the total load will be 36 units and at least 11 units need to be outsourced. This implies that one customer's demand will be fully outsourced and also one unit of another customer, hence, this path is not efficient. The same holds for an extension to customer 5, since exactly one customer's demand will be fully outsourced in which case it is better not to visit the customer at all in this path. On the contrary, the extension to customer 4 is potentially efficient since the total load becomes 34 and, therefore, 9 units have to be outsourced. Note that by Lemma 5.1 and the maximum demand in the path of 10, at most 9 units can be outsourced. Hence, we make use of the customer with the highest demand currently in the path to determine whether an extension is still possible. Concluding, the additional constraint is that the total demand after adding a customer may not exceed vehicle capacity plus the highest demand currently in the path. Moreover, the quantity dedicated to this customer that can still be delivered by the private vehicle can be computed by 'vehicle capacity - current load + highest demand' $= 25 - 34 + 10 = 1$.

Therefore, instead of explicitly creating labels with a split delivery, we propose an alternative labeling algorithm, in which the split customer and corresponding delivery quantity are determined in a post-processing step such that reduced costs are minimized. Note that if vehicle capacity is not exceeded, all units are delivered by the private vehicle. In the post-processing step, we know how many units need to be outsourced (the shortage in vehicle capacity). Then it remains to decide which customer's demand to split. Jin et al. [2008] have a similar issue for a variant of the SDVRP in which the number of vehicles to use is fixed upfront and they noted that it is best to split the demand of the customer with the smallest dual variable. Which customer to split in a path follows from Lemma 5.2 which can be proved by a simple interchange argument.

Lemma 5.2. *In an optimal solution to the pricing problem, each visited customer receives its full demand except for, at most, one customer with the smallest dual variable and demand higher than the shortage in vehicle capacity.*

Let a label L correspond to a partial path $p(L)$ in the graph G starting at the depot. For a type-k vehicle, associate the following attributes with a label L:

$i(L)$	Last node visited in partial path $p(L)$,
$c(L)$	Minimum reduced cost of partial path $p(L)$,
$q(L)$	Load in partial path $p(L)$,
$t(L)$	Ready time at node $i(L)$ when reached through partial path $p(L)$,
$s(L)$	Indicates whether or not a split delivery is necessary in partial path $p(L)$, if any $(\phi(L) = 0 \text{ if } s(L) = 0)$,
$d_{\max}(L)$	Maximum demand over the customers in path $p(L)$, updated until vehicle capacity is exceeded,
$V(L)$	Set of visited nodes along path $p(L)$.

For $d_{\max}(L)$ it is important to note that this value is no longer updated when vehicle capacity is already exceeded. Furthermore, let $\overline{V}(L)$ again denote the set of visited and unreachable nodes. Nodes can be unreachable because of time windows or vehicle capacity: an unreachable node j has one of the following properties $t(L) + t_{i(L),j} > l_j$ or $s(L) = 1$ and $d_j > Q + d_{\max}(L) - q(L)$. If necessary, the customer with the lowest dual that has a sufficiently high demand is split. Hence, the best candidate customer changes during the labeling algorithm and the actual reduced cost of a path is not known during the labeling algorithm. Therefore, in the reduced cost $c(L)$, we account for all visited customers the full demand d_i and subtract $d_i \pi_i^{5.1b}$. This results in keeping track of the minimum possible reduced cost of path $p(L)$. Note that in case of a split delivery, this value $c(L)$ is lower than the actual reduced cost since the dual is subtracted for too many units of demand. This needs to be accounted for when applying dominance rules. Also, at the end of the labeling algorithm the correct reduced cost has to be computed for a column to decide whether adding it is actually efficient.

Suppose we extend a label L' along arc $(i(L'), j)$ to node $j \in V \setminus \overline{V}(L')$ to generate a new label L. The resources for the new label L are established as follows:

$$\begin{aligned}
i(L) &= j,\\
c(L) &= c(L') + \bar{c}_{ij} - d_j \pi_j^{5.1b},\\
q(L) &= q(L') + d_j,\\
t(L) &= \max\{t(L') + t_{ij}, e_j\},\\
s(L) &= s(L') + 1 \text{ , if } \; q(L') < Q \wedge q(L) > Q,\\
d_{\max}(L) &= \begin{cases} \max\{d_{\max}(L), d_j\} & \text{if } s(L') = 0,\\ d_{\max}(L') & \text{if } s(L') = 1,\end{cases}\\
V(L) &= V(L') \cup \{j\}.
\end{aligned}$$

An extension from label L' to label L with $i(L) = j$ is feasible if $j \notin \overline{V}(L')$ and $e_j \leq t(L) \leq l_i$. To discard labels, sufficient dominance conditions are formulated. If a path needs a split delivery, the reduced cost is a function of the quantity delivered to the split customer, with a multiplication factor equal to the corresponding dual variable. On the one hand when deciding which customer to split, in a greedy way customers with the highest dual values will be served by the private vehicle since this results in the lowest reduced cost. On the other hand, the split customer must have sufficiently high demand (higher than the shortage in vehicle capacity) to obtain a feasible route. The dominance rules, again, must compare segments of reduced cost functions but in this case it is not immediately clear what the slope and ranges of these segments are. Therefore, to decide whether a label L_1 dominates L_2, we consider the 'worst' case in terms of cost and range for L_1 and the 'best' case for L_2, which corresponds to the situation where L_1 is least likely to dominate L_2. For the worst case for L_1, determine the highest dual value over the customers in the path that have a sufficiently high demand, i.e., $\hat{\pi}_1 = \max_{i \in V(L_1): d_i > q(L_1) - Q} \pi_i^{5.1b}$. Similarly, for L_2 determine the lowest dual value, i.e., $\check{\pi}_2 = \min_{i \in V(L_2): d_i > q(L_2) - Q} \pi_i^{5.1b}$. Define $\hat{\pi}_1 = 0$ and $\check{\pi}_2 = 0$ if a split is not necessary in the path. Note that if an extension of L_2 contains a customer with a lower dual value, this value cannot be lower than the lowest dual value of the extension of L_1 in which case the slopes of both reduced cost functions are the same. Moreover, for both labels, the demand of the 'split' customer is set to the highest demand of a customer in each path ($d_{\max}(L)$).

Define $\hat{q}(L) = q(L) - \mathbb{1}\{s(L) = 1\} d_{\max}(L)$, $\hat{c}(L_1) = c(L_1) + \mathbb{1}\{s(L_1) = 1\} d_{\max}(L_1)\hat{\pi}_1$ and $\hat{c}(L_2) = c(L_2) + \mathbb{1}\{s(L_2) = 1\} d_{\max}(L_2)\check{\pi}_2$. Also, compute for both labels the maximum quantity delivered by the private vehicle of the maximum demand, i.e., $\phi(L) = Q - q(L) + d_{\max}(L)$. The line segment of the reduced cost function extends from a zero delivery to a delivery of $\phi(L)$ units to the split customer, i.e., has a domain of zero to $\phi(L)$. See Figure 5.2 for an example of segments for labels L_1 and L_2.

Consider conditions (B1)-(B7) to dominate label L_2 by label L_1 that correspond to the same customer:

(B1) $t(L_1) \leq t(L_2)$;
(B2) $\hat{q}(L_1) \leq \hat{q}(L_2)$;
(B3) $s(L_1) \leq s(L_2)$;
(B4) $\overline{V}(L_1) \subseteq \overline{V}(L_2)$;
(B5) $\hat{c}(L_1) - \phi(L_1)\hat{\pi}_1 \leq \hat{c}(L_2) - \phi(L_2)\check{\pi}_2$;
(B6) $\hat{c}(L_1) - (\hat{q}(L_2) - \hat{q}(L_1))\, \hat{\pi}_1 \leq \hat{c}(L_2)$;
(B7) $\hat{c}(L_1) - (\hat{q}(L_2) + \phi(L_2) - \hat{q}(L_1))\, \hat{\pi}_1 \leq \hat{c}(L_2) - \phi(L_2)\check{\pi}_2$.

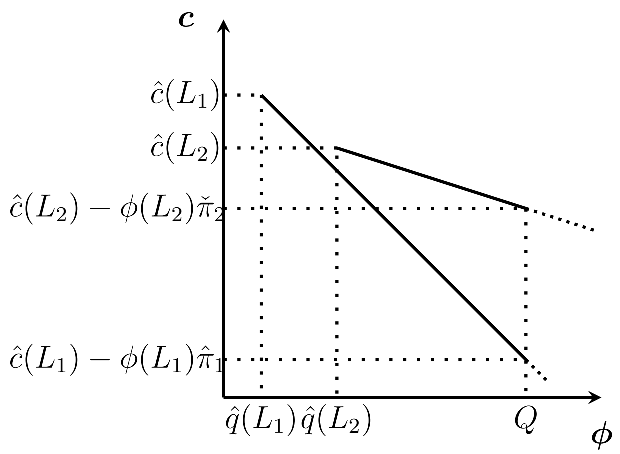


Figure 5.2 Reduced cost as a function of the delivered quantity

Proposition 2. *Conditions (B1)-(B7) are sufficient conditions to dominate label L_2 by label L_1.*

Proof. Any extension of label L_2 is feasible for label L_1 by conditions (B1)-(B4), in which (B1), (B2) and (B4) are the same as in Desaulniers [2010]. Note that in case of a split delivery in both labels L_1 and L_2, condition (B2) considers the load without the maximum demand, which allows for splitting the customer with the highest demand in extensions of both paths.

If neither labels has a split delivery, the dominance conditions are the same as in Desaulniers [2010]. If both labels have a split delivery, for label L_1, $d_{\max}(L_1)$ units are outsourced maximally. If the corresponding customer would have dual value $\hat{\pi}_1$, the reduced cost would be highest and the chance of dominating label L_2 is lowest. Therefore, we create the artificial segment for label L_1 from $(\hat{q}(L_1),\ \hat{c}(L_1))$ to $(Q = \hat{q}(L_1) + \phi(L_1),\ \hat{c}(L_1) - \phi(L_1)\hat{\pi}_1)$. Note that the choice of customer demand to create the artificial segment does not impact the lowest point of the slope which is used in condition (B5).

For label L_2, with a similar reasoning except for using the lowest possible reduced cost, we create artificial segment from $(\hat{q}(L_2),\ \hat{c}(L_2))$ to $(Q = \hat{q}(L_2) + \phi(L_2),\ \hat{c}(L_2) - \phi(L_2)\breve{\pi}_2)$. The dominance conditions by Desaulniers [2010] are applied to these artificial segments ((B5)-(B7)). By construction, using these artificial segments make it least likely that label L_1 will dominate L_2, and hence, if dominance is established by conditions (B1)-(B7) label L_2 will not result in a better path which completes the proof. $\square$

Note that each resulting path in the labeling algorithm now results in, at most, one column with negative reduced cost for MP1 since in the end we only split the customer with the lowest dual value. One could create multiple columns by splitting different customers if the corresponding dual variables still lead to a negative reduced cost column. We do not generate multiple columns since the cost of outsourcing is the same for each customer, therefore the objective value of MP1 remains the same no matter which customer receives a split delivery.

5.3.2.3 Subset-Row inequalities

SR inequalities are introduced by Jepsen et al. [2008] for the VRPTW and are Chvátal-Gomory rank 1 cuts. We can apply these valid inequalities to MP1. Remind that a_{ip} is the number of times customer i is visited in $p \in \Omega$. The SR inequalities for a subset of nodes $S \subseteq V$ and an integer $0 < \kappa \leq |S|$ can be formulated as follows:

$$\sum_{p \in \Omega} \left\lfloor \frac{1}{\kappa} \sum_{i \in S} a_{ip} \right\rfloor y_p \leq \left\lfloor \frac{|S|}{\kappa} \right\rfloor . \tag{5.4}$$

Separation of these valid inequalities is NP-complete. As suggested by e.g., Jepsen et al. [2008], we enumerate all inequalities for subsets of customers of size three ($|S| = 3$) with $\kappa = 2$. As clearly stated by Spliet and Desaulniers [2015], these valid inequalities ensure that for every triplet of customers, at most one route can be selected that contains more than one of these customers. Since these inequalities are defined on the master problem variables, they change the structure of the pricing problem. Let $\xi_I < 0$ be the dual variable corresponding to a valid inequality of type (5.4) for subset $S_I \subseteq V$. If a column is generated that contributes to the valid inequality, i.e., for every κ customers in S_I visited in the path, ξ_I is subtracted from the reduced cost of the path. However, only when ending a path in the pricing problem (extend to the end node), one knows exactly what the contribution of the valid inequalities to the reduced cost is. As described in Jepsen et al. [2008], the contribution of the SR inequalities can be accounted for in the costs to handle the SR inequalities in the dominance conditions.

5.3.3 Master problem 2: MP2

In MP1, the quantity delivered by the private vehicle to each visited customer in a route is decided upon in the pricing problem. The outsourced quantities are handled in the master problem by the β-variables. However, given a route with a split delivery retrieved from the pricing problem, the number of units delivered by the private vehicle to each customer is already known and hence, the number of units that are outsourced and the corresponding costs. Therefore, we introduce master problem 2 (MP2) in which the (outsourced) delivery quantities are not modeled explicitly via decision variables. For each customer there are two options: either it is visited by a route and from the pricing problem we know both the private and outsourced delivery quantities, or the customer is not visited at all and demand is fully outsourced. The latter case should be explicitly modeled in MP2.

Let Λ_k be the set of routes and associated delivery quantities for vehicle type $k \in K$ and outsourcing quantities and let $\Lambda = \bigcup_{k \in K} \Lambda_k$. $p \in \Lambda$ represents a route visiting a set of customers, the corresponding delivery quantities by the private vehicle, and the outsourced quantity ζ_p. Note that for a column p, it is not required to know to which customer ζ_p belongs. Let c_p, $p \in \Lambda$, be the total routing and outsourcing cost ($\zeta_p v$). Associate parameter a_{ip} with $p \in \Lambda$ representing the number of visits to customer $i \in V$ in the route. Define binary decision variables z_p which indicate whether $p \in \Lambda$ is in the solution of MP2 and define non-negative decision variables w_i indicating whether a customer $i \in V$ is fully outsourced or not. MP2 is formulated as follows:

$$\min \sum_{p \in \Lambda} c_p z_p + \sum_{i \in V} w_i d_i v \tag{5.5a}$$

$$\text{s.t.} \sum_{p \in \Lambda} a_{ip} z_p + w_i = 1, \quad \forall i \in V, \tag{5.5b}$$

$$\sum_{p \in \Lambda_k} z_p \leq m_k, \qquad \forall k \in K, \tag{5.5c}$$

$$w_i \geq 0, \qquad \forall i \in V_0, \tag{5.5d}$$

$$z_p \in \{0, 1\}, \qquad \forall p \in \Lambda. \tag{5.5e}$$

The objective function (5.5a) minimizes total routing and outsourcing costs, in which the outsourcing costs consist of the costs for split customers given by the pricing problem and of the costs for customers of which demand is fully outsourced. Constraints (5.5b) make sure that a customer is either visited by a private vehicle, potentially with a split delivery, or the customer's demand is fully outsourced. Constraints (5.5c) limit the number of vehicles used per type and the domains of the decision variables are restricted in constraints (5.5d) and (5.5e). Note that the w variables represent a binary decision but can be added to MP2 as non-negative variables because, if the z variables are integer, the w variables must be integer as well.

Associate dual variables $\mu_i^{5.5b} \in \mathbb{R}$ and $\mu_k^{5.5c} \leq 0$ with constraints (5.5b) and (5.5c) respectively. The reduced cost of a column $p \in \Lambda$ for a vehicle of type $k \in K$ can be expressed as follows:

$$\bar{c}_p = f_k + \zeta_p v + \sum_{(i,j) \in A} \left(c_{ij} - \mu_i^{5.5b} \right) x_{ijp} - \mu_k^{5.5c} \tag{5.6}$$

in which ζ_p is a non-negative variable being the quantity outsourced and x_{ijp} are integer variables counting the number of times arc (i, j) is traversed in the route of $p \in \Lambda$. For ease of notation, define

$$\bar{c}_{ij} = c_{ij} - \mu_i^{5.5b} + \mathbb{1}\{i = 0\} \left(f_k - \mu_k^{5.5c} \right), \tag{5.7}$$

in which $\mu_0^{5.5b} = 0$ and the vehicle costs are accounted for in the outgoing arcs of the depot.

5.3.3.1 Pricing algorithm 1 for MP2: MP2-PA1

The first pricing algorithm for MP2 (MP2-PA1) is based on the same reasoning as MP1-PA1 as described in Section 5.3.2.1, therefore, we only highlight the differences here. Note that there is a significant difference in the reduced cost functions between the two master problems. For MP1 the reduced cost is a function of the quantities delivered by the private vehicle while for MP2 the reduced cost is a function of the outsourced quantity. Accordingly, for MP1-PA1, the dual variable $\pi_i^{5.1b}$ should be *deducted* from the reduced cost for each privately delivered unit but this number of units is unknown for the split customer during the labeling algorithm. Conversely, for MP2-PA1, the outsourcing cost v should be *added* to the reduced cost for each outsourced unit but the outsourced quantity is unknown during the labeling algorithm.

MP2-PA1 uses resources $i(L)$, $q(L)$, $t(L)$, $r(L)$, $\phi(L)$, $V(L)$ and $\overline{V}(L)$ as defined for MP1-PA1 in Section 5.3.2.1. Only $c(L)$, the reduced cost, is redefined. The number of outsourced units cannot be established until the path is finished. Moreover, the reduced cost is lowest if all units are delivered by the private vehicle (no outsourcing costs). Therefore, during this labeling algorithm we keep track of the minimum possible reduced cost. This gives the following definition of the resource

$c(L)$ Minimum reduced cost of partial path $p(L)$ (i.e., all units delivered by private vehicle),

which is updated as follows for an extension of a label L' along arc $(i(L'), j)$ to a node $j \in V \setminus V(L')$ to generate a new label L

$$c(L) \quad = c(L') + \bar{c}_{ij}.$$

To reduce the number of labels, dominance rules can be applied to discard labels. The reduced cost of a path depends on the outsourced quantity, or equivalently, on the quantity delivered by the private vehicle to the split customer. Then we can express the reduced cost of a path in label L as a function of $\phi(L)$ as follows

$$\bar{c}_p = f_k + \left(d_{r(L)} - \phi(L)\right) v + \sum_{(i,j) \in A} \left(c_{ij} - \mu_j^{5.5b}\right) x_{ijp} - \mu_k^{5.5c}, \tag{5.8}$$

in which x_{ijp} again indicates the number of times arc (i, j) is in the path and recall that $r(L)$ is the split customer. The functional form of the rewritten reduced cost function is the same as for MP1. Therefore, to apply dominance rules to compare labels, again we have to compare segments of reduced cost functions. Similarly to Desaulniers [2010] and MP1, sufficient dominance conditions for dominance of label L_1 over L_2 associated with the same node are given by

(C1) $t(L_1) \leq t(L_2)$;
(C2) $q(L_1) \leq q(L_2)$;
(C3) $s(L_1) \leq s(L_2)$;
(C4) $\overline{V}(L_1) \subseteq \overline{V}(L_2)$;
(C5) $c(L_1) + d_{s(L_1)}v - \phi(L_1)v \leq c(L_2) + d_{r(L_2)}v - \phi(L_2)v$;
(C6) $c(L_1) + d_{s(L_1)}v - (q(L_2) - q(L_1))\, v \leq c(L_2) + d_{r(L_2)}v$;
(C7) $c(L_1) + d_{s(L_1)}v - (q(L_2) + \phi(L_2) - q(L_1))\, v \leq c(L_2) + d_{r(L_2)}v - \phi(L_2)v$;

in which $d_r(L) = 0$ and $\phi(L) = 0$ if there is no split delivery in the path at label L. In conditions (C5), (C6) and (C7) the minimum reduced cost is increased with the maximum outsourcing cost and subsequently, the unit outsourcing cost is deducted for the units that are not outsourced. Additionally, note that the slopes of the compared segments are equal (v), therefore, we can discard condition (C7) since it is redundant with conditions (C2), (C5) and (C6).

5.3.3.2 Pricing algorithm 2 for MP2: MP2-PA2

As for MP1-PA2 in Section 5.3.2.2, also for MP2 it can be observed that knowing which customer to split is not necessary during the labeling algorithm. Moreover, for MP2, even in the master problem this information is not necessary since only information on which customers are visited is needed. Therefore, MP1-PA2 can be adjusted for MP2 (MP2-PA2) and, again, we only highlight the differences here.

MP2-PA2 uses resources $i(L)$, $c(L)$, $q(L)$, $t(L)$, $s(L)$, $d_{\max}(L)$, $V(L)$ and $\overline{V}(L)$ as defined for MP1-PA2 in Section 5.3.2.2. Additionally $\phi(L)$ is defined as follows:

$\phi(L)$ The maximum quantity delivered to a split customer by the private vehicle in partial path $p(L)$, assuming that the customer with the highest demand is split.

Although the definition of $c(L)$ is the same, the computation is different because of the different reduced cost function. Therefore, for an extension of a label L' along arc $(i(L'), j)$ to node $j \in V \setminus V(L')$ to generate a new label L, resources $c(L)$ and $\phi(L)$ are updated as follows:

$$\begin{aligned} c(L) &= c(L') + \bar{c}_{ij}, \\ \phi(L) &= Q - q(L) + d_{\max}(L). \end{aligned}$$

To apply dominance, a similar reduced cost function as for MP2-PA1 can be used in which $d_{s(L)}$ is replaced by $d_{\max}$. Again segments of reduced cost functions have to be compared. Define $\hat{q}(L) = q(L) - \mathbb{1}\{s(L) = 1\}d_{\max}(L)$ and $\hat{c}(L) = c(L) + \mathbb{1}\{s(L) = 1\}d_{\max}(L)v$. Sufficient dominance conditions for dominance of label L_1 over L_2 associated with the same node are given by

(D1) $t(L_1) \leq t(L_2)$;
(D2) $\hat{q}(L_1) \leq \hat{q}(L_2)$;
(D3) $s(L_1) \leq s(L_2)$;
(D4) $\overline{V}(L_1) \subseteq \overline{V}(L_2)$;
(D5) $\hat{c}(L_1) - \phi(L_1)v \leq \hat{c}(L_2) - \phi(L_2)v$;
(D6) $\hat{c}(L_1) - (\hat{q}(L_2) - \hat{q}(L_1))\, v \leq \hat{c}(L_2)$;
(D7) $\hat{c}(L_1) - (\hat{q}(L_2) + \phi(L_2) - \hat{q}(L_1))\, v \leq \hat{c}(L_2) - \phi(L_2)v$.

These conditions are similar to the conditions for MP1-PA2. Since the slope of the segments is equal for both labels, condition (D7) is redundant analogous to MP2-PA1.

5.3.3.3 Generalized Subset-Row inequalities

As in Dabia et al. [2019], for MP2 we can still apply the SR inequalities as described in Section 5.3.2.3, however, these cuts do not make use of the fact that part of the demand can be outsourced to the common carrier. Dabia et al. [2019] propose so-called Generalized SR inequalities for this variant of the problem. These can be applied to the MP2 as well since constraint (5.5b) coincides with the first constraint of the second formulation in Dabia et al. [2019].

Let $d(S) = \sum_{i \in S} d_i$ be the sum of the demand for subset of customers $S \subseteq V_0$. Dabia et al. [2019] observe that splitting $d(S)$ in packages of size κ, will result in a number of privately delivered packages in an integer solution that is less than the total number of packages minus the outsourced packages of the customers in S. For a subset of customers $S \subseteq V_0$ and $0 < \kappa < d(S)$, they propose the following valid inequalities:

$$\sum_{p \in \Omega} \left\lfloor \frac{1}{\kappa} \sum_{i \in S} d_i a_{ip} \right\rfloor z_p + \sum_{i \in S} \left\lfloor \frac{d_i}{\kappa} \right\rfloor w_i \leq \left\lfloor \frac{d(S)}{\kappa} \right\rfloor. \tag{5.9}$$

In Dabia et al. [2019] two propositions are given to establish dominance rules that take the Generalized SR inequalities into account, in which the second proposition contains strengthened dominance rules compared to the first proposition. If the Generalized

SR cuts are combined with ng-paths (see Section 5.4.2), it is not possible to use the strengthened version since it makes use of the demand of the set of reachable customers which is changing during the algorithm when ng-paths are used. Therefore, similar dominance conditions as in the first proposition in Dabia et al. [2019] are used for the VRPPO.

5.3.4 Model extensions

In the above model we assumed that the costs of outsourcing is a customer-independent fixed fee per unit v. Although we designed the four solution algorithms to solve the VRPPO for a fixed fee per unit cost structure, several of our algorithms can be easily adjusted for handling alternative cost structures. In particular, for MP1-PA1, MP1-PA2 and MP2-PA1 a customer-dependent cost structure, in which the fixed fee per unit can be different per customer ($v_i \neq v_j$), can be easily integrated. In MP1 the outsourcing cost is only present in the master problem, hence, this can be easily changed to a customer-dependent fee without algorithmic changes in MP1-PA1 and MP1-PA2. However, in MP1-PA2, currently at most one column is generated per iteration since creating multiple columns would result in columns with the same costs. In case of customer-dependent outsourcing costs, it can be beneficial to generate multiple columns per pricing problem iteration. The reduced cost for MP2 does depend on the outsourcing cost. In MP2-PA1 the split customer, and hence its corresponding outsourcing cost, is decided during the labeling algorithm and, therefore, the customer-dependent outsourcing cost can easily be accounted for. On the contrary, for MP2-PA2, it is not straightforward to include customer-dependent outsourcing costs. This is caused by the fact that the reduced cost function is both dependent on the outsourced quantity and the outsourcing cost per unit. Hence, with customer-dependent outsourcing costs, in the dominance conditions both the range and the slope of the segments are unknown. Dominance conditions therefore have to be customized for MP2-PA2 to include customer-dependent outsourcing costs.

5.4 Implementation

5.4.1 Branching

Let x^* be the current fractional solution expressed in the arc flow variables in which x_{ij} is the arc flow variable of arc $(i,j) \in A$ of the underlying compact formulation. To result in a feasible solution, first, the algorithm branches on the total number of vehicles $\sum_{k \in K} \sum_{i \in V_0} x^k_{0i}$. In the results section we also consider to discard this first branching option, and start with the second branching strategy immediately. Secondly, the algorithm branches on the number of vehicles per vehicle type $\sum_{i \in V_0} x^k_{0i}$. If for all vehicle types the number used is integer, branch on the edge variables $x^k_{ij} + x^k_{ji}$ for some vehicle type $k \in K$. The algorithm looks for $i,\ j$ pairs such that ${x^*}^k_{ij} + {x^*}^k_{ji}$ is close to 0.5 and imposes the branches $x^k_{ij} + x^k_{ji} \leq 0$ and $x^k_{ij} + x^k_{ji} \geq 1$. Finally, branching on one fractional arc x^k_{ij} is performed for some arc $(i,j) \in A$ and vehicle type $k \in K$. As in Dabia et al. [2019] we apply strong branching which means that potential branches are evaluated quickly to decide which one to continue with first. In this case we solve the LP relaxation with only the columns already generated in the column generation algorithm. This way, for each branching candidate we estimate a lower bound in the two child nodes. The algorithm chooses the branch that maximizes the lower bound in the weakest of the two child nodes. In the first 15 nodes of the branch-and-bound tree we consider 30 branch candidates and 15 branch candidates in the remaining nodes.

5.4.2 Acceleration techniques

To speed up the labeling algorithm, we implement bidirectional labeling. This means that paths are both created forward from the starting node and backward from the ending node up to some bound in one of the resources which are, afterwards, merged to feasible paths. The dominance rules are also applied to backward labels. We apply an advanced version of bidirectional labeling in which the half-way point of the resource (up to which labels are extended) is determined dynamically during the labeling algorithm [Tilk et al., 2017]. As the boundary resource for the VRPPO we use the time.

We implemented the ng-path relaxation introduced by Baldacci et al. [2011] which allows for cycles in the labeling algorithm instead of finding only elementary routes. For each customer $i \in V_0$ define a neighborhood NG_i which contains node i itself and, at most, $b \in V_0$ other nodes that are closest to i. An ng-path can visit a customer i more than once only if at least one node $j \notin NG_i$ is visited between two visits to i. Allowing for such paths to be generated and added to the master problem may yield weaker lower bounds. However, the pricing problem may become easier to solve if b is sufficiently small. In the labeling algorithm, the number of times a customer is visited should now be counted instead of whether a customer is visited and also different label extensions are now possible with the ng-path relaxation. The labeling algorithms are adjusted accordingly.

The exact labeling algorithms as proposed in Sections 5.3.2.1, 5.3.2.2, 5.3.3.1 and 5.3.3.2 may be time consuming to fully execute. Therefore, before calling an exact pricing algorithm we first apply a heuristic labeling algorithm to more quickly generate negative reduced cost columns. The exact labeling algorithm is only called when the heuristic does not find any negative reduced cost paths. The heuristic performs the labeling algorithm on a reduced graph that keeps for each node, at most, the k outgoing arcs with the smallest reduced cost. The number of kept arcs is increased to $2k$, then to $4k$ until some bound (in our case set to 20) is reached.

5.5 Results

In the following sections, we will first compare, for both master problems, the two pricing algorithms and determine which algorithm performs best. Secondly, for the chosen algorithm the performance and the cost improvement of the VRPPO over the VRPPC on two sets of instances will be investigated. The first set of instances is derived from the instances used by Dabia et al. [2019] for the VRPPC (referred to as instance set $\mathcal{A}$). The second set of instances was constructed for the SDVRPTW and used by e.g., Desaulniers [2010] (referred to as instance set $\mathcal{B}$). We use two sets of instances since the benefit of outsourcing part of a customer's demand may differ for instances originally designed to examine either the impact of outsourcing in the absence of split delivery (set $\mathcal{A}$) or the reverse (set $\mathcal{B}$).

The instances in set $\mathcal{A}$ were originally constructed by Liu and Shen [1999] from the Solomon [1987] instances by adding information on heterogeneous vehicles. There are six types of instances, based on topology (R for randomly dispersed customers, C for clustered customers and RC for a combination) and time window size (type 1 for tight time windows and type 2 for wide time windows). The instances contain heterogeneous vehicles and there are 3 vehicles per vehicle type. The algorithms are tested on instances with 25, 50 and 100 customers. As in Dabia et al. [2019] and Liu and Shen [1999], three different vehicle cost levels are considered. Types a, b and c have high, medium and low vehicle costs respectively. We refer to Liu and Shen [1999] for more details on the vehicle compositions and vehicle fixed costs. The outsourcing cost is derived from Dabia et al. [2019]. We do not consider all unit discounts for outsourcing as in Dabia et al. [2019], but rather a fixed fee per unit outsourced as argued in Section 5.3. Therefore, we consider the highest cost and the lowest cost from Dabia et al. [2019] in these experiments to examine the impact of different outsourcing cost levels. This means we set: $v = 5.00,\ 3.50$ for R instances, $v = 2.00,\ 0.50$ for C instances and $v = 3.50,\ 2.00$ for RC instances. There are 56 different instances, each with three vehicle costs, for the three different number of customers and for two outsourcing cost levels; this gives 1008 instances in total.

The instances in set $\mathcal{B}$ are also derived from the Solomon [1987] instances by allowing split deliveries. These instances contain homogeneous vehicles without fixed vehicle costs. Since the original vehicle capacity in the Solomon instances is relatively high, the vehicle capacity for the SDVRP is set to $Q = 30,\ 50,\ 100$, respectively. Based on preliminary experiments, for the VRPPO we only use $Q = 30,\ 50$ since results for $Q = 50$ and $Q = 100$ are comparable in terms of cost improvements. Desaulniers [2010] points out that since demand is randomly generated between 1 and 50, split deliveries can be necessary in the SDVRP for $Q = 30$. This implies that outsourcing can be necessary for the VRPPO and the VRPPC. The considered outsourcing costs are the same as for instance set $\mathcal{A}$ as described above. Again, there are 56 different instances, two vehicle capacities, three different number of customers, and two outsourcing cost levels resulting in 672 instances in total.

For all instances, the master problems are initialized with a column that represents the solution in which delivery of all demand is fully outsourced. The cost of this column provides a valid upper bound on the solution. The branch-price-and-cut algorithms are implemented using Java and Gurobi 8.0. All tests are performed on a desktop computer running Windows 10, using a single core from an eight core Intel(R) Core(TM) i7-6700K, CPU 4.00GHz processor with 24GB of RAM. The maximum computation time is one hour per instance.

The outline of the computational experiments is as follows. First, in Section 5.5.1, we compare the two master problems for each of the two pricing algorithms on a subset of the instances in set $\mathcal{A}$ to evaluate their performance. Secondly, Section 5.5.2 presents aggregated results on extensive tests on both instance sets $\mathcal{A}$ and $\mathcal{B}$ for the selected algorithm from Section 5.5.1. Next, in Section 5.5.3, we compare for both instance sets the results of the VRPPO and the VRPPC to gain insight in the potential cost improvements of allowing a part of the demand to be outsourced. Finally, Section 5.5.4 presents some figures to gain insight into the structure of the solutions of the VRPPO. The results per instance are given in Appendix G.

5.5.1 Comparing the algorithms

Both for MP1 (Section 5.3.2) and MP2 (Section 5.3.3), two algorithms have been proposed for the pricing problem (PA1 and PA2). Since the pricing algorithms differ in several aspects, it is hard to assess their performance on theoretical grounds. The first pricing algorithm can create multiple labels per node extension, which results in many labels, but vehicle capacity cannot be exceeded. On the contrary, the second pricing algorithm creates at most one label per node extension, but can result in longer paths since vehicle capacity can be exceeded. Furthermore, since artificial segments for the reduced cost function are necessary to apply dominance in the second pricing algorithm, the dominance criteria cannot be compared between the pricing algorithms. Moreover, for the second pricing algorithm a post-processing step is required. Therefore, it cannot be stated upfront which algorithm will perform best in terms of running time and number of instances that can be solved. Hence, we test all four algorithms MP1-PA1, MP1-PA2, MP2-PA1, and MP2-PA2 for multiple parameter settings on a subset of the instances in set $\mathcal{A}$.

Preliminary experiments showed that the differences between algorithms and parameter settings are quite substantial. To find the best performing algorithm (master problem, pricing algorithm and parameter setting), the R, C, and RC instances with time window type 1 (tight time windows) were selected, with 25 customers, for vehicle cost a, and with high outsourcing cost to test on. This resulted in a set of 29 instances. Next, we ran the algorithms on 11 additional instances with time window type 2 (25 customers, vehicle cost a, high outsourcing cost) that are easier to solve compared with other instances with type 2 time windows.

In these experiments three parameters that are likely to have an impact on the performance of the algorithms are evaluated. These are an indicator signaling if branching on the total number of vehicles is used, the maximum number of active (Generalized) SR inequalities at any point during the execution of the algorithm (value 30 or 40), and the size of the neighborhood of the ng-paths (value 7, 8, or 9). Column 'S' in Table 5.1 indicates the results for eight different scenarios. The values of the parameters are indicated in the columns 'Br', '#SR', and 'NG' respectively. For each algorithm, per parameter setting, the average CPU time for solving the instances with time window type 1 to optimality ('T1(s)'), the number of solved instances out of the 40 instances ('#Opt.'), and the average CPU time of solving the 40 instances to optimality ('T(s)') are given.

Table 5.1 Comparison algorithms for different parameter settings

				Master Problem 1						Master Problem 2					
				PA1			PA2			PA1			PA2		
S	Br	#SR	NG	T1(s)	#Opt.	T(s)	T1(s)	#Opt.	T(s)	T1(s)	#Opt.	T(s)	T1(s)	#Opt.	T(s)
1	1	40	7	117	37	297	**31**	**40**	**195**	103	37	290	**32**	**40**	**185**
2	1	40	8	110	37	304	29	39	139	84	37	297	22	39	86
3	1	40	9	110	37	320	31	39	138	99	36	235	21	39	87
4	0	40	7	95	37	273	24	39	115	94	37	281	**22**	**40**	**186**
5	0	40	8	82	37	283	25	38	56	71	37	282	20	39	100
6	0	30	7	122	37	295	**30**	**40**	**173**	146	37	321	**28**	**40**	**189**
7	0	30	8	98	37	293	25	39	123	98	37	302	25	39	94
8	1	30	7	175	37	336	**35**	**40**	**174**	120	37	301	**36**	**40**	**167**

The first obvious conclusion is that PA2 performs much better than PA1 for both master problems. Both the running time with PA1 for time window type 1 instances is much higher than for PA2 (columns 'T1(s)') and PA1 solves, at most, 37 out of 40 instances (columns '#Opt'). Comparing MP1-PA2 and MP2-PA2 suggests that the performance of these algorithms is quite similar as for both MP1 and MP2 all 40 instances are solved in three and four scenarios respectively (indicated in bold). Because scenario 4 for MP2-PA2 clearly gives the lowest running time for the type 1 instances and does not perform worst on all 40 instances, it is selected for the remaining experiments.

5.5.2 Aggregated results VRPPO

Table 5.2 reports on results on the performance of MP2-PA2 for instance set $\mathcal{A}$, aggregated over time window type and vehicle costs. Table 5.3 gives the results for instance set $\mathcal{B}$, aggregated over time window type and vehicle capacity. Presented in both tables, for each topology R, C, and RC ('Topology', number of instances between brackets) and per instance size ('N') are the number of instances solved to optimality ('#Opt.'), the number of solved instances in which the solution consists of outsourcing all demand ('#All out.'), the average computation time in seconds ('T(s)'), the average size of the branch-and-bound tree of the instances solved to optimality ('Tree'), and the average integrality gap ('Gap(%)'). The integrality gap indicates the percentage difference between the optimal integer solution (IP) and the solution of the linear relaxation at the root node (LB_{root}) which is calculated by (IP - LB_{root})/IP. Remind that if in a solution all demand is outsourced, this solution is the initial solution of the master problem.

Instance set $\mathcal{A}$
Instances in set $\mathcal{A}$ are easier to solve for low outsourcing costs than for high outsourcing costs as shown by the number of instances solved, the computation times, and the size of the branch-and-bound tree indicated in Table 5.2. As can be expected, the number of solved instances decreases if the number of customers increases. The number of solved instances is approximately the same for high and low outsourcing cost for the R instances. For the C and RC instances, lower outsourcing costs allow for solving more instances to optimality. This observation can be explained by the fact that for low outsourcing cost the optimal solution is regularly to outsource all demand which is a relatively easy solution to find since it is the initial solution. On average, all instances are solved within 17 minutes with more than half of the instances being solved within 10 seconds, however, for some instances the optimal solution is only found very close to the time limit (see Appendix G for the results per instance). For all solved instances, the integrality gap is low with all averages below 1.5% and a maximum of 4.40%.

Instance set $\mathcal{B}$
Table 5.3 gives the performance results on the VRPPO for instance set $\mathcal{B}$ aggregated on vehicle capacity and time window type. The time to solve these instances to optimality is smaller than those in set $\mathcal{A}$, probably because the instances in set $\mathcal{B}$ do not contain heterogeneous vehicles and, therefore, less pricing problems need to be solved. As a result, all 25 and 50 customer instances are solved to optimality. The difference in solution time between the instances with high and low outsourcing cost is smaller

Table 5.2 Aggregated results on instance set $\mathcal{A}$

		High outsourcing cost					Low outsourcing cost				
Topology	N	#Opt.	#All out.	T(s)	Tree	Gap(%)	#Opt.	#All out.	T(s)	Tree	Gap(%)
R (69)	25	43	0	29	1.7	0.13	42	0	14	0.7	0.05
	50	26	0	485	12.2	0.44	30	0	391	5.7	0.26
	100	3	0	78	10.7	0.13	4	0	299	15.0	0.11
C (51)	25	33	17	273	2.9	1.12	51	51	0	0	0.00
	50	26	16	457	16.6	0.62	51	51	27	0	0.00
	100	25	15	354	0.9	0.01	48	48	110	0	0.00
RC (48)	25	35	0	127	1.1	0.14	40	15	106	0.1	0.06
	50	18	0	963	99.2	1.43	26	12	338	7.8	0.68
	100	9	0	870	6.7	0.08	17	9	671	4.8	0.08

for these instances than those in set $\mathcal{A}$. Again, instances with low outsourcing costs regularly result in solutions in which all demand is outsourced, for both C and RC instances. For the instances in set $\mathcal{B}$, the RC instances are not harder to solve than the R and C instances opposed to set $\mathcal{A}$. On average, the instances in set $\mathcal{B}$ are solved within 6 minutes with more than three-quarters of them being solved within 10 seconds (see Appendix G). Integrality gaps for the instances in set $\mathcal{B}$ are even lower than for those in set $\mathcal{A}$ with all averages being below 1.5% and with a maximum gap of only 3.23%.

Table 5.3 Aggregated results on instance set $\mathcal{B}$

		High outsourcing cost					Low outsourcing cost				
Topology	N	#Opt.	#All out.	T(s)	Tree	Gap(%)	#opt	#All out.	T(s)	Tree	Gap(%)
R (46)	25	46	0	0	0	0.00	46	0	0	0.3	0.03
	50	46	0	17	4.3	0.11	46	0	19	5.0	0.10
	100	29	0	250	11.9	0.04	30	0	212	13.1	0.05
C (34)	25	34	0	3	12.1	0.52	34	34	0	0	0.00
	50	34	0	255	492.0	0.49	34	34	0	0	0.00
	100	25	0	161	34.8	0.08	34	26	0	0	0.00
RC (32)	25	32	0	7	40.5	1.48	32	16	2	7.6	0.26
	50	32	0	2	3.9	0.53	32	16	29	47.8	0.23
	100	31	0	357	16.5	0.07	32	0	75	5.9	0.05

5.5.3 VRPPO vs VRPPC

In Sections 5.5.3.1 and 5.5.3.2 the solutions of the VRPPO are compared with the solutions of VRPPC on the total costs for instance sets $\mathcal{A}$ and $\mathcal{B}$ respectively. In Section 5.5.4 we explore the impact of allowing splitting on the structure of the routes. The solutions of the VRPPC are obtained by running the algorithm by Dabia et al. [2019] with the fixed fee outsourcing cost. The results are aggregated over all instance sizes. The columns in the tables report the topology of the instances ('T'), the type of time windows ('TW'), the fixed vehicle costs ('C') for set $\mathcal{A}$, and the vehicle capacity ('Q') for set $\mathcal{B}$. For both outsourcing costs (high and low), the table gives in the first three columns the average percentage cost improvement ('Avg. %'), the highest improvement ('Max. %'), and the number of solved instances in the category ('#Opt.') respectively.

Finally, the fourth column ('All out.') indicates the fraction of the solved instances in which all demand is outsourced in the optimal solution, all (1), none (0), or two-third ($^{2}/_{3}$) of the instances. If for an instance all demand is outsourced in the optimal solution of the VRPPO, then this is also the case for the VRPPC. Hence, for these instances allowing part of a customer's demand to be outsourced does not result in a cost improvement of the VRPPO compared with the VRPPC. For each instance, the percentage improvement in total cost between VRPPO and VRPPC is computed as (cost VRPPC - cost VRPPO)/cost VRPPC. Only instances for which both the VRPPO and the VRPPC result in an optimal solution are included in the averages. Note that there are some instances for which an optimal VRPPO solution is found, but no optimal VRPPC solution.

5.5.3.1 Improvements on set $\mathcal{A}$ instances

In Table 5.4 the solutions of the VRPPO are compared with the solutions of the VRPPC for instance set $\mathcal{A}$.

Table 5.4 Comparison VRPPO and VRPPC for instance set $\mathcal{A}$

			High outsourcing cost				Low outsourcing cost			
T	TW	C	Avg. %	Max. %	#Opt.	All out.	Avg. %	Max. %	#Opt.	All out.
R	1	a	0.32	1.30	20	0	0.68	1.29	25	0
		b	0.12	0.93	22	0	0.08	0.66	22	0
		c	0.08	1.05	22	0	0.03	0.76	22	0
	2	a	0	0	1	0	0	0	1	0
		b	0	0	2	0	0	0	2	0
		c	0	0	4	0	0	0	4	0
C	1	a	0	0	24	1	0	0	27	1
		b	0	0.02	17	0	0	0	27	1
		c	0	0.02	17	0	0	0	27	1
	2	a	0	0	16	1	0	0	21	1
		b	0	0	1	0	0	0	21	1
		c	0	0	1	0	0	0	21	1
RC	1	a	0.12	0.73	15	0	0	0	24	1
		b	0.19	1.31	17	0	0.12	0.60	18	0
		c	0.21	1.46	17	0	0.14	0.69	18	0
	2	a	0	0	1	0	0	0	9	1
		b	0	0	5	0	0	0	4	0
		c	0	0	7	0	0.01	0.06	7	0

First, it is observed that the improvements that can be achieved by allowing a part of the demand to be outsourced are relatively small. For high outsourcing costs, a better improvement can be achieved than for low outsourcing cost for most instance categories. This can be explained by the fact that for a high outsourcing cost, covering more distance with a private vehicle is more likely to be cost efficient compared with a situation in which outsourcing is cheap. For the R and RC instances higher improvements are reached than for the C instances which can be explained by the fact that if one customer in a cluster is outsourced, the whole cluster of customers tends to be outsourced as for the VRPPC [Dabia et al., 2019]. For the limited number of solved instances with wide time windows (type 2), hardly any improvement is achieved by allowing outsourcing. There are no consistent results for the different vehicle costs since, for the R instances, better improvements are achieved for high vehicle costs (a) while for the RC instances, better improvements are found for low vehicle costs (c).

5.5.3.2 Improvements on set $\mathcal{B}$ instances

Table 5.5 shows that for set $\mathcal{B}$ substantially larger improvements are obtained by allowing outsourcing part of the demand compared to set $\mathcal{A}$. Improvements of approximately 10% are achieved. Also observe that higher improvements are obtained when vehicle capacity is tight ($Q = 30$) as the percentage improvements are much higher for $Q = 30$ than for $Q = 50$. This implies that splitting demand over a private vehicle and a common carrier is more beneficial if the customer demands are closer to or higher than vehicle capacity. For vehicle capacity $Q = 30$, larger improvements can be achieved for high outsourcing costs than for low outsourcing costs for the same reasons as indicated for set $\mathcal{A}$. For vehicle capacity $Q = 50$, the improvements are smaller and comparable for both outsourcing cost levels. For high outsourcing costs, the best improvements can be achieved for instances with both clustered and randomly located customers (RC), while for low outsourcing costs, the best improvements are found for instances with only randomly located customers (R). The type of time windows does not have a big impact on the improvements for the instances in set $\mathcal{B}$.

Table 5.5 Comparison VRPPO and VRPPC for instance set $\mathcal{B}$

			High outsourcing cost				Low outsourcing cost			
T	TW	Q	Avg. %	Max. %	#Opt.	All out.	Avg. %	Max. %	#Opt.	All out.
r	1	30	3.87	10.05	36	0	2.55	5.49	36	0
		50	0.61	2.69	28	0	0.88	2.22	29	0
	2	30	3.86	10.05	33	0	2.55	5.49	33	0
		50	0.54	1.29	24	0	0.84	1.76	24	0
c	1	30	2.87	3.99	27	0	0	0	27	1
		50	0	0	18	0	0	0	27	1
	2	30	1.82	3.21	24	0	0	0	24	1
		50	0	0	24	0	0	0	24	$^2/_3$
rc	1	30	4.56	7.95	24	0	0.03	0.09	24	$^2/_3$
		50	0.15	0.88	24	0	0.11	0.48	24	0
	2	30	4.56	7.95	24	0	0.03	0.09	24	$^2/_3$
		50	0.13	0.76	23	0	0.10	0.39	24	0

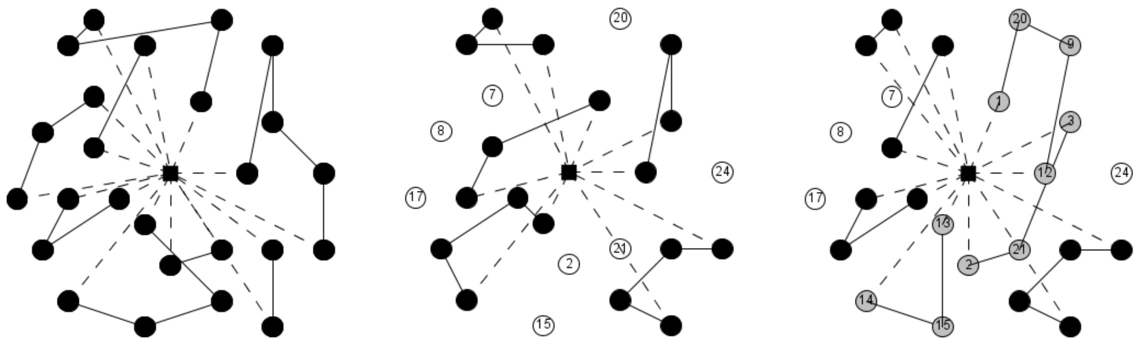
To see the impact of time windows, we also conducted experiments for a subset of instances in data set $\mathcal{B}$ in which we discarded the time windows and adjusted the algorithm accordingly by e.g., removing the time condition from the dominance criteria. We observed that the results are quite comparable to those with time windows for these instances.

5.5.4 Insights

To get further insight in the obtained results, we examine the structure of some individual optimal solutions by visualizing them in Figures 5.3 and 5.4. In both figures, routes are indicated by lines connecting the visited customers. The customers that are not connected by the lines have their demand fully outsourced. The customers indicated in gray are in a route that requires a split delivery since total demand exceeds vehicle capacity. Note that any customer with sufficiently high demand can be the split customer without changing the total costs, therefore, all customers in these routes are colored gray.

In a solution of the VRPPC some routes may not fully utilize the vehicle capacity. One might expect that the VRPPO solution contains the same routes as the VRPPC solution in which more customers are added to the routes and a split delivery is performed to fully utilize the vehicle capacity. However, the results show that this is not necessarily the case. Rather, the VRPPO solution of an instance can contain completely different routes than the corresponding VRPPC solution. As an example, consider Figure 5.3 which shows the optimal VRP, VRPPC and VRPPO solutions of set $\mathcal{A}$ instance R101a with 25 customers and high outsourcing costs. Consider customer 20 in the upper part of Figure 5.3b. Customer 20 is not added to the route 0-3-9-12-0 of the VRPPC solution to find the optimal VRPPO solution, but rather, it is combined with customers 9, 12 and 1, both because of efficiency and time windows. Note that customers 3 and 20 cannot be in the same route because of their time windows; temporarily widening the time windows of customer 20 to allow for customers 3 and 20 in the same route did not result in a different solution. Hence, in the VRPPO solution, customer 20 is not added to route 0-3-9-12-0 of the VRPPC solution because of route efficiency reasons. At the same time, customer 3 in the VRPPO solution is served together with customers whose demand was fully outsourced in the VRPPC solution. One can observe there is only one route that is the same in both solutions (in the lower right area). The VRPPO solution contains seven routes, of which three contain a split delivery.

Next, it is also interesting to look at the number of units of demand outsourced. In both the routes 0-2-21-3-0 and 0-14-15-13-0 just one unit of demand needs to be outsourced, hence, all customers are candidate to split the delivery since their demand is larger than one. In route 0-12-9-20-1-0, four units of demand need to be outsourced. Also for this route, all customers are candidate to be split and four units of demand represents between 21% and 44% of each customer's demand.

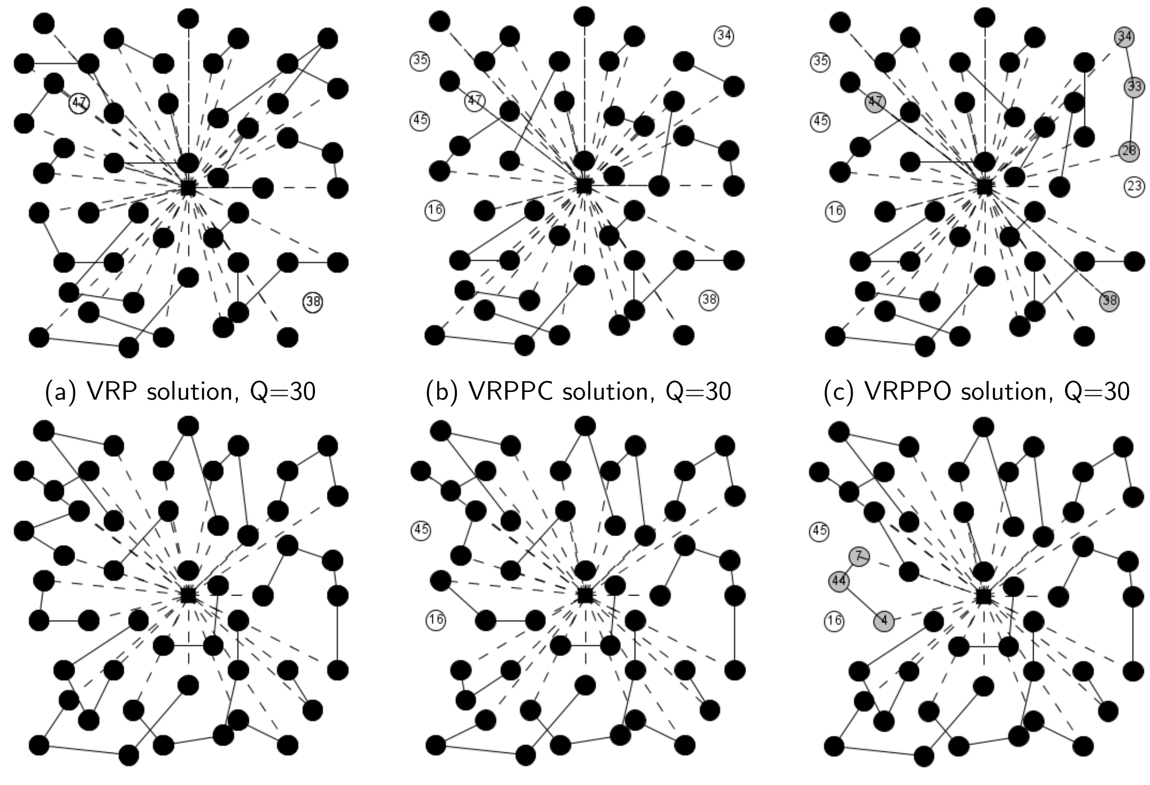


(a) VRP solution (b) VRPPC solution (c) VRPPO solution

Figure 5.3 Set $\mathcal{A}$ instance R101a, 25 customers, high outsourcing cost. Routes with gray customers require a split delivery which can be assigned to any customer with sufficiently high demand.

Figure 5.4 presents the VRP, VRPPC and VRPPO results for set $\mathcal{B}$ instance R102 with high outsourcing costs. The improvement in costs of the VRPPO compared with the VRPPC is 9.79% for $Q = 30$ and 1.55% for $Q = 50$ respectively. The VRP solution for $Q = 30$ is infeasible, since customers 38 and 47 have demands higher than the vehicle capacity. For $Q = 30$, two customers (38 and 47) with demand larger than the vehicle capacity must be fully outsourced in the VRPPC solution (Figure 5.4b) but in the VRPPO solution full trucks can be sent to these customers which reduces costs substantially (Figure 5.4c). For $Q = 50$, both customers 38 and 47 are combined with another customer in the route in both the VRPPC and the VRPPO solution (Figure 5.4e and 5.4f respectively). Note that for $Q = 30$ improvements are not only achieved because customers with a demand higher than vehicle capacity can be partially served by a private vehicle in the VRPPO, but also other adjustments can be made to improve efficiency. For example, customer 34 that has demand of eight is outsourced in the VRPPC solution but is served by a private vehicle in the VRPPO solution. Moreover, note that customer 23 for $Q = 30$ is served in the VRPPC solution while in the VRPPO solution it is more efficient to fully outsource this customer to service customer 34 by the private vehicle (with a split in the corresponding route). For $Q = 50$, the VRPPO solution contains one route less than the VRPPC solution. Hence, by allowing a split between private and outsourced delivery, the number of used private vehicles can be reduced in some cases.

For $Q = 30$, the quantities outsourced are one unit for routes 0-39-0 and 0-29-34-35-0, and six units for route 0-48-0 which is 17% of the demand. For $Q = 50$, only route 0-4-44-7-0 requires a split and one unit of demand is outsourced which is between 4% and 11% of the demand.



(a) VRP solution, Q=30 (b) VRPPC solution, Q=30 (c) VRPPO solution, Q=30

(d) VRP solution, Q=50 (e) VRPPC solution, Q=50 (f) VRPPO solution, Q=50

Figure 5.4 Set $\mathcal{B}$ instance R102, 50 customers, high outsourcing cost. Routes with gray customers require a split delivery which can be assigned to any customer with sufficiently high demand.

Concluding, a solution of the VRPPO can be rather different than the corresponding VRPPC solution. The routes are structurally different, customers fully served in a VRPPC solution can be fully outsourced in the VRPPO solution, and customers with demand higher than vehicle capacity can receive a full truck load delivery in the VRPPO solution. Moreover, both small and large shares of the demand are being outsourced in the split delivery in the considered examples.

5.6 Conclusion and Future Research

This paper is the first to formally describe a vehicle routing problem in which splitting the delivery of demand to customers between the private and common fleet is allowed. For the so-called Vehicle Routing Problem with Partial Outsourcing (VRPPO), we developed a branch-and-price-and-cut solution framework. We proposed two master problem formulations for the VRPPO, and for both master problems we designed two pricing algorithms. In the first master problem, all outsourcing decisions are taken in the master problem, while in the second master problem the decision on partially outsourcing a demand is referred to the pricing problem. The first pricing algorithm was inspired by a pricing algorithm for the SDVRPTW by Desaulniers [2010] in which multiple labels per extension are created to decide which customer is split. The second pricing algorithm exploits specific problem characteristics by creating at most one label per extension and by taking the splitting decision after creating a path. The first pricing algorithm leads to a higher number of labels, while in the second pricing algorithm the paths can be longer since vehicle capacity can be exceeded during the labeling algorithm. The performance of the algorithms is enhanced by applying (Generalized) Subset-Row inequalities and dominance rules in the labeling algorithms. For the first pricing algorithm the dominance rules suggested by Desaulniers [2010] are applicable. For the second pricing algorithm, non-trivial problem specific adjustments are made to be able to handle the postponed decision on splitting.

Extensive testing on two sets of instances derived from the literature provided insight in the different algorithms and the possible cost improvements of the VRPPO over the VRPPC. The results show that the second pricing algorithm performs much better than the first pricing algorithm and that the difference between the master problems is small. Moreover, the results show that higher cost improvements can be achieved through outsourcing and split deliveries if customer demand is close to or higher than vehicle capacity. If outsourcing costs are low, it can be more beneficial to outsource all demand of a certain customer instead of having a split, thus resulting in larger cost improvements of the VRPPO over the VRPPC for high outsourcing costs. Finally, a topology with randomly located customers gives more room for improvement than settings with only clustered customers since outsourcing one customer in a cluster tends to lead to outsourcing all customers in the cluster, which was also observed for the VRPPC [Dabia et al., 2019].

Since both routing problems with outsourcing or split deliveries are rich problems, multiple directions for future research can be identified. First, similar to Dabia et al. [2019] and Gahm et al. [2017], the outsourcing cost structure could be extended to include for example quantity discounts. Secondly, one could consider customer inconvenience constraints such as a minimum delivery amount [Gulczynski et al., 2010, Han and Chu, 2016]. Finally, also allowing for splits between private vehicles in the VRPPO could offer interesting research challenges.

6

Conclusions

This dissertation explored several distribution problems inspired by industry practice. Four separate studies were presented in Chapters 2 to 5. Although these studies relate to different practical problems, they each contribute to the understanding of distribution problems and help to increase efficiency in such problems. The studies focus on gaining insight in fundamental distribution problems, developing efficient distribution strategies and analyzing the benefit of novel distribution strategies. This concluding chapter summarizes the main findings, discusses implications and reflects on limitations and further research.

6.1 Summary of main findings

The following sections summarize the main findings of chapters 2 to 5 by focusing on the key issues addressed and by highlighting the main results obtained.

6.1.1 Understanding the Computational Complexity of the Inventory Routing Problem

To understand the difficulties that arise in solving distribution problems, it is important to comprehend the underlying computational complexity of a problem. This allows to reveal the structure of a problem which contributes to developing solution methods that exploit this structure. Therefore, Chapter 2 investigates the sources of computational complexity of the Inventory Routing Problem (IRP) by looking for complexity proofs for different variants of the problem. Since the Traveling Salesman Problem (TSP) is an NP-hard problem and it is a special case of the IRP, it can be concluded immediately that the IRP is NP-hard [Karp, 1972]. However, the underlying routing problem is not necessarily the only complicating aspect of the IRP. Therefore, Chapter 2 studies the IRP on metrics on which the TSP is easy or even trivial and, hence, NP-hardness through the TSP is avoided. The IRP on a point (the depot and all customers at one location), on a half-line and in a Euclidean plane are considered. The objective of the study is to find a borderline between easy and hard problems. Similar problems on alike metrics have been studied before, but those studies mainly focused on establishing approximation algorithms rather than on understanding the computational complexity [Das et al., 2011, Bosman et al., 2018].

The analysis shows that, next to routing, also the time horizon, the service times, the customer demand combined with vehicle capacity, and the number of vehicles contribute to the complexity of the IRP. The main finding is a polynomial time algorithm for the studied IRP on the half-line with uniform service times and a planning horizon of two days. Since this variant is a borderline problem, this result implies that the studied IRP becomes hard or has an open complexity if one of the features is generalized. Moreover, analysis shows that almost any IRP variant with arbitrary service times (i.e., the service times are different per customer) is NP-hard. The same result holds if vehicle capacity is considered and the customers have arbitrary demand. If the planning horizon is extended to an arbitrary number of days while all other aspects of the problem are simple (i.e., not implying NP-hardness of the problem), it turns out that the Pinwheel Scheduling Problem [Holte et al., 1989] determines the complexity of the IRP. Unfortunately, the complexity of this problem has not been established so far, but it is unlikely that this problem is easy to solve [Jacobs and Longo, 2014]. Finally, Chapter 2 considers a variant of the IRP in the Euclidean plane for which the objective is to minimize the total tour length. The tour length is not given by optimized TSP tours, but rather by approximations. The approximation of the tour length takes the number of locations and an area in which these locations are spread as input [Beardwood et al., 1959, Chien, 1992]. In the literature several types of areas have been considered. In Chapter 2 the convex hull of the involved locations is used as area. This approximation of the TSP allows for showing NP-hardness of the considered IRP in the Euclidean plane. Therefore, this chapter shows NP-hardness of a generalization of the Pinwheel Scheduling Problem in which tasks are executed at different locations without using the hardness of TSP.

6.1.2 Considering delivery aspects when taking ordering decisions

Chapter 3 considers the case in which a supplier outsources its customers replenishment deliveries to an external carrier. The business partner of this Ph.D. project is such a supplier. The business partner Geldmaat decides upon the replenishment of ATMs in the Netherlands and issues replenishment orders to Cash-in-Transit companies. In the terminology of the replenishment literature, Geldmaat acts as a supplier that outsources delivery of goods. Different cost structures can be applicable for outsourcing the distribution of goods. Chapter 3 considers a cost structure in which a fixed fee is incurred per customer replenishment and per day on which at least one replenishment takes place. Given this cost structure, the optimization problem faced by the supplier is a Dynamic-Demand Joint Replenishment Problem (DJRP) [Khouja and Goyal, 2008] in which it has to be determined in which period to replenish each customer. Chapter 3 argues that solving a DJRP does not generate efficient distribution plans from a transportation or supply chain point of view. In practice, however, suppliers do face a DJRP and likely solve this problem to optimize their business, because, the supplier cannot take customer locations into account when deciding on the replenishments given the fixed fee-cost structure. As a result, the carrier is forced to perform inefficient delivery routes which leads to higher transportation costs which will result in higher fixed transportation fees in future contracts. Moreover, if the carrier has a limited fleet, it can occur that not all customers can be served on the same day due to longer travel times between distant locations.

To address this shortcoming of the DJRP, Chapter 3 studies an extension of a DJRP by including transportation costs. This extension allows to assess the efficiency improvement if customer locations would be taken into account by the supplier when deciding on which customers the carrier has to service. To this end, the Dynamic-Demand Joint Replenishment Problem with Approximated Transportation Costs (DJRP-AT) is defined and a solution method based on branch-and-price-and-cut is developed. The transportation costs are computed as an approximation of the optimal tour length with the number of locations and an area containing the locations as inputs [Beardwood et al., 1959]. A similar tour length approximation was considered in Chapter 2. At first, in Chapter 3, the convex hull covering all locations was used as area similar to Chapter 2, but eventually Chapter 3 adopted the more straightforward computation of the smallest rectangle covering all locations to establish the area. For solving the DJRP-AT, using approximations for the transportation costs implies that sets of customers that are visited by one vehicle on one day have to be generated. Generating customer sets is expected to be easier than generating actual vehicle routes since the sequence of the customers is not important. However, although problem specific dominance rules were developed to discard labels, the DJRP-AT showed to be hard to solve. This will be discussed in more detail in Section 6.3. Still, the solutions found with the designed solution method could support the analysis and could prove the point that was being made.

The distribution plans and costs found with the DJRP and the DJRP-AT are compared. The results show that significant cost savings can be achieved by deviating from the traditional DJRP cost structure. It is shown that a collaborative ordering strategy can be beneficial to both the supplier and the carrier and this insight provides support for future negotiations between the involved parties. Also, since approximated transportation costs are used, the results can be compared to a variant of the IRP. This IRP contains different constraints than the IRP typically addressed in the literature and hence, there is no state-of-the-art solution method for this variant of the IRP. Therefore, an IRP equivalent to the DJRP-AT with optimized routing is implemented which can be solved for the smallest instances. Comparison with the DJRP-AT shows that the average deviation of the DJRP-AT from the IRP is small, and hence, that using route length approximations results in solutions close to those found by fully optimizing the delivery routes.

6.1.3 More Efficient Replenishment by Introducing Demand Moves in the Inventory Routing Problem

Chapter 4 addresses an approach for inventory replenishment in which customers can fulfill (part of) the demand of a nearby customer. As an example, consider ATMs that are often located in close proximity of each other in urban areas which provides the opportunity to redirect or proactively steer ATM-users (end-users) to a certain ATM to make a cash withdrawal. An ATM-user can, for instance, be redirected when arriving at an ATM or be informed upfront via a mobile application, possibly incurring a reward or penalty for using a certain ATM. This redirection option can be incorporated in the optimization of ATM replenishment. Therefore, Chapter 4 extends the IRP with *demand moves* which precisely represent the redirections of end-users between customers which leads to the definition of the Inventory Routing Problem with Demand Moves (IRPDM). The aim of the chapter is to introduce the novel concept of demand moves, to model the demand moves in the IRP and to assess the impact on the solutions and costs compared to the traditional IRP. For each demand move a service fee/cost is incurred which depends on the distance between the involved customers and quantity moved.

To model the IRPDM, a problem formulation for the IRP from the literature [Desaulniers et al., 2016] is extended and a branch-and-price-and-cut solution approach is developed. The technical analysis of the IRPDM showed that using the initial inventory at the customers to satisfy the moved demand added complexity to the use of valid inequalities in the solution method. Therefore, the solution method in Chapter 4 is restricted to the case in which the initial inventory at a customer can only be used to satisfy the demand of the customer itself. Multiple families of valid inequalities have been developed for the IRP in existing literature. However, only one of these families is directly applicable to the IRPDM, for the others non-trivial adjustments are required. The adjustments imply that the valid inequalities are not as strong as the original ones, this is for example caused by the fact that there are more possibilities to satisfy a customer's demand. An IRPDM solution can result in the situation that a certain customer is never replenished by the vehicle since all demand (after consuming the initial inventory) is moved to another customer. This might not be desirable in practice, hence, in the model the option is included to limit the percentage of demand of a customer that can be moved per day. The results show that substantial cost improvements can be achieved if there is no maximum imposed on the demand that can be moved per customer per day. The results also indicate that only a limited number of demand moves per day is applied in the solutions. Hence, to achieve these substantial cost improvements not many customers have to be involved, but are rather only a few customers per day are affected. Furthermore, sensitivity analysis is performed on both the demand move fee and the maximum percentage of demand that can be moved. Limiting the demand that can be moved to 75% approximately halves the potential cost improvement, which seems a large reduction. However, even by allowing only 25% of the demand of a customer to be moved per day still results in cost improvements that are worthwhile in practice.

6.1.4 The Vehicle Routing Problem with Partial Outsourcing

Chapter 5 formally defines the Vehicle Routing Problem with Partial Outsourcing (VRPPO) which is an extension of the Vehicle Routing Problem with Time Windows. In the VRPPO, a customer can either be served by a single private vehicle, by a common carrier or the service can be split between a private vehicle and the common carrier. For outsourcing, a fixed fee per unit of goods is paid which is independent of the customer. Both VRPs with split deliveries (SDVRP) (see for example Archetti and Speranza [2012]) and VRPs with outsourcing (VRPPC) (e.g., Chu [2005] and Dabia et al. [2019]) have been defined and studied before in the literature, but a formal definition of a VRP with both these distribution strategies was not present in the literature.

To address the VRPPO, Chapter 5 proposes two formulations and designs branch-and-price-and-cut solution methods, with for each formulation two different exact pricing mechanisms. The aim of the chapter is to analyze the two problem formulations and the corresponding solution methods. Besides that, the solutions and the associated costs are compared to those of the VRPPC in which a split between the delivery options is not possible. Comparison of the solution methods, clearly shows that, per problem formulation, one exact pricing mechanism is more efficient than the other pricing mechanism. Moreover, it shows that the difference between the problem formulations is small for the best pricing mechanisms. Testing on two sets of instances to analyze the cost difference between the VRPPO and the VRPPC shows higher cost improvements of the VRPPO over the VRPPC if customer demand is close to or higher than the vehicle capacity. Also, higher outsourcing costs results in higher cost improvements of the VRPPO over the VRPPC than lower outsourcing costs. A possible explanation is that if outsourcing costs are low, then it is more beneficial to outsource all units of one customer and hence, with low outsourcing costs, the benefit of allowing for splits declines. Finally, if customers are located in clusters, cost improvements are lower than if customers are randomly spread over an area. Visualization of some solutions shows that a VRPPO solution can contain completely different routes than the corresponding VRPPC solution.

6.2 Implications

Solution methods for the IRP are often focused on hardness of the underlying routing problem. Chapter 2 shows that other aspects should receive sufficient attention as well since routing is not the only factor determining the computational complexity of the IRP. For example, selecting sets of customers to be served together each by one vehicle on one day combined with vehicle constraints is difficult. This is because bin packing aspects are present in the problem and bin packing is a hard problem. Another example is the question on which days to serve a customer, since this causes difficulties through the relation with the pinwheel scheduling problem. Chapter 2 suggests that it is worthwhile to give sufficient attention to other aspects than routing when developing solution methods for the IRP, for both exact and heuristic methods.

Some solution methods in both the literature and this thesis already incorporate these observations. For example, Desaulniers [2010] studies the SDVRP and during the construction of the potential vehicle routes, the delivery quantity for the split customer is not decided upon until finishing the route construction. Hence, the vehicle capacity and delivery quantities are directly considered in the solution method. For the IRP, Desaulniers et al. [2016] use delivery patterns representing all possible combinations of delivery quantities addressing the vehicle capacity issue. Moreover, several delivery patterns can be discarded in a clever way by applying dominance criteria which reduces the number of possible delivery quantities and hence improves the performance of the solution method. The same idea is applied for the IRPDM in Chapter 4 of this thesis. Also in Chapter 5, the delivery quantities and vehicle capacities are addressed explicitly for the VRPPO. In the firstly proposed pricing algorithms for both problem formulations, the potential delivery quantity by the private vehicle for a split customer is stored during the execution of the algorithm and is therefore directly considered. In the second proposed pricing algorithms, the privately delivered quantity for the split customer is only explicitly considered at the end of a route creation. Although it cannot be decided upfront which of the pricing algorithms is more efficient (among others because of a trade-off between number of labels and the length of the created routes), the computational results clearly show that the second type of pricing algorithm outperforms the first.

In recent literature, only limited attention was paid to dynamic-demand joint replenishment problems. This is shown by the systematic literature review by Bastos et al. [2017] in which the authors report only two published studies on the DJRP in the years 2006-2015, while they found more studies for the static (36) and stochastic (19) demand variants of the JRP. Still, the DJRP is a very relevant problem in practice since the cost structure is present in many business applications that face varying demand, for example in ATM replenishment and for retailers that outsource their storage and replenishment activities. Chapter 3 contributes to the insight that more traditional problems, such as the DJRP, can provide a good starting point for decision support in practice and remain important in academic research. Therefore, it can be beneficial to both research and practice to determine the underlying fundamental problems faced by the industry and to study these problems in more detail.

Chapter 3 shows that splitting the optimization of ordering and of delivery decisions is insufficient to find good solutions for practical supply chain problems. However, complete integrated optimization is not always necessary, nor possible, in practice. Chapter 3 shows that the supply chain can benefit if delivery aspects are considered when making ordering decisions. This has multiple implications. First, it is important, both for research and practice, to analyze which decisions can be taken by each party under which cost structure. This provides insight in which formal problem, such as the DJRP, relates to the practical setting. Secondly, to improve the supply chain, one can consider integrating aspects of the business partner's objectives when optimizing one party's decisions. In the DJRP-AT in Chapter 3, the locations of the customers are explicitly considered when optimizing the ordering decisions at the supplier. This allows logistics service providers to construct better delivery routes which lowers costs and increases resource utilization. It is crucial that in contracts between the different parties, the inclusion of the transportation aspects in the ordering process is reflected in the cost structure. If a contract does not contain mutual incentives to cooperate, the parties in a supply chain will most likely not collaborate since there is no benefit for them. Future research can support the process of collaboration by providing insight in the consequences of alternative cost structures.

Chapter 4 shows that incorporating demand moves in the IRP is useful for reducing replenishment costs. It should be noted that implementing demand moves in practice, for example for ATM replenishment, is rather involved. It has to be determined under which circumstances demand moves are appropriate, which includes establishing the service cost for the end-user and appropriate 'neighbor' sets. This latter aspect implicates the question what distance between locations is permissible to redirect an end-user. Hence, the results on the IRPDM in Chapter 4 pose interesting follow-up questions both for research and practice. These will be discussed in more detail in Section 6.3.

Novel distribution strategies are considered in this dissertation, such as demand moves in the IRP and allowing for splits between a private vehicle and outsourcing in the VRPPO. New distribution strategies are considered to enhance efficiency in order to save costs. Sensitivity analysis, such as in Chapters 4 and 5, shows that it is important to analyze under which circumstances these distribution strategies actually provide improvements in practice. For example, if the loss in service due to demand moves (Chapter 4) is very high, then it may not be beneficial to consider demand moves, while even at a maximum of 25% demand moved per customer per day cost reductions can be achieved that are substantial in practice. And Chapter 5 showed that for certain data sets using splits between private and common vehicles does not result in a cost improvement over not using splits. Hence, a careful analysis is necessary before implementing innovative distribution strategies in practice to see whether a novel strategy is actually profitable.

6.3 Limitations and Further Research

Chapter 3 considers an extension of the DJRP which includes transportation costs. The transportation costs are computed by approximating the length of the tour that visits a given sets of customers. By using an approximation of the tour length which uses the combination of customers, the sequence of the customers is not important. It was expected that generating customer sets would result in relatively easy to solve pricing problems. This expectation is based on the fact that a limited number of labels is needed for finding customer sets compared to solving an actual routing problem which requires determining the actual customer sequence. However, the dominance criteria that can be used in generating vehicle routes are not applicable to the DJRP-AT pricing problems. Problem specific dominance rules are developed for the designed pricing problems, but these dominance criteria are not strong which makes the solution method less efficient since many labels are kept during the labeling algorithm. Also, the integrality gaps for the DJRP-AT master problem are large despite using two families of valid inequalities and hence, many iterations between solving the master and pricing problems are required to close the gaps. The results in Chapter 3 show that calculating approximated transportation costs gives solutions close to the ones achieved by calculating actual routing costs. Therefore, the above observations can inspire future research to design more efficient solution methods for the DJRP-AT by potentially developing different solution methods for the pricing problems and also by designing new valid inequalities to reduce the integrality gaps.

Chapter 4 is the first research on including the novel concept of demand moves in the IRP. Being the first study comes with multiple limitations and therefore various directions for further research can be identified.

First, because of algorithmic issues, the initial inventory at a customer at the beginning of the planning horizon is not used to fulfill moved demand in the solutions found in Chapter 4. Including the option to satisfy moved demand from initial inventory can further reduce the costs of the IRPDM compared to the IRP. Hence, further research on the IRPDM includes the design of a solution method that does include the option to fulfill moved demand from initial inventory. Secondly, Chapter 4 analyses the impact of imposing a maximum on the demand that can be moved per customer per day. This represents the practical aspect that it is probably not desirable that all demand of a customer is moved to another customer every day. Currently, this chapter only considers to have the same maximum for each customer on all days. In practice, it can be useful to consider a more flexible system. For example, on a limited number of days it is allowed to move all demand but in the following days the demand that can be moved is limited. Thirdly, in the model it is assumed that a demand move can only take place if there is no inventory left. Hence, only out-of-stock situations are exploited while keeping customer service sufficiently high. If one wants to influence end-user behavior, e.g., by suggesting alternative ATMs for cash withdrawals or by offering financial benefits for using a different ATM than the preferred option, then the model can be extended to include the case in which customers are not necessarily out of stock when demand is moved. Finally, the modeled IRPDM is designed to take decisions on the total volume delivered to a customer in each period, and how much demand is satisfied by each customer. This implies that decisions are not taken on end-user level and hence, an IRPDM solution does not indicate which end-users to redirect to another customer. The IRPDM can be extended to include decisions on end-user level, which can, e.g., be used in a mobile application to inform the end-users.

Furthermore, several implementation issues arise when demand moves are going to be used in practice. These issues also came up during discussions with the business partner in the case of ATM replenishment. Currently, users of Dutch ATMs are accustomed to having the possibility of withdrawing cash free of charge from each ATM. Therefore, further research can investigate appropriate stimuli (i.e., rewards/penalties for cash withdrawals at certain ATMs) to influence end-user behavior while considering that the change should be limited with respect to the current situation faced by the end-user. Moreover, safety is always an important aspect in cash supply chains, therefore, when implementing a reward system it should be considered how to prevent abuse.

This dissertation contains several chapters that study problems that are inspired by practice. When investigating problems inspired by real-life supply chains, it is important to analyze which party takes what decisions and to keep in mind from which perspective the studied problem is to be modeled. Moreover, it is interesting to study what the consequences are of one party's decisions on the choices of other parties in the chain. Chapter 3 on the DJRP-AT explicitly considers the situation in which one party in the supply chain cannot directly influence all decisions in the chain. If the considered supplier would take customer locations into account when taking the ordering decisions, a cost reduction can be achieved by the carrier which could potentially increase the efficiency of the whole supply chain. Although it is clear the supply chain can benefit, it remains to be studied for the DJRP-AT how the supplier can benefit from considering transportation costs when making ordering decisions, i.e., how can a party in a supply chain benefit from considering multiple parts of the chain in their decision making. By contrast, Chapters 4 and 5 assume that one party can take all decisions that are in scope. Concluding, in academic research that is inspired by practice, careful consideration is needed on which parties take what decisions and what processes each party affects. An informed choice should be made which decisions to include in a model. In practice it is important that a party realizes which decisions it can take and how these decisions influence their business partner. And also the other way around: how do the decisions taken by a partner influence your business.

As stated above the research in this dissertation is inspired by practice, at the same time a deliberate choice is made to apply exact solution methods. In some cases it is possible to solve real-life problems via an exact method, however, more often exact methods are incapable of solving real-life sized problems to optimality. Exact approaches do, however, provide insights that heuristic approaches cannot.

First, several exact approaches require a mathematical formulation for the considered problem which provides insight in the structure of the problem which aids the development of solution methods. Chapter 5 also underlines that it is important for some exact solution methods to analyze several problem formulations and corresponding solution methods, since efficiency can differ significantly. Secondly, optimal solutions can reveal structural solution aspects. For example, the number of locations involved in demand moves in the IRPDM in Chapter 4 and the circumstances under which splitting deliveries between a private and common vehicles in the VRPPO in Chapter 5 is beneficial. Thirdly, solutions that are guaranteed to be optimal can be used to assess the quality of a heuristic method, which cannot be fully evaluated otherwise. Additionally, in some cases it is sufficient to develop an exact solution method to prove a point or to show a potential improvement. Finally, an exact approach can provide a starting point for developing a matheuristic which can make use of very fast exact MILP solvers and combines the strengths of exact and heuristic solution methods.

Although exact solution approaches are a good starting point to address problems, it is evident that the new problems introduced in this thesis could benefit from the development of heuristic solution approaches, in particular for solving larger problem instances. Several insights can be obtained from this thesis that are useful in the development of heuristic solution methods. Chapter 2 raises awareness that other aspects than routing cause complexity in the IRP, which can be useful knowledge when designing heuristic solution methods. For example, sufficient emphasis should be put on when a customer is replenished during the planning horizon. Chapter 3 uses an approximation to represent the transportation costs involved in visting a set of locations instead of finding the optimal route. The comparison of the DJRP-AT with the equivalent IRP shows that the deviation of the DJRP-AT solution of the IRP solution is small. Hence, although an approximation of the route length is used, the model is capable of balancing the different costs. The insight that an approximation of certain costs can be sufficiently good can be used in the development of heuristic solution methods, for example if many long routes have to be optimized. Furthermore, the solutions in Chapter 4 for the IRPDM show that the optimal solutions do not necessarily contain many demand moves, and also, on average, approximately half of the demand of a customer in one period is moved if a demand move takes place. These observations which are based on optimal solutions can be exploited when designing heuristic solution methods by not having too many demand moves in a heuristically obtained solution and if there is a demand move, not all demand of a period should necessarily be moved. An approach for a construction heuristic for the VRPPO in Chapter 5 could use a VRPPC solution as starting point and subsequently add customers to the routes in that solution, possibly with a split delivery between a private and common vehicles. However, the solutions obtained in Chapter 5 show that some routes in an optimal VRPPO solution are rather different than in the corresponding VRPPC solution. Therefore, this insight can be used to design heuristics that base the routes less on a VRPPC solution.

Concluding, this dissertation studies the computational complexity of a class of distribution problems, models both fundamental and more practical distribution problems, and develops exact solution methods for such problems. The problems are inspired by real-life optimization problems from a cash supply chain, but are more widely applicable. The studies provide insight in problem structures and solution aspects and contribute to the development of alternative solution methods.

Bibliography

T. F. Abdelmaguid, M. M. Dessouky, and F. Ordóñez. Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering*, 56(4):1519–1534, 2009.

N. Absi, C. Archetti, S. Dauzère-Pérès, and D. Feillet. A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795, 2015.

Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26(1):103–120, 2014.

A. Alvarez, P. Munari, and R. Morabito. Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25(6):1785–1809, 2018.

H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

S. Anily and M. Haviv. The cost allocation problem for the first order interaction joint replenishment model. *Operations Research*, 55(2):292–302, 2007.

S. Anily and M. Tzur. Shipping multiple items by capacitated vehicles: An optimal dynamic programming approach. *Transportation Science*, 39(2):233–248, 2005.

C. Archetti and M. G. Speranza. Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, 19(1-2):3–22, 2012.

C. Archetti and M. G. Speranza. The inventory routing problem: The value of integration. *International Transactions in Operational Research*, 23(3):393–407, 2016.

C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A Branch-and-Cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

C. Archetti, M. W. P. Savelsbergh, and M. G. Speranza. To split or not to split: That is the question. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):114–123, 2008.

C. Archetti, N. Bianchessi, and M. G. Speranza. A column generation approach for the split delivery vehicle routing problem. *Networks*, 58(4):241–254, 2011.

C. Archetti, N. Bianchessi, S. Irnich, and M. G. Speranza. Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21(3):353–374, 2014a.

C. Archetti, M. G. Speranza, and D. Vigo. Vehicle routing problems with profits. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 273–297. MOS-SIAM Series on Optimization (SIAM, Philadelphia), 2014b.

C. Archetti, N. Bianchessi, and M. G. Speranza. A branch-price-and-cut algorithm for the commodity constrained split delivery vehicle routing problem. *Computers & Operations Research*, 64:1–10, 2015.

C. Archetti, N. Boland, and M. G. Speranza. A matheuristic for the multivehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3):377–387, 2017.

E. Arkin, D. Joneja, and R. Roundy. Computational complexity of uncapacitated multi-echelon production planning problems. *Operations Research Letters*, 8(2):61–66, 1989.

P. Avella, M. Boccia, and L. A. Wolsey. Single-period cutting planes for inventory routing problems. *Transportation Science*, 52(3):497–508, 2018.

F. Baita, W. Ukovich, R. Pesenti, and D. Favaretto. Dynamic routing-and-inventory problems: a review. *Transportation Research Part A: Policy and Practice*, 32(8): 585–598, 1998.

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

A. C. Baller, S. Dabia, G. Desaulniers, and W. E. H. Dullaert. The inventory routing problem with demand moves. 2019a.

A. C. Baller, S. Dabia, W. E. H. Dullaert, and D. Vigo. The dynamic-demand joint replenishment problem with approximated transportation costs. *European Journal of Operational Research*, 276:1013–1033, 2019b.

A. C. Baller, S. Dabia, W. E. H. Dullaert, and D. Vigo. The vehicle routing problem with partial outsourcing. *Transportation Science*, 2019c. to appear.

A. C. Baller, M. van Ee, M. Hoogeboom, and L. Stougie. On the complexity of inventory routing problems when routing is easy. *Networks*, 2019d. to appear.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

L. S. L. Bastos, M. L. Mendes, D. R. L. Nunes, A. C. S. Melo, and M. P. Carneiro. A systematic literature review on the joint replenishment problem solutions: 2006-2015. *Production*, 27, 2017.

Ş. Batı and D. Gözüpek. Joint optimization of cash management and routing for new-generation automated teller machine networks. Forthcoming in IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017.

J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Mathematical Proceedings of the Cambridge Philosophical Society*, 55(4): 299–327, 1959.

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

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

L. Bertazzi, M. Savelsbergh, and M. G. Speranza. Inventory routing. In *The vehicle routing problem: latest advances and new challenges*, pages 49–72. Springer, 2008.

N. Bianchessi, M. Drexl, and S. Irnich. The split delivery vehicle routing problem with time windows and customer inconvenience constraints. *Transportation Science*, 2019. Published online in Articles in Advance 13 May 2019.

F. F. Boctor, G. Laporte, and J. Renaud. Models and algorithms for the dynamic-demand joint replenishment problem. *International Journal of Production Research*, 42(13):2667–2678, 2004.

M.-C. Bolduc, J. Renaud, F. Boctor, and G. Laporte. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, 59:776–787, 2008.

M.-C. Bolduc, G. Laporte, J. Renaud, and F. F. Boctor. A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars. *European Journal of Operational Research*, 202(1):122–130, 2010.

T. Bosman, M. van Ee, Y. Jiao, A. Marchetti-Spaccamela, R. Ravi, and L. Stougie. Approximation algorithms for replenishment problems with fixed turnover times. In M. Bender, M. Farach-Colton, and M. Mosteiro, editors, *LATIN 2018: Theoretical Informatics. LATIN 2018. Lecture Notes in Computer Science, vol 10807*, pages 217–230. Springer, Cham, 2018.

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

B. Çavdar and J. Sokol. A distribution-free tsp tour length estimation model for random graphs. *European Journal of Operational Research*, 243(2):588–598, 2015.

P. Chandra and M. L. Fisher. Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517, 1994.

S. Chen, B. Golden, and E. Wasil. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks*, 49(4): 318–329, 2007.

T. W. Chien. Operational estimators for the length of a traveling salesman tour. *Computers & Operations Research*, 19(6):469–478, 1992.

C.-W. Chu. A heuristic algorithm for the truckload and less-than-truckload problem. *European Journal of Operational Research*, 165:657–667, 2005.

L. C. Coelho. Problem instances inventory routing, n.d. URL `http://www.leandro-coelho.com/instances/inventory-routing/`. Accessed: 24 May 2016.

L. C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565, 2013.

L. C. Coelho and G. Laporte. Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391–397, 2014.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548, 2012.

L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2014.

J.-F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo. Vehicle routing. *Handbooks in operations research and management science*, 14:367–428, 2007.

J.-F. Cordeau, D. Laganà, R. Musmanno, and F. Vocaturo. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, 55:153–166, 2015.

J.-F. Côté and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. *European Journal of Operational Research*, 198:464–469, 2009.

J. O. Cunha and R. A. Melo. On reformulations for the one-warehouse multi-retailer problem. *Annals of Operations Research*, 238(1):99–122, 2016.

S. Dabia, D. Lai, and D. Vigo. An exact algorithm for a rich vehicle routing problem with private fleet and common carrier. *Transportation Science*, 2019. Published online in Articles in Advance 21 May 2019.

C. F. Daganzo. The distance traveled to visit n points with a maximum of c stops per vehicle: An analytic model and an application. *Transportation science*, 18(4): 331–350, 1984a.

C. F. Daganzo. The length of tours in zones of different shapes. *Transportation Research Part B: Methodological*, 18(2):135–145, 1984b.

G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.

A. Das, C. Mathieu, and S. Mozes. The train delivery problem - vehicle routing meets bin packing. In K. Jansen and R. Solis-Oba, editors, *Approximation and Online Algorithms. WAOA 2010. Lecture Notes in Computer Science, vol 6534*, pages 94–105. Springer, Berlin, Heidelberg, 2011. ISBN 978-3-642-18318-8.

A. Dasci and V. Verter. A continuous model for production–distribution system design. *European Journal of Operational Research*, 129(2):287–298, 2001.

W. E. de Paepe, J. K. Lenstra, J. Sgall, R. A. Sitters, and L. Stougie. Computer-aided complexity classification of dial-a-ride problems. *INFORMS Journal on Computing*, 16(2):120–132, 2004.

G. Desaulniers. Branch-and-price-and-cut for the split-delivery vehicle routing problem with time windows. *Operations Research*, 58(1):179–192, 2010.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*. Springer Science & Business Media, 2005.

G. Desaulniers, F. Lessard, and A. Hadjar. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

G. Desaulniers, O. B. Madsen, and S. Ropke. The vehicle routing problem with time windows. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications, 2nd edition*, pages 119–159. MOS-SIAM Series on Optimization (SIAM, Philadelphia), 2014.

G. Desaulniers, J. G. Rakke, and L. C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50(3):1060–1076, 2016.

J. Desrosiers and M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 1–32. Springer, 2005.

S. Eilon, C. D. T. Watson-Gandy, and N. Christofides. *Distribution management: Mathematical Modelling & Practical Analysis*. Griffin London, 1971.

A. Federgruen and D. Simchi-Levi. Analysis of vehicle routing and inventory-routing problems. *Handbooks in operations research and management science*, 8:297–373, 1995.

A. Federgruen, J. Meissner, and M. Tzur. Progressive interval heuristics for multi-item capacitated lot-sizing problems. *Operations Research*, 55(3):490–502, 2007.

D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

M. Figliozzi. Planning approximations to the average length of vehicle routing problems with varying customer demands and routing constraints. *Transportation Research Record: Journal of the Transportation Research Board*, 2089:1–8, 2008.

A. Franceschetti, O. Jabali, and G. Laporte. Continuous approximation models in freight distribution management. *TOP*, 25(3):413–433, 2017.

C. Gahm, C. Brabänder, and A. Tuma. Vehicle routing with private fleet, multiple common carriers offering volume discounts, and rental options. *Transportation Research Part E: Logistics and Transportation Review*, 97:192–216, 2017.

M. R. Garey and D. S. Johnson. *Computers and intractability: a guide to NP-completeness*. WH Freeman New York, 1979.

H. N. Geismar, C. Sriskandarajah, and Y. Zhu. A review of operational issues in managing physical currency supply chains. *Production and Operations Management*, 26(6):976–996, 2017.

D. Goeke, T. Gschwind, and M. Schneider. Upper and lower bounds for the vehicle-routing problem with private fleet and common carrier. *Discrete Applied Mathematics*, 2018.

R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of discrete mathematics*, 5:287–326, 1979.

D. Gulczynski, B. Golden, and E. Wasil. The split delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):612–626, 2010.

M. Haimovich and A. H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.

A. F.-W. Han and Y.-C. Chu. A multi-start heuristic approach for the split-delivery vehicle routing problem with minimum delivery amounts. *Transportation Research Part E: Logistics and Transportation Review*, 88:11–31, 2016.

M. Hariga, M. Gumus, A. Daghfous, and S. K. Goyal. A vendor managed inventory model under contractual storage agreement. *Computers & Operations Research*, 40 (8):2138–2144, 2013.

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

A. Hindle and D. Worthington. Models to estimate average route lengths in different geographical environments. *Journal of the Operational Research Society*, 55(6):662–666, 2004.

R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel. The pinwheel: A real-time scheduling problem. In *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences. Volume II: Software Track*, volume 2, pages 693–702. IEEE, 1989.

M. A. Hoque. An optimal solution technique for the joint replenishment problem with storage and transport capacities and budget constraints. *European Journal of Operational Research*, 175(2):1033–1042, 2006.

Y. Huang, H. N. Geismar, D. Rajamani, S. Sethi, C. Sriskandarajah, and M. Carlos. Optimizing logistics operations in a country's currency supply network. *IISE Transactions*, 49(2):223–237, 2017.

S. Huijink. *Cooperation: Vehicle routing and outsouring, games and nucleoni*. PhD thesis, Tilburg University, 2016.

S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In *Column generation*, pages 33–65. Springer, 2005.

O. Jabali, M. Gendreau, and G. Laporte. A continuous approximation model for the fleet composition problem. *Transportation Research Part B: Methodological*, 46(10): 1591–1606, 2012.

T. Jacobs and S. Longo. A new perspective on the windows scheduling problem. arXiv:1410.7237, 2014.

M. Jepsen, B. Petersen, S. Spoorendonk, and D. Pisinger. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56 (2):497–511, 2008.

M. Jin, K. Liu, and B. Eksioglu. A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 36(2):265–270, 2008.

H.-Y. Kang, A. H. I. Lee, C.-W. Wu, and C.-H. Lee. An efficient method for dynamic-demand joint replenishment problem with multiple suppliers and multiple vehicles. *International Journal of Production Research*, 55(4):1065–1084, 2017.

R. M. Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103, 1972.

B. B. Keskin, İ. Çapar, C. R. Sox, and N. K. Freeman. An integrated load-planning algorithm for outbound logistics at webb wheel. *Interfaces*, 44(5):480–497, 2014.

M. Khouja and S. Goyal. A review of the joint replenishment problem literature: 1989–2005. *European Journal of Operational Research*, 186(1):1–16, 2008.

O. Kwon, B. Golden, and E. Wasil. Estimating the length of the optimal tsp tour: An empirical study using regression and neural networks. *Computers & Operations Research*, 22(10):1039–1046, 1995.

A. Langevin and F. Soumis. Design of multiple-vehicle delivery tours satisfying time constraints. *Transportation Research Part B: Methodological*, 23(2):123–138, 1989.

A. Langevin, P. Mbaraga, and J. F. Campbell. Continuous approximation models in freight distribution: An overview. *Transportation Research Part B: Methodological*, 30(3):163–188, 1996.

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

G. Laporte, Y. Nobert, and M. Desrochers. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.

H. Larrain, L. C. Coelho, and A. Cataldo. A variable MIP neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines. *Computers & Operations Research*, 85:22–31, 2017.

C. Larsen and M. Turkensteen. A vendor managed inventory model using continuous approximations for route length estimates and markov chain modeling for cost estimates. *International Journal of Production Economics*, 157:120–132, 2014.

J. L. Lázaro, Á. B. Jiménez, and A. Takeda. Improving cash logistics in bank branches by coupling machine learning and robust optimization. *Expert Systems With Applications*, 92:236–255, 2018.

J. Lee and B.-I. Kim. Industrial ship routing problem with split delivery and two types of vessels. *Expert Systems with Applications*, 42(22):9012–9023, 2015.

W. Lefever, E.-H. Aghezzaf, K. Hadj-Hamou, and B. Penz. Analysis of an improved branch-and-cut formulation for the inventory-routing problem with transshipment. *Computers & Operations Research*, 98:137–148, 2018.

F.-H. Liu and S.-Y. Shen. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research society*, 50(7):721–732, 1999.

M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

Z. Luo, H. Qin, W. Zhu, and A. Lim. Branch and price and cut for the split-delivery vehicle routing problem with time windows and linear weight-related cost. *Transportation Science*, 51(2):668–687, 2016.

J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2): 423–445, 2004.

N. H. Moin and S. Salhi. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, 2007.

A. Narayanan and E. P. Robinson. More on 'Models and algorithms for the dynamic-demand joint replenishment problem'. *International Journal of Production Research*, 44(2):383–397, 2006.

A. Narayanan and P. Robinson. Efficient and effective heuristics for the coordinated capacitated lot-size problem. *European Journal of Operational Research*, 203(3): 583–592, 2010.

G. L. Nemhauser and S. Park. A polyhedral approach to edge coloring. *Operations Research Letters*, 10(6):315–322, 1991.

A. L. Olsen. Inventory replenishment with interdependent ordering costs: An evolutionary algorithm solution. *International Journal of Production Economics*, 113(1): 359–369, 2008.

G. Ozbaygin, O. Karasan, and H. Yaman. New exact solution approaches for the split delivery vehicle routing problem. *EURO Journal on Computational Optimization*, 6 (1):85–115, 2018.

C. H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

J.-Y. Potvin and M.-A. Naud. Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society*, 62:326–336, 2011.

M. Rahmouni and J.-C. Hennet. Determining the optimal routes in the multi-product multi-site joint delivery problem. In *2015 International Conference on Industrial Engineering and Systems Management (IESM)*, pages 1361–1366. IEEE, 2015.

G. Righini and M. Salani. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273, 2006.

E. P. Robinson, A. Narayanan, and L.-L. Gao. Effective heuristics for the dynamic demand joint replenishment problem. *Journal of the Operational Research Society*, 58(6):808–815, 2007.

P. Robinson, A. Narayanan, and F. Sahin. Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37(1):3–15, 2009.

M. Salani and I. Vacca. Branch and price for the vehicle routing problem with discrete split deliveries and time windows. *European Journal of Operational Research*, 213 (3):470–477, 2011.

A. M. Sarmiento and R. Nagi. A review of integrated analysis of production-distribution systems. *IIE transactions*, 31(11):1061–1074, 1999.

A. Senoussi, N. K. Mouss, B. Penz, N. Brahimi, and S. Dauzère-Pérès. Modeling and solving a one-supplier multi-vehicle production-inventory-distribution problem with clustered retailers. *The International Journal of Advanced Manufacturing Technology*, 85(5):971–989, 2016.

Z.-J. M. Shen and L. Qi. Incorporating inventory and routing costs in strategic location models. *European Journal of Operational Research*, 179(2):372–389, 2007.

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

O. Solyalı and H. Süral. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. *Annals of Operations Research*, 196(1): 517–541, 2012.

M. G. Speranza. Trends in transportation and logistics. *European Journal of Operational Research*, 264(3):830–836, 2018.

R. Spliet and G. Desaulniers. The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, 244(2):379–391, 2015.

A. Stenger, D. Vigo, S. Enz, and M. Schwind. A variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, 47:64–80, 2013.

C. Tilk, A.-K. Rothenbächer, T. Gschwind, and S. Irnich. Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research*, 261(2):530–539, 2017.

P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*, volume 18 of *MOS-SIAM Series on Optimization*. SIAM, Philadelphia, PA, 2nd edition, 2014.

M. Turkensteen and A. Klose. Demand dispersion and logistics costs in one-to-many distribution systems. *European Journal of Operational Research*, 223(2):499–507, 2012.

R. G. Van Anholt. *Optimizing logistics processes in cash supply chains*. PhD thesis, Vrije Universiteit Amsterdam, 2014.

R. G. Van Anholt, L. C. Coelho, G. Laporte, and I. F. A. Vis. An inventory-routing problem with pickups and deliveries arising in the replenishment of automated teller machines. *Transportation Science*, 50(3):1077–1091, 2016.

L. M. Van der Heide, L. C. Coelho, I. F. A. Vis, and R. G. Van Anholt. Replenishment and denomination mix of automated teller machines with dynamic and stochastic demands. Technical Report CIRRELT-2017-47, CIRRELT, August 2017.

J. A. Ventura, V. A. Valdebenito, and B. Golany. A dynamic inventory model with supplier selection in a serial supply chain structure. *European Journal of Operational Research*, 230(2):258–271, 2013.

T. Vidal, N. Maculan, L. S. Ochi, and P. H. Vaz Penna. Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science*, 50(2):720–734, 2016. doi: 10.1287/trsc.2015.0584.

L. Wang, J. He, D. Wu, and Y.-R. Zeng. A novel differential evolution algorithm for joint replenishment problem under interdependence and its application. *International Journal of Production Economics*, 135(1):190–198, 2012.

I. R. Webb, B. R. Buzby, and G. M. Campbell. Cyclical schedules for the joint replenishment problem with dynamic demands. *Naval Research Logistics (NRL)*, 44(6):577–589, 1997.

S. Yan, J. C. Chu, F.-Y. Hsiao, and H.-J. Huang. A planning model and solution algorithm for multi-trip split-delivery vehicle routing and scheduling problems with time windows. *Computers & Industrial Engineering*, 87:383–393, 2015.

Appendices

A

Tour Length Approximations

Several models to approximate the length of a traveling salesman tour have been proposed in the literature. The first type of model assumes that no information is available on the customer's exact location; the second type assumes that the locations are known. Within the first type of models, Beardwood et al. [1959] presented a simple formula to approximate the length of a tour and they showed that their approximation is asymptotically equal to the shortest traveling salesman tour for random points in a given area. The tour length D for an area A and M uniformly distributed points is approximated by:

$$D \approx \phi\sqrt{AM} \tag{A.1}$$

in which ϕ is a constant, approximately 0.75 for the Euclidean space. Eilon et al. [1971] presented a more accessible proof for this formula. An extension was proposed by Daganzo [1984a] for the case in which the depot is not positioned in the same area as the customers. Therefore, a term for the line-haul distance from the depot to the customer's area is included. A variant of this formula was studied by Daganzo [1984b] who introduced a strip-strategy. In this method, non-overlapping strips of an optimized width cover the area in which the customers are located. The expected length of a route in one strip is easy to compute, hence, the routes for all strips together provide a tour length approximation.

Chien [1992] tested seven different approximations for the traveling salesman tour length that have the same functional form as (A.1). The author compared approximations that vary in the calculation of the area A and considered models both with and without an extra term for the depot. The considered shapes for the area were the smallest rectangle covering the customers, the smallest rectangle that covers both the depot and the customers, a circular sector that covers both the depot and the customers and finally a lune shaped area covering all the customers. For these models the best values for the constants were derived by testing the models on instances up to 30 customers. The approximations of Chien [1992] allow for a comparison between subsets of customers that have the same cardinality. In comparison, the approximation by Beardwood et al. [1959] in equation (A.1) assumes the same area (A) for each subset and therefore cannot be used to compare equally sized subsets of customers.

The parameters of formula (A.1) and Chien's model (1992), that includes a term for the depot, were reassessed by Kwon et al. [1995] by considering instances with up to 80 customers located in rectangular areas having different length-to-width ratios. Two new models were introduced that include this ratio and the performance of these newly introduced models seems good. However, the models have only been tested for rectangular areas and very specific information on the input, the ratio between length and width of the area, has to be known for these tour length approximations limiting their practical use. Hindle and Worthington [2004] used simulation to refine equation (A.1) for a 100 x 100 square area by including a term with the natural logarithm of the number of customers. The results indicated that the formula by Beardwood et al. [1959] could be improved by using a different functional form, however, this new result was not generalized for other sizes and shapes of areas. More recently, Çavdar and Sokol [2015] tested several existing tour length approximations, including those of Beardwood et al. [1959] and Chien [1992], and introduced a new model incorporating the standard deviation of the horizontal and vertical customer coordinates. The tested instances have different node dispersions and the areas in which the customers are located have different shapes. The computations for the newly introduced approximation are more complicated than the models proposed by Beardwood et al. [1959] and Chien [1992]. Moreover, the experiments showed that the new model accurately approximates the actual tour length for large numbers of customers, however, the approximation deviates significantly from the optimal tour length for small numbers of customers.

The previous mentioned models are all approximations for the length of a single tour. Extensions to multiple, capacitated, vehicles can be found in Daganzo [1984a]; Langevin and Soumis [1989]; Figliozzi [2008] and Turkensteen and Klose [2012]. These publications include similar ideas as in the single tour approximation models and they are extended to handle multiple tours. Note that Langevin and Soumis [1989] also considered a time constraint on the tours.

Applications of the above mentioned tour length approximation models are mainly found in optimization of passenger transportation systems [Langevin et al. 1996] and in location optimization models [Shen and Qi 2007]. Shen and Qi [2007] used an approximation for the Vehicle Routing Problem (VRP) by Haimovich and Rinnooy Kan [1985] that requires the length of a tour as input, which in turn is approximated by (A.1). Other applications include fleet composition models, e.g., Jabali et al. [2012] who applied a VRP approximation, and production and distribution system design, e.g., Dasci and Verter [2001] who used the approximation in equation (A.1). For a more elaborate overview of continuous approximation models and applications we refer to Franceschetti et al. [2017].

B

Computational Complexity DJRP-AT

Consider problem formulation (3.3a)-(3.3h) with the following function for the approximated transportation cost:

$$f(s) = B + \phi\sqrt{AM} \tag{B.1}$$

in which A is the area of the predefined rectangle in which all customers and the depot in the instance are located and M is the number of points in the tour (depot and customers). This means A is identical for every subset of customers s. Furthermore, assume the inventory holding rates are zero and that $B = 0$. The objective function of the DJRP-AT becomes $\sum_{t \in T} \phi\sqrt{AM_t} = \sum_{t \in T} \phi'\sqrt{M_t}$ with M_t the number of points visited in period $t \in T$ ($\phi' = \phi\sqrt{A}$). This objective is minimized if $\sum_t M_t$ is minimal.

Now, consider two different types of subset composition constraints: limited tour duration and a maximum on the number of customers in the subset. For this cost structure, these two side constraints are equivalent. The maximum number of customers in a set is imposed by $M_t \leq k_M + 1$. The duration is computed by the second part of (B.1), hence, given the fixed value of A the constraint $\phi'\sqrt{M_t} \leq k'_D$ can be written as $M_t \leq k_D = (k'_D/\phi')^2$. Therefore the structure of the constraint is the same for a maximum on the tour duration and on the number of customers.

Considering the functional forms of the objective and the constraints, the problem is to minimize the sum of the number of visits per period under the constraints that a maximum number of customers can be visited per period and that customers cannot run out of stock. A special case of this problem is when only one customer can be visited per period ($k_M = 1$), also known as the so-called Pinwheel Scheduling Problem (or Windows Scheduling Problem). In the Pinwheel Scheduling Problem, a feasible schedule must be found to repeatedly process a set of jobs; for each job j a time limit between two executions is given which is the period of a job p_j. This is similar to replenishing customers in such a way that they do not run out of stock: after a replenishment, calculations can determine the latest possible timing of the next delivery. That Pinwheel Scheduling is a special case of the DJRP-AT can be seen by assuming that inventory holding costs are zero, $B = 0$ and by introducing customers who must be replenished every period, whose locations define the rectangle containing all customers. Then, the area in the transportation cost function is the same in every period, which means that selecting which customers to replenish is based solely on inventory levels and the maximum number of customers to replenish. This requires finding a feasible replenishment schedule, which is equivalent to the Pinwheel Scheduling Problem.

It was recently shown by Jacobs and Longo [2014] that the Pinwheel Scheduling Problem cannot be solved in pseudopolynomial time, unless there is a randomized algorithm solving the classical problem Satisfiability in quasipolynomial time. Since this is unlikely, it is plausible that the Pinwheel Scheduling Problem is not solvable in polynomial time. Therefore, a final conclusion on the complexity of the DJRP-AT with cost function (B.1) cannot be given, but it is very unlikely that this problem is mathematically easy.

Now consider the case in which the area in the cost function $R(s)$ is the smallest rectangle that covers the depot and the customers in the subset. Then $R(s)$ is dependent on the specific subset s that must be visited. The problem with A equal to the complete area, is a special case of the problem with the smallest rectangle $R(s)$. This can be easily seen by creating some customers in the corners of the complete area requiring replenishment every period; then the smallest rectangle is equal to the complete area in every period, resulting in a reduction from one problem to the other. Therefore, if the problem is NP-complete with A (the whole area), then the problem is also NP-complete with $R(s)$ (the smallest rectangle covering the points).

C

Derivation $\Delta(L, L')$

The value of $\Delta(L, L')$ in condition P.3 of Proposition 1 should be such that for every possible extension $P \subseteq \mathcal{N} \setminus s'$ condition D.2 holds. Therefore, we examine $\bar{c}_t(L \oplus P)$ and $\bar{c}_t(L' \oplus P)$. For ease of notation denote $s(L)$ by s and let $R(s)$ denote the area of the smallest rectangle covering the depot and the customers in s. Furthermore, define $\pi_j := \pi^1_{jt}$ and $\pi_t := \pi^2_t$.

$$\begin{aligned}
\bar{c}_t(L \oplus P) &= \phi\sqrt{R(s \cup P)|s \cup P|} + \sum_{j \in s} u_j \pi_j + \sum_{j \in P} u_j \pi_j - \pi_t \\
&= \phi\sqrt{(R(s \cup P) - R(s))|s \cup P| + R(s)|s| + R(s)|P|} \\
&\qquad + \sum_{j \in s} u_j \pi_j + \sum_{j \in P} u_j \pi_j - \pi_t \\
&\leq \phi\sqrt{(R(s \cup P) - R(s))|s \cup P|} + \phi\sqrt{R(s)|s|} + \phi\sqrt{R(s)|P|} \\
&\qquad + \sum_{j \in s} u_j \pi_j + \sum_{j \in P} u_j \pi_j - \pi_t \\
&= \bar{c}_t(L) + \phi\sqrt{(R(s \cup P) - R(s))|s \cup P|} + \phi\sqrt{R(s)|P|} + \sum_{j \in P} u_j \pi_j \qquad \text{(C.1)}
\end{aligned}$$

and similarly

$$\bar{c}_t(L' \oplus P) \leq \bar{c}_t(L') + \phi\sqrt{(R(s' \cup P) - R(s'))|s' \cup P|} + \phi\sqrt{R(s')|P|} + \sum_{j \in P} u_j \pi_j \qquad \text{(C.2)}$$

Hence, we can express $\bar{c}_t(L \oplus P)$ and $\bar{c}_t(L' \oplus P)$ in terms of $\bar{c}_t(L)$ and $\bar{c}_t(L')$. It is already known that $\bar{c}_t(L) < \bar{c}_t(L')$, for dominance also $\bar{c}_t(L \oplus P) \leq \bar{c}_t(L' \oplus P)$ has to hold.

$$\begin{aligned}
&\bar{c}_t(L \oplus P) - \bar{c}_t(L' \oplus P) \\
&\qquad = \bar{c}_t(L) + \phi\sqrt{(R(s \cup P) - R(s))|s \cup P|} + \phi\sqrt{R(s)|P|} + \sum_{j \in P} u_j \pi_j \\
&\qquad\quad - \bar{c}_t(L') - \phi\sqrt{(R(s' \cup P) - R(s'))|s' \cup P|} - \phi\sqrt{R(s')|P|} - \sum_{j \in P} u_j \pi_j \\
&\qquad = \bar{c}_t(L) - \bar{c}_t(L') + \boldsymbol{\phi\sqrt{(R(s \cup P) - R(s))|s \cup P|}} \\
&\qquad\quad \boldsymbol{-\phi\sqrt{(R(s' \cup P) - R(s'))|s' \cup P|} + \phi\sqrt{R(s)|P|} - \phi\sqrt{R(s')|P|}}
\end{aligned} \tag{C.3}$$

To find a dominance rule, an upper bound (UB) on the bold part in the last expression must be determined to guarantee dominance.

$$\begin{aligned}
UB &= \phi\sqrt{(R(s \cup P) - R(s))|s \cup P|} - \phi\sqrt{(R(s' \cup P) - R(s'))|s' \cup P|} \\
&\quad + \phi\sqrt{R(s)|P|} - \phi\sqrt{R(s')|P|} \\
&\leq \phi\sqrt{(R(s \cup P) - R(s))|s \cup P|} - \phi\sqrt{(R(s' \cup P) - R(s'))|s' \cup P|} \\
&\leq \phi\sqrt{(R(s \cup P) - R(s))|s \cup P| - (R(s' \cup P) - R(s'))|s' \cup P|} \\
&= \phi\sqrt{R(s \cup P)|s \cup P| - R(s' \cup P)|s' \cup P| + R(s')|s' \cup P| - R(s)|s \cup P|} \\
&\leq \phi\sqrt{R(s')|s' \cup P| - R(s)|s \cup P|}
\end{aligned} \tag{C.4}$$

The first inequality follows from $R(s) - R(s') < 0$ and the second inequality follows from $\sqrt{a} - \sqrt{b} \leq \sqrt{a - b}$ given that $a \geq b \geq 0$. Then, by rearranging terms the equality is found and the last inequality follows from $R(s \cup P) \leq R(s' \cup P)$ and $|s \cup P| < |s' \cup P|$. Hence,

$$\Delta(L, L') = \phi\sqrt{R(s')|s' \cup P| - R(s)|s \cup P|}. \tag{C.5}$$

D

Results per instance DJRP-AT

Tables D.1 - D.7 show the results per instance for the DJRP-AT. In Tables D.1 - D.4 results are presented for the model with a constraint on the duration; the remaining tables show the results of the model with limit on the number of customers served per period. In all tables, the instance number, number of customers N, number of periods T, and the upper bound k on the extra constraint are given. Furthermore, the solution time ('Time (s)') in seconds, the objective value ('DJRP-AT Solution'), the size of the branch-and-bound tree ('Tree'), the number of columns in the final model ('Cols') and the integrality gap ('Gap (%)') are presented. For all tables, the total costs computed with the traveling salesman tour solution are given in the column indicated by 'TSP Solution'. For Tables D.1 - D.4 the final column 'Difference' indicates the percentage difference between the model objective value and the actual costs with the tour costs. This difference indicates the cost underestimation of the route length approximation, as opposed to the actual shortest tour. In Tables D.5 - D.7 a comparison with the solution of the DJRP is shown for individual replenishment costs $m = 25$, 100, prop, zones, quad by giving the total costs and the percentage difference with the DJRP-AT solution ('TSP Sol'). In Table D.8 the results per instance of the comparison between the DJRP-AT and the equivalent IRP are presented.

Table D.1 Results per instance for duration constraint for $T = 3$ and $N = 5,\ 10$.

Instance	N	T	k_D	Time (s)	DJRP-AT Solution	Tree	Columns	Gap (%)	TSP Solution	Difference (%)
abs1n5	5	3	600	0	2868	12	65	15	3471	−17.4
abs2n5	5	3	600	0	2628	6	58	24	3146	−16.5
abs3n5	5	3	600	0	4310	2	30	21	5250	−17.9
abs4n5	5	3	600	0	2657	4	52	21	3050	−12.9
abs5n5	5	3	600	0	1753	0	40	0	2030	−13.7
abs1n5	5	3	800	0	2798	8	68	22	3391	−17.5
abs2n5	5	3	800	0	1704	0	44	0	1975	−13.7
abs3n5	5	3	800	1	3330	6	53	12	4169	−20.1
abs4n5	5	3	800	0	1746	0	30	0	1983	−11.9
abs5n5	5	3	800	0	1753	0	40	0	2030	−13.7
abs1n5	5	3	1000	0	1892	0	59	0	2225	−15.0
abs2n5	5	3	1000	0	1704	0	44	0	1975	−13.7
abs3n5	5	3	1000	0	3330	20	64	24	4169	−20.1
abs4n5	5	3	1000	0	1746	0	30	0	1983	−11.9
abs5n5	5	3	1000	0	1753	0	40	0	2030	−13.7
abs1n5	5	3	1200	0	1892	0	59	0	2225	−15.0
abs2n5	5	3	1200	0	1704	0	44	0	1975	−13.7
abs3n5	5	3	1200	0	2240	0	37	0	2592	−13.6
abs4n5	5	3	1200	0	1746	0	30	0	1983	−11.9
abs5n5	5	3	1200	0	1753	0	40	0	2030	−13.7
abs1n10	10	3	600	1	4802	18	311	12	5935	−19.1
abs2n10	10	3	600	0	x	2	144	-	-	-
abs3n10	10	3	600	0	4272	6	205	9	5485	−22.1
abs4n10	10	3	600	0	x	2	199	-	-	-
abs5n10	10	3	600	1	4736	78	468	23	5481	−13.6
abs1n10	10	3	800	0	3642	6	306	3	4332	−15.9
abs2n10	10	3	800	2	5192	40	401	25	6422	−19.1
abs3n10	10	3	800	0	3429	0	304	0	4073	−15.8
abs4n10	10	3	800	1	4974	94	660	29	6128	−18.8
abs5n10	10	3	800	1	4678	24	668	30	5442	−14.0
abs1n10	10	3	1000	0	3642	14	808	13	4332	−15.9
abs2n10	10	3	1000	1	3938	12	750	9	4585	−14.1
abs3n10	10	3	1000	0	3429	6	760	8	4073	−15.8
abs4n10	10	3	1000	1	4820	38	1210	34	5783	−16.7
abs5n10	10	3	1000	1	3591	12	847	19	4107	−12.6
abs1n10	10	3	1200	0	3474	8	1077	18	4406	−21.2
abs2n10	10	3	1200	1	3830	20	1051	16	4452	−14.0
abs3n10	10	3	1200	0	3429	6	1238	12	4073	−15.8
abs4n10	10	3	1200	1	3572	20	1350	22	4237	−15.7
abs5n10	10	3	1200	0	2520	0	1034	0	2728	−7.6

x: instance infeasible.

Table D.2 Results per instance for duration constraint for $T = 3$ and $N = 15,\ 20$.

Instance	N	T	k_D	Time (s)	DJRP-AT Solution	Tree	Columns	Gap (%)	TSP Solution	Difference (%)
abs1n15	15	3	600	0	x	2	607	-	-	-
abs2n15	15	3	600	1	x	2	541	-	-	-
abs3n15	15	3	600	0	x	2	531	-	-	-
abs4n15	15	3	600	0	x	2	484	-	-	-
abs5n15	15	3	600	1	x	2	491	-	-	-
abs1n15	15	3	800	10	5361	108	2901	19	6208	−13.6
abs2n15	15	3	800	5	5498	50	1991	17	6182	−11.1
abs3n15	15	3	800	1	x	14	1082	-	-	-
abs4n15	15	3	800	1	x	6	1047	-	-	-
abs5n15	15	3	800	3	5411	46	1888	15	6067	−10.8
abs1n15	15	3	1000	6	5159	28	6395	26	5698	−9.5
abs2n15	15	3	1000	8	5387	124	6223	25	5972	−9.8
abs3n15	15	3	1000	7	5480	52	2389	15	6580	−16.7
abs4n15	15	3	1000	11	5438	200	3914	27	6061	−10.3
abs5n15	15	3	1000	12	5401	164	6278	24	6038	−10.5
abs1n15	15	3	1200	11	4194	80	21 348	18	4873	−13.9
abs2n15	15	3	1200	3	4477	14	12 174	16	4753	−5.8
abs3n15	15	3	1200	1	4394	10	4647	6	5035	−12.7
abs4n15	15	3	1200	12	5401	78	13 773	31	5824	−7.3
abs5n15	15	3	1200	6	4234	24	15 278	15	4588	−7.7
abs1n20	20	3	600	1	x	2	490	-	-	-
abs2n20	20	3	600	0	x	2	1393	-	-	-
abs3n20	20	3	600	0	x	2	776	-	-	-
abs4n20	20	3	600	1	x	2	603	-	-	-
abs5n20	20	3	600	0	x	2	450	-	-	-
abs1n20	20	3	800	1	x	2	1373	-	-	-
abs2n20	20	3	800	22	5755	110	7714	10	6518	−11.7
abs3n20	20	3	800	1	x	2	1909	-	-	-
abs4n20	20	3	800	1	x	2	2137	-	-	-
abs5n20	20	3	800	1	x	2	1520	-	-	-
abs1n20	20	3	1000	34	x	94	8085	-	-	-
abs2n20	20	3	1000	268	5648	530	33 518	19	6228	−9.3
abs3n20	20	3	1000	18	6161	36	12 315	16	7073	−12.9
abs4n20	20	3	1000	4	x	10	9090	-	-	-
abs5n20	20	3	1000	1	x	2	9487	-	-	-
abs1n20	20	3	1200	2122	6312	984	107 421	27	6944	−9.1
abs2n20	20	3	1200	138	5546	34	72 993	20	6160	−10.0
abs3n20	20	3	1200	248	5979	148	63 819	25	6649	−10.1
abs4n20	20	3	1200	1235	6532	984	71 117	26	7341	−11.0
abs5n20	20	3	1200	4479	6714	3540	145 401	20	7592	−11.6

x: instance infeasible.

Table D.3 Results per instance for duration constraint for $T = 6$ and $N = 5,\ 10$.

Instance	N	T	k_D	Time (s)	DJRP-AT Solution	Tree	Columns	Gap (%)	TSP Solution	Difference (%)
abs1n5	5	6	600	1	6252	28	106	3	7518	−16.8
abs2n5	5	6	600	1	5527	52	145	16	6694	−17.4
abs3n5	5	6	600	0	9178	0	36	0	11 474	−20.0
abs4n5	5	6	600	0	6061	18	82	5	7255	−16.5
abs5n5	5	6	600	0	4481	2	109	0	5281	−15.1
abs1n5	5	6	800	1	6032	26	117	12	7305	−17.4
abs2n5	5	6	800	0	4565	0	74	0	5462	−16.4
abs3n5	5	6	800	0	8163	46	97	9	10 328	−21.0
abs4n5	5	6	800	0	5126	2	91	1	5916	−13.3
abs5n5	5	6	800	0	4481	2	109	0	5281	−15.1
abs1n5	5	6	1000	0	5156	2	109	1	6124	−15.8
abs2n5	5	6	1000	0	4565	0	74	0	5462	−16.4
abs3n5	5	6	1000	1	7223	58	129	11	9033	−20.0
abs4n5	5	6	1000	0	5126	2	91	1	5916	−13.3
abs5n5	5	6	1000	0	4481	2	109	0	5281	−15.1
abs1n5	5	6	1200	0	5156	2	109	1	6124	−15.8
abs2n5	5	6	1200	0	4565	0	74	0	5462	−16.4
abs3n5	5	6	1200	0	6149	0	69	0	7235	−15.0
abs4n5	5	6	1200	0	5126	2	91	1	5916	−13.3
abs5n5	5	6	1200	1	4481	2	109	0	5281	−15.1
abs1n10	10	6	600	6	x	28	479	-	-	-
abs2n10	10	6	600	1	x	2	194	-	-	-
abs3n10	10	6	600	39	x	280	759	-	-	-
abs4n10	10	6	600	1	x	2	255	-	-	-
abs5n10	10	6	600	39	9785	266	899	8	11 330	−13.6
abs1n10	10	6	800	16	8699	170	974	7	10 203	−14.7
abs2n10	10	6	800	20	10 435	132	854	12	12 832	−18.7
abs3n10	10	6	800	3	8456	34	752	5	9997	−15.4
abs4n10	10	6	800	4	10 044	36	653	5	12 388	−18.9
abs5n10	10	6	800	105	9681	678	1546	17	11 303	−14.3
abs1n10	10	6	1000	8	7832	66	1384	7	8828	−11.3
abs2n10	10	6	1000	26	9225	208	1343	11	11 047	−16.5
abs3n10	10	6	1000	9	7674	58	2090	6	8790	−12.7
abs4n10	10	6	1000	13	9988	102	1307	16	12 187	−18.0
abs5n10	10	6	1000	30	7904	184	2207	10	8885	−11.0
abs1n10	10	6	1200	16	7672	152	2182	14	8951	−14.3
abs2n10	10	6	1200	20	8174	130	2084	9	9510	−14.0
abs3n10	10	6	1200	5	7467	72	2729	11	8670	−13.9
abs4n10	10	6	1200	19	8095	148	2179	9	9334	−13.3
abs5n10	10	6	1200	1	6785	10	1698	3	7526	−9.8

x: instance infeasible.

Table D.4 Results per instance for duration constraint for $T = 6$ and $N = 15,\ 20$.

Instance	N	T	k_D	Time (s)	DJRP-AT Solution	Tree	Columns	Gap (%)	TSP Solution	Difference (%)
abs1n15	15	6	600	2	x	2	906	-	-	-
abs2n15	15	6	600	0	x	2	665	-	-	-
abs3n15	15	6	600	2	x	2	650	-	-	-
abs4n15	15	6	600	1	x	2	707	-	-	-
abs5n15	15	6	600	1	x	2	565	-	-	-
abs1n15	15	6	800	7201	+	12 454	29 997	-	-	-
abs2n15	15	6	800	123	x	392	4239	-	-	-
abs3n15	15	6	800	1	x	2	1338	-	-	-
abs4n15	15	6	800	2	x	2	1499	-	-	-
abs5n15	15	6	800	23	x	76	3169	-	-	-
abs1n15	15	6	1000	7201	+	17 038	32 513	-	-	-
abs2n15	15	6	1000	154	11 027	532	7814	11	12 274	−10.2
abs3n15	15	6	1000	7201	+	19 738	18 391	-	-	-
abs4n15	15	6	1000	48	11 018	156	6099	9	12 132	−9.2
abs5n15	15	6	1000	259	11 106	602	11 452	11	12 379	−10.3
abs1n15	15	6	1200	1851	9194	3610	47 858	13	10 180	−9.7
abs2n15	15	6	1200	7201	+	15 552	42 386	-	-	-
abs3n15	15	6	1200	7203	+	11 598	46 260	-	-	-
abs4n15	15	6	1200	3312	11 018	7030	37 617	15	12 132	−9.2
abs5n15	15	6	1200	132	9322	334	23 432	9	9682	−3.7
abs1n20	20	6	600	1	x	2	591	-	-	-
abs2n20	20	6	600	3	x	2	2306	-	-	-
abs3n20	20	6	600	2	x	2	758	-	-	-
abs4n20	20	6	600	1	x	2	605	-	-	-
abs5n20	20	6	600	0	x	2	342	-	-	-
abs1n20	20	6	800	2	x	2	1617	-	-	-
abs2n20	20	6	800	127	x	170	21 133	-	-	-
abs3n20	20	6	800	3	x	2	2933	-	-	-
abs4n20	20	6	800	3	x	2	2822	-	-	-
abs5n20	20	6	800	2	x	2	1550	-	-	-
abs1n20	20	6	1000	4	x	2	8483	-	-	-
abs2n20	20	6	1000	7201	+	4702	121 980	-	-	-
abs3n20	20	6	1000	1850	x	1716	35 683	-	-	-
abs4n20	20	6	1000	4	x	2	9973	-	-	-
abs5n20	20	6	1000	6	x	2	14 757	-	-	-
abs1n20	20	6	1200	7202	+	1596	151 670	-	-	-
abs2n20	20	6	1200	7203	+	1010	240 387	-	-	-
abs3n20	20	6	1200	7202	+	1862	233 434	-	-	-
abs4n20	20	6	1200	7201	+	2894	145 444	-	-	-
abs5n20	20	6	1200	7201	+	2788	239 743	-	-	-

x: instance infeasible. +: instance not solved within two hours.

Table D.5 Results per instance for maximum on number of customers constraint for $T = 3$ and $N = 5,\ 10$.

										DJRP Solution					Difference (%)				
Instance	N	T	k_M	Time (s)	DJRP-AT Solution	Tree	Cols	Gap (%)	TSP Sol	25	100	m prop	zones	quad	25	100	m prop	zones	quad
abs1n5	5	3	3	0	2868	12	62	11	3471	3581	3581	3581	3581	3581	3.1	3.1	3.1	3.1	3.1
abs2n5	5	3	3	0	2628	6	59	9	3146	3413	3413	3413	3413	3413	7.8	7.8	7.8	7.8	7.8
abs3n5	5	3	3	0	3330	8	53	10	4169	4568	4169	4169	4140	4140	8.7	0.0	0.0	−0.7	−0.7
abs4n5	5	3	3	0	3740	10	63	36	4335	4335	4335	4335	4335	4335	0.0	0.0	0.0	0.0	0.0
abs5n5	5	3	3	0	2673	6	57	11	3260	3246	3246	3246	3246	3246	−0.4	−0.4	−0.4	−0.4	−0.4
abs1n5	5	3	4	1	2798	8	68	22	3391	3581	3581	3583	3581	3581	5.3	5.3	5.4	5.3	5.3
abs2n5	5	3	4	0	2628	6	58	24	3146	3392	3392	3392	3392	3392	7.2	7.2	7.2	7.2	7.2
abs3n5	5	3	4	0	3330	20	64	24	4169	4563	4169	4169	4127	4127	8.6	0.0	0.0	−1.0	−1.0
abs4n5	5	3	4	0	2657	4	55	24	3050	3050	3050	3050	3050	3050	0.0	0.0	0.0	0.0	0.0
abs5n5	5	3	4	0	2668	6	67	25	3187	3187	3187	3187	3187	3187	0.0	0.0	0.0	0.0	0.0
abs1n10	10	3	5	0	3737	4	531	2	4315	4315	4315	4315	4315	4315	0.0	0.0	0.0	0.0	0.0
abs2n10	10	3	5	1	4045	6	675	5	4842	4842	4842	4842	4842	4842	0.0	0.0	0.0	0.0	0.0
abs3n10	10	3	5	0	3429	0	676	0	4073	4502	4502	4502	4502	4502	9.5	9.5	9.5	9.5	9.5
abs4n10	10	3	5	1	5003	58	979	28	6163	6297	6654	6032	6032	6124	2.1	7.4	−2.2	−2.2	−0.6
abs5n10	10	3	5	1	3859	8	545	6	4351	4351	4351	4351	4351	4351	0.0	0.0	0.0	0.0	0.0
abs1n10	10	3	6	0	3642	8	707	8	4332	4315	4315	4315	4315	4315	−0.4	−0.4	−0.4	−0.4	−0.4
abs2n10	10	3	6	1	3939	16	788	11	4559	4830	4842	4842	4830	4842	5.6	5.8	5.8	5.6	5.8
abs3n10	10	3	6	0	3429	6	962	3	4073	4386	4386	4386	4386	4386	7.1	7.1	7.1	7.1	7.1
abs4n10	10	3	6	1	4820	18	1061	32	5783	5785	6258	5665	5785	5665	0.0	7.6	−2.1	0.0	−2.1
abs5n10	10	3	6	0	3652	12	820	10	4192	4436	4351	4351	4431	4374	5.5	3.7	3.7	5.4	4.2
abs1n10	10	3	7	1	3642	16	1035	14	4332	4315	4315	4315	4315	4315	−0.4	−0.4	−0.4	−0.4	−0.4
abs2n10	10	3	7	1	3830	16	1018	15	4452	4830	4842	4842	4830	4842	7.8	8.1	8.1	7.8	8.1
abs3n10	10	3	7	0	3429	6	1598	9	4073	4399	4386	4386	4386	4386	7.4	7.1	7.1	7.1	7.1
abs4n10	10	3	7	1	3716	16	1178	18	4336	4339	4336	4336	4339	4339	0.1	0.0	0.0	0.1	0.1
abs5n10	10	3	7	0	3591	10	961	15	4107	4458	4351	4351	4431	4374	7.9	5.6	5.6	7.3	6.1
abs1n10	10	3	8	0	3545	8	1117	18	4299	4315	4315	4315	4315	4315	0.4	0.4	0.4	0.4	0.4
abs2n10	10	3	8	1	3719	14	1080	18	4430	4830	4842	4842	4830	4842	8.3	8.5	8.5	8.3	8.5
abs3n10	10	3	8	0	3429	6	1366	13	4073	4399	4386	4386	4386	4386	7.4	7.1	7.1	7.1	7.1
abs4n10	10	3	8	1	3572	22	1262	20	4237	4339	4336	4336	4339	4339	2.3	2.3	2.3	2.3	2.3
abs5n10	10	3	8	0	3543	8	1080	19	3962	4458	4351	4351	4431	4374	11.1	8.9	8.9	10.6	9.4

Table D.6 Results per instance for maximum on number of customers constraint for $T = 3$ and $N = 15,\ 20$.

Instance	N	T	k_M	Time (s)	DJRP-AT Solution	Tree	Cols	Gap (%)	TSP Sol	DJRP Solution 25	DJRP Solution 100	DJRP Solution m prop	DJRP Solution zones	DJRP Solution quad	Difference (%) 25	Difference (%) 100	Difference (%) m prop	Difference (%) zones	Difference (%) quad
abs1n15	15	3	7	20	5228	178	5765	19	5843	6422	6571	6108	6225	6108	9.0	11.1	4.3	6.1	4.3
abs2n15	15	3	7	15	5396	146	5061	19	5965	6793	6316	6674	6405	6397	12.2	5.6	10.6	6.9	6.8
abs3n15	15	3	7	7	5419	66	2974	16	6006	6729	6729	6729	6729	6729	10.8	10.8	10.8	10.8	10.8
abs4n15	15	3	7	22	5499	154	6378	21	5972	6781	6703	6645	6756	6799	11.9	10.9	10.1	11.6	12.2
abs5n15	15	3	7	24	5465	186	10 481	21	5996	6646	6555	6561	6521	6455	9.8	8.5	8.6	8.1	7.1
abs1n15	15	3	8	7	4198	46	9429	10	4862	4836	4862	4836	4836	4836	−0.5	0.0	−0.5	−0.5	-0.5
abs2n15	15	3	8	4	4477	18	12 731	9	4753	4763	4753	4753	4753	5019	0.2	0.0	0.0	0.0	5.3
abs3n15	15	3	8	2	4416	14	7909	3	4967	5323	4989	4989	4989	4989	6.7	0.4	0.4	0.4	0.4
abs4n15	15	3	8	25	5453	188	14 584	27	5957	6112	6164	5978	6120	6120	2.5	3.4	0.4	2.7	2.7
abs5n15	15	3	8	12	5282	102	11 395	23	6155	6550	6303	5975	6000	6000	6.0	2.3	−3.0	−2.6	−2.6
abs1n15	15	3	9	14	4198	100	17 735	15	4636	4863	4862	4836	4863	4836	4.7	4.6	4.1	4.7	4.1
abs2n15	15	3	9	5	4403	24	14 836	13	4735	5017	4753	4753	4753	5019	5.6	0.4	0.4	0.4	5.7
abs3n15	15	3	9	5	4400	16	12 516	8	4994	5193	4921	4921	4984	4984	3.8	−1.5	−1.5	−0.2	−0.2
abs4n15	15	3	9	6	4478	26	17 787	16	4813	4918	4813	4813	4927	4927	2.1	0.0	0.0	2.3	2.3
abs5n15	15	3	9	5	4265	16	16 694	9	4655	4729	4655	4655	4719	4692	1.6	0.0	0.0	1.4	0.8
abs1n15	15	3	10	2	3911	16	17 735	13	4394	4863	4862	4836	4863	4836	9.6	9.6	9.1	9.6	9.1
abs2n15	15	3	10	3	4272	16	22 755	14	4463	5017	4753	4753	4753	5019	11.0	6.1	6.1	6.1	11.1
abs3n15	15	3	10	4	4370	16	14 612	11	4970	5188	4921	4921	4965	4965	4.2	−1.0	−1.0	−0.1	−0.1
abs4n15	15	3	10	14	4451	60	24 897	19	4767	4918	4813	4813	4927	4927	3.1	1.0	1.0	3.2	3.2
abs5n15	15	3	10	8	4234	32	26 594	14	4588	4706	4655	4655	4719	4692	2.5	1.4	1.4	2.8	2.2
abs1n15	15	3	11	4	3894	26	24 961	15	4164	4863	4862	4836	4863	4836	14.4	14.3	13.9	14.4	13.9
abs2n15	15	3	11	4	4221	20	27 849	16	4417	5017	4753	4753	4753	5019	12.0	7.1	7.1	7.1	12.0
abs3n15	15	3	11	7	4348	24	19 962	15	5054	5169	4921	4921	4897	4897	2.2	−2.7	−2.7	−3.2	−3.2
abs4n15	15	3	11	10	4285	42	21 741	20	4627	4918	4813	4813	4927	4927	5.9	3.9	3.9	6.1	6.1
abs5n15	15	3	11	6	4135	26	21 417	16	4410	4706	4655	4655	4719	4692	6.3	5.3	5.3	6.5	6.0
abs1n20	20	3	10	1536	5980	196	257 354	27	6257	6837	6837	6257	6696	6257	8.5	8.5	0.0	6.6	0.0
abs2n20	20	3	10	55	4661	4	159 085	3	4760	5211	5211	5211	5211	5211	8.7	8.7	8.7	8.7	8.7
abs3n20	20	3	10	207	4886	22	321 432	8	5418	5496	5496	5496	5496	5496	1.4	1.4	1.4	1.4	1.4
abs4n20	20	3	10	4121	6033	640	282 808	21	7157	7768	7604	7100	7209	7129	7.9	5.9	−0.8	0.7	−0.4
abs5n20	20	3	10	2116	6202	296	671 812	16	6929	7502	6929	6929	7247	7247	7.6	0.0	0.0	4.4	4.4
abs1n20	20	3	11	2060	5090	146	294 071	9	5158	5172	5158	5158	5172	5172	0.3	0.0	0.0	0.3	0.3
abs2n20	20	3	11	322	4655	18	160 968	8	4815	5157	5202	5202	5247	5247	6.6	7.4	7.4	8.2	8.2
abs3n20	20	3	11	612	4870	38	237 531	9	5416	5550	5543	5543	5543	5543	2.4	2.3	2.3	2.3	2.3
abs4n20	20	3	11	7206	+	492	506 631	-	-	-	-	-	-	-	-	-	-	-	-
abs5n20	20	3	11	425	5108	28	425 524	5	5746	5717	5746	5746	5746	5746	−0.5	0.0	0.0	0.0	0.0
abs1n20	20	3	12	3686	5038	146	723 553	12	5150	5172	5158	5158	5172	5172	0.4	0.1	0.1	0.4	0.4
abs2n20	20	3	12	899	4655	38	244 041	8	4815	5261	5082	5082	5238	5238	8.5	5.2	5.2	8.1	8.1
abs3n20	20	3	12	1283	4870	48	397 099	11	5416	5597	5540	5540	5540	5540	3.2	2.2	2.2	2.2	2.2
abs4n20	20	3	12	1550	5153	58	713 459	15	5707	5819	5707	5707	5707	5707	1.9	0.0	0.0	0.0	0.0
abs5n20	20	3	12	1454	5106	64	833 293	9	5743	5693	5746	5746	5746	5746	−0.9	0.0	0.0	0.0	0.0
abs1n20	20	3	13	3388	4991	112	990 314	14	5151	5172	5158	5158	5172	5172	0.4	0.1	0.1	0.4	0.4
abs2n20	20	3	13	559	4632	16	253 365	10	4721	5252	5082	5082	5118	5118	10.1	7.1	7.1	7.8	7.8
abs3n20	20	3	13	423	4690	14	280 999	10	5051	5594	5540	5540	5540	5540	9.7	8.8	8.8	8.8	8.8
abs4n20	20	3	13	1673	5068	58	486 108	17	5654	5819	5707	5707	5707	5707	2.8	0.9	0.9	0.9	0.9
abs5n20	20	3	13	3730	5106	136	901 660	13	5743	5693	5746	5746	5746	5746	−0.9	0.0	0.0	0.0	0.0

+: instance not solved within two hours.

Table D.7 Results per instance for maximum on number of customers constraint for $T = 6$ and $N = 5,\ 10,\ 15$.

										DJRP Solution					Difference (%)				
Instance	N	T	k_M	**Time (s)**	**DJRP-AT Solution**	**Tree**	**Cols**	**Gap (%)**	**TSP Sol**	**25**	**100**	**m prop**	**zones**	**quad**	**25**	**100**	**m prop**	**zones**	**quad**
abs1n5	5	6	3	0	6161	6	91	1	7430	7783	7518	7783	7783	7783	4.5	1.2	4.5	4.5	4.5
abs2n5	5	6	3	0	5527	18	99	5	6694	7052	7052	7052	7052	7052	5.1	5.1	5.1	5.1	5.1
abs3n5	5	6	3	0	8059	28	94	6	9866	11 246	9866	10 396	10 398	10 398	12.3	0.0	5.1	5.1	5.1
abs4n5	5	6	3	1	8031	26	113	16	9510	10 125	10 107	10 182	10 182	10 182	6.1	5.9	6.6	6.6	6.6
abs5n5	5	6	3	1	5460	20	112	6	6680	6750	6721	6721	6750	6750	1.0	0.6	0.6	1.0	1.0
abs1n5	5	6	4	0	6032	26	117	12	7305	7789	7623	7619	7623	7623	6.2	4.2	4.1	4.2	4.2
abs2n5	5	6	4	0	5527	52	145	16	6694	6932	6941	6941	6932	6941	3.4	3.6	3.6	3.4	3.6
abs3n5	5	6	4	1	7223	58	129	11	9033	9444	9019	9045	9020	9020	4.3	−0.2	0.1	−0.1	−0.1
abs4n5	5	6	4	1	6061	26	113	7	7255	7352	7352	7150	7150	7150	1.3	1.3	−1.5	−1.5	−1.5
abs5n5	5	6	4	0	5459	24	140	17	6566	6602	6558	6558	6527	6633	0.5	−0.1	−0.1	−0.6	1.0
abs1n10	10	6	5	11	8980	54	1022	4	10 421	10 631	10 631	10 631	10 631	10 631	2.0	2.0	2.0	2.0	2.0
abs2n10	10	6	5	11	9547	48	1162	4	11 283	11 731	11 731	11 731	11 731	11 731	3.8	3.8	3.8	3.8	3.8
abs3n10	10	6	5	6	8456	52	1395	4	9997	11 155	10 856	10 856	10 889	11 155	10.4	7.9	7.9	8.2	10.4
abs4n10	10	6	5	5	10 116	38	771	6	12 517	13 317	13 317	13 317	13 317	13 317	6.0	6.0	6.0	6.0	6.0
abs5n10	10	6	5	22	9243	124	1206	8	10 529	10 992	10 992	10 992	10 992	10 992	4.2	4.2	4.2	4.2	4.2
abs1n10	10	6	6	10	8629	102	1475	10	10 126	10 674	10 631	10 631	10 589	10 631	5.1	4.8	4.8	4.4	4.8
abs2n10	10	6	6	34	9211	254	1947	11	10 875	11 655	11 738	11 646	11 655	11 740	6.7	7.4	6.6	6.7	7.4
abs3n10	10	6	6	20	7846	146	2019	3	8956	9116	9116	9255	9255	9305	1.8	1.8	3.2	3.2	3.8
abs4n10	10	6	6	40	9988	260	1675	13	12 187	12 896	12 980	13 076	12 896	12 858	5.5	6.1	6.8	5.5	5.2
abs5n10	10	6	6	68	8761	484	2173	15	10 097	11 160	11 018	10 992	11 086	10 971	9.5	8.4	8.1	8.9	8.0
abs1n10	10	6	7	16	7774	120	1753	7	8773	8877	8871	8854	8877	8854	1.2	1.1	0.9	1.2	0.9
abs2n10	10	6	7	13	8174	84	1985	6	9510	9457	9701	9564	9552	9731	−0.6	2.0	0.6	0.4	2.3
abs3n10	10	6	7	8	7662	52	2360	6	8805	9461	9027	9104	9134	9344	6.9	2.5	3.3	3.6	5.8
abs4n10	10	6	7	12	8321	74	1600	4	9483	9620	9650	9588	9586	9586	1.4	1.7	1.1	1.1	1.1
abs5n10	10	6	7	21	7904	136	2011	10	8885	9471	9386	9360	9279	9291	6.2	5.3	5.1	4.2	4.4
abs1n10	10	6	8	25	7698	200	2271	13	8826	9001	8865	8919	8922	8930	1.9	0.4	1.0	1.1	1.2
abs2n10	10	6	8	18	8114	120	2504	11	9294	9599	9773	9828	9783	9801	3.2	4.9	5.4	5.0	5.2
abs3n10	10	6	8	14	7583	126	2572	11	8727	9441	8907	8950	8888	9451	7.6	2.0	2.5	1.8	7.7
abs4n10	10	6	8	17	8095	124	2010	8	9334	9528	9420	9583	9527	9493	2.0	0.9	2.6	2.0	1.7
abs5n10	10	6	8	33	7809	244	2336	13	8857	9526	9278	9278	9313	9495	7.0	4.5	4.5	4.9	6.7
abs1n15	15	6	8	3592	10 140	7730	35 017	9	11 601	12 163	12 215	12 163	12 163	12 163	4.6	5.0	4.6	4.6	4.6
abs2n15	15	6	8	7201	+	736	110 588	-	-	-	-	-	-	-	-	-	-	-	-
abs3n15	15	6	8	7201	+	13 558	45 565	-	-	-	-	-	-	-	-	-	-	-	-
abs4n15	15	6	8	5014	11 146	8704	41 826	12	12 287	13 846	12 933	13 276	13 846	13 846	11.3	5.0	7.4	11.3	11.3
abs5n15	15	6	8	722	11 084	1568	26 574	15	12 269	13 613	13 069	13 136	13 231	13 231	9.9	6.1	6.6	7.3	7.3
abs1n15	15	6	9	7201	+	10 642	61 282	-	-	-	-	-	-	-	-	-	-	-	-
abs2n15	15	6	9	3313	10 281	6384	46 828	11	10 993	12 444	12 111	11 749	11 953	12 434	11.7	9.2	6.4	8.0	11.6
abs3n15	15	6	9	247	9536	512	27 663	7	10 833	11 231	11 143	11 176	11 231	11 231	3.5	2.8	3.1	3.5	3.5
abs4n15	15	6	9	134	10 133	286	29 297	10	10 920	12 309	12 072	12 035	12 036	12 035	11.3	9.5	9.3	9.3	9.3
abs5n15	15	6	9	934	10 145	1864	40 232	14	10 846	11 535	11 662	11 724	11 708	11 699	6.0	7.0	7.5	7.4	7.3
abs1n15	15	6	10	193	8895	390	43 861	8	9734	10 334	10 361	10 384	10 266	10 378	5.8	6.1	6.3	5.2	6.2
abs2n15	15	6	10	191	9422	388	35 056	7	9859	10 270	10 207	10 212	10 203	10 176	4.0	3.4	3.5	3.4	3.1
abs3n15	15	6	10	1472	9504	2476	54 081	13	10 690	11 385	10 775	10 808	11 264	11 264	6.1	0.8	1.1	5.1	5.1
abs4n15	15	6	10	6231	9728	9122	72 982	8	10 206	10 367	10 336	10 252	10 368	10 382	1.6	1.3	0.4	1.6	1.7
abs5n15	15	6	10	795	9311	1536	33 943	9	9664	9785	9732	9716	9732	9741	1.2	0.7	0.5	0.7	0.8
abs1n15	15	6	11	592	8848	1096	53 819	13	9735	10 306	10 279	10 114	10 351	10 236	5.5	5.3	3.7	5.9	4.9
abs2n15	15	6	11	231	9300	478	33 939	10	9830	10 414	10 330	10 324	10 326	10 407	5.6	4.8	4.8	4.8	5.5
abs3n15	15	6	11	1851	9407	3214	62 292	12	10 569	11 443	10 965	10 898	11 015	11 015	7.6	3.6	3.0	4.0	4.0
abs4n15	15	6	11	2479	9370	4216	58 445	10	9776	10 180	10 017	10 008	10 189	10 158	4.0	2.4	2.3	4.1	3.8
abs5n15	15	6	11	546	9160	932	41 385	12	9520	9781	9817	9682	9785	9799	2.7	3.0	1.7	2.7	2.8

+: instance not solved within two hours.

Table D.8 Results per instance comparison DJRP-AT and IRP.

Instance	N	T	k_M	DJRP-AT Solution	IRP Solution	Difference (%)	Time DJRP-AT (s)	Time IRP (s)
abs1n5	5	3	3	3471	3471	0.00	0	1
abs2n5	5	3	3	3146	3146	0.00	0	1
abs3n5	5	3	3	4169	4140	−0.70	0	1
abs4n5	5	3	3	4335	4335	0.00	0	1
abs5n5	5	3	3	3260	3237	−0.69	0	1
abs1n5	5	3	4	3391	3358	−0.96	1	2
abs2n5	5	3	4	3146	3031	−3.67	0	1
abs3n5	5	3	4	4169	4127	−1.01	0	4
abs4n5	5	3	4	3050	3050	0.00	0	0
abs5n5	5	3	4	3187	3160	−0.85	0	1
abs1n10	10	3	5	4315	4315	0.00	0	324
abs2n10	10	3	5	4842	4842	0.00	1	960
abs3n10	10	3	5	4073	4073	0.00	0	81
abs4n10	10	3	5	6163	5996	−2.70	1	3800
abs5n10	10	3	5	4351	4351	0.00	1	614
abs1n10	10	3	6	4332	4296	−0.84	0	11 567
abs2n10	10	3	6	4559	4533	−0.56	1	9625
abs3n10	10	3	6	4073	4038	−0.85	0	5763
abs4n10	10	3	6	5783	+	-	1	-
abs5n10	10	3	6	4192	4192	0.00	0	10 440
abs1n10	10	3	7	4332	+	-	1	-
abs2n10	10	3	7	4452	+	-	1	-
abs3n10	10	3	7	4073	+	-	0	-
abs4n10	10	3	7	4336	+	-	1	-
abs5n10	10	3	7	4107	+	-	0	-
abs1n5	5	6	3	7430	7430	0.00	0	1
abs2n5	5	6	3	6694	6694	0.00	0	4
abs3n5	5	6	3	9866	9866	0.00	0	4
abs4n5	5	6	3	9510	9510	0.00	1	5
abs5n5	5	6	3	6680	6603	−1.16	1	9
abs1n5	5	6	4	7305	7299	−0.07	0	25
abs2n5	5	6	4	6694	6556	−2.07	0	10
abs3n5	5	6	4	9033	8844	−2.09	1	18
abs4n5	5	6	4	7255	7150	−1.44	1	5
abs5n5	5	6	4	6566	6456	−1.69	0	9
abs1n10	10	6	5	10 421	10 421	0.00	11	2577
abs2n10	10	6	5	11 283	11 283	0.00	11	12 021
abs3n10	10	6	5	9997	9944	−0.54	6	1759
abs4n10	10	6	5	12 517	12 201	−2.53	5	3422
abs5n10	10	6	5	10 529	10 529	0.00	22	4941
abs1n10	10	6	6	10 126	+	-	10	-
abs2n10	10	6	6	10 875	+	-	34	-
abs3n10	10	6	6	8956	8849	−1.19	20	13 738
abs4n10	10	6	6	12 187	+	-	40	-
abs5n10	10	6	6	10 097	+	-	68	-

+: instance not solved within four hours.

E

Results per Instance IRPDM

Tables E.1 - E.3 show detailed results on the instances used for Section 4.5.2 when $m = 0.01$ and no maximum on the moved demand. The fleet size differs per table (K=3,4,5). The three tables report per instance the computation time for the IRPDM if the instance is solved within two hours of running time and whether the instance was solved to optimality (y/n). Thereafter, the upper bound ('UB'), the root lower bound ('$\mathrm{LB}_{\mathrm{root}}$') and the lower bound after adding valid inequalities ('$\mathrm{LB}_{\mathrm{cuts}}$') is given. The size of the tree ('Tree'), the number of added capacity inequalities ('CI') and MCS inequalities ('MCS') are then given, followed by the percentage cost improvement over the IRP ('Impr.'), the number of demand moves ('Nr.DM') and the average size of the demand moves ('Sz.DM') of the solution. Finally, the best upper bound found for the IRP is reported ('UB'), retrieved from Coelho [n.d.] and an indication ('Opt.') whether the instance was solved to optimality (y/n). The instance is indicated in the following format C_Hp_Nc_Kv_i with C the level of holding costs, p the number of periods, c the number of customers, v the number of vehicles and i the index of the instance. If no solution is available because the instance is not solved to optimality, a dash is filled out. We only include instances for which at least the root node is solved. For two IRP instances there is no feasible solution possible, their upper bounds are therefore unknown (Unk).

Table E.1 Detailed results on K3 instances

	IRPDM											IRP	
Instance	T(s)	Opt.	UB	LB_{root}	LB_{cuts}	Tree	CI	MCS	Impr.	Nr.DM	Sz.DM	UB	Opt.
High_H3_N5_K3_1	0.1	y	2212.2	2212.2	2212.2	7	0	0	3.76	1	50.0	2298.7	y
High_H3_N5_K3_2	0.4	y	2334.5	2081.6	2299.1	65	0	2	1.49	1	14.0	2369.9	y
High_H3_N5_K3_3	0.1	y	4019.3	3996.5	3996.5	15	2	0	4.10	3	40.3	4191.3	y
High_H3_N5_K3_4	0.4	y	2772.3	2719.6	2720.6	32	2	2	3.54	4	14.3	2874.1	y
High_H3_N5_K3_5	0.0	y	2561.5	2561.5	2561.5	4	0	0	3.83	2	81.0	2663.7	y
High_H3_N10_K3_1	49.7	y	5497.4	5377.8	5427.6	79	7	10	0.16	1	34.0	5506.1	y
High_H3_N10_K3_2	52.2	y	5680.7	5522.0	5628.4	177	4	4	1.09	1	44.0	5743.3	y
High_H3_N10_K3_3	7.9	y	4735.3	4623.3	4710.2	28	0	9	1.51	2	22.5	4808.1	y
High_H3_N10_K3_4	21.9	y	5145.5	4952.2	5060.0	56	0	9	3.56	7	32.3	5335.3	y
High_H3_N10_K3_5	3.0	y	5182.1	5038.1	5106.4	26	0	6	0.81	2	54.0	5224.5	y
High_H3_N15_K3_1	77.3	y	6184.9	6074.0	6184.6	12	6	6	0.93	2	18.5	6242.9	y
High_H3_N15_K3_2	607.8	y	6038.6	5969.1	6034.7	31	1	5	0.54	2	10.0	6071.3	y
High_H3_N15_K3_3	154.6	y	6908.1	6895.3	6908.1	8	0	5	0.26	2	8.5	6926.2	y
High_H3_N15_K3_4	7270.0	n	-	5471.7	5544.2	12	0	9	-	-	-	5705.2	y
High_H3_N15_K3_5	7277.4	n	-	5554.8	5583.2	124	9	10	-	-	-	5967.3	y
High_H3_N20_K3_1	7201.0	n	-	7730.8	7909.2	1	6	8	-	-	-	8165.4	y
High_H3_N20_K3_2	7485.9	n	-	7172.0	7278.9	3	0	12	-	-	-	7499.5	y
High_H3_N20_K3_3	2402.9	y	7833.9	7788.9	7824.5	13	0	4	0.08	2	16.5	7840.5	y
High_H3_N20_K3_4	7200.8	n	-	7598.0	7687.4	16	0	12	-	-	-	7919.1	y
High_H3_N20_K3_5	7200.5	n	-	8964.2	9030.6	1	0	11	-	-	-	9149.2	y
High_H3_N25_K3_4	7201.9	n	-	8804.3	8854.9	4	0	8	-	-	-	9049.1	y
Low_H3_N5_K3_1	0.0	y	1335.0	1335.0	1335.0	5	0	0	6.68	1	50.0	1430.5	y
Low_H3_N5_K3_2	0.7	y	1549.2	1286.0	1511.0	93	1	3	2.12	1	14.0	1582.7	y
Low_H3_N5_K3_3	0.1	y	2835.1	2800.0	2800.0	15	2	0	5.42	3	40.3	2997.4	y
Low_H3_N5_K3_4	0.5	y	2175.7	2124.7	2129.0	33	3	2	5.04	4	14.3	2291.2	y
Low_H3_N5_K3_5	0.0	y	1408.0	1408.0	1408.0	4	0	0	6.98	2	81.0	1513.8	y
Low_H3_N10_K3_1	92.0	y	2725.9	2605.6	2646.4	135	9	10	0.24	1	34.0	2732.6	y
Low_H3_N10_K3_2	48.6	y	3395.0	3230.8	3344.8	177	2	4	2.17	1	44.0	3470.2	y
Low_H3_N10_K3_3	9.3	y	2574.4	2458.5	2546.2	27	0	9	2.83	2	22.5	2649.3	y
Low_H3_N10_K3_4	33.2	y	2977.3	2783.2	2893.9	83	0	9	6.47	7	32.3	3183.4	y
Low_H3_N10_K3_5	5.7	y	2422.4	2278.6	2334.6	53	1	8	1.57	1	77.0	2461.1	y
Low_H3_N15_K3_1	165.2	y	2728.9	2603.9	2724.1	12	3	10	1.97	2	18.5	2783.8	y
Low_H3_N15_K3_2	851.8	y	2725.2	2664.8	2721.1	34	3	5	1.18	2	10.0	2757.8	y
Low_H3_N15_K3_3	120.8	y	3055.3	3044.7	3055.3	10	0	6	0.57	1	11.0	3072.8	y
Low_H3_N15_K3_4	7201.5	n	-	2636.5	2712.7	29	1	8	-	-	-	2886.3	y
Low_H3_N15_K3_5	7201.1	n	-	2839.0	2871.2	90	17	10	-	-	-	3260.6	y
Low_H3_N20_K3_1	7200.5	n	-	3148.3	3326.8	3	1	17	-	-	-	3605.7	y
Low_H3_N20_K3_2	8942.2	n	-	2585.0	2697.1	3	1	11	-	-	-	2908.5	y
Low_H3_N20_K3_3	737.5	y	3049.4	3008.3	3045.0	14	0	6	0.50	2	16.5	3064.8	y
Low_H3_N20_K3_4	7201.1	n	-	3754.7	3843.1	18	4	9	-	-	-	4088.9	y
Low_H3_N20_K3_5	7200.7	n	-	3938.7	4056.7	7	2	15	-	-	-	4124.2	y
Low_H3_N25_K3_4	7200.8	n	-	3398.2	3451.4	8	3	10	-	-	-	3659.1	y
Low_H3_N25_K3_5	7200.7	n	-	3943.1	4035.1	1	0	4	-	-	-	4120.3	y
High_H6_N5_K3_1	38.5	y	7259.1	7166.0	7233.3	1425	0	3	0.00	0	-	7259.1	y
High_H6_N5_K3_2	1394.6	y	6331.5	6149.5	6162.7	32 258	1	7	3.04	8	42.1	6530.4	y
High_H6_N5_K3_3	7200.1	n	9876.2	9267.5	9510.2	89 083	7	6	-	3	12.0	9862.9	y
High_H6_N5_K3_4	20.1	y	6242.1	5982.5	6020.5	525	1	6	2.56	6	55.8	6405.7	y
High_H6_N5_K3_5	1337.1	y	5871.2	5739.8	5739.8	18 955	0	1	1.70	5	64.6	5972.8	y
High_H6_N10_K3_1	7200.5	n	-	11 095.7	11 098.5	1551	0	8	-	-	-	11 440.9	y
High_H6_N10_K3_2	7200.3	n	-	11 039.7	11 123.5	2559	0	18	-	-	-	11 584.8	n
High_H6_N10_K3_3	7200.3	n	-	9752.7	9825.1	2572	3	16	-	-	-	9982.7	y
High_H6_N10_K3_4	7200.7	n	-	10 551.6	10 585.4	845	5	13	-	-	-	11 168.5	y
High_H6_N10_K3_5	7200.3	n	10 604.7	10 339.6	10 365.3	3749	0	11	-	2	34.0	10 702.1	y
Low_H6_N5_K3_1	73.5	y	4650.5	4559.5	4630.1	2442	0	3	0.14	1	14.0	4657.2	y
Low_H6_N5_K3_2	2003.8	y	4007.3	3834.0	3842.8	45 914	1	7	5.06	8	42.1	4221.0	y
Low_H6_N5_K3_3	7200.1	n	7729.6	7114.2	7356.3	86 419	2	7	-	0	-	7703.5	y
Low_H6_N5_K3_4	16.5	y	4298.7	4042.1	4077.4	447	0	6	4.37	6	55.8	4495.3	y
Low_H6_N5_K3_5	5241.9	y	3792.0	3656.1	3656.1	67 053	0	0	2.25	7	44.7	3879.1	y
Low_H6_N10_K3_1	7200.1	n	-	6757.0	6759.4	1658	1	5	-	-	-	7138.6	n
Low_H6_N10_K3_2	7200.4	n	-	7725.4	7807.8	2670	1	12	-	-	-	8276.2	n
Low_H6_N10_K3_3	7200.8	n	-	5910.4	5980.6	2339	4	14	-	-	-	6185.3	y
Low_H6_N10_K3_4	7200.4	n	-	6860.1	6900.0	1040	2	14	-	-	-	7483.3	y
Low_H6_N10_K3_5	7200.4	n	5774.0	5474.0	5503.1	4470	0	7	-	2	34.0	5824.1	y

Table E.2 Detailed results on K4 instances

	IRPDM											IRP	
Instance	T(s)	Opt.	UB	LB_{root}	LB_{cuts}	Tree	CI	MCS	Impr.	Nr.DM	Sz.DM	UB	Opt.
High_H3_N5_K4_1	0.0	y	2395.9	2383.7	2395.8	6	0	1	3.08	1	50.0	2472.1	y
High_H3_N5_K4_2	18.9	y	2577.2	2328.6	2397.3	3251	1	3	0.86	2	12.0	2599.6	y
High_H3_N5_K4_3	0.1	y	4557.1	4307.6	4534.9	10	1	1	5.21	3	27.7	4807.5	y
High_H3_N5_K4_4	0.1	y	3032.2	3012.0	3023.0	4	0	2	5.55	4	14.0	3210.2	y
High_H3_N5_K4_5	0.0	y	2755.8	2755.8	2755.8	4	0	0	2.43	4	40.5	2824.5	y
High_H3_N10_K4_1	7.3	y	6021.1	5816.4	6004.3	36	2	7	0.00	0	-	6021.1	y
High_H3_N10_K4_2	19.2	y	6416.6	6095.8	6370.9	217	0	6	1.88	4	59.5	6539.3	y
High_H3_N10_K4_3	4.6	y	5109.4	4960.4	5079.4	41	1	9	0.34	2	22.0	5127.0	y
High_H3_N10_K4_4	40.4	y	5660.1	5432.9	5569.3	218	2	12	2.99	7	33.3	5834.7	y
High_H3_N10_K4_5	20.7	y	5594.2	5285.4	5448.8	201	6	8	0.89	3	43.3	5644.4	y
High_H3_N15_K4_1	7201.3	n	6931.3	6376.2	6461.4	1192	11	14	-	0	-	6611.3	y
High_H3_N15_K4_2	7204.8	n	-	6299.8	6370.1	446	2	8	-	-	-	6705.9	y
High_H3_N15_K4_3	7201.8	n	-	7249.7	7452.8	470	3	9	-	-	-	7607.7	y
High_H3_N15_K4_4	7155.5	y	5970.6	5822.5	5915.0	307	0	4	0.78	3	26.0	6017.5	y
High_H3_N15_K4_5	7217.2	n	-	5923.9	6168.8	516	9	12	-	-	-	6375.4	y
High_H3_N20_K4_1	7469.9	n	-	8246.4	8459.9	20	4	12	-	-	-	8717.8	y
High_H3_N20_K4_2	7287.6	n	-	7357.5	7459.6	3	1	14	-	-	-	7710.9	y
High_H3_N20_K4_3	7201.9	n	-	8144.6	8290.0	114	1	15	-	-	-	8414.1	y
High_H3_N20_K4_4	7255.0	n	-	8131.7	8350.9	28	2	14	-	-	-	8589.3	y
High_H3_N20_K4_5	7204.3	n	-	9569.2	9697.8	68	5	11	-	-	-	9782.6	y
High_H3_N25_K4_1	7201.9	n	-	8826.2	9031.2	1	0	18	-	-	-	9287.9	y
High_H3_N25_K4_2	7202.9	n	-	9982.4	10 114.5	14	2	15	-	-	-	10 264.0	n
High_H3_N25_K4_3	7200.8	n	-	10 763.7	10 814.6	1	0	3	-	-	-	11 026.8	y
High_H3_N25_K4_4	7216.4	n	-	9127.3	9201.5	32	6	11	-	-	-	9436.9	y
High_H3_N25_K4_5	7201.4	n	-	11 633.4	11 711.7	5	0	21	-	-	-	11 806.0	y
Low_H3_N5_K4_1	0.1	y	1518.7	1507.2	1518.6	12	0	1	5.17	1	50.0	1601.6	y
Low_H3_N5_K4_2	20.4	y	1791.3	1535.1	1607.4	3502	0	3	1.18	3	8.0	1812.7	y
Low_H3_N5_K4_3	0.1	y	3360.9	3112.8	3341.5	13	1	1	6.74	3	27.7	3603.7	y
Low_H3_N5_K4_4	0.1	y	2441.5	2420.8	2436.3	11	0	2	7.22	4	14.0	2631.6	y
Low_H3_N5_K4_5	0.0	y	1590.2	1590.2	1590.2	4	0	0	5.14	5	32.4	1676.4	y
Low_H3_N10_K4_1	16.1	y	3261.9	3046.2	3234.5	86	0	7	0.00	0	-	3261.9	y
Low_H3_N10_K4_2	12.4	y	4131.6	3811.3	4087.2	124	5	4	3.26	6	35.2	4271.0	y
Low_H3_N10_K4_3	7.2	y	2947.0	2793.6	2913.7	67	2	12	0.73	3	25.3	2968.7	y
Low_H3_N10_K4_4	55.2	y	3490.0	3253.6	3398.9	249	1	12	5.25	8	29.1	3683.2	y
Low_H3_N10_K4_5	42.6	y	2832.0	2509.5	2675.2	374	9	6	1.39	2	49.5	2872.1	y
Low_H3_N15_K4_1	7201.6	n	3349.4	2912.6	3003.1	1573	11	13	-	2	45.5	3166.4	y
Low_H3_N15_K4_2	7200.6	n	3309.8	2996.2	3064.8	655	3	8	-	5	24.4	3396.7	y
Low_H3_N15_K4_3	7200.9	n	-	3404.9	3610.8	293	4	8	-	-	-	3757.4	y
Low_H3_N15_K4_4	3454.7	y	3142.5	2979.5	3084.4	293	1	5	1.80	7	14.3	3200.2	y
Low_H3_N15_K4_5	7201.2	n	3851.9	3208.1	3451.6	562	23	9	-	4	24.8	3671.1	y
Low_H3_N20_K4_1	7338.8	n	-	3663.6	3891.6	20	4	16	-	-	-	4148.0	y
Low_H3_N20_K4_2	7200.7	n	-	2756.8	2860.5	32	5	17	-	-	-	3128.1	y
Low_H3_N20_K4_3	7248.7	n	-	3366.8	3508.9	24	1	23	-	-	-	3645.5	y
Low_H3_N20_K4_4	7201.3	n	-	4288.9	4516.5	84	7	16	-	-	-	4787.9	n
Low_H3_N20_K4_5	7210.8	n	-	4542.4	4672.2	84	11	15	-	-	-	4764.8	y
Low_H3_N25_K4_1	7215.1	n	-	3457.6	3554.8	1	0	3	-	-	-	3949.7	y
Low_H3_N25_K4_2	7233.3	n	-	4219.2	4353.8	16	17	8	-	-	-	4502.9	n
Low_H3_N25_K4_3	7200.7	n	-	4422.7	4530.4	1	0	8	-	-	-	4687.6	y
Low_H3_N25_K4_4	7204.9	n	-	3727.9	3804.8	34	6	15	-	-	-	4044.8	y
Low_H3_N25_K4_5	7202.3	n	-	4519.2	4587.5	12	0	20	-	-	-	4672.8	y
High_H6_N5_K4_1	23.9	y	8074.3	7904.6	8058.6	986	1	8	0.46	2	50.0	8111.3	y
High_H6_N5_K4_2	3045.6	y	7136.7	6871.6	6879.0	91 012	9	4	3.71	8	31.4	7411.3	y
High_H6_N5_K4_3	112.0	y	11 179.6	10 957.5	11 027.7	3394	0	1	2.58	9	12.8	11 475.3	y
High_H6_N5_K4_4	12.7	y	6686.1	6568.3	6580.4	675	3	4	4.29	7	32.1	6985.4	y
High_H6_N5_K4_5	3469.1	y	6904.5	6743.1	6792.3	71 871	0	1	1.31	2	82.0	6996.4	y
High_H6_N10_K4_1	7200.1	n	-	12 431.1	12 519.0	4868	0	14	-	-	-	12 801.4	n
High_H6_N10_K4_2	7200.1	n	-	12 727.7	12 771.1	7152	3	14	-	-	-	13 190.4	n
High_H6_N10_K4_3	7200.3	n	-	10 633.1	10 815.1	7520	2	22	-	-	-	11 067.9	y
High_H6_N10_K4_4	7200.4	n	-	11 652.0	11 760.7	4105	3	17	-	-	-	12 323.9	y
High_H6_N10_K4_5	7200.1	n	-	11 046.8	11 097.2	9728	1	12	-	-	-	11 471.5	y
Low_H6_N5_K4_1	21.6	y	5457.4	5295.4	5451.7	885	7	8	0.89	4	56.7	5506.2	y
Low_H6_N5_K4_2	4518.2	y	4814.3	4568.9	4576.1	136 841	5	2	6.08	7	35.9	5125.8	y
Low_H6_N5_K4_3	116.2	y	9030.5	8806.9	8877.3	3424	0	1	3.11	7	15.0	9320.7	y
Low_H6_N5_K4_4	17.1	y	4761.9	4644.5	4650.4	908	4	2	6.37	7	32.1	5085.8	y
Low_H6_N5_K4_5	7200.1	n	4825.1	4660.4	4713.6	148 679	0	1	-	3	53.7	4913.4	y
Low_H6_N10_K4_1	7200.1	n	-	8090.4	8180.9	4490	0	17	-	-	-	8421.9	n
Low_H6_N10_K4_2	7200.1	n	-	9431.4	9467.5	8072	1	12	-	-	-	9875.1	n
Low_H6_N10_K4_3	7200.2	n	-	6800.1	6970.1	7256	3	21	-	-	-	7255.6	y
Low_H6_N10_K4_4	7200.2	n	-	7966.3	8081.3	4876	2	19	-	-	-	8645.1	y
Low_H6_N10_K4_5	7200.2	n	-	6187.5	6232.1	10 028	1	10	-	-	-	6604.9	y

Table E.3 Detailed results on K5 instances

	IRPDM											IRP	
Instance	T(s)	Opt.	UB	LB_{root}	LB_{cuts}	Tree	CI	MCS	Impr.	Nr.DM	Sz.DM	UB	Opt.
High_H3_N5_K5_1	0.0	y	2480.8	2434.5	2480.8	5	0	1	3.75	1	30.0	2577.5	y
High_H3_N5_K5_2	1.7	y	2675.2	2543.8	2570.5	358	1	2	4.70	1	28.0	2807.3	y
High_H3_N5_K5_3	0.2	y	4858.0	4744.5	4827.4	25	1	3	5.87	2	32.5	5160.9	y
High_H3_N5_K5_4	0.1	y	3720.3	3647.8	3710.4	7	3	1	4.60	2	21.0	3899.7	y
High_H3_N5_K5_5	0.0	y	2974.7	2974.7	2974.7	4	0	0	6.07	4	38.5	3166.8	y
High_H3_N10_K5_1	3235.5	y	6392.1	6239.7	6275.8	18 095	1	6	1.61	2	44.0	6496.7	y
High_H3_N10_K5_2	294.4	y	6857.0	6667.9	6772.3	1704	1	7	1.49	4	51.7	6960.9	y
High_H3_N10_K5_3	2.3	y	5467.2	5285.2	5438.9	17	6	18	1.62	5	15.2	5557.2	y
High_H3_N10_K5_4	361.3	y	6056.0	5853.4	5924.1	1667	12	10	4.01	6	27.3	6308.8	y
High_H3_N10_K5_5	0.6	y	5634.9	5499.8	5621.4	11	0	6	2.37	1	31.0	5771.5	y
High_H3_N15_K5_1	5860.0	y	6985.1	6718.4	6900.3	803	0	14	0.55	2	36.0	7023.7	y
High_H3_N15_K5_2	7201.1	n	7067.4	6603.7	6942.2	1199	3	6	-	4	23.0	7194.3	y
High_H3_N15_K5_3	7211.3	n	7960.6	7680.2	7825.7	578	2	14	-	3	20.7	7981.7	y
High_H3_N15_K5_4	7200.5	n	-	6152.6	6332.3	228	2	5	-	-	-	6388.1	y
High_H3_N15_K5_5	7221.9	n	-	6383.7	6678.1	514	10	14	-	-	-	6963.6	y
High_H3_N20_K5_1	4858.2	y	8883.2	8708.2	8832.9	69	13	14	0.92	2	15.0	8965.2	y
High_H3_N20_K5_2	7227.2	n	-	7583.2	7676.6	18	10	15	-	-	-	7910.5	y
High_H3_N20_K5_3	7201.3	n	-	8461.8	8654.6	243	9	16	-	-	-	8787.5	y
High_H3_N20_K5_4	7267.5	n	-	8647.4	8840.9	202	7	14	-	-	-	9047.3	y
High_H3_N20_K5_5	7202.5	n	-	10 215.0	10 381.4	110	11	22	-	-	-	10 523.8	y
High_H3_N25_K5_1	7233.7	n	-	9142.5	9257.3	19	13	15	-	-	-	9473.2	y
High_H3_N25_K5_2	7204.6	n	-	10 503.0	10 614.4	24	2	17	-	-	-	10 757.7	y
High_H3_N25_K5_3	7202.3	n	-	11 345.9	11 481.0	9	1	12	-	-	-	11 576.3	y
High_H3_N25_K5_4	7202.7	n	-	9437.5	9513.7	90	11	14	-	-	-	9719.3	y
High_H3_N25_K5_5	7211.2	n	-	12 215.8	12 362.3	14	2	25	-	-	-	12 451.2	y
Low_H3_N5_K5_1	0.0	y	1606.8	1556.6	1606.8	4	0	1	6.05	1	30.0	1710.3	y
Low_H3_N5_K5_2	2.2	y	1885.7	1755.7	1781.7	480	3	3	6.63	1	28.0	2019.6	y
Low_H3_N5_K5_3	0.2	y	3656.1	3555.5	3625.8	34	1	3	7.81	2	30.5	3965.8	y
Low_H3_N5_K5_4	0.2	y	3136.5	3056.3	3124.8	25	1	3	5.50	2	21.0	3319.0	y
Low_H3_N5_K5_5	0.0	y	1807.2	1807.2	1807.2	5	0	0	10.02	4	38.5	2008.5	y
Low_H3_N10_K5_1	2979.2	y	3623.8	3475.3	3514.2	18 209	4	6	2.82	2	44.0	3728.8	y
Low_H3_N10_K5_2	229.5	y	4572.5	4385.6	4489.5	1832	1	7	2.33	4	51.7	4681.3	y
Low_H3_N10_K5_3	3.3	y	3313.5	3119.2	3271.2	39	8	19	2.58	5	15.2	3401.2	y
Low_H3_N10_K5_4	356.3	y	3871.4	3673.5	3740.3	1827	10	9	6.87	5	32.8	4157.0	y
Low_H3_N10_K5_5	0.8	y	2862.0	2729.3	2850.7	14	0	9	4.41	1	31.0	2993.9	y
Low_H3_N15_K5_1	1821.5	y	3517.8	3251.5	3437.4	718	2	16	1.76	2	36.0	3580.8	y
Low_H3_N15_K5_2	7201.1	n	3736.5	3302.7	3642.2	1102	3	6	-	4	20.0	3889.4	y
Low_H3_N15_K5_3	7201.1	n	4073.0	3845.3	3989.4	694	5	8	-	2	16.5	4133.6	y
Low_H3_N15_K5_4	7201.2	n	-	3311.0	3489.9	153	1	5	-	-	-	3572.5	y
Low_H3_N15_K5_5	7200.6	n	-	3663.7	3967.9	634	4	13	-	-	-	4256.2	y
Low_H3_N20_K5_1	5899.1	y	4310.5	4137.9	4287.6	76	9	18	2.14	3	30.7	4404.8	y
Low_H3_N20_K5_2	7268.9	n	-	2989.8	3081.3	26	7	12	-	-	-	3344.1	y
Low_H3_N20_K5_3	7207.0	n	-	3686.7	3874.7	272	6	22	-	-	-	4016.5	y
Low_H3_N20_K5_4	7203.6	n	-	4802.3	4999.5	90	5	18	-	-	-	5215.8	y
Low_H3_N20_K5_5	7231.0	n	-	5185.1	5361.8	54	25	20	-	-	-	5506.0	y
Low_H3_N25_K5_1	7220.6	n	-	3759.8	3880.7	12	20	21	-	-	-	4095.2	y
Low_H3_N25_K5_2	7233.8	n	-	4739.7	4851.6	61	12	8	-	-	-	5009.8	y
Low_H3_N25_K5_3	7226.8	n	-	5009.1	5130.1	8	1	12	-	-	-	5229.3	y
Low_H3_N25_K5_4	7203.3	n	-	4033.6	4122.3	116	12	20	-	-	-	4378.8	y
Low_H3_N25_K5_5	7272.3	n	-	5109.3	5242.0	28	0	25	-	-	-	5314.1	y
High_H6_N5_K5_1	443.4	y	8948.5	8834.4	8909.3	21 623	8	7	1.04	3	48.7	9043.0	y
High_H6_N5_K5_2	3116.5	y	7952.5	7626.7	7677.8	128 143	4	3	4.17	8	28.5	8298.9	y
High_H6_N5_K5_3	7200.1	n	13 332.6	12 885.3	13 005.0	194 178	0	6	-	2	10.5	13 399.1	y
High_H6_N5_K5_4	6.7	y	7872.5	7631.1	7853.6	479	0	11	3.75	5	34.0	8179.6	y
High_H6_N5_K5_5	181.8	y	7918.0	7860.5	7860.5	6611	0	0	-	9	15.2	Unk	n
High_H6_N10_K5_1	7200.1	n	-	13 622.8	13 729.7	8772	0	11	-	-	-	14 146.1	n
High_H6_N10_K5_2	7200.1	n	-	14 378.7	14 435.0	13 973	0	11	-	-	-	14 949.5	n
High_H6_N10_K5_3	7200.1	n	-	11 606.2	11 779.0	11 015	1	18	-	-	-	12 054.3	y
High_H6_N10_K5_4	7200.1	n	-	12 854.9	13 001.2	8236	10	25	-	-	-	13 754.9	y
High_H6_N10_K5_5	7200.2	n	-	11 709.5	11 746.9	14 993	2	13	-	-	-	12 068.5	y
Low_H6_N5_K5_1	427.5	y	6327.4	6224.6	6297.7	20 562	9	6	1.83	3	48.7	6445.7	y
Low_H6_N5_K5_2	7200.2	n	5692.2	5317.8	5374.3	288 586	3	3	-	7	24.0	6009.4	y
Low_H6_N5_K5_3	7200.1	n	11 157.6	10 773.0	10 892.4	202 314	0	5	-	7	12.0	11 282.7	y
Low_H6_N5_K5_4	4.9	y	5954.7	5727.7	5952.0	344	0	10	5.26	5	33.4	6285.0	y
Low_H6_N5_K5_5	1581.8	y	5830.2	5784.2	5784.2	49 449	0	0	-	10	13.7	Unk	n
Low_H6_N10_K5_1	7200.1	n	-	9289.8	9414.6	9312	0	15	-	-	-	9914.7	n
Low_H6_N10_K5_2	7200.1	n	-	11 080.0	11 138.7	14 965	0	10	-	-	-	11 633.8	n
Low_H6_N10_K5_3	7200.1	n	-	7785.3	7954.8	14 293	0	16	-	-	-	8239.6	y
Low_H6_N10_K5_4	7200.1	n	-	9175.4	9325.4	10 152	5	21	-	-	-	10 093.5	n
Low_H6_N10_K5_5	7200.3	n	8122.8	6858.9	6895.3	15 403	6	13	-	5	18.8	7214.5	y

F

Results per Instance IRPDM: Impact of m and Maximum on Moved Demand

Tables F.1 - F.3 show detailed results on the instances used for Sections 4.5.2.1 and 4.5.2.2. The fleet size differs per table (K=3,4,5). The three tables show for each instance details on the time ('T(s)'), the upper bound ('UB') and the best lower bound ('LB_{best}'). The details are shown for different values of m (0.01, 0.005, 0.05, 0.1) and varying maxima on the moved demand (100%, 25%, 50%, 75%). Remind that instances with up to 15 customers and horizon three, and with five customers and horizon six are tested. The formatting of the instance number is the same as in Appendix E.

Table F.1 Detailed results impact demand move cost and maximum on moved demand, K3 instances

	$m = 0.01$&Max.100%			$m = 0.005$&Max.100%			$m = 0.05$&Max.100%			$m = 0.1$&Max.100%			$m = 0.01$&Max.25%			$m = 0.01$&Max.50%			$m = 0.01$&Max.75%		
Instance	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}
High_H3_N5_K3_1	0	2212.2	2212.2	0	2186.7	2186.7	0	2298.7	2298.7	0	2298.7	2298.7	0	2298.7	2298.7	0	2287.9	2287.9	0	2287.9	2287.9
High_H3_N5_K3_2	0	2334.5	2334.5	1	2328.3	2328.3	0	2369.9	2369.9	0	2369.9	2369.9	2	2369.9	2369.9	0	2334.5	2334.5	0	2334.5	2334.5
High_H3_N5_K3_3	0	4019.3	4019.3	0	3930.3	3930.3	0	4121.4	4121.4	0	4191.3	4191.3	1	4191.3	4191.3	1	4191.3	4191.3	1	4157.4	4157.4
High_H3_N5_K3_4	0	2772.3	2772.3	0	2736.4	2736.4	0	2852.2	2852.2	1	2859.2	2859.2	1	2846.6	2846.6	1	2846.6	2846.6	1	2846.6	2846.6
High_H3_N5_K3_5	0	2561.5	2561.5	0	2507.1	2507.1	0	2617.4	2617.4	0	2647.1	2647.1	0	2593.6	2593.6	0	2593.6	2593.6	0	2578.4	2578.4
High_H3_N10_K3_1	50	5497.4	5497.4	99	5471.3	5471.3	4	5506.1	5506.1	2	5506.1	5506.1	48	5506.1	5506.1	106	5506.1	5506.1	56	5497.4	5497.4
High_H3_N10_K3_2	52	5680.7	5680.7	48	5669.9	5669.9	27	5743.3	5743.3	8	5743.3	5743.3	119	5743.3	5743.3	326	5743.3	5743.3	40	5680.7	5680.7
High_H3_N10_K3_3	8	4735.3	4735.3	22	4724.3	4724.3	77	4808.1	4808.1	8	4808.1	4808.1	135	4808.1	4808.1	761	4808.1	4808.1	34	4778.1	4778.1
High_H3_N10_K3_4	22	5145.5	5145.5	22	5034.4	5034.4	4875	5335.3	5335.3	857	5335.3	5335.3	6240	5335.3	5335.3	7009	5301.6	5301.6	3000	5275.9	5275.9
High_H3_N10_K3_5	3	5182.1	5182.1	5	5121.7	5121.7	1	5224.5	5224.5	1	5224.5	5224.5	5	5224.5	5224.5	9	5224.5	5224.5	32	5224.5	5224.5
High_H3_N15_K3_1	77	6184.9	6184.9	200	6172.4	6172.4	40	6196.5	6196.5	60	6199.6	6199.6	156	6194.1	6194.1	144	6194.1	6194.1	99	6194.1	6194.1
High_H3_N15_K3_2	608	6038.6	6038.6	-	-	6030.1	104	6071.3	6071.3	27	6071.3	6071.3	902	6071.3	6071.3	1264	6071.3	6071.3	635	6071.3	6071.3
High_H3_N15_K3_3	155	6908.1	6908.1	269	6903.9	6903.9	99	6926.2	6926.2	27	6926.2	6926.2	244	6924.9	6924.9	263	6924.9	6924.9	389	6924.9	6924.9
High_H3_N15_K3_4	-	-	5558.5	-	-	5545.6	-	6820.6	5606.4	-	-	5603.9	-	-	5595.7	-	-	5604.1	-	-	5585.6
High_H3_N15_K3_5	-	-	5684.6	-	-	5651.8	-	-	5795.0	-	6154.8	5802.7	-	-	5732.8	-	-	5665.1	-	-	5705.3
Low_H3_N5_K3_1	0	1335.0	1335.0	0	1309.5	1309.5	0	1430.5	1430.5	0	1430.5	1430.5	0	1430.5	1430.5	0	1417.1	1417.1	1	1417.1	1417.1
Low_H3_N5_K3_2	1	1549.2	1549.2	1	1542.9	1542.9	0	1582.7	1582.7	0	1582.7	1582.7	2	1582.7	1582.7	0	1549.1	1549.1	1	1549.1	1549.1
Low_H3_N5_K3_3	0	2835.1	2835.1	0	2746.1	2746.1	0	2927.6	2927.6	0	2997.4	2997.4	1	2997.4	2997.4	1	2997.4	2997.4	1	2973.1	2973.1
Low_H3_N5_K3_4	1	2175.7	2175.7	0	2139.8	2139.8	1	2269.2	2269.2	1	2275.6	2275.6	1	2263.6	2263.6	1	2263.6	2263.6	2	2263.6	2263.6
Low_H3_N5_K3_5	0	1408.0	1408.0	0	1354.6	1354.6	0	1467.5	1467.5	1	1497.2	1497.2	0	1443.7	1443.7	1	1443.7	1443.7	0	1428.5	1428.5
Low_H3_N10_K3_1	92	2725.9	2725.9	124	2697.4	2697.4	6	2732.6	2732.6	3	2732.6	2732.6	74	2732.6	2732.6	255	2732.6	2732.6	78	2725.9	2725.9
Low_H3_N10_K3_2	49	3395.0	3395.0	121	3384.2	3384.2	45	3470.2	3470.2	21	3470.2	3470.2	166	3470.2	3470.2	475	3470.2	3470.2	30	3395.0	3395.0
Low_H3_N10_K3_3	9	2574.4	2574.4	34	2563.4	2563.4	16	2649.3	2649.3	7	2649.3	2649.3	151	2649.3	2649.3	929	2649.3	2649.3	91	2625.9	2625.9
Low_H3_N10_K3_4	33	2977.3	2977.3	20	2866.2	2866.2	1631	3183.4	3183.4	1473	3183.4	3183.4	-	3201.4	3158.1	3612	3131.8	3131.8	1920	3109.7	3109.7
Low_H3_N10_K3_5	6	2422.4	2422.4	10	2363.0	2363.0	2	2461.1	2461.1	1	2461.1	2461.1	8	2461.1	2461.1	27	2461.1	2461.1	91	2457.9	2457.9
Low_H3_N15_K3_1	165	2728.9	2728.9	162	2716.3	2716.3	41	2740.8	2740.8	69	2743.9	2743.9	157	2738.4	2738.4	340	2738.4	2738.4	272	2738.4	2738.4
Low_H3_N15_K3_2	852	2725.2	2725.2	-	2720.6	2719.6	128	2757.8	2757.8	26	2757.8	2757.8	845	2757.8	2757.8	1076	2757.8	2757.8	681	2757.8	2757.8
Low_H3_N15_K3_3	121	3055.3	3055.3	455	3051.6	3051.6	38	3072.8	3072.8	23	3072.8	3072.8	212	3072.8	3072.8	220	3072.8	3072.8	317	3072.8	3072.8
Low_H3_N15_K3_4	-	-	2740.0	-	-	2702.5	-	-	2801.6	-	-	2822.1	-	-	2797.2	-	-	2780.7	-	-	2751.7
Low_H3_N15_K3_5	-	-	2963.7	-	-	2921.3	-	-	3065.4	-	-	3088.4	-	-	2988.4	-	-	2995.8	-	-	2971.4
High_H6_N5_K3_1	39	7259.1	7259.1	18	7214.1	7214.1	34	7259.1	7259.1	35	7259.1	7259.1	42	7259.1	7259.1	45	7259.1	7259.1	40	7259.1	7259.1
High_H6_N5_K3_2	1395	6331.5	6331.5	86	6148.5	6148.5	999	6433.4	6433.4	2387	6460.1	6460.1	2454	6399.4	6399.4	3478	6399.4	6399.4	6561	6399.4	6399.4
High_H6_N5_K3_3	-	9876.2	9825.0	-	9904.0	9760.7	542	9862.9	9862.9	440	9862.9	9862.9	3224	9848.8	9848.8	3597	9848.8	9848.8	4633	9848.8	9848.8
High_H6_N5_K3_4	20	6242.1	6242.1	9	6045.2	6045.2	21	6405.7	6405.7	20	6405.7	6405.7	44	6378.2	6378.2	70	6377.5	6377.5	166	6364.8	6364.8
High_H6_N5_K3_5	1337	5871.2	5871.2	320	5764.6	5764.6	1520	5972.8	5972.8	1410	5972.8	5972.8	5020	5951.0	5951.0	-	5946.3	5938.0	5236	5914.4	5914.4
Low_H6_N5_K3_1	74	4650.5	4650.5	28	4595.4	4595.4	98	4657.2	4657.2	93	4657.2	4657.2	49	4650.5	4650.5	61	4650.5	4650.5	61	4650.5	4650.5
Low_H6_N5_K3_2	2004	4007.3	4007.3	147	3821.6	3821.6	2320	4120.6	4120.6	5502	4147.3	4147.3	5335	4088.8	4088.8	-	4125.9	4082.8	-	4111.4	4068.7
Low_H6_N5_K3_3	-	7729.6	7673.1	-	7746.0	7610.0	559	7703.5	7703.5	474	7703.5	7703.5	2573	7685.2	7685.2	3431	7685.2	7685.2	4375	7685.2	7685.2
Low_H6_N5_K3_4	17	4298.7	4298.7	10	4102.3	4102.3	22	4495.3	4495.3	23	4495.3	4495.3	31	4454.8	4454.8	75	4452.3	4452.3	203	4437.9	4437.9
Low_H6_N5_K3_5	5242	3792.0	3792.0	1496	3688.1	3688.1	4909	3879.1	3879.1	4571	3879.1	3879.1	-	3880.6	3840.4	-	3871.5	3821.5	-	3833.8	3805.9

Table F.2 Detailed results impact demand move cost and maximum on moved demand, K4 instances

	$m = 0.01$&Max.100%			$m = 0.005$&Max.100%			$m = 0.05$&Max.100%			$m = 0.1$&Max.100%			$m = 0.01$&Max.25%			$m = 0.01$&Max.50%			$m = 0.01$&Max.75%		
Instance	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}
High_H3_N5_K4_1	0	2395.9	2395.9	0	2370.4	2370.4	0	2421.7	2421.7	0	2431.9	2431.9	0	2413.5	2413.5	0	2413.5	2413.5	0	2413.5	2413.5
High_H3_N5_K4_2	19	2577.2	2577.2	14	2557.3	2557.3	5	2599.6	2599.6	5	2599.6	2599.6	16	2599.6	2599.6	18	2599.6	2599.6	25	2599.6	2599.6
High_H3_N5_K4_3	0	4557.1	4557.1	0	4496.4	4496.4	1	4775.7	4775.7	1	4807.5	4807.5	2	4794.2	4794.2	0	4668.0	4668.0	1	4668.0	4668.0
High_H3_N5_K4_4	0	3032.2	3032.2	0	2997.0	2997.0	0	3210.2	3210.2	0	3210.2	3210.2	0	3210.2	3210.2	1	3210.2	3210.2	4	3210.2	3210.2
High_H3_N5_K4_5	0	2755.8	2755.8	0	2701.4	2701.4	0	2824.5	2824.5	0	2824.5	2824.5	0	2824.5	2824.5	0	2824.5	2824.5	1	2806.8	2806.8
High_H3_N10_K4_1	7	6021.1	6021.1	9	6002.9	6002.9	2	6021.1	6021.1	2	6021.1	6021.1	6	6021.1	6021.1	4	6021.1	6021.1	8	6021.1	6021.1
High_H3_N10_K4_2	19	6416.6	6416.6	11	6325.1	6325.1	21	6485.3	6485.3	80	6492.6	6492.6	21	6479.4	6479.4	72	6478.5	6478.5	213	6474.2	6474.2
High_H3_N10_K4_3	5	5109.4	5109.4	7	5092.9	5092.9	2	5127.0	5127.0	2	5127.0	5127.0	7	5127.0	5127.0	8	5127.0	5127.0	9	5127.0	5127.0
High_H3_N10_K4_4	40	5660.1	5660.1	17	5545.4	5545.4	20	5827.9	5827.9	18	5832.6	5832.6	118	5824.2	5824.2	35	5784.2	5784.2	97	5784.2	5784.2
High_H3_N10_K4_5	21	5594.2	5594.2	22	5496.1	5496.1	9	5644.4	5644.4	6	5644.4	5644.4	18	5644.4	5644.4	40	5636.6	5636.6	66	5636.6	5636.6
High_H3_N15_K4_1	-	6931.3	6555.0	-	6928.5	6515.1	-	6625.6	6593.5	-	6660.0	6601.4	-	-	6535.4	-	6984.5	6563.1	-	6688.0	6558.6
High_H3_N15_K4_2	-	-	6486.8	-	6578.1	6418.6	-	6705.0	6612.1	-	6793.2	6649.6	-	-	6590.6	-	-	6563.9	-	-	6549.6
High_H3_N15_K4_3	-	-	7522.0	-	-	7492.9	-	7779.6	7581.8	-	7618.4	7594.3	-	-	7540.1	-	7824.4	7531.3	-	-	7521.6
High_H3_N15_K4_4	7156	5970.6	5970.6	-	-	5942.0	1359	6017.5	6017.5	614	6017.5	6017.5	-	6160.3	6003.4	-	-	5996.2	-	-	6002.3
High_H3_N15_K4_5	-	-	6284.6	-	-	6247.0	1058	6351.5	6351.5	3464	6370.5	6370.5	-	-	6329.3	-	-	6323.6	-	-	6293.7
Low_H3_N5_K4_1	0	1518.7	1518.7	0	1493.2	1493.2	0	1552.0	1552.0	0	1562.2	1562.2	0	1543.8	1543.8	0	1543.8	1543.8	0	1543.8	1543.8
Low_H3_N5_K4_2	20	1791.3	1791.3	16	1770.5	1770.5	7	1812.7	1812.7	6	1812.7	1812.7	19	1812.7	1812.7	18	1812.7	1812.7	25	1812.7	1812.7
Low_H3_N5_K4_3	0	3360.9	3360.9	0	3300.2	3300.2	1	3571.9	3571.9	1	3603.7	3603.7	3	3603.7	3603.7	0	3473.6	3473.6	1	3473.6	3473.6
Low_H3_N5_K4_4	0	2441.5	2441.5	0	2406.3	2406.3	0	2631.6	2631.6	0	2631.6	2631.6	1	2631.6	2631.6	1	2631.6	2631.6	5	2631.6	2631.6
Low_H3_N5_K4_5	0	1590.2	1590.2	0	1536.5	1536.5	0	1676.4	1676.4	0	1676.4	1676.4	1	1676.4	1676.4	2	1676.4	1676.4	2	1658.4	1658.4
Low_H3_N10_K4_1	16	3261.9	3261.9	13	3235.5	3235.5	3	3261.9	3261.9	3	3261.9	3261.9	11	3261.9	3261.9	16	3261.9	3261.9	17	3261.9	3261.9
Low_H3_N10_K4_2	12	4131.6	4131.6	10	4046.2	4046.2	62	4216.2	4216.2	147	4223.5	4223.5	48	4206.0	4206.0	215	4204.3	4204.3	454	4198.3	4198.3
Low_H3_N10_K4_3	7	2947.0	2947.0	5	2924.4	2924.4	4	2968.7	2968.7	4	2968.7	2968.7	12	2968.7	2968.7	20	2968.7	2968.7	25	2968.7	2968.7
Low_H3_N10_K4_4	55	3490.0	3490.0	22	3375.2	3375.2	28	3670.1	3670.1	46	3674.8	3674.8	613	3666.4	3666.4	34	3615.7	3615.7	161	3615.7	3615.7
Low_H3_N10_K4_5	43	2832.0	2832.0	41	2728.8	2728.8	7	2872.1	2872.1	5	2872.1	2872.1	19	2872.1	2872.1	57	2870.8	2870.8	105	2870.8	2870.8
Low_H3_N15_K4_1	-	3349.4	3092.0	-	3417.4	3056.3	-	3172.9	3131.4	-	3169.7	3141.7	-	3188.9	3099.2	-	3475.1	3101.8	-	-	3093.5
Low_H3_N15_K4_2	-	3309.8	3186.2	-	-	3111.3	-	3402.9	3313.9	-	3588.4	3353.7	-	-	3255.0	-	-	3241.4	-	3694.7	3240.1
Low_H3_N15_K4_3	-	-	3666.4	-	-	3645.0	-	3794.3	3740.4	-	3760.4	3746.9	-	-	3705.1	-	-	3677.1	-	-	3672.4
Low_H3_N15_K4_4	3455	3142.5	3142.5	3440	3105.7	3105.7	1375	3200.2	3200.2	567	3200.2	3200.2	3103	3180.0	3180.0	7108	3180.0	3180.0	5550	3180.0	3180.0
Low_H3_N15_K4_5	-	3851.9	3579.2	-	-	3542.6	1403	3643.1	3643.1	3671	3661.4	3661.4	-	-	3622.2	-	-	3600.9	-	-	3588.1
High_H6_N5_K4_1	24	8074.3	8074.3	9	7965.5	7965.5	350	8111.3	8111.3	349	8111.3	8111.3	530	8111.3	8111.3	1122	8111.3	8111.3	1953	8111.3	8111.3
High_H6_N5_K4_2	3046	7136.7	7136.7	3385	7018.1	7018.1	-	7440.9	7345.5	-	7444.2	7351.5	-	7469.9	7249.5	-	-	7211.2	-	-	7177.2
High_H6_N5_K4_3	112	11 179.6	11 179.6	141	11 101.5	11 101.5	457	11 440.2	11 440.2	692	11 474.2	11 474.2	885	11 344.9	11 344.9	605	11 319.0	11 319.0	1131	11 319.0	11 319.0
High_H6_N5_K4_4	13	6686.1	6686.1	78	6553.2	6553.2	53	6961.9	6961.9	86	6985.4	6985.4	106	6887.0	6887.0	145	6887.0	6887.0	301	6864.6	6864.6
High_H6_N5_K4_5	3469	6904.5	6904.5	5340	6850.3	6850.3	707	6996.4	6996.4	603	6996.4	6996.4	-	6999.2	6974.9	-	6998.9	6941.2	-	6974.2	6932.9
Low_H6_N5_K4_1	22	5457.4	5457.4	13	5341.7	5341.7	740	5506.2	5506.2	753	5506.2	5506.2	1003	5506.2	5506.2	2014	5506.2	5506.2	5291	5506.2	5506.2
Low_H6_N5_K4_2	4518	4814.3	4814.3	5453	4695.7	4695.7	-	5130.5	5016.8	-	5108.7	5022.2	-	-	4915.8	-	-	4874.4	-	-	4842.0
Low_H6_N5_K4_3	116	9030.5	9030.5	179	8959.8	8959.8	542	9284.4	9284.4	945	9320.7	9320.7	912	9189.1	9189.1	678	9160.7	9160.7	1379	9160.7	9160.7
Low_H6_N5_K4_4	17	4761.9	4761.9	107	4629.0	4629.0	83	5052.2	5052.2	144	5085.8	5085.8	171	4977.3	4977.3	247	4977.3	4977.3	369	4945.8	4945.8
Low_H6_N5_K4_5	-	4825.1	4805.5	-	4773.5	4740.1	-	4913.4	4913.2	5862	4913.4	4913.4	-	4901.4	4854.8	-	4867.4	4826.2	-	4954.0	4814.3

Table F.3 Detailed results impact demand move cost and maximum on moved demand, K5 instances

	$m = 0.01$&Max.100%			$m = 0.005$&Max.100%			$m = 0.05$&Max.100%			$m = 0.1$&Max.100%			$m = 0.01$&Max.25%			$m = 0.01$&Max.50%			$m = 0.01$&Max.75%		
Instance	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}	T(s)	UB	LB_{best}
High_H3_N5_K5_1	0	2480.8	2480.8	0	2465.2	2465.2	0	2536.8	2536.8	1	2567.4	2567.4	0	2512.3	2512.3	0	2480.8	2480.8	0	2480.8	2480.8
High_H3_N5_K5_2	2	2675.2	2675.2	2	2658.9	2658.9	1	2774.9	2774.9	4	2807.3	2807.3	8	2807.3	2807.3	11	2801.9	2801.9	2	2675.2	2675.2
High_H3_N5_K5_3	0	4858.0	4858.0	0	4810.8	4810.8	1	5117.5	5117.5	2	5160.9	5160.9	6	5145.4	5145.4	12	5145.4	5145.4	1	4971.0	4971.0
High_H3_N5_K5_4	0	3720.3	3720.3	0	3695.7	3695.7	1	3899.7	3899.7	0	3899.7	3899.7	1	3899.7	3899.7	1	3899.7	3899.7	4	3899.7	3899.7
High_H3_N5_K5_5	0	2974.7	2974.7	0	2917.4	2917.4	2	3136.6	3136.6	4	3166.8	3166.8	0	3064.1	3064.1	1	3064.1	3064.1	3	3064.1	3064.1
High_H3_N10_K5_1	3236	6392.1	6392.1	5977	6332.1	6332.1	450	6496.7	6496.7	350	6496.7	6496.7	4925	6496.7	6496.7	-	6487.9	6466.6	3419	6429.8	6429.8
High_H3_N10_K5_2	294	6857.0	6857.0	736	6790.7	6790.7	21	6960.9	6960.9	17	6960.9	6960.9	234	6960.9	6960.9	953	6960.9	6960.9	651	6920.2	6920.2
High_H3_N10_K5_3	2	5467.2	5467.2	3	5437.4	5437.4	6	5534.8	5534.8	16	5556.7	5556.7	226	5535.6	5535.6	6	5503.3	5503.3	7	5503.3	5503.3
High_H3_N10_K5_4	361	6056.0	6056.0	832	5974.6	5974.6	97	6308.8	6308.8	641	6308.8	6308.8	805	6282.2	6282.2	1249	6282.2	6282.2	3857	6282.2	6282.2
High_H3_N10_K5_5	1	5634.9	5634.9	1	5596.0	5596.0	3	5713.6	5713.6	10	5771.5	5771.5	1	5656.4	5656.4	1	5656.4	5656.4	1	5656.4	5656.4
High_H3_N15_K5_1	5860	6985.1	6985.1	-	7609.5	6918.1	-	7084.1	6966.7	-	7068.7	6974.7	-	7486.4	6997.3	6000	7018.5	7018.5	-	7025.7	7013.3
High_H3_N15_K5_2	-	7067.4	7019.4	-	7000.4	6931.6	4189	7118.8	7118.8	-	7218.3	7163.1	-	7191.1	7116.4	3871	7085.8	7085.8	-	-	7061.5
High_H3_N15_K5_3	-	7960.6	7905.5	-	-	7862.2	754	7949.7	7949.7	2446	7966.5	7966.5	6510	7936.2	7936.2	-	-	7930.4	-	-	7920.3
High_H3_N15_K5_4	-	-	6361.7	-	6642.0	6333.7	6214	6388.1	6388.1	2137	6388.1	6388.1	2157	6388.1	6388.1	4405	6388.1	6388.1	-	-	6381.4
High_H3_N15_K5_5	-	-	6771.3	-	-	6713.9	-	-	6867.0	-	7076.0	6889.9	-	-	6803.3	-	-	6781.4	-	-	6773.0
Low_H3_N5_K5_1	0	1606.8	1606.8	0	1591.0	1591.0	0	1667.1	1667.1	1	1697.7	1697.7	0	1642.6	1642.6	0	1606.8	1606.8	0	1606.8	1606.8
Low_H3_N5_K5_2	2	1885.7	1885.7	1	1870.5	1870.5	1	1985.4	1985.4	3	2019.6	2019.6	8	2019.6	2019.6	12	2013.9	2013.9	1	1885.7	1885.7
Low_H3_N5_K5_3	0	3656.1	3656.1	1	3611.8	3611.8	2	3934.0	3934.0	2	3965.8	3965.8	7	3947.7	3947.7	15	3947.7	3947.7	1	3768.2	3768.2
Low_H3_N5_K5_4	0	3136.5	3136.5	0	3111.9	3111.9	1	3319.0	3319.0	0	3319.0	3319.0	1	3319.0	3319.0	2	3319.0	3319.0	6	3319.0	3319.0
Low_H3_N5_K5_5	0	1807.2	1807.2	0	1753.6	1753.6	3	1976.7	1976.7	6	2008.5	2008.5	1	1912.1	1912.1	2	1912.1	1912.1	4	1912.1	1912.1
Low_H3_N10_K5_1	2979	3623.8	3623.8	4869	3563.5	3563.5	699	3728.8	3728.8	540	3728.8	3728.8	5056	3728.8	3728.8	-	3713.6	3709.3	3696	3660.9	3660.9
Low_H3_N10_K5_2	230	4572.5	4572.5	584	4506.2	4506.2	25	4681.3	4681.3	18	4681.3	4681.3	262	4681.3	4681.3	1130	4681.3	4681.3	876	4647.0	4647.0
Low_H3_N10_K5_3	3	3313.5	3313.5	4	3283.7	3283.7	10	3377.7	3377.7	40	3401.2	3401.2	721	3381.3	3381.3	7	3341.9	3341.9	25	3341.9	3341.9
Low_H3_N10_K5_4	356	3871.4	3871.4	820	3790.1	3790.1	127	4139.2	4139.2	192	4157.0	4157.0	393	4101.6	4101.6	1442	4101.6	4101.6	3556	4101.6	4101.6
Low_H3_N10_K5_5	1	2862.0	2862.0	1	2831.7	2831.7	3	2937.9	2937.9	9	2993.9	2993.9	0	2877.1	2877.1	1	2877.1	2877.1	1	2877.1	2877.1
Low_H3_N15_K5_1	1822	3517.8	3517.8	-	3495.2	3457.2	2020	3576.1	3576.1	5691	3579.5	3579.5	5673	3559.0	3559.0	-	3584.7	3555.0	-	3584.5	3553.2
Low_H3_N15_K5_2	-	3736.5	3720.5	-	3705.0	3625.4	5861	3811.3	3811.3	-	3889.4	3852.9	6455	3807.2	3807.2	2882	3778.2	3778.2	-	-	3762.2
Low_H3_N15_K5_3	-	4073.0	4065.6	-	4076.4	4032.3	670	4098.4	4098.4	695	4115.0	4115.0	-	-	4080.3	-	-	4077.9	-	-	4076.1
Low_H3_N15_K5_4	-	-	3513.8	-	-	3491.0	6145	3572.5	3572.5	5338	3572.5	3572.5	-	-	3570.4	-	-	3569.5	-	-	3546.4
Low_H3_N15_K5_5	-	-	4062.6	-	-	4009.5	-	-	4145.3	-	4431.1	4158.6	-	-	4089.5	-	-	4068.8	-	-	4054.6
High_H6_N5_K5_1	443	8948.5	8948.5	232	8874.1	8874.1	6744	9030.1	9030.1	-	9043.7	9034.2	-	8998.6	8998.4	1256	8966.2	8966.2	902	8956.4	8956.4
High_H6_N5_K5_2	3117	7952.5	7952.5	3094	7839.5	7839.5	1434	8176.1	8176.1	-	8249.5	8211.7	-	-	8069.5	-	-	8015.2	6789	7989.1	7989.1
High_H6_N5_K5_3	-	13 332.6	13 254.1	-	13 207.6	13 172.1	327	13 399.1	13 399.1	257	13 399.1	13 399.1	-	13 928.7	13 351.4	-	13 438.0	13 321.6	-	-	13 294.7
High_H6_N5_K5_4	7	7872.5	7872.5	2	7558.1	7558.1	768	8179.6	8179.6	692	8179.6	8179.6	412	8091.8	8091.8	1864	8087.1	8087.1	94	7975.6	7975.6
High_H6_N5_K5_5	182	7918.0	7918.0	725	7866.6	7866.6	749	8086.4	8086.4	713	8102.1	8102.1	130	7965.1	7965.1	730	7965.1	7965.1	1831	7965.1	7965.1
Low_H6_N5_K5_1	428	6327.4	6327.4	261	6253.0	6253.0	-	6422.0	6415.5	-	6445.8	6414.1	-	6412.8	6378.5	1495	6347.1	6347.1	2056	6345.9	6345.9
Low_H6_N5_K5_2	-	5692.2	5653.3	-	5613.0	5538.8	-	5982.1	5875.3	-	6023.5	5878.3	-	-	5728.8	-	-	5672.8	-	-	5656.0
Low_H6_N5_K5_3	-	11 157.6	11 106.8	-	11 120.0	11 022.6	1141	11 282.7	11 282.7	872	11 282.7	11 282.7	-	-	11 203.5	-	-	11 174.1	-	-	11 140.8
Low_H6_N5_K5_4	5	5954.7	5954.7	1	5622.2	5622.2	1389	6285.0	6285.0	1182	6285.0	6285.0	326	6169.0	6169.0	1557	6168.4	6168.4	102	6054.1	6054.1
Low_H6_N5_K5_5	1582	5830.2	5830.2	4029	5778.9	5778.9	3036	5999.5	5999.5	2726	6015.9	6015.9	1117	5879.5	5879.5	4470	5879.5	5879.5	-	5881.2	5873.3

G

Results per instance VRPPO

Tables G.1 - G.12 show the results per instance for both the VRPPO and the VRPPC for the set $\mathcal{A}$ instances derived from Dabia et al. [2019]. Tables G.1 - G.6 give the results for high outsourcing costs and Tables G.7 - G.12 give the results for low outsourcing cost. All these tables report per instance number ('Inst.') and number of customers ('N'), the solution time ('Time') in seconds, the best upper and lower bound ('UB' and 'best LB' respectively), the root lower bound ('root LB'), the size of the branch-and-bound tree ('Tree') and the number of generated SR inequalities ('#SR'). Tables G.13 - G.18 report the same results for the set $\mathcal{B}$ instances as used in Desaulniers [2010]. In all tables the solutions for an instance are reported if at least a lower bound is found for the VRPPO or an upper bound for the VRPPO is established which is lower than the optimal VRPPC solution.

Table G.1 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost a, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101a	25	1	1197.4	1196.8	1197.4	4	0	0	1203.2	1203.2	1203.2	0	0
R102a	25	0	1078.2	1078.2	1078.2	0	0	1	1078.2	1078.2	1078.2	0	0
R103a	25	6	1036.6	1036.0	1036.6	2	0	2	1038.2	1038.2	1038.2	0	1
R104a	25	67	1019.8	1014.3	1019.8	6	0	3	1019.8	1019.8	1019.8	0	0
R105a	25	1	1104.3	1104.3	1104.3	0	0	1	1108.2	1108.2	1108.2	0	0
R106a	25	9	1051.2	1046.1	1051.2	8	2	5	1051.7	1049.6	1051.7	4	0
R107a	25	31	1029.9	1026.2	1029.9	6	1	4	1029.9	1029.9	1029.9	0	0
R108a	25	77	1019.7	1013.0	1019.7	6	0	15	1019.7	1016.7	1019.7	8	1
R109a	25	5	1027.5	1025.1	1027.5	8	1	6	1029.5	1026.1	1029.5	8	0
R110a	25	9	1026.7	1022.7	1026.7	4	0	6	1026.7	1025.3	1026.7	4	0
R111a	25	35	1032.4	1021.6	1032.4	8	0	20	1032.4	1026.7	1032.4	18	0
R112a	25	51	1019.2	1010.2	1019.2	6	1	14	1019.2	1014.9	1019.2	8	0
C101a	25	0	920	920	920	0	0	0	920	920	920	0	0
C102a	25	1	920	920	920	0	0	1	920	920	920	0	0
C103a	25	0	920	920	920	0	0	2	920	920	920	0	0
C104a	25	2	920	920	920	0	0	54	920	920	920	0	0
C105a	25	0	920	920	920	0	0	0	920	920	920	0	0
C106a	25	0	920	920	920	0	0	0	920	920	920	0	0
C107a	25	0	920	920	920	0	0	0	920	920	920	0	0
C108a	25	0	920	920	920	0	0	0	920	920	920	0	0
C109a	25	0	920	920	920	0	0	0	920	920	920	0	0
RC101a	25	4	1528.4	1519.0	1528.4	4	0	2	1528.4	1523.4	1528.4	2	0
RC102a	25	6	1502.0	1500.7	1502.0	2	0	4	1502.0	1501.8	1502.0	2	1
RC103a	25	24	1496.4	1487.4	1496.4	2	0	5	1496.4	1496.4	1496.4	0	0
RC104a	25	106	1490.2	1489.2	1490.2	2	0	18	1490.2	1489.1	1490.2	2	1
RC105a	25	1	1520.9	1520.9	1520.9	0	0	2	1520.9	1520.9	1520.9	0	0
RC106a	25	6	1498.1	1495.5	1498.1	2	0	5	1498.1	1496.1	1498.1	2	0
RC107a	25	46	1486.1	1485.9	1486.1	2	1	10	1486.1	1486.1	1486.1	0	0
RC108a	25	145	1485.1	1483.7	1485.1	2	0	16	1485.1	1485.1	1485.1	0	0
R201a	25	7	1085.0	1085.0	1085.0	0	0	4	1085.0	1085.0	1085.0	0	0
C201a	25	0	920	920	920	0	0	0	920	920	920	0	0
C202a	25	13	920	920	920	0	0	11	920	920	920	0	0
C203a	25	106	920	920	920	0	0	-	920	-	-	0	0
C204a	25	309	920	920	920	0	0	-	920	-	-	0	0
C205a	25	0	920	920	920	0	0	0	920	920	920	0	0
C206a	25	2	920	920	920	0	0	1	920	920	920	0	0
C207a	25	193	920	920	920	0	0	62	920	920	920	0	0
C208a	25	10	920	920	920	0	0	1	920	920	920	0	0
RC201a	25	2833	1377.7	1362.7	1377.7	18	7	80	1377.7	1361.7	1377.7	16	4

Table G.2 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost a, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101a	50	10	2360.1	2344.1	2360.1	12	3	8	2365.9	2358.5	2365.9	8	0
R102a	50	589	2239.2	2229.0	2239.2	22	13	108	2254.9	2238.5	2254.9	44	6
R103a	50	2219	2156.7	2150.6	2156.7	4	1	142	2165.3	2158.8	2165.3	12	2
R105a	50	25	2253.3	2245.1	2253.3	16	9	65	2283.1	2267.4	2283.1	44	8
R106a	50	1608	2187.5	2174.9	2187.5	34	32	180	2199.3	2184.4	2199.3	42	12
R107a	50	-	3605	2120.8	2120.8	6	0	486	2147.2	2129.5	2147.2	40	10
R109a	50	183	2174.8	2167.0	2174.8	18	11	92	2184.2	2169.6	2184.2	22	9
R110a	50	946	2128.8	2120.3	2128.8	10	3	94	2139.1	2123.8	2139.1	14	2
R111a	50	-	3605	2125.3	2125.3	6	0	397	2144.4	2126.2	2144.4	40	16
R112a	50	-	3605	2087.9	2087.9	4	1	377	2109.0	2094.0	2109.0	16	3
C101a	50	0	1720	1720	1720	0	0	0	1720	1720	1720	0	0
C102a	50	1	1720	1720	1720	0	0	2	1720	1720	1720	0	0
C103a	50	2	1720	1720	1720	0	0	181	1720	1720	1720	0	0
C104a	50	7	1720	1720	1720	0	0	-	1720	-	-	0	0
C105a	50	0	1720	1720	1720	0	0	1	1720	1720	1720	0	0
C106a	50	0	1720	1720	1720	0	0	0	1720	1720	1720	0	0
C107a	50	0	1720	1720	1720	0	0	0	1720	1720	1720	0	0
C108a	50	0	1720	1720	1720	0	0	1	1720	1720	1720	0	0
C109a	50	1	1720	1720	1720	0	0	1	1720	1720	1720	0	0
RC101a	50	10	2893.2	2891.3	2893.2	2	0	15	2893.9	2891.9	2893.9	6	0
RC102a	50	590	2788.2	2780.4	2788.2	24	37	39	2788.2	2779.5	2788.2	10	0
RC103a	50	-	2758.1	2743.7	2752.8	34	66	312	2759.5	2746.5	2759.5	32	25
RC104a	50	-	3395	2680.0	2680.0	2	4	-	2708.5	2688.7	2707.6	100	95
RC105a	50	2668	2837.2	2807.3	2837.2	518	437	242	2837.9	2819.2	2837.9	116	31
RC106a	50	1126	2771.3	2755.9	2771.3	112	123	217	2771.3	2761.6	2771.3	56	13
RC107a	50	-	2733.7	2704.2	2713.0	38	50	-	2730.0	2712.4	2729.7	366	337
RC108a	50	-	3395	2668.9	2668.9	4	1	-	2703.9	2683.0	2696.9	96	86
C201a	50	0	1720	1720	1720	0	0	1	1720	1720	1720	0	0
C202a	50	101	1720	1720	1720	0	0	207	1720	1720	1720	0	0
C203a	50	1031	1720	1720	1720	0	0	-	1720	-	-	0	0
C205a	50	4	1720	1720	1720	0	0	1	1720	1720	1720	0	0
C206a	50	15	1720	1720	1720	0	0	4	1720	1720	1720	0	0
C207a	50	401	1720	1720	1720	0	0	1940	1720	1720	1720	0	0
C208a	50	54	1720	1720	1720	0	0	17	1720	1720	1720	0	0
R101a	100	21	4996.3	4995.6	4996.3	2	1	93	5053.9	5046.1	5053.9	12	7
R105a	100	-	7290	4545.7	4573.4	14	32	-	4595.0	4555.3	4593.1	332	480
C101a	100	0	3620	3620	3620	0	0	1	3620	3620	3620	0	0
C102a	100	5	3620	3620	3620	0	0	42	3620	3620	3620	0	0
C103a	100	18	3620	3620	3620	0	0	-	3620	-	-	0	0
C104a	100	50	3620	3620	3620	0	0	-	3620	-	-	0	0
C105a	100	0	3620	3620	3620	0	0	1	3620	3620	3620	0	0
C106a	100	0	3620	3620	3620	0	0	2	3620	3620	3620	0	0
C107a	100	1	3620	3620	3620	0	0	1	3620	3620	3620	0	0
C108a	100	2	3620	3620	3620	0	0	3	3620	3620	3620	0	0
C109a	100	4	3620	3620	3620	0	0	6	3620	3620	3620	0	0
RC101a	100	248	5124.2	5118.1	5124.2	10	15	120	5161.7	5156.7	5161.7	14	6
RC102a	100	-	6034	4987.0	4989.9	6	11	221	5012.2	5009.6	5012.2	6	4
RC105a	100	194	5012.7	5009.4	5012.7	2	4	161	5046.1	5045.3	5046.1	4	4
RC106a	100	1611	4964.0	4962.4	4964.0	4	6	152	4980.9	4979.7	4980.9	4	2
C201a	100	11	3620	3620	3620	0	0	6	3620	3620	3620	0	0
C202a	100	563	3620	3620	3620	0	0	-	3620	-	-	0	0
C205a	100	61	3620	3620	3620	0	0	11	3620	3620	3620	0	0
C206a	100	278	3620	3620	3620	0	0	32	3620	3620	3620	0	0
C207a	100	1917	3620	3620	3620	0	0	-	3620	-	-	0	0
C208a	100	1080	3620	3620	3620	0	0	1024	3620	3620	3620	0	0

Table G.3 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost b, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101b	25	0	739.9	738.1	739.9	4	0	1	740.9	740.9	740.9	0	0
R102b	25	0	638.3	638.3	638.3	0	0	0	638.3	638.3	638.3	0	0
R103b	25	1	577.4	577.4	577.4	0	0	1	577.4	577.4	577.4	0	0
R104b	25	3	552.9	552.9	552.9	0	0	2	553.6	553.6	553.6	0	0
R105b	25	1	649.9	649.0	649.9	4	0	1	649.9	648.7	649.9	2	0
R106b	25	1	578.1	578.1	578.1	0	0	2	578.1	578.1	578.1	0	0
R107b	25	1	554.9	554.9	554.9	0	0	1	554.9	554.9	554.9	0	0
R108b	25	5	541.5	541.5	541.5	0	0	2	541.5	541.5	541.5	0	0
R109b	25	0	563.5	563.5	563.5	0	0	1	565.5	565.5	565.5	0	0
R110b	25	1	557.4	557.4	557.4	0	0	2	557.4	557.4	557.4	0	0
R111b	25	2	564.2	564.2	564.2	0	0	2	566.1	566.1	566.1	0	0
R112b	25	20	532.4	531.3	532.4	4	0	5	532.4	532.4	532.4	0	0
C101b	25	22	570.1	552.7	570.1	6	2	10	570.1	553.7	570.1	2	1
C102b	25	2578	570.1	545.7	570.1	8	7	48	570.1	546.5	570.1	6	0
C105b	25	29	570.1	549.7	570.1	6	0	11	570.1	550.1	570.1	2	0
C106b	25	19	570.1	548.9	570.1	6	1	8	570.1	551.0	570.1	2	0
C107b	25	45	563.9	541.0	563.9	8	0	17	563.9	542.3	563.9	2	0
C108b	25	232	563.9	540.3	563.9	8	2	24	563.9	540.9	563.9	2	0
C109b	25	1505	563.9	539.3	563.9	8	6	47	563.9	540.4	563.9	6	2
RC101b	25	5	710.7	695.9	710.7	4	1	6	710.7	710.0	710.7	4	0
RC102b	25	0	617.0	617.0	617.0	0	0	1	617.0	617.0	617.0	0	0
RC103b	25	3	603.9	603.9	603.9	0	0	1	603.9	603.9	603.9	0	0
RC104b	25	8	577.1	577.1	577.1	0	0	2	577.1	577.1	577.1	0	0
RC105b	25	2	667.6	667.6	667.6	0	0	2	667.6	666.9	667.6	2	0
RC106b	25	1	616.5	616.5	616.5	0	0	1	616.5	616.5	616.5	0	0
RC107b	25	2	568.9	568.9	568.9	0	0	1	568.9	568.9	568.9	0	0
RC108b	25	5	565.0	565.0	565.0	0	0	2	565.0	565.0	565.0	0	0
R201b	25	21	670.6	670.6	670.6	0	0	12	670.6	657.2	670.6	2	0
R202b	25	351	603.3	603.3	603.3	0	2	46	603.3	603.3	603.3	0	1
C201b	25	4	503.4	487.7	503.4	6	1	3	503.4	487.7	503.4	2	0
RC201b	25	1	571.2	571.2	571.2	0	0	3	571.2	571.2	571.2	0	0
RC202b	25	1098	548.8	548.8	548.8	0	0	17	548.8	548.8	548.8	0	0
RC205b	25	11	548.9	548.9	548.9	0	0	3	548.9	548.9	548.9	0	0
RC206b	25	15	535.1	535.1	535.1	0	0	5	535.1	535.1	535.1	0	0

Table G.4 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost b, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101b	50	9	1308.2	1304.1	1308.2	14	8	7	1308.2	1304.3	1308.2	8	0
R102b	50	23	1180.8	1180.0	1180.8	2	0	10	1180.8	1180.8	1180.8	0	0
R103b	50	2589	1074.4	1065.2	1074.4	26	5	304	1078.1	1067.4	1078.1	68	5
R105b	50	37	1177.4	1175.6	1177.4	18	19	20	1182.6	1179.2	1182.6	4	3
R106b	50	358	1098.1	1088.0	1098.1	18	10	156	1098.1	1086.6	1098.1	40	3
R107b	50	933	1026.2	1020.2	1026.2	4	4	72	1026.2	1020.7	1026.2	4	0
R109b	50	159	1086.8	1073.0	1086.8	24	14	147	1086.8	1072.8	1086.8	42	5
R110b	50	1183	1030.6	1015.0	1030.6	26	13	305	1030.6	1015.5	1030.6	52	11
R111b	50	223	1014.8	1013.7	1014.8	2	0	46	1014.8	1013.6	1014.8	2	0
R112b	50	-	983.5	960.2	961.3	10	11	668	971.9	960.8	971.9	30	29
C101b	50	591	1090.7	1072.7	1090.7	54	113	250	1090.7	1081.1	1090.7	50	74
C102b	50	-	1720	1051.0	1051.0	2	2	518	1081.9	1053.6	1081.9	34	65
C105b	50	1146	1090.7	1064.9	1090.7	62	171	340	1090.7	1073.9	1090.7	56	63
C106b	50	936	1090.7	1071.2	1090.7	58	157	271	1090.7	1074.4	1090.7	48	62
C107b	50	633	1079.7	1065.7	1079.7	16	47	165	1079.7	1069.5	1079.7	18	22
C108b	50	2241	1078.1	1059.8	1078.1	24	88	366	1078.1	1063.3	1078.1	32	51
C109b	50	-	1720	1051.1	1051.1	2	3	612	1076.7	1053.3	1076.7	30	53
RC101b	50	528	1397.8	1344.4	1397.8	158	207	273	1397.8	1358.7	1397.8	96	42
RC102b	50	2456	1199.1	1171.0	1199.1	186	275	339	1199.1	1180.5	1199.1	60	47
RC103b	50	548	1127.0	1120.2	1127.0	6	7	41	1127.0	1127.0	1127.0	0	1
RC104b	50	-	3395	1026.3	1026.3	2	2	-	1071.1	1041.8	1064.1	56	112
RC105b	50	779	1247.2	1222.9	1247.2	84	133	129	1247.2	1239.5	1247.2	24	11
RC106b	50	1314	1171.4	1140.4	1171.4	122	205	569	1171.4	1151.1	1171.4	76	65
RC107b	50	191	1077.1	1077.1	1077.1	0	1	35	1077.1	1077.1	1077.1	0	0
RC108b	50	-	3395	1031.3	1031.3	6	13	875	1054.6	1041.8	1054.6	10	8
RC201b	50	71	1058.1	1058.1	1058.1	0	0	31	1058.1	1058.1	1058.1	0	2
R101b	100	113	2623.1	2618.5	2623.1	16	48	203	2647.7	2643.6	2647.7	24	24
R105b	100	-	7290	2122.1	2130.7	16	60	3239	2147.0	2125.4	2147.0	248	425
C101b	100	12	2609.8	2609.8	2609.8	0	0	30	2609.8	2609.8	2609.8	0	1
C105b	100	115	2548.8	2548.8	2548.8	0	3	24	2548.8	2548.8	2548.8	0	2
C106b	100	35	2570.1	2570.1	2570.1	0	0	48	2570.1	2570.1	2570.1	0	1
C107b	100	24	2537.7	2537.7	2537.7	0	4	46	2537.7	2537.7	2537.7	0	0
C108b	100	2899	2527.2	2523.6	2527.2	14	39	362	2527.6	2523.3	2527.6	8	11
RC101b	100	523	2820.2	2815.6	2820.2	18	47	98	2857.7	2855.1	2857.7	6	4
RC102b	100	-	6034	2682.6	2682.6	2	6	285	2708.2	2707.1	2708.2	6	6
RC105b	100	1017	2708.7	2707.2	2708.7	6	16	191	2742.1	2741.2	2742.1	6	7
RC106b	100	1264	2660.0	2658.6	2660.0	2	5	89	2676.9	2676.9	2676.9	0	0

Table G.5 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost c, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101c	25	0	669.9	669.9	669.9	0	0	0	669.9	669.9	669.9	0	0
R102c	25	0	575.3	575.3	575.3	0	0	1	575.3	575.3	575.3	0	0
R103c	25	1	517.6	517.6	517.6	0	0	1	517.6	517.6	517.6	0	0
R104c	25	1	484.9	484.9	484.9	0	0	1	484.9	484.9	484.9	0	0
R105c	25	1	580.9	580.9	580.9	0	0	0	580.9	580.9	580.9	0	0
R106c	25	0	511.1	511.1	511.1	0	0	1	511.1	511.1	511.1	0	0
R107c	25	1	485.9	485.9	485.9	0	0	1	485.9	485.9	485.9	0	0
R108c	25	3	472.7	472.7	472.7	0	0	2	472.7	472.7	472.7	0	0
R109c	25	1	501.5	501.5	501.5	0	0	1	501.5	501.5	501.5	0	0
R110c	25	1	492.8	492.8	492.8	0	0	1	492.8	492.8	492.8	0	0
R111c	25	2	497.2	497.2	497.2	0	0	4	500.2	498.2	500.2	2	0
R112c	25	10	465.4	465.0	465.4	4	0	9	465.4	465.2	465.4	2	0
C101c	25	24	397.6	393.1	397.6	6	2	10	397.6	393.9	397.6	2	0
C102c	25	1605	392.1	386.6	392.1	6	4	56	392.1	388.3	392.1	4	0
C105c	25	42	397.6	392.0	397.6	6	1	12	397.6	392.7	397.6	2	0
C106c	25	23	397.6	391.3	397.6	6	1	10	397.6	392.8	397.6	2	0
C107c	25	25	382.6	382.6	382.6	0	1	11	382.6	382.6	382.6	2	0
C108c	25	109	382.6	381.5	382.6	6	6	17	382.6	382.6	382.6	2	0
C109c	25	983	382.6	380.6	382.6	6	3	37	382.6	382.1	382.6	2	0
RC101c	25	1	581.3	581.3	581.3	0	0	2	581.3	581.3	581.3	0	0
RC102c	25	1	482.0	482.0	482.0	0	0	0	482.0	482.0	482.0	0	0
RC103c	25	1	468.9	468.9	468.9	0	0	1	468.9	468.9	468.9	0	0
RC104c	25	1	442.1	442.1	442.1	0	1	1	442.1	442.1	442.1	0	0
RC105c	25	2	541.6	541.6	541.6	0	0	1	541.6	541.6	541.6	0	0
RC106c	25	1	481.5	481.5	481.5	0	0	1	481.5	481.5	481.5	0	0
RC107c	25	1	433.9	433.9	433.9	0	0	1	433.9	433.9	433.9	0	0
RC108c	25	2	430.0	430.0	430.0	0	0	1	430.0	430.0	430.0	0	0
R201c	25	3	580.6	580.6	580.6	0	0	4	580.6	580.6	580.6	0	0
R202c	25	16	513.3	513.3	513.3	0	1	4	513.3	513.3	513.3	0	0
R203c	25	499	478.7	478.7	478.7	0	1	-	478.7	478.7	478.7	2	0
R205c	25	10	450.9	450.9	450.9	0	1	4	450.9	450.9	450.9	0	0
C201c	25	1141	403.4	401.8	403.4	4	4	52	403.4	403.4	403.4	0	1
RC201c	25	1	466.2	466.2	466.2	0	0	3	466.2	466.2	466.2	0	0
RC202c	25	8	443.8	443.8	443.8	0	0	2	443.8	443.8	443.8	0	0
RC203c	25	21	432.7	432.7	432.7	0	1	56	432.7	432.7	432.7	0	0
RC205c	25	1	443.9	443.9	443.9	0	0	3	443.9	443.9	443.9	0	0
RC206c	25	4	430.1	430.1	430.1	0	0	5	430.1	430.1	430.1	0	0
RC207c	25	92	403.9	403.9	403.9	0	1	11	403.9	403.9	403.9	0	0

Table G.6 Set $\mathcal{A}$ instances, high outsourcing cost, vehicle cost c, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101c	50	7	1157.2	1150.7	1157.2	14	5	8	1157.2	1151.1	1157.2	6	3
R102c	50	4	1034.8	1034.8	1034.8	0	0	5	1034.8	1034.8	1034.8	0	0
R103c	50	180	917.8	917.1	917.8	2	4	25	917.8	917.0	917.8	2	1
R104c	50	-	3605	801.5	801.5	2	2	1847	810.9	803.0	810.9	28	5
R105c	50	47	1036.4	1032.5	1036.4	24	19	42	1037.9	1034.0	1037.9	18	2
R106c	50	71	938.5	935.0	938.5	6	1	67	938.5	934.1	938.5	8	0
R107c	50	578	872.1	870.0	872.1	6	4	69	872.1	870.4	872.1	4	2
R108c	50	-	3605	783.7	783.7	2	3	1458	792.1	785.1	792.1	22	17
R109c	50	91	928.7	922.5	928.7	12	6	47	928.7	922.2	928.7	8	3
R110c	50	167	867.1	865.0	867.1	2	0	75	867.1	865.2	867.1	4	2
R111c	50	15	853.8	853.8	853.8	0	0	14	853.8	853.8	853.8	0	0
R112c	50	-	3605	803.1	803.9	14	14	797	813.5	804.6	813.5	34	32
C101c	50	519	760.7	748.3	760.7	50	115	234	760.7	752.5	760.7	48	58
C102c	50	-	1720	735.6	735.6	4	2	630	751.9	740.0	751.9	48	92
C105c	50	984	760.7	744.5	760.7	68	166	449	760.7	749.3	760.7	80	93
C106c	50	671	760.7	746.7	760.7	56	135	265	760.7	748.8	760.7	48	62
C107c	50	310	749.7	744.6	749.7	12	25	158	749.7	747.8	749.7	18	28
C108c	50	2240	748.1	741.0	748.1	32	120	215	748.1	744.4	748.1	18	17
C109c	50	-	746.7	737.1	743.1	12	31	576	746.7	738.2	746.7	30	49
RC101c	50	541	1172.8	1130.7	1172.8	160	232	195	1172.8	1147.0	1172.8	54	29
RC102c	50	3082	989.1	960.1	989.1	232	501	489	989.1	969.4	989.1	82	71
RC103c	50	1300	917.0	905.4	917.0	22	20	64	917.0	917.0	917.0	0	0
RC104c	50	-	3395	815.6	815.6	6	17	-	886.8	828.4	851.0	56	80
RC105c	50	169	1022.2	1014.3	1022.2	16	10	35	1022.2	1021.0	1022.2	2	0
RC106c	50	1584	961.4	926.0	961.4	142	270	503	961.4	935.2	961.4	60	74
RC107c	50	355	867.1	865.2	867.1	2	4	51	867.1	866.4	867.1	2	1
RC108c	50	-	874.4	821.7	833.2	10	21	730	844.6	828.7	844.6	10	9
R201c	50	347	956.8	955.7	956.8	2	22	190	956.8	955.5	956.8	2	11
RC201c	50	19	893.1	893.1	893.1	0	0	33	893.1	893.1	893.1	0	0
R101c	100	100	2317.1	2312.4	2317.1	14	39	195	2341.7	2336.7	2341.7	22	26
R105c	100	-	7290	1816.5	1823.9	16	61	-	1841.0	1819.9	1840.7	274	467
C101c	100	11	1874.8	1874.8	1874.8	0	2	21	1874.8	1874.8	1874.8	0	0
C105c	100	18	1813.8	1813.8	1813.8	0	1	18	1813.8	1813.8	1813.8	0	0
C106c	100	54	1835.1	1835.1	1835.1	0	2	29	1835.1	1835.1	1835.1	0	0
C107c	100	22	1802.7	1802.7	1802.7	0	2	41	1802.7	1802.7	1802.7	0	1
C108c	100	1665	1792.2	1789.6	1792.2	8	34	309	1792.6	1789.4	1792.6	8	11
RC101c	100	323	2532.2	2528.4	2532.2	8	22	101	2569.7	2567.3	2569.7	6	4
RC102c	100	-	6034	2394.7	2398.1	4	9	182	2420.2	2418.9	2420.2	4	4
RC105c	100	1242	2420.7	2419.8	2420.7	6	13	177	2454.1	2453.2	2454.1	4	5
RC106c	100	1406	2372.0	2370.2	2372.0	4	11	199	2388.9	2388.6	2388.9	2	4

Table G.7 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost a, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101a	25	0	1081.6	1080.7	1081.6	2	0	1	1086.2	1086.2	1086.2	0	0
R102a	25	0	1021.7	1021.7	1021.7	0	0	1	1033.9	1033.9	1033.9	0	0
R103a	25	1	993.5	993.5	993.5	0	0	1	996.2	996.2	996.2	0	0
R104a	25	3	987.0	987.0	987.0	0	0	8	995.6	995.2	995.6	4	0
R105a	25	0	1028.8	1028.8	1028.8	0	0	0	1042.3	1042.3	1042.3	0	0
R106a	25	0	1002.4	1002.4	1002.4	0	0	1	1005.2	1005.2	1005.2	0	0
R107a	25	1	993.5	993.5	993.5	0	0	1	996.2	996.2	996.2	0	0
R108a	25	2	983.2	983.2	983.2	0	0	6	995.6	991.3	995.6	4	0
R109a	25	0	997.2	997.2	997.2	0	0	1	1000.6	1000.6	1000.6	0	0
R110a	25	1	997.2	997.2	997.2	0	0	1	997.5	997.5	997.5	0	0
R111a	25	3	998.5	996.6	998.5	2	0	2	1001.2	1001.2	1001.2	0	0
R112a	25	4	985.7	985.7	985.7	0	0	3	992.4	992.4	992.4	0	0
C101a	25	0	230	230	230	0	0	0	230	230	230	0	0
C102a	25	0	230	230	230	0	0	0	230	230	230	0	0
C103a	25	0	230	230	230	0	0	0	230	230	230	0	0
C104a	25	0	230	230	230	0	0	1	230	230	230	0	0
C105a	25	0	230	230	230	0	0	0	230	230	230	0	0
C106a	25	0	230	230	230	0	0	0	230	230	230	0	0
C107a	25	0	230	230	230	0	0	0	230	230	230	0	0
C108a	25	0	230	230	230	0	0	0	230	230	230	0	0
C109a	25	0	230	230	230	0	0	0	230	230	230	0	0
RC101a	25	0	1080	1080	1080	0	0	1	1080	1080	1080	0	0
RC102a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC103a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC104a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC105a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC106a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC107a	25	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0
RC108a	25	1	1080	1080	1080	0	0	1	1080	1080	1080	0	0
R201a	25	1	1023.4	1023.4	1023.4	0	0	2	1023.4	1023.4	1023.4	0	0
C201a	25	0	230	230	230	0	0	0	230	230	230	0	0
C202a	25	0	230	230	230	0	0	0	230	230	230	0	0
C203a	25	0	230	230	230	0	0	1	230	230	230	0	0
C204a	25	2	230	230	230	0	0	3	230	230	230	0	0
C205a	25	0	230	230	230	0	0	0	230	230	230	0	0
C206a	25	1	230	230	230	0	0	0	230	230	230	0	0
C207a	25	4	230	230	230	0	0	2	230	230	230	0	0
C208a	25	0	230	230	230	0	0	0	230	230	230	0	0
RC201a	25	0	1080	1080	1080	0	0	1	1080	1080	1080	0	0
RC202a	25	173	1080	1080	1080	0	0	27	1080	1080	1080	0	0
RC203a	25	481	1080	1080	1080	0	0	-	1080	-	-	0	0
RC204a	25	1798	1080	1080	1080	0	0	-	1080	-	-	0	0
RC205a	25	7	1080	1080	1080	0	0	2	1080	1080	1080	0	0
RC206a	25	28	1080	1080	1080	0	0	8	1080	1080	1080	0	0
RC207a	25	1484	1080	1080	1080	0	0	891	1080	1080	1080	0	0

Table G.8 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost a, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101a	50	0	2170.1	2170.1	2170.1	0	0	1	2196.2	2196.2	2196.2	0	0
R102a	50	139	2126.2	2122.7	2126.2	4	0	15	2143.6	2142.2	2143.6	2	0
R103a	50	352	2079.9	2079.7	2079.9	2	2	233	2100.8	2093.9	2100.8	34	8
R104a	50	2251	2050.3	2050.2	2050.3	2	1	847	2063.4	2058.4	2063.4	8	1
R105a	50	8	2149.3	2143.9	2149.3	4	2	9	2175.1	2170.4	2175.1	4	0
R106a	50	216	2101.9	2100.1	2101.9	6	2	67	2121.3	2117.0	2121.3	16	1
R107a	50	770	2072.8	2068.7	2072.8	2	0	128	2087.2	2083.2	2087.2	14	2
R108a	50	2332	2048.5	2047.5	2048.5	2	0	1423	2063.4	2058.9	2063.4	16	2
R109a	50	57	2101.2	2099.9	2101.2	12	2	14	2118.1	2118.1	2118.1	0	0
R110a	50	48	2069.3	2069.3	2069.3	0	0	40	2083.7	2082.7	2083.7	4	0
R111a	50	225	2062.9	2062.9	2062.9	0	0	50	2073.9	2069.9	2073.9	4	0
R112a	50	-	2523.5	2052.6	2052.9	14	7	278	2068.1	2061.4	2068.1	22	3
C101a	50	0	430	430	430	0	0	0	430	430	430	0	0
C102a	50	0	430	430	430	0	0	1	430	430	430	0	0
C103a	50	1	430	430	430	0	0	1	430	430	430	0	0
C104a	50	2	430	430	430	0	0	74	430	430	430	0	0
C105a	50	0	430	430	430	0	0	0	430	430	430	0	0
C106a	50	0	430	430	430	0	0	0	430	430	430	0	0
C107a	50	0	430	430	430	0	0	1	430	430	430	0	0
C108a	50	0	430	430	430	0	0	0	430	430	430	0	0
C109a	50	1	430	430	430	0	0	1	430	430	430	0	0
RC101a	50	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0
RC102a	50	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0
RC103a	50	1	1940	1940	1940	0	0	1	1940	1940	1940	0	0
RC104a	50	3	1940	1940	1940	0	0	1	1940	1940	1940	0	0
RC105a	50	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0
RC106a	50	0	1940	1940	1940	0	0	1	1940	1940	1940	0	0
RC107a	50	2	1940	1940	1940	0	0	0	1940	1940	1940	0	0
RC108a	50	6	1940	1940	1940	0	0	2	1940	1940	1940	0	0
C201a	50	1	430	430	430	0	0	1	430	430	430	0	0
C202a	50	3	430	430	430	0	0	2	430	430	430	0	0
C203a	50	20	430	430	430	0	0	814	430	430	430	0	0
C204a	50	370	430	430	430	0	0	-	430	-	-	0	0
C205a	50	1	430	430	430	0	0	1	430	430	430	0	0
C206a	50	1	430	430	430	0	0	2	430	430	430	0	0
C207a	50	31	430	430	430	0	0	10	430	430	430	0	0
C208a	50	5	430	430	430	0	0	2	430	430	430	0	0
RC201a	50	4	1940	1940	1940	0	0	2	1940	1940	1940	0	0
RC202a	50	817	1940	1940	1940	0	0	-	1940	-	-	0	0
RC205a	50	602	1940	1940	1940	0	0	329	1940	1940	1940	0	0
RC206a	50	420	1940	1940	1940	0	0	85	1940	1940	1940	0	0
R101a	100	21	4368.9	4368.6	4368.9	4	2	20	4376.8	4376.8	4376.8	0	0
R105a	100	883	4281.7	4278.3	4281.7	12	13	325	4309.6	4299.7	4309.6	36	15
R109a	100	-	5103	4191.5	4191.5	2	0	340	4206.6	4203.8	4206.6	6	2
C101a	100	0	905	905	905	0	0	1	905	905	905	0	0
C102a	100	2	905	905	905	0	0	3	905	905	905	0	0
C103a	100	4	905	905	905	0	0	16	905	905	905	0	0
C104a	100	12	905	905	905	0	0	715	905	905	905	0	0
C105a	100	0	905	905	905	0	0	1	905	905	905	0	0
C106a	100	1	905	905	905	0	0	1	905	905	905	0	0
C107a	100	0	905	905	905	0	0	1	905	905	905	0	0
C108a	100	1	905	905	905	0	0	2	905	905	905	0	0
C109a	100	1	905	905	905	0	0	3	905	905	905	0	0
RC101a	100	1	3448	3448	3448	0	0	1	3448	3448	3448	0	0
RC102a	100	4	3448	3448	3448	0	0	2	3448	3448	3448	0	0
RC103a	100	18	3448	3448	3448	0	0	14	3448	3448	3448	0	0
RC104a	100	173	3448	3448	3448	0	0	807	3448	3448	3448	0	0
RC105a	100	2	3448	3448	3448	0	0	2	3448	3448	3448	0	0
RC106a	100	3	3448	3448	3448	0	0	2	3448	3448	3448	0	0
RC107a	100	23	3448	3448	3448	0	0	8	3448	3448	3448	0	0
RC108a	100	156	3448	3448	3448	0	0	37	3448	3448	3448	0	0
C201a	100	3	905	905	905	0	0	4	905	905	905	0	0
C202a	100	30	905	905	905	0	0	182	905	905	905	0	0
C203a	100	1285	905	905	905	0	0	-	905	-	-	0	0
C205a	100	8	905	905	905	0	0	6	905	905	905	0	0
C206a	100	18	905	905	905	0	0	8	905	905	905	0	0
C207a	100	230	905	905	905	0	0	46	905	905	905	0	0
C208a	100	144	905	905	905	0	0	19	905	905	905	0	0
RC201a	100	1740	3448	3448	3448	0	0	85	3448	3448	3448	0	0

Table G.9 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost b, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101b	25	0	694.7	692.7	694.7	2	0	0	694.7	692.7	694.7	2	0
R102b	25	1	604.7	604.7	604.7	0	1	1	604.7	604.7	604.7	0	0
R103b	25	1	561.9	561.9	561.9	0	0	1	561.9	561.9	561.9	0	0
R104b	25	4	543.0	540.4	543.0	4	0	1	543.0	543.0	543.0	0	0
R105b	25	0	620.3	620.3	620.3	0	0	1	620.3	620.3	620.3	0	0
R106b	25	0	569.6	569.6	569.6	0	0	1	569.6	569.6	569.6	0	0
R107b	25	3	547.4	547.3	547.4	4	0	1	547.4	547.4	547.4	0	0
R108b	25	8	530.8	530.8	530.8	0	0	3	530.8	530.8	530.8	0	0
R109b	25	1	549.0	549.0	549.0	0	0	0	549.0	549.0	549.0	0	0
R110b	25	2	549.0	547.1	549.0	4	1	4	549.0	547.1	549.0	4	0
R111b	25	3	552.2	551.1	552.2	2	0	1	552.2	552.2	552.2	0	0
R112b	25	12	524.9	522.7	524.9	4	1	5	524.9	524.0	524.9	2	0
C101b	25	0	230	230	230	0	0	0	230	230	230	0	0
C102b	25	0	230	230	230	0	0	0	230	230	230	0	0
C103b	25	0	230	230	230	0	0	0	230	230	230	0	0
C104b	25	0	230	230	230	0	0	1	230	230	230	0	0
C105b	25	0	230	230	230	0	0	0	230	230	230	0	0
C106b	25	0	230	230	230	0	0	0	230	230	230	0	0
C107b	25	0	230	230	230	0	0	0	230	230	230	0	0
C108b	25	0	230	230	230	0	0	1	230	230	230	0	0
C109b	25	0	230	230	230	0	0	0	230	230	230	0	0
RC101b	25	1	673.4	671.8	673.4	2	0	3	673.4	669.9	673.4	2	0
RC102b	25	2	617.0	617.0	617.0	0	0	1	617.0	617.0	617.0	0	0
RC103b	25	1	603.9	603.9	603.9	0	0	2	603.9	603.9	603.9	0	0
RC104b	25	5	577.1	577.1	577.1	0	0	1	577.1	577.1	577.1	0	0
RC105b	25	2	661.3	661.3	661.3	0	0	4	661.3	661.3	661.3	0	0
RC106b	25	1	612.9	612.9	612.9	0	0	1	612.9	612.9	612.9	0	0
RC107b	25	2	568.9	568.9	568.9	0	0	1	568.9	568.9	568.9	0	0
RC108b	25	13	565.0	565.0	565.0	0	0	2	565.0	565.0	565.0	0	0
R201b	25	9	653.9	653.9	653.9	0	1	12	653.9	643.4	653.9	2	1
R202b	25	148	594.6	594.6	594.6	0	1	35	594.6	594.6	594.6	0	0
C201b	25	0	230	230	230	0	0	0	230	230	230	0	0
C202b	25	0	230	230	230	0	0	1	230	230	230	0	0
C203b	25	0	230	230	230	0	0	0	230	230	230	0	0
C204b	25	2	230	230	230	0	0	3	230	230	230	0	0
C205b	25	0	230	230	230	0	0	1	230	230	230	0	0
C206b	25	0	230	230	230	0	0	0	230	230	230	0	0
C207b	25	4	230	230	230	0	0	1	230	230	230	0	0
C208b	25	1	230	230	230	0	0	1	230	230	230	0	0
RC201b	25	0	571.2	571.2	571.2	0	0	3	571.2	571.2	571.2	0	0
RC205b	25	4	548.9	548.9	548.9	0	0	3	548.9	548.9	548.9	0	0
RC206b	25	22	535.1	535.1	535.1	0	0	5	535.1	535.1	535.1	0	0

Table G.10 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost b, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101b	50	3	1257.7	1254.7	1257.7	8	1	4	1257.7	1255.2	1257.7	4	0
R102b	50	151	1147.8	1145.5	1147.8	26	15	34	1147.8	1145.9	1147.8	10	1
R103b	50	867	1047.6	1040.2	1047.6	10	0	166	1054.0	1042.9	1054.0	44	4
R105b	50	5	1142.8	1139.3	1142.8	4	1	17	1147.8	1145.8	1147.8	8	0
R106b	50	255	1072.9	1061.8	1072.9	14	1	119	1072.9	1062.8	1072.9	34	3
R107b	50	1160	1005.2	1002.4	1005.2	2	0	58	1005.2	1003.9	1005.2	2	0
R109b	50	93	1069.0	1055.5	1069.0	12	1	82	1069.0	1056.0	1069.0	26	1
R110b	50	468	1018.6	1009.6	1018.6	8	0	201	1018.6	1010.1	1018.6	36	4
R111b	50	252	1005.8	1002.9	1005.8	2	0	53	1005.8	1003.5	1005.8	2	1
R112b	50	-	977.5	952.9	954.2	10	0	251	957.5	954.9	957.5	10	6
C101b	50	0	430	430	430	0	0	0	430	430	430	0	0
C102b	50	0	430	430	430	0	0	1	430	430	430	0	0
C103b	50	1	430	430	430	0	0	1	430	430	430	0	0
C104b	50	2	430	430	430	0	0	78	430	430	430	0	0
C105b	50	0	430	430	430	0	0	1	430	430	430	0	0
C106b	50	0	430	430	430	0	0	0	430	430	430	0	0
C107b	50	0	430	430	430	0	0	0	430	430	430	0	0
C108b	50	0	430	430	430	0	0	1	430	430	430	0	0
C109b	50	0	430	430	430	0	0	0	430	430	430	0	0
RC101b	50	24	1305.1	1303.0	1305.1	8	5	17	1305.1	1304.3	1305.1	2	0
RC102b	50	175	1139.3	1134.2	1139.3	8	10	19	1139.3	1127.9	1139.3	2	1
RC103b	50	817	1108.4	1098.6	1108.4	8	11	62	1108.4	1104.5	1108.4	2	1
RC104b	50	-	1940	1025.5	1025.5	4	2	1598	1060.0	1033.6	1060.0	18	19
RC105b	50	93	1189.7	1185.8	1189.7	10	3	30	1189.7	1186.4	1189.7	6	0
RC106b	50	131	1125.6	1103.9	1125.6	12	10	25	1125.6	1103.1	1125.6	2	0
RC107b	50	1533	1068.7	1049.6	1068.7	14	17	203	1068.7	1056.0	1068.7	2	1
RC108b	50	-	1940	1019.4	1019.4	6	10	421	1043.1	1024.3	1043.1	4	2
C201b	50	0	430	430	430	0	0	1	430	430	430	0	0
C202b	50	3	430	430	430	0	0	3	430	430	430	0	0
C203b	50	20	430	430	430	0	0	810	430	430	430	0	0
C204b	50	400	430	430	430	0	0	-	430	-	-	0	0
C205b	50	0	430	430	430	0	0	1	430	430	430	0	0
C206b	50	2	430	430	430	0	0	1	430	430	430	0	0
C207b	50	31	430	430	430	0	0	11	430	430	430	0	0
C208b	50	4	430	430	430	0	0	2	430	430	430	0	0
RC201b	50	79	1054.2	1054.2	1054.2	0	0	35	1054.2	1054.2	1054.2	0	0
R101b	100	161	2377.5	2373.7	2377.5	24	60	165	2393.3	2394.6	2393.3	22	20
R105b	100	-	5103	2046.0	2053.3	28	107	2559	2064.9	2048.5	2064.9	242	345
C101b	100	0	905	905	905	0	0	1	905	905	905	0	0
C102b	100	2	905	905	905	0	0	4	905	905	905	0	0
C103b	100	5	905	905	905	0	0	16	905	905	905	0	0
C104b	100	12	905	905	905	0	0	697	905	905	905	0	0
C105b	100	0	905	905	905	0	0	1	905	905	905	0	0
C106b	100	1	905	905	905	0	0	1	905	905	905	0	0
C107b	100	0	905	905	905	0	0	1	905	905	905	0	0
C108b	100	1	905	905	905	0	0	2	905	905	905	0	0
C109b	100	2	905	905	905	0	0	3	905	905	905	0	0
RC101b	100	261	2331.5	2324.6	2331.5	20	28	70	2344.1	2342.9	2344.1	4	2
RC102b	100	3560	2209.4	2205.3	2209.4	16	29	139	2221.1	2220.7	2221.1	4	2
RC105b	100	632	2229.8	2226.2	2229.8	6	15	78	2241.5	2241.5	2241.5	0	0
RC106b	100	394	2183.8	2182.3	2183.8	2	3	180	2197.0	2196.7	2197.0	6	6
C201b	100	3	905	905	905	0	0	4	905	905	905	0	0
C202b	100	34	905	905	905	0	0	190	905	905	905	0	0
C203b	100	1349	905	905	905	0	0	-	905	-	-	0	0
C205b	100	7	905	905	905	0	0	5	905	905	905	0	0
C206b	100	18	905	905	905	0	0	8	905	905	905	0	0
C207b	100	230	905	905	905	0	0	45	905	905	905	0	0
C208b	100	140	905	905	905	0	0	19	905	905	905	0	0

Table G.11 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost c, 25 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101c	25	0	631.7	631.2	631.7	2	0	0	631.7	631.2	631.7	2	0
R102c	25	0	542.3	542.3	542.3	0	0	0	542.3	542.3	542.3	0	0
R103c	25	1	494.9	494.9	494.9	0	0	1	494.9	494.9	494.9	0	0
R104c	25	1	476.0	476.0	476.0	0	0	1	476.0	476.0	476.0	0	0
R105c	25	0	559.3	559.3	559.3	0	0	0	559.3	559.3	559.3	0	0
R106c	25	0	503.6	503.6	503.6	0	0	0	503.6	503.6	503.6	0	0
R107c	25	1	478.4	478.4	478.4	0	0	2	478.4	478.4	478.4	0	0
R108c	25	1	463.8	463.8	463.8	0	0	1	463.8	463.8	463.8	0	0
R109c	25	0	491.0	491.0	491.0	0	0	1	491.0	491.0	491.0	0	0
R110c	25	1	482.5	482.5	482.5	0	0	1	482.5	482.5	482.5	0	0
R111c	25	2	488.2	488.1	488.2	2	0	1	488.2	488.2	488.2	0	0
R112c	25	5	457.9	457.9	457.9	0	0	6	457.9	457.0	457.9	2	0
C101c	25	0	230	230	230	0	0	0	230	230	230	0	0
C102c	25	0	230	230	230	0	0	0	230	230	230	0	0
C103c	25	0	230	230	230	0	0	1	230	230	230	0	0
C104c	25	0	230	230	230	0	0	1	230	230	230	0	0
C105c	25	0	230	230	230	0	0	0	230	230	230	0	0
C106c	25	0	230	230	230	0	0	0	230	230	230	0	0
C107c	25	0	230	230	230	0	0	0	230	230	230	0	0
C108c	25	0	230	230	230	0	0	0	230	230	230	0	0
C109c	25	0	230	230	230	0	0	0	230	230	230	0	0
RC101c	25	2	553.4	541.0	553.4	2	1	3	553.4	552.3	553.4	2	0
RC102c	25	1	482.0	482.0	482.0	0	0	0	482.0	482.0	482.0	0	0
RC103c	25	1	468.9	468.9	468.9	0	0	1	468.9	468.9	468.9	0	0
RC104c	25	1	442.1	442.1	442.1	0	0	1	442.1	442.1	442.1	0	0
RC105c	25	1	530.1	530.1	530.1	0	0	1	530.1	530.1	530.1	0	0
RC106c	25	0	481.5	481.5	481.5	0	0	0	481.5	481.5	481.5	0	0
RC107c	25	1	433.9	433.9	433.9	0	0	1	433.9	433.9	433.9	0	0
RC108c	25	3	430.0	430.0	430.0	0	0	1	430.0	430.0	430.0	0	0
R201c	25	1	563.9	563.9	563.9	0	0	4	563.9	563.9	563.9	0	0
R202c	25	333	509.7	509.7	509.7	0	0	14	509.7	509.7	509.7	0	0
R205c	25	13	438.9	438.9	438.9	0	0	4	438.9	438.9	438.9	0	0
C201c	25	0	230	230	230	0	0	0	230	230	230	0	0
C202c	25	0	230	230	230	0	0	0	230	230	230	0	0
C203c	25	1	230	230	230	0	0	0	230	230	230	0	0
C204c	25	2	230	230	230	0	0	3	230	230	230	0	0
C205c	25	0	230	230	230	0	0	1	230	230	230	0	0
C206c	25	0	230	230	230	0	0	0	230	230	230	0	0
C207c	25	4	230	230	230	0	0	1	230	230	230	0	0
C208c	25	0	230	230	230	0	0	1	230	230	230	0	0
RC201c	25	1	466.2	466.2	466.2	0	0	2	466.2	466.2	466.2	0	0
RC202c	25	21	443.8	443.8	443.8	0	0	7	443.8	443.8	443.8	0	0
RC203c	25	83	432.7	432.7	432.7	0	0	598	432.7	432.7	432.7	0	1
RC205c	25	2	443.9	443.9	443.9	0	0	3	443.9	443.9	443.9	0	0
RC206c	25	2	430.1	430.1	430.1	0	0	3	430.1	430.1	430.1	0	0
RC207c	25	93	403.9	403.9	403.9	0	1	5	403.9	403.9	403.9	0	0

Table G.12 Set $\mathcal{A}$ instances, low outsourcing cost, vehicle cost c, 50 & 100 customers

		VRPPO						VRPPC					
Inst.	N	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101c	50	0	1111.7	1111.7	1111.7	0	0	3	1111.7	1111.7	1111.7	0	0
R102c	50	46	1009.8	1008.7	1009.8	14	5	37	1009.8	1008.4	1009.8	14	2
R103c	50	341	896.3	893.9	896.3	6	2	26	896.3	896.3	896.3	0	0
R104c	50	-	2523.5	790.6	790.6	2	3	1348	797.5	792.5	797.5	28	20
R105c	50	5	1002.3	995.7	1002.3	6	1	12	1002.3	1001.4	1002.3	4	0
R106c	50	120	914.9	911.2	914.9	12	8	93	914.9	912.8	914.9	26	1
R107c	50	705	852.2	851.8	852.2	4	1	26	852.2	852.2	852.2	0	0
R109c	50	22	906.8	904.8	906.8	2	1	38	906.8	904.8	906.8	4	1
R110c	50	209	858.0	857.3	858.0	4	3	89	858.0	857.5	858.0	8	1
R111c	50	25	844.8	844.8	844.8	0	1	16	844.8	844.8	844.8	0	1
R112c	50	-	808.8	795.0	796.2	14	11	706	804.5	797.5	804.5	34	26
C101c	50	0	430	430	430	0	0	0	430	430	430	0	0
C102c	50	0	430	430	430	0	0	0	430	430	430	0	0
C103c	50	1	430	430	430	0	0	2	430	430	430	0	0
C104c	50	2	430	430	430	0	0	78	430	430	430	0	0
C105c	50	0	430	430	430	0	0	0	430	430	430	0	0
C106c	50	0	430	430	430	0	0	0	430	430	430	0	0
C107c	50	0	430	430	430	0	0	0	430	430	430	0	0
C108c	50	0	430	430	430	0	0	1	430	430	430	0	0
C109c	50	0	430	430	430	0	0	0	430	430	430	0	0
RC101c	50	156	1109.7	1097.1	1109.7	48	48	68	1109.7	1097.2	1109.7	20	10
RC102c	50	185	944.1	929.5	944.1	12	9	33	944.1	929.6	944.1	2	1
RC103c	50	2148	911.6	885.8	911.6	40	93	112	911.6	900.5	911.6	4	1
RC104c	50	-	1940	812.1	812.1	4	2	-	865.0	823.0	851.2	74	129
RC105c	50	127	981.8	978.2	981.8	14	5	30	981.8	981.1	981.8	4	0
RC106c	50	195	926.0	896.4	926.0	16	23	32	926.0	898.2	926.0	2	2
RC107c	50	1260	867.1	839.9	867.1	14	12	269	867.1	849.4	867.1	6	1
RC108c	50	-	848.1	806.3	824.8	8	11	2672	844.6	819.1	844.6	52	72
R201c	50	606	937.8	936.9	937.8	2	26	137	937.8	936.9	937.8	2	5
C201c	50	1	430	430	430	0	0	1	430	430	430	0	0
C202c	50	3	430	430	430	0	0	3	430	430	430	0	0
C203c	50	20	430	430	430	0	0	909	430	430	430	0	0
C204c	50	428	430	430	430	0	0	-	430	-	-	0	0
C205c	50	1	430	430	430	0	0	2	430	430	430	0	0
C206c	50	1	430	430	430	0	0	1	430	430	430	0	0
C207c	50	30	430	430	430	0	0	11	430	430	430	0	0
C208c	50	4	430	430	430	0	0	2	430	430	430	0	0
RC201c	50	12	889.2	889.2	889.2	0	0	27	889.8	889.8	889.8	0	0
R101c	100	132	2071.5	2067.4	2071.5	20	53	157	2087.3	2085.1	2087.3	22	17
R105c	100	-	5103	1739.7	1748.4	34	128	2661	1758.5	1742.4	1758.5	228	354
C101c	100	0	905	905	905	0	0	1	905	905	905	0	0
C102c	100	2	905	905	905	0	0	3	905	905	905	0	0
C103c	100	5	905	905	905	0	0	16	905	905	905	0	0
C104c	100	12	905	905	905	0	0	716	905	905	905	0	0
C105c	100	0	905	905	905	0	0	1	905	905	905	0	0
C106c	100	1	905	905	905	0	0	1	905	905	905	0	0
C107c	100	0	905	905	905	0	0	1	905	905	905	0	0
C108c	100	1	905	905	905	0	0	2	905	905	905	0	0
C109c	100	1	905	905	905	0	0	3	905	905	905	0	0
RC101c	100	242	2043.5	2038.9	2043.5	18	27	95	2056.1	2054.3	2056.1	8	9
RC102c	100	3355	1921.4	1916.7	1921.4	14	27	164	1933.1	1932.3	1933.1	4	1
RC105c	100	194	1941.8	1939.0	1941.8	2	2	60	1953.5	1953.5	1953.5	0	0
RC106c	100	650	1895.8	1895.4	1895.8	4	7	238	1909.0	1909.4	1909.0	12	6
C201c	100	3	905	905	905	0	0	4	905	905	905	0	0
C202c	100	31	905	905	905	0	0	184	905	905	905	0	0
C203c	100	1300	905	905	905	0	0	-	905	-	-	0	0
C205c	100	8	905	905	905	0	0	5	905	905	905	0	0
C206c	100	17	905	905	905	0	0	8	905	905	905	0	0
C207c	100	230	905	905	905	0	0	47	905	905	905	0	0
C208c	100	144	905	905	905	0	0	19	905	905	905	0	0

Table G.13 Set $\mathcal{B}$ instances, high outsourcing cost, 25 customers

	$Q = 30$, VRPPO						$Q = 30$, VRPPC						$Q = 50$, VRPPO						$Q = 50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	0	788.3	788.3	788.3	0	0	0	788.3	788.3	788.3	0	0	0	617.2	617.2	617.2	0	0	0	617.2	617.2	617.2	0	0
R102	0	773.7	773.7	773.7	0	1	0	781.2	781.2	781.2	0	3	0	571.9	571.9	571.9	0	5	0	571.9	571.9	571.9	0	5
R103	0	753.1	753.1	753.1	0	3	0	753.1	753.1	753.1	0	1	0	530.3	530.3	530.3	0	2	0	530.3	530.3	530.3	0	2
R104	0	753.1	753.1	753.1	0	4	0	753.1	753.1	753.1	0	1	0	523.3	523.3	523.3	0	2	0	523.3	523.3	523.3	0	2
R105	0	759.0	759.0	759.0	0	3	0	759.0	759.0	759.0	0	0	1	577.6	577.6	577.6	0	11	1	579.6	578.5	579.6	6	20
R106	0	756.3	756.3	756.3	0	0	0	756.3	756.3	756.3	0	0	0	532.6	532.6	532.6	0	9	0	532.6	532.6	532.6	0	3
R107	0	749.5	749.5	749.5	0	3	0	749.5	749.5	749.5	0	1	0	529.1	529.1	529.1	0	4	0	529.1	529.1	529.1	0	6
R108	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	0	522.9	522.9	522.9	0	2	1	522.9	522.9	522.9	0	3
R109	0	752.8	752.8	752.8	0	2	0	752.8	752.8	752.8	0	0	0	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R110	0	749.5	749.5	749.5	0	4	1	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R111	0	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	1	522.9	522.9	522.9	0	3	0	522.9	522.9	522.9	0	3
R112	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	4	0	517.1	517.1	517.1	0	0
C101	2	714.9	706.9	714.9	20	17	1	744.6	736.6	744.6	20	14	4	509.3	505.5	509.3	16	53	1	509.3	505.0	509.3	16	57
C102	4	714.9	706.9	714.9	30	23	2	744.6	736.6	744.6	26	23	13	509.2	504.3	509.2	32	122	3	509.2	504.4	509.2	18	53
C103	3	714.9	706.9	714.9	26	27	1	744.6	736.6	744.6	24	24	10	509.2	504.3	509.2	26	51	5	509.2	504.3	509.2	34	96
C104	3	714.9	706.9	714.9	26	28	1	744.6	736.6	744.6	24	19	17	509.2	504.3	509.2	38	106	4	509.2	504.3	509.2	30	81
C105	2	714.9	706.9	714.9	22	24	1	744.6	736.6	744.6	20	16	5	509.3	505.6	509.3	18	54	2	509.3	504.8	509.3	12	39
C106	2	714.9	706.9	714.9	20	17	1	744.6	736.6	744.6	26	31	5	509.3	505.4	509.3	18	61	1	509.3	505.0	509.3	16	48
C107	3	714.9	706.7	714.9	22	25	1	744.6	736.6	744.6	20	16	5	509.3	505.4	509.3	18	44	2	509.3	505.6	509.3	16	44
C108	2	714.9	706.9	714.9	22	23	2	744.6	736.6	744.6	32	21	6	509.3	504.9	509.3	20	28	1	509.3	504.9	509.3	14	15
C109	2	714.9	706.9	714.9	22	27	1	744.6	736.6	744.6	22	20	4	509.3	504.9	509.3	16	15	1	509.3	504.9	509.3	14	14
RC101	0	1514.2	1514.2	1514.2	0	2	0	1645.0	1645.0	1645.0	0	2	15	944.4	914.5	944.4	96	150	4	944.7	915.1	944.7	72	78
RC102	1	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	12	934.7	908.5	934.7	64	97	2	934.7	908.5	934.7	46	42
RC103	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	16	934.4	908.5	934.4	92	143	4	934.4	908.5	934.4	52	63
RC104	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	17	934.4	908.5	934.4	94	144	4	934.4	908.5	934.4	56	70
RC105	1	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	14	938.8	908.5	938.8	82	119	3	938.8	908.5	938.8	50	40
RC106	0	1514.2	1514.2	1514.2	0	2	1	1645.0	1645.0	1645.0	0	2	14	938.8	908.5	938.8	82	118	3	938.8	908.5	938.8	52	52
RC107	0	1514.2	1514.2	1514.2	0	2	0	1645.0	1645.0	1645.0	0	2	15	934.4	908.5	934.4	86	131	4	934.4	908.5	934.4	58	65
RC108	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	19	934.4	908.5	934.4	90	136	4	934.4	908.5	934.4	58	68
R201	1	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	0	557.2	557.2	557.2	0	9	0	557.2	557.2	557.2	0	8
R202	0	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	1	517.1	517.1	517.1	0	2	0	517.1	517.1	517.1	0	0
R203	0	749.5	749.5	749.5	0	3	0	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R204	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R205	0	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	0	517.1	517.1	517.1	0	3	0	517.1	517.1	517.1	0	0
R206	0	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	0	517.1	517.1	517.1	0	2	0	517.1	517.1	517.1	0	0
R207	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	1	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R208	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	0	0	517.1	517.1	517.1	0	0
R209	0	749.5	749.5	749.5	0	3	0	749.5	749.5	749.5	0	1	0	517.1	517.1	517.1	0	1	1	517.1	517.1	517.1	0	0
R210	0	752.8	752.8	752.8	0	0	0	752.8	752.8	752.8	0	0	0	522.9	522.9	522.9	0	4	0	522.9	522.1	522.9	2	5
R211	0	749.5	749.5	749.5	0	4	0	749.5	749.5	749.5	0	1	1	517.1	517.1	517.1	0	2	0	517.1	517.1	517.1	0	0
C201	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	593.7	593.7	593.7	0	6	0	593.7	593.7	593.7	0	4
C202	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	2	0	588.9	588.9	588.9	0	2
C203	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	2	0	588.9	588.9	588.9	0	2
C204	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	2	0	588.9	588.9	588.9	0	2
C205	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	9	0	588.9	588.9	588.9	0	2
C206	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	12	0	588.9	588.9	588.9	0	2
C207	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	4	1	588.9	588.9	588.9	0	2
C208	0	791.1	791.1	791.1	0	1	0	802.9	802.9	802.9	0	1	0	588.9	588.9	588.9	0	5	0	588.9	588.9	588.9	0	2
RC201	0	1514.2	1514.2	1514.2	0	2	0	1645.0	1645.0	1645.0	0	2	4	943.3	914.5	943.3	24	47	2	943.3	914.5	943.3	22	44
RC202	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	8	934.4	908.5	934.4	42	73	2	934.4	908.5	934.4	28	33
RC203	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	18	934.4	908.5	934.4	94	152	3	934.4	908.5	934.4	52	61
RC204	1	1514.2	1513.2	1514.2	6	2	1	1645.0	1644.0	1645.0	4	2	18	934.4	908.5	934.4	94	141	4	934.4	908.5	934.4	58	64
RC205	0	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	15	937.4	908.5	937.4	82	128	3	937.4	908.5	937.4	50	51
RC206	0	1514.2	1514.2	1514.2	0	2	0	1645.0	1645.0	1645.0	0	2	7	938.8	908.5	938.8	32	59	1	938.8	908.5	938.8	20	38
RC207	0	1514.2	1514.2	1514.2	0	2	0	1645.0	1645.0	1645.0	0	2	17	934.4	908.5	934.4	88	104	4	934.4	908.5	934.4	60	65
RC208	1	1514.2	1513.2	1514.2	6	2	0	1645.0	1644.0	1645.0	4	2	20	934.4	908.5	934.4	94	145	4	934.4	908.5	934.4	58	67

Table G.14 Set $\mathcal{B}$ instances, high outsourcing cost, 50 customers

	$Q = 30$, VRPPO						$Q = 30$, VRPPC						$Q = 50$, VRPPO						$Q = 50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	0	1658.5	1658.5	1658.5	0	0	0	1838.4	1838.4	1838.4	0	0	1	1188.0	1187.6	1188.0	2	7	0	1191.8	1191.8	1191.8	0	8
R102	1	1648.2	1648.2	1648.2	0	5	0	1827.1	1827.1	1827.1	0	1	2	1111.6	1111.5	1111.6	2	7	5	1129.1	1121.3	1129.1	8	42
R103	0	1639.7	1639.7	1639.7	0	2	1	1820.3	1819.4	1820.3	2	5	4	1096.7	1096.6	1096.7	2	34	6	1105.4	1102.7	1105.4	6	58
R104	1	1622.0	1622.0	1622.0	0	0	0	1799.5	1799.5	1799.5	0	0	23	1074.9	1073.0	1074.9	4	46	8	1079.8	1076.5	1079.8	6	58
R105	0	1631.5	1631.5	1631.5	0	0	1	1807.2	1807.2	1807.2	0	0	12	1142.4	1136.6	1142.4	18	80	5	1144.3	1141.2	1144.3	12	51
R106	0	1625.0	1625.0	1625.0	0	0	0	1802.4	1802.4	1802.4	0	0	8	1097.7	1089.4	1097.7	8	54	10	1112.8	1102.7	1112.8	16	124
R107	1	1618.7	1618.7	1618.7	0	0	0	1799.5	1799.5	1799.5	0	0	86	1090.9	1081.9	1090.9	50	312	23	1098.3	1089.9	1098.3	34	182
R108	0	1618.7	1618.7	1618.7	0	0	1	1799.5	1799.5	1799.5	0	0	52	1072.5	1071.5	1072.5	8	73	23	1079.8	1076.3	1079.8	22	142
R109	1	1626.7	1626.7	1626.7	0	0	0	1802.4	1802.4	1802.4	0	0	1	1094.7	1094.7	1094.7	2	24	11	1104.2	1098.2	1104.2	18	111
R110	0	1620.4	1620.4	1620.4	0	1	0	1799.5	1799.5	1799.5	0	0	29	1078.3	1074.3	1078.3	14	93	36	1085.1	1076.8	1085.1	50	309
R111	0	1625.0	1625.0	1625.0	0	0	1	1802.4	1802.4	1802.4	0	0	6	1081.5	1079.3	1081.5	4	22	18	1096.4	1090.0	1096.4	26	140
R112	1	1618.7	1618.7	1618.7	0	0	0	1799.5	1799.5	1799.5	0	1	48	1072.5	1070.4	1072.5	12	93	26	1079.8	1073.2	1079.8	24	190
C101	371	1402.3	1391.1	1402.3	1472	1960	230	1432.0	1421.1	1432.0	1378	1423	93	1006.8	999.0	1006.8	142	571	34	1006.8	998.7	1006.8	96	374
C102	455	1402.3	1391.1	1402.3	1668	2099	291	1432.0	1420.9	1432.0	1618	1652	285	1006.6	998.0	1006.6	386	1539	122	1006.6	999.4	1006.6	300	1302
C103	552	1401.0	1390.5	1401.0	1834	2344	384	1430.8	1420.1	1430.8	1924	2218	822	1006.2	997.8	1006.2	634	3076	294	1006.2	997.7	1006.2	626	2448
C104	825	1401.0	1389.9	1401.0	2214	3121	544	1430.8	1420.3	1430.8	2544	3025	2663	1005.1	996.7	1005.1	990	3997	457	1005.1	996.4	1005.1	762	2907
C105	219	1401.7	1390.8	1401.7	910	1084	163	1431.5	1420.4	1431.5	988	1068	89	1006.8	998.0	1006.8	142	545	40	1006.8	997.8	1006.8	112	401
C106	282	1401.7	1390.8	1401.7	1192	1488	208	1431.5	1420.5	1431.5	1302	1411	93	1006.8	998.2	1006.8	172	643	39	1006.8	998.2	1006.8	120	411
C107	299	1401.7	1390.8	1401.7	1210	1508	227	1431.5	1420.3	1431.5	1330	1533	90	1006.8	997.9	1006.8	156	685	66	1006.8	998.2	1006.8	188	556
C108	295	1401.0	1390.6	1401.0	1148	1355	185	1430.8	1420.1	1430.8	1030	1083	231	1005.6	996.9	1005.6	360	1351	113	1005.6	996.9	1005.6	320	1070
C109	371	1401.0	1390.6	1401.0	1344	1670	390	1430.8	1419.9	1430.8	2096	2402	521	1005.6	997.3	1005.6	552	1918	162	1005.6	997.1	1005.6	368	1508
RC101	1	2727.8	2725.7	2727.8	4	3	3	2877.0	2871.7	2877.0	12	7	0	1715.1	1701.1	1715.1	2	25	1	1715.1	1701.4	1715.1	2	22
RC102	1	2727.8	2725.7	2727.8	2	5	3	2877.0	2871.4	2877.0	16	9	1	1706.3	1690.2	1706.3	2	21	1	1706.3	1690.3	1706.3	2	19
RC103	1	2725.8	2723.8	2725.8	2	6	3	2875.0	2869.7	2875.0	16	7	3	1705.3	1688.0	1705.3	4	29	2	1705.3	1687.8	1705.3	4	24
RC104	3	2725.8	2723.3	2725.8	8	7	4	2875.0	2869.2	2875.0	16	8	6	1704.2	1686.7	1704.2	4	37	3	1704.2	1687.0	1704.2	4	22
RC105	2	2725.8	2723.8	2725.8	4	4	2	2875.0	2869.7	2875.0	12	7	1	1706.0	1688.8	1706.0	2	20	1	1706.0	1689.0	1706.0	2	19
RC106	1	2725.8	2723.2	2725.8	6	6	3	2875.0	2869.2	2875.0	14	8	1	1707.4	1690.7	1707.4	2	23	2	1707.4	1690.4	1707.4	4	22
RC107	3	2725.8	2723.3	2725.8	6	5	4	2875.0	2869.2	2875.0	20	6	1	1706.0	1688.8	1706.0	2	23	2	1706.0	1688.9	1706.0	4	27
RC108	2	2725.8	2723.3	2725.8	6	5	5	2875.0	2869.2	2875.0	20	8	2	1704.2	1687.2	1704.2	2	23	2	1704.2	1687.4	1704.2	2	19
R201	1	1631.5	1631.5	1631.5	0	0	0	1807.2	1807.2	1807.2	0	0	2	1120.8	1118.9	1120.8	2	36	6	1129.5	1126.6	1129.5	12	45
R202	0	1625.0	1625.0	1625.0	0	0	1	1802.4	1802.4	1802.4	0	0	11	1094.9	1090.5	1094.9	6	67	21	1108.7	1098.6	1108.7	28	238
R203	1	1618.7	1618.7	1618.7	0	0	0	1799.5	1799.5	1799.5	0	0	16	1077.9	1077.1	1077.9	4	60	28	1089.7	1084.2	1089.7	36	197
R204	0	1618.7	1618.7	1618.7	0	0	0	1799.5	1799.5	1799.5	0	0	122	1072.5	1070.8	1072.5	12	84	47	1079.8	1073.7	1079.8	42	274
R205	1	1626.7	1626.7	1626.7	0	0	1	1802.4	1802.4	1802.4	0	0	2	1093.2	1093.2	1093.2	0	33	13	1106.8	1102.0	1106.8	18	134
R206	0	1625.0	1625.0	1625.0	0	0	0	1802.4	1802.4	1802.4	0	0	20	1087.0	1083.4	1087.0	6	64	31	1099.8	1091.8	1099.8	46	260
R207	1	1618.7	1618.7	1618.7	0	0	1	1799.5	1799.5	1799.5	0	1	32	1077.9	1076.7	1077.9	6	64	37	1089.7	1082.9	1089.7	42	259
R208	0	1618.7	1618.7	1618.7	0	0	0	1799.5	1799.5	1799.5	0	0	127	1072.5	1071.2	1072.5	12	74	21	1079.8	1073.9	1079.8	18	132
R209	1	1623.8	1623.8	1623.8	0	0	1	1799.5	1799.5	1799.5	0	0	25	1074.1	1072.3	1074.1	10	74	21	1084.4	1080.2	1084.4	26	141
R210	0	1625.0	1625.0	1625.0	0	0	0	1802.4	1802.4	1802.4	0	0	16	1088.1	1085.8	1088.1	4	55	17	1099.5	1094.5	1099.5	22	175
R211	1	1618.7	1618.7	1618.7	0	0	1	1799.5	1799.5	1799.5	0	0	113	1072.5	1071.0	1072.5	8	89	23	1079.8	1073.5	1079.8	22	155
C201	1	1526.5	1524.8	1526.5	8	15	1	1538.3	1536.6	1538.3	8	13	22	1153.9	1150.3	1153.9	30	131	6	1153.9	1150.6	1153.9	12	58
C202	2	1526.5	1523.7	1526.5	10	22	3	1538.3	1536.1	1538.3	12	15	2	1148.8	1148.8	1148.8	0	28	1	1148.8	1148.8	1148.8	0	23
C203	3	1526.5	1524.3	1526.5	12	16	3	1538.3	1536.1	1538.3	12	16	12	1147.3	1146.3	1147.3	8	35	7	1147.3	1146.3	1147.3	8	43
C204	3	1526.5	1523.9	1526.5	12	17	3	1538.3	1536.1	1538.3	12	17	23	1147.3	1146.2	1147.3	6	44	11	1147.3	1145.7	1147.3	14	48
C205	4	1526.5	1523.8	1526.5	24	41	2	1538.3	1536.1	1538.3	14	18	6	1148.2	1147.6	1148.2	6	52	3	1148.2	1146.5	1148.2	6	29
C206	5	1526.5	1523.8	1526.5	24	38	3	1538.3	1535.6	1538.3	14	21	3	1147.3	1146.9	1147.3	2	40	1	1147.3	1146.3	1147.3	2	25
C207	6	1526.5	1523.8	1526.5	24	41	2	1538.3	1535.7	1538.3	14	18	9	1147.3	1146.2	1147.3	6	38	7	1147.3	1145.6	1147.3	10	60
C208	5	1526.5	1523.8	1526.5	24	42	3	1538.3	1535.7	1538.3	16	19	7	1147.3	1145.9	1147.3	6	44	5	1147.3	1146.1	1147.3	6	53
RC201	1	2725.8	2723.8	2725.8	4	3	2	2875.0	2869.7	2875.0	12	7	1	1715.1	1701.2	1715.1	2	28	2	1715.1	1700.5	1715.1	2	27
RC202	1	2725.8	2723.8	2725.8	2	6	4	2875.0	2869.7	2875.0	20	8	1	1706.3	1689.7	1706.3	2	18	2	1706.3	1690.0	1706.3	2	23
RC203	3	2725.8	2723.8	2725.8	8	7	5	2875.0	2869.4	2875.0	20	11	6	1705.3	1688.1	1705.3	4	29	2	1705.3	1688.0	1705.3	4	27
RC204	3	2725.8	2723.3	2725.8	8	7	4	2875.0	2869.2	2875.0	16	8	8	1704.2	1687.1	1704.2	4	33	3	1704.2	1686.7	1704.2	4	27
RC205	2	2725.8	2723.8	2725.8	4	3	2	2875.0	2869.7	2875.0	12	7	1	1706.2	1689.4	1706.2	2	24	2	1706.2	1689.2	1706.2	4	24
RC206	2	2725.8	2723.3	2725.8	6	3	3	2875.0	2869.1	2875.0	14	10	1	1707.4	1690.5	1707.4	2	23	2	1707.4	1690.9	1707.4	2	19
RC207	3	2725.8	2723.3	2725.8	6	4	4	2875.0	2869.1	2875.0	20	10	2	1706.0	1689.4	1706.0	2	22	1	1706.0	1689.1	1706.0	2	22
RC208	2	2725.8	2723.3	2725.8	6	5	5	2875.0	2869.1	2875.0	20	10	10	1704.2	1686.8	1704.2	4	31	2	1704.2	1687.4	1704.2	2	12

Table G.15 Set $\mathcal{B}$ instances, high outsourcing cost, 100 customers

	$Q = 30$, VRPPO						$Q = 30$, VRPPC						$Q = 50$, VRPPO						$Q = 50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	10	4708.4	4707.7	4708.4	4	6	10	4782.1	4781.5	4782.1	2	2	96	2565.4	2561.1	2565.4	40	122	19	2636.4	2632.0	2636.4	12	45
R102	3	4679.1	4679.1	4679.1	0	2	3	4760.4	4760.4	4760.4	0	1	358	2511.7	2509.0	2511.7	18	186	185	2561.8	2555.5	2561.8	104	418
R103	5	4678.8	4678.8	4678.8	0	5	6	4757.6	4757.6	4757.6	0	3	-	7290.0	2485.3	2490.4	60	310	107	2526.5	2522.7	2526.5	22	146
R104	6	4678.8	4678.8	4678.8	0	2	6	4757.6	4757.6	4757.6	0	4	-	7290.0	2483.0	2484.9	8	175	325	2520.1	2516.2	2520.1	30	233
R105	0	4685.5	4685.5	4685.5	0	2	3	4765.9	4765.9	4765.9	0	4	462	2514.3	2507.2	2514.3	118	464	43	2552.5	2548.9	2552.5	18	99
R106	5	4679.1	4679.1	4679.1	0	4	3	4757.6	4757.6	4757.6	0	4	-	7290.0	2494.2	2498.8	96	465	201	2533.0	2526.8	2533.0	86	339
R107	6	4678.8	4678.8	4678.8	0	2	5	4757.6	4757.6	4757.6	0	3	-	7290.0	2485.1	2488.1	28	254	239	2525.2	2519.0	2525.2	56	400
R108	8	4678.8	4678.8	4678.8	0	1	4	4757.6	4757.6	4757.6	0	3	-	7290.0	2483.1	2484.4	10	163	604	2520.1	2515.8	2520.1	56	314
R109	15	4682.3	4681.7	4682.3	6	4	4	4758.7	4758.7	4758.7	0	3	1450	2497.9	2493.7	2497.9	68	422	77	2527.7	2524.1	2527.7	26	176
R110	7	4680.7	4680.7	4680.7	0	2	3	4758.7	4758.7	4758.7	0	3	-	7290.0	2486.6	2491.0	56	351	116	2521.0	2515.7	2521.0	30	212
R111	8	4679.1	4679.1	4679.1	0	3	7	4757.6	4757.6	4757.6	0	4	-	7290.0	2485.4	2489.3	32	287	432	2525.2	2518.0	2525.2	116	713
R112	11	4678.8	4678.8	4678.8	0	6	3	4757.6	4757.6	4757.6	0	3	-	7290.0	2481.0	2482.7	8	172	464	2518.1	2514.0	2518.1	38	220
C101	31	3134.6	3131.7	3134.6	26	57	78	3216.3	3211.2	3216.3	64	80	-	3620.0	2465.0	2478.2	2746	7141	-	2485.9	2464.1	2479.1	4190	12022
C102	71	3134.6	3131.7	3134.6	40	58	75	3216.3	3212.8	3216.3	56	76	-	3620.0	2464.0	2472.9	866	2230	-	3620.0	2464.4	2475.6	2864	6321
C103	105	3133.9	3129.7	3133.9	68	74	81	3215.6	3212.6	3215.6	74	75	-	3620.0	2463.3	2470.2	516	1867	-	3620.0	2463.4	2474.0	2274	4845
C104	74	3133.9	3130.9	3133.9	42	62	110	3215.6	3212.7	3215.6	76	78	-	3620.0	2461.2	2467.3	202	785	-	3620.0	2462.4	2470.6	1496	3639
C105	60	3134.6	3129.9	3134.6	46	66	56	3216.3	3211.1	3216.3	64	90	-	3620.0	2465.1	2477.6	1984	5180	-	2488.5	2464.9	2478.3	3448	9085
C106	50	3134.6	3129.8	3134.6	42	67	93	3216.3	3211.1	3216.3	80	99	-	3620.0	2465.6	2475.9	2008	5652	-	2484.0	2463.5	2477.5	3544	9582
C107	41	3134.6	3129.9	3134.6	54	79	77	3216.3	3212.3	3216.3	68	78	-	3620.0	2464.5	2477.4	2252	7166	-	2484.6	2465.2	2478.9	3476	9609
C108	37	3133.9	3130.6	3133.9	34	58	58	3215.6	3210.6	3215.6	64	95	-	3620.0	2462.1	2472.3	1508	3984	-	3620.0	2461.9	2473.5	3410	9968
C109	57	3133.9	3131.3	3133.9	44	65	83	3215.6	3212.7	3215.6	66	70	-	3620.0	2461.2	2469.9	966	3326	-	3620.0	2460.8	2472.5	2966	7215
RC101	5	4833.4	4833.4	4833.4	0	0	2	4857.0	4857.0	4857.0	0	0	16	3415.8	3413.6	3415.8	8	41	81	3446.2	3439.7	3446.2	48	87
RC102	17	4828.8	4826.7	4828.8	6	3	3	4857.0	4857.0	4857.0	0	0	49	3399.8	3397.3	3399.8	6	27	13	3418.3	3416.8	3418.3	4	25
RC103	17	4828.3	4826.7	4828.3	6	2	3	4857.0	4857.0	4857.0	0	0	960	3390.0	3382.7	3390.0	52	183	8	3397.4	3397.4	3397.4	0	10
RC104	27	4827.3	4826.7	4827.3	8	3	3	4853.7	4853.7	4853.7	0	0	1658	3386.8	3381.8	3386.8	46	139	17	3397.4	3396.7	3397.4	2	12
RC105	11	4830.1	4828.9	4830.1	4	3	3	4857.0	4857.0	4857.0	0	0	174	3413.1	3409.2	3413.1	38	71	14	3425.9	3424.1	3425.9	4	9
RC106	1	4823.3	4823.3	4823.3	0	3	2	4849.7	4849.7	4849.7	0	0	151	3398.8	3394.0	3398.8	34	107	57	3420.3	3415.2	3420.3	20	54
RC107	7	4823.3	4822.9	4823.3	2	3	3	4849.7	4849.7	4849.7	0	0	709	3389.5	3384.3	3389.5	62	147	13	3400.1	3399.1	3400.1	4	5
RC108	9	4823.3	4822.9	4823.3	2	2	3	4849.7	4849.7	4849.7	0	0	1452	3386.8	3381.9	3386.8	46	171	27	3397.4	3395.9	3397.4	6	23
R201	8	4685.5	4684.6	4685.5	2	1	7	4765.9	4765.3	4765.9	2	5	1376	2508.5	2505.0	2508.5	50	290	40	2538.3	2536.2	2538.3	12	107
R202	7	4679.1	4679.1	4679.1	0	2	2	4757.6	4757.6	4757.6	0	3	-	7290.0	2491.1	2494.0	34	275	262	2525.9	2521.9	2525.9	54	378
R203	7	4678.8	4678.8	4678.8	0	3	3	4757.6	4757.6	4757.6	0	3	-	7290.0	2484.3	2485.7	10	198	598	2522.1	2517.8	2522.1	44	317
R204	10	4678.8	4678.8	4678.8	0	1	4	4757.6	4757.6	4757.6	0	3	-	7290.0	2481.5	2483.4	8	176	1709	2518.1	2514.1	2518.1	32	156
R205	7	4680.9	4680.9	4680.9	0	2	6	4761.4	4761.4	4761.4	0	3	3331	2494.4	2491.2	2494.4	40	289	96	2527.0	2522.5	2527.0	18	138
R206	8	4679.1	4679.1	4679.1	0	1	6	4757.6	4757.6	4757.6	0	2	-	7290.0	2485.7	2488.7	12	210	439	2521.7	2516.9	2521.7	50	246
R207	10	4678.8	4678.8	4678.8	0	2	5	4757.6	4757.6	4757.6	0	3	-	7290.0	2483.8	2485.2	8	182	1448	2521.7	2516.5	2521.7	62	312
R208	16	4678.8	4678.8	4678.8	0	4	5	4757.6	4757.6	4757.6	0	3	-	7290.0	2480.8	2483.7	8	199	-	7290.0	2514.0	2517.9	38	205
R209	7	4680.7	4680.7	4680.7	0	2	7	4761.4	4761.4	4761.4	0	4	-	7290.0	2487.5	2489.6	14	186	158	2521.0	2518.0	2521.0	12	120
R210	8	4679.1	4679.1	4679.1	0	3	2	4757.6	4757.6	4757.6	0	3	-	7290.0	2485.6	2486.8	8	183	327	2522.1	2517.7	2522.1	40	267
R211	18	4678.8	4678.8	4678.8	0	5	8	4757.6	4757.6	4757.6	0	4	-	7290.0	2481.6	2481.6	2	144	-	2518.1	2514.2	2517.9	52	274
C201	3	3234.6	3234.6	3234.6	0	6	3	3341.9	3341.9	3341.9	0	4	84	2576.4	2573.6	2576.4	24	104	32	2576.4	2573.7	2576.4	18	98
C202	7	3234.6	3234.4	3234.6	2	11	3	3341.9	3341.9	3341.9	0	4	281	2575.5	2572.5	2575.5	40	171	128	2575.5	2572.1	2575.5	46	167
C203	4	3234.6	3234.6	3234.6	0	4	4	3341.9	3341.9	3341.9	0	4	451	2574.4	2571.3	2574.4	40	201	116	2574.4	2571.3	2574.4	40	173
C204	3	3234.6	3234.6	3234.6	0	4	4	3341.9	3341.9	3341.9	0	5	1283	2572.6	2568.9	2572.6	82	394	211	2572.6	2569.6	2572.6	60	182
C205	7	3234.6	3234.6	3234.6	0	7	5	3341.9	3341.9	3341.9	0	4	190	2573.9	2570.3	2573.9	58	270	125	2573.9	2570.1	2573.9	74	298
C206	5	3234.6	3234.6	3234.6	0	7	5	3341.9	3341.9	3341.9	0	3	405	2573.8	2570.4	2573.8	94	385	126	2573.8	2570.0	2573.8	74	287
C207	3	3234.6	3234.6	3234.6	0	9	5	3341.9	3341.9	3341.9	0	4	348	2572.6	2569.5	2572.6	60	242	130	2572.6	2569.4	2572.6	70	277
C208	3	3234.6	3234.6	3234.6	0	9	5	3341.9	3341.9	3341.9	0	4	415	2572.6	2569.6	2572.6	74	301	117	2572.6	2569.4	2572.6	60	222
RC201	5	4827.3	4827.3	4827.3	0	0	2	4849.7	4849.7	4849.7	0	0	52	3411.9	3409.9	3411.9	8	53	18	3438.1	3435.0	3438.1	8	38
RC202	8	4823.3	4822.9	4823.3	2	2	3	4849.7	4849.7	4849.7	0	0	254	3399.4	3397.0	3399.4	10	64	19	3418.3	3415.6	3418.3	6	30
RC203	7	4823.3	4822.9	4823.3	2	3	3	4849.7	4849.7	4849.7	0	0	1998	3390.0	3383.5	3390.0	52	188	13	3397.4	3397.4	3397.4	0	6
RC204	11	4823.3	4822.9	4823.3	8	2	3	4849.7	4849.7	4849.7	0	0	1620	3386.8	3380.5	3386.8	32	106	30	3397.4	3396.5	3397.4	4	17
RC205	16	4828.8	4826.7	4828.8	0	3	3	4857.0	4857.0	4857.0	0	0	413	3393.4	3390.6	3393.4	26	94	11	3406.4	3406.1	3406.4	2	17
RC206	2	4823.3	4823.3	4823.3	0	3	3	4849.7	4849.7	4849.7	0	0	517	3394.7	3391.0	3394.7	22	99	44	3412.3	3408.0	3412.3	14	51
RC207	11	4823.3	4822.9	4823.3	2	2	3	4849.7	4849.7	4849.7	0	0	866	3386.8	3382.5	3386.8	22	113	17	3397.4	3396.4	3397.4	4	6
RC208	14	4823.3	4822.9	4823.3	2	2	3	4849.7	4849.7	4849.7	0	0	-	3388.1	3382.1	3383.4	26	112	33	3397.4	3396.5	3397.4	4	16

Table G.16 Set $\mathcal{B}$ instances, low outsourcing cost, 25 customers

	$Q = 30$, VRPPO						$Q = 30$, VRPPC						$Q = 50$, VRPPO						$Q = 50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	0	754.0	754.0	754.0	0	0	0	754.0	754.0	754.0	0	0	0	597.6	595.5	597.6	2	6	0	597.6	597.6	597.6	0	2
R102	0	748.8	748.8	748.8	0	1	0	754.0	754.0	754.0	0	0	0	549.2	549.2	549.2	0	0	0	549.2	549.2	549.2	0	0
R103	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	518.6	518.6	518.6	0	1	0	518.6	518.6	518.6	0	1
R104	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.8	512.8	0	1	1	513.9	513.9	513.9	0	3
R105	0	737.9	737.9	737.9	0	0	0	743.1	743.1	743.1	0	2	0	566.4	566.4	566.4	0	7	0	566.4	566.4	566.4	0	7
R106	0	737.9	737.9	737.9	0	0	1	740.4	740.1	740.4	2	1	0	518.6	518.6	518.6	0	0	0	518.6	518.6	518.6	0	0
R107	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	1	518.6	516.1	518.6	2	3	0	518.6	516.1	518.6	2	3
R108	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.1	512.8	2	3	1	513.9	512.1	513.9	4	3
R109	0	737.9	737.9	737.9	0	0	0	740.4	740.1	740.4	2	1	0	514.1	514.1	514.1	0	2	0	514.1	514.1	514.1	0	0
R110	0	733.0	733.0	733.0	0	0	0	738.2	738.2	738.2	0	3	0	513.9	513.5	513.9	2	6	0	513.9	513.5	513.9	2	4
R111	0	737.9	737.9	737.9	0	0	0	740.4	740.1	740.4	2	1	1	512.8	512.1	512.8	2	2	1	513.9	512.1	513.9	4	3
R112	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.8	512.8	0	4	0	513.9	512.1	513.9	2	3
C101	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C102	0	230	230	230	0	0	1	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C103	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C104	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C105	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C106	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C107	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C108	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C109	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
RC101	0	1080	1080	1080	0	0	1	1080	1080	1080	0	0	1	885.1	879.8	885.1	8	9	0	886.1	885.5	886.1	2	3
RC102	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	877.0	881.4	8	16	0	881.4	881.4	881.4	0	1
RC103	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	877.0	881.4	12	29	0	881.4	881.4	881.4	0	1
RC104	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	3	881.4	877.0	881.4	16	25	0	881.4	881.4	881.4	0	1
RC105	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	3	881.4	877.0	881.4	14	43	0	881.4	881.4	881.4	0	1
RC106	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	877.0	881.4	12	31	0	881.4	881.4	881.4	0	1
RC107	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	4	881.4	877.0	881.4	20	22	0	881.4	881.4	881.4	0	1
RC108	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	5	881.4	877.0	881.4	24	28	0	881.4	881.4	881.4	0	1
R201	0	737.9	737.9	737.9	0	0	0	740.4	740.1	740.4	2	1	1	550.4	549.8	550.4	2	14	0	550.4	550.4	550.4	0	10
R202	0	737.9	737.9	737.9	0	1	0	740.4	740.1	740.4	2	1	0	512.8	512.8	512.8	0	3	0	513.9	512.1	513.9	2	3
R203	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.1	512.8	2	2	1	513.9	513.5	513.9	2	4
R204	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	1	512.8	512.1	512.8	2	3	0	513.9	512.1	513.9	2	3
R205	0	737.9	737.9	737.9	0	0	1	740.4	740.1	740.4	2	1	0	514.1	514.1	514.1	0	5	0	514.1	514.1	514.1	0	0
R206	0	737.9	737.9	737.9	0	0	0	740.4	740.1	740.4	2	1	1	512.8	512.8	512.8	0	3	0	513.9	512.1	513.9	2	3
R207	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.8	512.8	0	2	0	513.9	513.5	513.9	2	3
R208	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.8	512.8	0	3	1	513.9	512.1	513.9	2	3
R209	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	514.1	514.1	514.1	0	3	0	514.1	514.1	514.1	0	0
R210	0	737.9	737.9	737.9	0	0	0	740.4	740.1	740.4	2	1	1	512.8	512.8	512.8	0	2	0	513.9	512.1	513.9	2	3
R211	0	733.0	733.0	733.0	0	0	0	738.2	737.7	738.2	2	3	0	512.8	512.8	512.8	0	4	0	513.9	512.1	513.9	2	3
C201	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	1	230	230	230	0	0
C202	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C203	1	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C204	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C205	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C206	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C207	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
C208	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0	0	230	230	230	0	0
RC201	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	3	885.1	879.4	885.1	18	19	1	886.1	885.5	886.1	2	3
RC202	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	877.0	881.4	8	17	0	881.4	881.4	881.4	0	1
RC203	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	3	881.4	877.0	881.4	16	26	0	881.4	881.4	881.4	0	1
RC204	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	4	881.4	877.0	881.4	18	28	0	881.4	881.4	881.4	0	1
RC205	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	877.0	881.4	14	33	0	881.4	881.4	881.4	0	1
RC206	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	2	881.4	876.3	881.4	12	31	0	881.4	881.4	881.4	0	1
RC207	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	4	881.4	877.0	881.4	20	23	0	881.4	881.4	881.4	0	1
RC208	0	1080	1080	1080	0	0	0	1080	1080	1080	0	0	6	881.4	877.0	881.4	24	27	0	881.4	881.4	881.4	0	1

Table G.17 Set $\mathcal{B}$ instances, low outsourcing cost, 50 customers

	$Q=30$, VRPPO						$Q=30$, VRPPC						$Q=50$, VRPPO						$Q=50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	0	1597.3	1597.3	1597.3	0	0	0	1688.3	1687.6	1688.3	2	0	0	1151.9	1151.9	1151.9	0	3	1	1166.3	1164.4	1166.3	2	14
R102	0	1592.5	1592.5	1592.5	0	1	1	1679.0	1679.0	1679.0	0	0	0	1104.2	1104.2	1104.2	0	9	16	1123.2	1115.9	1123.2	38	90
R103	1	1586.1	1586.1	1586.1	0	2	0	1672.6	1672.6	1672.6	0	0	8	1086.3	1085.4	1086.3	4	45	6	1099.9	1096.6	1099.9	10	64
R104	0	1572.9	1572.9	1572.9	0	2	0	1659.4	1659.4	1659.4	0	0	24	1068.1	1063.9	1068.1	8	38	9	1078.3	1071.9	1078.3	8	47
R105	0	1582.1	1582.1	1582.1	0	0	1	1674.0	1674.0	1674.0	0	0	2	1119.5	1118.4	1119.5	6	34	5	1133.9	1129.6	1133.9	12	47
R106	1	1577.4	1577.4	1577.4	0	1	0	1665.7	1665.7	1665.7	0	0	7	1088.3	1083.4	1088.3	8	45	18	1109.7	1099.7	1109.7	36	146
R107	0	1572.9	1572.9	1572.9	0	1	0	1659.4	1659.4	1659.4	0	0	25	1075.9	1072.9	1075.9	16	81	32	1095.8	1084.4	1095.8	60	254
R108	1	1572.9	1572.9	1572.9	0	2	0	1659.4	1659.4	1659.4	0	0	68	1064.5	1062.7	1064.5	14	68	26	1078.3	1071.6	1078.3	26	170
R109	0	1576.9	1576.9	1576.9	0	1	0	1664.9	1664.9	1664.9	0	0	0	1083.3	1083.3	1083.3	0	7	9	1098.2	1092.2	1098.2	16	77
R110	0	1572.9	1572.9	1572.9	0	2	1	1659.4	1659.4	1659.4	0	0	19	1067.8	1065.1	1067.8	10	83	36	1081.6	1072.5	1081.6	54	243
R111	2	1575.1	1573.9	1575.1	2	2	0	1664.9	1664.9	1664.9	0	0	14	1074.7	1071.8	1074.7	8	58	18	1091.9	1084.7	1091.9	26	135
R112	0	1572.9	1572.9	1572.9	0	2	0	1659.4	1659.4	1659.4	0	0	36	1064.5	1061.8	1064.5	12	76	38	1078.3	1069.7	1078.3	48	267
C101	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C102	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C103	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C104	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C105	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C106	0	430	430	430	0	0	0	430	430	430	0	0	1	430	430	430	0	0	0	430	430	430	0	0
C107	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	1	430	430	430	0	0
C108	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C109	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
RC101	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	17	1663.1	1656.4	1663.1	54	59	10	1663.1	1656.1	1663.1	46	48
RC102	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	33	1654.6	1647.7	1654.6	90	121	9	1654.6	1647.6	1654.6	34	36
RC103	0	1940	1940	1940	0	0	1	1940	1940	1940	0	0	51	1653.9	1645.8	1653.9	104	138	13	1653.9	1645.8	1653.9	44	48
RC104	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	87	1652.7	1645.1	1652.7	120	151	15	1652.7	1645.1	1652.7	38	44
RC105	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	30	1654.6	1647.1	1654.6	78	93	10	1654.6	1646.9	1654.6	36	37
RC106	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	50	1655.7	1648.0	1655.7	108	141	14	1655.7	1648.0	1655.7	48	62
RC107	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	66	1654.5	1646.7	1654.5	122	147	12	1654.5	1646.7	1654.5	40	44
RC108	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	81	1652.7	1645.1	1652.7	102	121	14	1652.7	1645.1	1652.7	38	43
R201	2	1582.1	1582.1	1582.1	2	1	0	1674.0	1674.0	1674.0	0	1	1	1108.7	1108.7	1108.7	0	30	8	1124.3	1117.8	1124.3	18	76
R202	0	1577.4	1577.4	1577.4	0	1	1	1665.7	1665.7	1665.7	0	0	16	1085.8	1082.6	1085.8	10	59	17	1105.2	1097.5	1105.2	30	154
R203	0	1572.9	1572.9	1572.9	0	2	0	1659.4	1659.4	1659.4	0	0	80	1071.9	1069.3	1071.9	30	139	39	1088.2	1079.5	1088.2	56	300
R204	1	1572.9	1572.9	1572.9	0	3	0	1659.4	1659.4	1659.4	0	0	89	1064.5	1062.5	1064.5	8	65	37	1078.3	1069.9	1078.3	40	241
R205	0	1576.9	1576.9	1576.9	0	1	1	1664.9	1664.9	1664.9	0	0	4	1083.4	1083.0	1083.4	2	30	11	1100.6	1094.9	1100.6	18	117
R206	1	1576.9	1576.9	1576.9	0	4	0	1664.9	1664.9	1664.9	0	0	19	1080.1	1077.0	1080.1	6	63	27	1096.5	1088.9	1096.5	46	209
R207	0	1572.9	1572.9	1572.9	0	3	0	1659.4	1659.4	1659.4	0	0	146	1071.9	1067.8	1071.9	34	188	42	1088.2	1077.7	1088.2	58	356
R208	1	1572.9	1572.9	1572.9	0	3	1	1659.4	1659.4	1659.4	0	0	113	1064.5	1061.6	1064.5	10	87	41	1078.3	1069.7	1078.3	42	308
R209	0	1572.9	1572.9	1572.9	0	3	0	1659.4	1659.4	1659.4	0	0	45	1068.1	1065.5	1068.1	22	103	23	1082.9	1077.7	1082.9	32	167
R210	2	1575.1	1573.9	1575.1	2	2	0	1664.9	1664.9	1664.9	0	0	6	1078.0	1076.4	1078.0	2	31	20	1096.5	1088.8	1096.5	30	225
R211	0	1572.9	1572.9	1572.9	0	2	1	1659.4	1659.4	1659.4	0	0	151	1064.5	1061.6	1064.5	14	90	44	1078.3	1069.9	1078.3	50	240
C201	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C202	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C203	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C204	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	1	430	430	430	0	0
C205	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C206	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C207	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
C208	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0	0	430	430	430	0	0
RC201	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	19	1663.1	1656.1	1663.1	56	76	7	1663.1	1656.4	1663.1	28	39
RC202	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	38	1654.6	1647.7	1654.6	86	145	12	1654.6	1647.6	1654.6	40	38
RC203	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	69	1653.9	1646.3	1653.9	110	153	13	1653.9	1646.1	1653.9	40	42
RC204	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	96	1652.7	1645.1	1652.7	110	153	13	1652.7	1644.6	1652.7	38	45
RC205	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	39	1654.6	1647.1	1654.6	84	106	12	1654.6	1647.1	1654.6	36	34
RC206	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	49	1655.7	1648.1	1655.7	102	152	15	1655.7	1648.1	1655.7	46	60
RC207	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	65	1654.5	1647.0	1654.5	98	144	13	1654.5	1646.7	1654.5	36	42
RC208	0	1940	1940	1940	0	0	0	1940	1940	1940	0	0	137	1652.7	1644.8	1652.7	106	142	16	1652.7	1645.1	1652.7	38	43

Table G.18 Set $\mathcal{B}$ instances, low outsourcing cost, 100 customers

	$Q = 30$, VRPPO						$Q = 30$, VRPPC						$Q = 50$, VRPPO						$Q = 50$, VRPPC					
Inst.	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR	Time	UB	root LB	best LB	Tree	#SR
R101	10	3619.7	3617.2	3619.7	4	4	16	3683.1	3682.7	3683.1	6	2	57	2249.4	2245.4	2249.4	24	66	25	2300.5	2297.1	2300.5	16	56
R102	2	3600.6	3600.6	3600.6	0	5	6	3669.8	3669.1	3669.8	2	4	568	2198.2	2195.4	2198.2	44	233	228	2234.7	2228.1	2234.7	156	676
R103	4	3600.3	3600.3	3600.3	0	3	5	3667.0	3666.3	3667.0	2	4	-	5103	2172.7	2177.2	58	364	227	2208.6	2202.4	2208.6	82	396
R104	5	3599.6	3599.6	3599.6	0	2	12	3667.0	3666.3	3667.0	4	4	-	5103	2169.9	2172.3	16	209	227	2200.4	2195.5	2200.4	36	252
R105	2	3605.5	3605.5	3605.5	0	1	6	3673.1	3673.1	3673.1	0	1	316	2195.6	2188.9	2195.6	104	428	47	2230.9	2226.6	2230.9	32	148
R106	3	3599.8	3599.8	3599.8	0	2	9	3667.0	3666.3	3667.0	4	6	2107	2183.4	2178.9	2183.4	82	521	65	2209.0	2205.2	2209.0	24	198
R107	5	3599.6	3599.6	3599.6	0	1	9	3667.0	3666.3	3667.0	4	5	-	5103	2171.3	2174.6	46	349	438	2206.1	2199.4	2206.1	156	672
R108	5	3599.6	3599.6	3599.6	0	1	12	3667.0	3666.3	3667.0	4	4	-	5103	2169.1	2171.9	18	206	483	2200.4	2194.9	2200.4	76	402
R109	10	3602.6	3602.5	3602.6	4	2	12	3668.1	3667.3	3668.1	4	5	847	2181.3	2175.7	2181.3	60	373	101	2208.1	2203.2	2208.1	48	219
R110	4	3601.4	3601.4	3601.4	0	2	13	3668.1	3667.3	3668.1	4	5	-	5103	2172.3	2177.2	76	444	419	2202.9	2196.2	2202.9	172	793
R111	4	3599.8	3599.8	3599.8	0	1	13	3667.0	3666.3	3667.0	4	4	-	5103	2171.2	2175.1	50	410	530	2206.1	2197.6	2206.1	198	991
R112	8	3599.6	3599.6	3599.6	0	1	14	3667.0	3666.3	3667.0	4	4	-	5103	2167.9	2170.3	10	113	515	2198.9	2193.3	2198.9	82	365
C101	0	905	905	905	0	0	1	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C102	0	905	905	905	0	0	0	905	905	905	0	0	1	905	905	905	0	0	0	905	905	905	0	0
C103	1	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C104	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	1	905	905	905	0	0
C105	0	905	905	905	0	0	1	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C106	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C107	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C108	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
C109	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0	0	905	905	905	0	0
RC101	1	3213.1	3213.1	3213.1	0	0	2	3216.1	3215.8	3216.1	2	1	16	2691.9	2689.8	2691.9	8	16	16	2704.9	2703.9	2704.9	6	15
RC102	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	28	2676.6	2674.9	2676.6	8	26	7	2684.5	2683.9	2684.5	2	5
RC103	0	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	78	2666.4	2664.3	2666.4	6	28	17	2674.4	2674.0	2674.4	2	12
RC104	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	257	2666.3	2663.6	2666.3	8	58	20	2674.4	2673.1	2674.4	6	23
RC105	0	3215.4	3215.4	3215.4	0	1	2	3216.1	3215.8	3216.1	2	1	28	2690.0	2687.3	2690.0	10	60	4	2694.3	2694.3	2694.3	0	5
RC106	0	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	94	2677.6	2673.9	2677.6	30	69	23	2688.9	2687.1	2688.9	8	21
RC107	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	268	2669.3	2665.3	2669.3	40	66	20	2677.1	2676.2	2677.1	6	12
RC108	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	333	2666.3	2663.8	2666.3	16	73	15	2674.4	2673.6	2674.4	4	21
R201	6	3605.5	3605.5	3605.5	2	0	9	3673.1	3672.4	3673.1	2	2	606	2191.0	2187.6	2191.0	36	180	12	2217.1	2216.6	2217.1	2	51
R202	4	3599.8	3599.8	3599.8	0	2	11	3667.0	3666.3	3667.0	4	6	-	2181.2	2176.7	2180.6	62	412	115	2204.8	2200.6	2204.8	32	244
R203	5	3599.6	3599.6	3599.6	0	2	8	3667.0	3666.3	3667.0	4	5	-	5103	2170.1	2172.9	20	198	401	2202.0	2197.1	2202.0	66	430
R204	8	3599.6	3599.6	3599.6	0	3	13	3667.0	3666.3	3667.0	4	6	-	5103	2167.7	2169.9	12	192	931	2198.9	2193.2	2198.9	68	406
R205	5	3602.4	3602.4	3602.4	0	2	10	3670.8	3669.9	3670.8	2	2	1721	2180.9	2176.7	2180.9	32	254	56	2207.5	2203.2	2207.5	16	133
R206	4	3599.8	3599.8	3599.8	0	4	9	3667.0	3666.3	3667.0	4	6	-	5103	2172.7	2175.5	22	315	169	2200.4	2195.5	2200.4	30	230
R207	6	3599.6	3599.6	3599.6	0	1	19	3667.0	3666.0	3667.0	8	5	-	5103	2169.7	2171.7	14	164	419	2200.4	2196.1	2200.4	42	272
R208	8	3599.6	3599.6	3599.6	0	1	12	3667.0	3666.3	3667.0	4	6	-	5103	2168.3	2169.5	8	128	1340	2198.9	2193.2	2198.9	74	368
R209	7	3601.4	3601.4	3601.4	0	4	10	3670.8	3669.9	3670.8	2	2	-	5103	2172.6	2176.4	20	235	383	2203.0	2197.7	2203.0	86	427
R210	6	3599.8	3599.8	3599.8	0	2	12	3667.0	3666.3	3667.0	4	6	-	5103	2170.9	2173.3	16	207	133	2200.4	2195.9	2200.4	22	178
R211	10	3599.6	3599.6	3599.6	0	1	13	3667.0	3666.3	3667.0	4	4	-	5103	2168.4	2169.3	4	128	1199	2198.9	2193.7	2198.9	72	364
C201	0	905	905	905	0	0	0	905	905	905	0	0	0	901.5	901.5	901.5	0	0	0	901.5	901.5	901.5	0	0
C202	0	905	905	905	0	0	0	905	905	905	0	0	1	901.5	901.5	901.5	0	0	1	901.5	901.5	901.5	0	0
C203	0	905	905	905	0	0	0	905	905	905	0	0	0	901.5	901.5	901.5	0	0	0	901.5	901.5	901.5	0	0
C204	0	905	905	905	0	0	0	905	905	905	0	0	1	901.5	901.5	901.5	0	0	1	901.5	901.5	901.5	0	0
C205	0	905	905	905	0	0	0	905	905	905	0	0	0	901.5	901.5	901.5	0	0	1	901.5	901.5	901.5	0	0
C206	0	905	905	905	0	0	0	905	905	905	0	0	0	901.5	901.5	901.5	0	0	0	901.5	901.5	901.5	0	0
C207	0	905	905	905	0	0	0	905	905	905	0	0	0	901.5	901.5	901.5	0	0	0	901.5	901.5	901.5	0	0
C208	0	905	905	905	0	0	1	905	905	905	0	0	1	901.5	901.5	901.5	0	0	0	901.5	901.5	901.5	0	0
RC201	1	3213.1	3213.1	3213.1	0	0	3	3216.1	3215.8	3216.1	2	1	56	2690.6	2685.1	2690.6	16	57	4	2697.8	2697.8	2697.8	0	4
RC202	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	143	2676.0	2672.9	2676.0	14	73	10	2681.4	2681.0	2681.4	2	7
RC203	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	51	2666.4	2663.9	2666.4	2	21	16	2674.4	2674.0	2674.4	2	11
RC204	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	314	2666.3	2663.9	2666.3	4	49	16	2674.4	2672.7	2674.4	4	19
RC205	0	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	98	2671.1	2668.2	2671.1	14	60	6	2678.6	2678.6	2678.6	0	8
RC206	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	38	2674.9	2673.2	2674.9	2	31	8	2685.3	2685.3	2685.3	0	16
RC207	2	3213.1	3213.1	3213.1	0	1	3	3216.1	3215.8	3216.1	2	1	62	2666.3	2664.0	2666.3	2	24	14	2674.4	2673.5	2674.4	4	7
RC208	1	3213.1	3213.1	3213.1	0	1	2	3216.1	3215.8	3216.1	2	1	529	2666.3	2663.7	2666.3	8	41	21	2674.4	2672.7	2674.4	6	21

Summary

How to distribute goods is a question that is faced daily by many companies and, therefore, these questions are solved regularly in practice either with or without supporting technologies. A general aim is to keep costs low and customer service high, for example, by minimizing delivery costs and making sure that customer demand is satisfied. Increasing resource utilization and transportation efficiency can lead to cost savings and service improvements for all parties involved (e.g., manufacturers, transportation companies and customers). Efficiency can, for example, be enhanced by finding improved distribution plans, i.e., plans with lower costs, or by exploring new distribution strategies. This dissertation focuses on gaining insight in fundamental distribution problems, developing efficient distribution strategies and analyzing the benefit of novel distribution strategies.

Chapter 2 investigates which aspects influence the computational complexity of the Inventory Routing Problem (IRP) by looking for complexity proofs for several variants of the problem. The IRP combines the optimization of inventory management and routing of the vehicles that perform the replenishments for a set of customers over a given time horizon. Understanding the computational complexity of problems helps to reveal the structure of a problem which contributes to the development of solution methods. The Travelling Salesman Problem (TSP) is a special case of the IRP and since the TSP is NP-hard, it can immediately be concluded that the IRP is NP-hard. However, the underlying routing problem is not necessarily the only complicating aspect in the IRP. Therefore, Chapter 2 studies the IRP on metrics on which the TSP is easy or even trivial, hence NP-hardness through the TSP is avoided. First, problem variants on a point and on a half-line are studied. The problems differ in the number of vehicles, the number of days in the planning horizon and the service times of the customers. The main result is a polynomial time dynamic programming algorithm for the variant on the half-line with uniform service times and a planning horizon of two days. Second, for nearly any problem in the class with non-fixed planning horizon, we show that the complexity is dictated by the complexity of the Pinwheel Scheduling Problem, of which the complexity is a long-standing open research question. Third, NP-hardness is shown for problem variants with non-uniform servicing times. Concluding, the analysis shows that, next to routing, also the time horizon, service times, the customer demand combined with vehicle capacity, and the number of available vehicles contribute to the complexity of the IRP. Finally, we prove strong NP-hardness of a Euclidean variant with uniform service times and an easily computable routing cost approximation, avoiding immediate NP-hardness via the TSP.

In Chapter 3 a vendor-managed inventory setting is considered in which a supplier determines the timing and size of replenishments for its customers and the deliveries are

outsourced to an external carrier. In practical settings, it is sometimes the case that the supplier pays the transportation company a fixed fee per performed delivery. Geldmaat, the business partner of this Ph.D. project is such a supplier. Geldmaat decides upon the replenishment of ATMs in the Netherlands and issues replenishment orders to Cash-in-Transit companies. In the terminology of the replenishment literature: Geldmaat acts as a supplier that outsources delivery of goods. Chapter 3 considers an outsourcing cost structure in which a fixed fee is incurred per customer replenishment and per day on which at least one replenishment takes place. The corresponding optimization problem faced by the supplier is a Dynamic-Demand Joint Replenishment Problem (DJRP). Because of the fixed fee cost structure, there is no incentive for the supplier to schedule replenishments for nearby customers in the same period. As a result, the carrier is forced to perform inefficient delivery routes which leads to higher transportation costs, which will result in higher fixed transportation fees for the supplier in future contracts. Moreover, if the carrier has a limited fleet, it can occur that not all customers can be served on the same day due to longer travel times between distant locations. To assess the efficiency improvement if customer locations would be considered by the supplier, in Chapter 3 the DJRP is extended to the DJRP with Approximated Transportation Costs (DJRP-AT). Finding actual transportation costs requires to solve a routing problem. However, such problems are relatively hard to solve and for the purpose of this chapter, it is not necessary to know the sequence of the customers in a route as the deliveries are outsourced to an external carrier. Therefore, in Chapter 3 the transportation costs to service a given set of customers is approximated. A solution approach for the DJRP-AT based on branch-and-cut-and-price is validated using test instances from the literature. The distribution plans and costs found by solving the DJRP and the DJRP-AT are compared. The results show that significant cost savings can be achieved by deviating from the DJRP cost structure by considering the proximity of customers. Results show improvements of 4% on average and up to 14.4% for individual instances compared with the DJRP.

Chapter 4 addresses an approach for inventory replenishment in which customers can fulfill (part of) the demand of a nearby customer. If the customers are located relatively close to each other, one has the opportunity to satisfy a part of the demand of a customer by the inventory stored at another nearby customer. This redirection option can be included in the optimization of the customer replenishments in the Inventory Routing Problem to lower total costs. This idea can for example be applied to ATMs in urban areas where an ATM-user, who wants to withdraw money, can be redirected to another ATM which is within walking distance. One ATM then satisfies part of the demand of another ATM in close proximity. To the best of our knowledge, the possibility of redirecting end-users (e.g., ATM-users) is new to the Operations Research literature and has not been implemented, but is being considered in the industry. In Chapter 4 the so-called Inventory Routing Problem with Demand Moves (IRPDM) is defined and formulated in which *demand moves* represent the redirection of end-users between customers. For each demand move a service fee/cost is incurred which depends on the distance between the involved customers and quantity moved. A branch-price-and-cut solution approach is proposed to solve the IRPDM. The results show that substantial cost improvements can be achieved compared with the IRP. The results also indicate that only a limited number of demand moves per day is applied in the solutions. For instances from the literature cost improvements of the IRPDM over the IRP of up to

10% are observed with average savings around 3%.

In Chapter 5 the Vehicle Routing Problem with Partial Outsourcing (VRPPO) is introduced. In the Vehicle Routing Problem a set of customers that each have a certain demand need to be served in one day by a set of capacitated vehicles. In the VRPPO, a customer can either be served by a single private vehicle (a vehicle owned by the supplier), by a common carrier (the delivery is fully outsourced), or by both a single private vehicle and a common carrier. As such, it is a variant of the Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC) in which the delivery to a customer can either be fully outsourced or not. We propose two different path-based formulations for the VRPPO and solve these with a branch-and-price-and-cut solution method. For each path-based formulation, two different pricing procedures are designed. To assess the quality of the solution methods and gain insight in potential cost improvements compared with the VRPPC, tests are performed on two instance sets with up to 100 customers from the literature. Analyzing the cost difference between the VRPPO and the VRPPC shows higher cost improvements of the VRPPO over the VRPPC if customer demand is close to or higher than the vehicle capacity. Furthermore, if customers are located in clusters, cost improvements are lower than if customers are randomly spread over an area. Visualization of some solutions shows that a VRPPO solution can contain completely different routes than the corresponding VRPPC solution.

This dissertation studies the computational complexity of a class of distribution problems, models both fundamental and more practical distribution problems, and develops exact solution methods for such problems. The problems are inspired by real-life optimization problems from a cash supply chain, but are more widely applicable. The studies provide insight in problem structures and solution aspects, and contribute to the development of alternative solution methods.

Samenvatting

Veel bedrijven worden dagelijks geconfronteerd met de vraag hoe zij het beste hun goederen kunnen distribueren. Daarom worden dit soort vraagstukken regelmatig in de praktijk opgelost, met of zonder ondersteunende technologieën. Een algemeen doel is om kosten laag te houden en de klanttevredenheid hoog, bijvoorbeeld door de leveringskosten te minimaliseren terwijl aan alle klantvraag wordt voldaan. Betere benutting van middelen, zoals vrachtwagens, en efficiënter transport kan leiden tot lagere kosten en verbeterde service voor alle betrokken partijen (bv., fabrikanten, transportbedrijven en klanten). Efficiëntie kan worden vergroot door bijvoorbeeld betere distributieschema's te vinden, dat wil zeggen, schema's met lagere kosten, of door nieuwe distributiestrategieën te verkennen. Dit proefschrift beoogt om inzicht te bieden in complexe distributie problemen, om efficiëntere distributieschema's te ontwikkelen en om het potentiële voordeel van vernieuwende distributiestrategieën te analyseren.

De 'Inventory Routing Problem' (IRP) is een optimalisatievraagstuk waarbij bepaald moet worden welke klanten wanneer te bevoorraden en welke hoeveelheid goederen dan te leveren terwijl de kosten worden geminimaliseerd en in alle klantvraag kan worden voorzien. Hoofdstuk 2 onderzoekt welke aspecten de computationele complexiteit van de IRP beïnvloeden door naar complexiteitsbewijzen te zoeken voor verschillende varianten van het probleem. Het begrijpen van de computationele complexiteit van problemen helpt om probleemstructuren te ontdekken wat bijdraagt aan de ontwikkeling van oplossingsmethodes. De 'Travelling Salesman Problem' (TSP) is een speciaal geval van de IRP (vind een route met één vrachtwagen die alle klanten in één route bedient, i.e., het routeringsprobleem). Aangezien de TSP NP-hard is (computationeel moeilijk), kan onmiddellijk geconcludeerd worden dat de IRP ook NP-hard is. Echter, het onderliggende routeringsprobleem is niet noodzakelijk het enige complicerende aspect. Daarom bestudeert hoofdstuk 2 de IRP op metrieken (onderliggende structuren) waarop de TSP gemakkelijk of zelfs triviaal is, waardoor de IRP op deze metrieken niet noodzakelijk NP-hard is. In het hoofdstuk worden ten eerste probleemvarianten op een punt en op een half-lijn bestudeerd. De problemen verschillen in het aantal vrachtwagens, het aantal dagen in de planningshorizon en de servicetijden bij de klanten. Het belangrijkste resultaat is een dynamisch programmeer algoritme in polynomiale tijd voor de probleemvariant op de half-lijn met uniforme (gelijke) servicetijden en een planningshorizon van twee dagen. Ten tweede, voor vrijwel elk probleem in de klasse met niet-vastgestelde planningshorizon, laten we zien dat de computationele complexiteit bepaald wordt door de complexiteit van de 'Pinwheel Scheduling Problem' waarvan de complexiteit een tot op heden in de literatuur onbeantwoorde onderzoeksvraag is. Ten derde wordt aangetoond dat probleemvarianten met niet-uniforme servicetijden NP-hard zijn. Concluderend, de analyse toont aan dat, naast het routeringsprobleem,

ook de planningshorizon, de servicetijden, de klantvraag in combinatie met de capaciteit van de vrachtwagens en het aantal beschikbare vrachtwagens bijdragen aan de complexiteit van de IRP. Ten slotte bewijzen we dat een probleemvariant in de Euclidische ruimte met uniforme servicetijden *strong NP-hard* is, waarbij de TSP als onderliggend probleem wordt vermeden door een gemakkelijk te berekenen benadering van de transportkosten.

In hoofdstuk 3 wordt een 'vendor-managed inventory' setting bestudeerd waarbij een leverancier de timing en grootte bepaald van een klantlevering en de leveringen worden uitbesteed aan een externe transporteur. In de praktijk wordt soms door de leverancier een vast bedrag per levering betaald aan de transporteur. Geldmaat, de partner van dit Ph.D. project is zo'n 'leverancier'. Geldmaat bepaalt wanneer geldautomaten in Nederland bevoorraad worden en geeft orders aan een waarde transporteur om de bevoorrading uit te voeren. In de terminologie van de literatuur op het gebied van voorraadbeheer: Geldmaat is de leverancier die de leveringen aan zijn klanten (geldautomaten) uitbesteed. Hoofdstuk 3 beschouwt een uitbestedingskostenstructuur waarin een vast tarief wordt betaald per bevoorrading en per dag waarop ten minste één bevoorrading plaatsvindt. Het overeenkomstige optimalisatieprobleem van de leverancier is de 'Dynamic-Demand Joint Replenishment Problem' (DJRP). Door de kostenstructuur met vaste tarieven is er voor de leverancier geen stimulans om klanten die dicht bij elkaar zijn op dezelfde dag te bevoorraden. Hierdoor wordt de transporteur gedwongen om inefficiënte bezorgroutes uit te voeren wat leidt tot hogere transportkosten, wat zal resulteren in hogere vaste tarieven voor de leverancier welke worden overeengekomen in toekomstige contractonderhandelingen. Als de transporteur bovendien een beperkt aantal vrachtwagens heeft, kan het bovendien voorkomen dat niet alle klantorders kunnen worden uitgevoerd door de langere reistijden tussen klanten. Om de verbetering in efficiëntie te evalueren als de leverancier de locaties van klanten wel overweegt, wordt in hoofdstuk 3 de DJRP uitgebreid naar de 'DJRP with Approximated Transportation Costs' (DJRP-AT). Het berekenen van werkelijke transportkosten vereist de oplossing van een routeringsprobleem. Dergelijke problemen zijn relatief moeilijk om op te lossen en voor het doel van dit hoofdstuk is het niet noodzakelijk om de exacte volgorde van de klanten in een route te weten aangezien de leveringen worden uitbesteed. Daarom worden in hoofdstuk 3 de transportkosten benaderd. Een oplossingsmethode voor de DJRP-AT op basis van 'branch-and-cut-and-price' wordt gevalideerd met behulp van testdata uit de literatuur. De distributieschema's en kosten gevonden met de DJRP en de DJRP-AT worden vergeleken. De resultaten laten zien dat significante kostenbesparingen kunnen worden behaald als de leverancier transportkosten overweegt in zijn beslissingen. De resultaten tonen verder een kostenverbetering van gemiddeld 4% en tot 14,4%.

Hoofdstuk 4 beschouwt een aanpak voor een voorraadprobleem waarin klanten (een deel van) de vraag van nabij gelegen klanten kunnen vervullen. Als klanten zich relatief dicht bij elkaar bevinden, heeft men de mogelijkheid om in een deel van de vraag te voorzien door de voorraad van een andere, dichtbij gelegen klant aan te spreken. Deze optie kan worden opgenomen in de optimalisatie van de bevoorrading van alle klanten in de IRP om kosten te verlagen welke gebruikelijk bestaan uit opslag- en transportkosten. Dit idee kan bijvoorbeeld worden toegepast bij geldautomaten: een consument die geld wil opnemen kan doorgestuurd worden naar een andere geldautomaat die zich op loopafstand bevindt. Een geldautomaat voldoet dan een deel van de vraag van een andere geldautomaat die dichtbij gelegen is. Voor zover bij ons bekend, is de mogelijkheid om eindgebruikers door te sturen nieuw in de Operations Research literatuur en wordt dit wel overwogen maar nog niet toegepast in de praktijk. In hoofdstuk 4 wordt de zogenaamde 'Inventory Routing Problem with Demand Moves' (IRPDM) gedefinieerd en geformuleerd waarin de 'demand moves' (vraagbewegingen) het doorsturen van eindgebruikers tussen klanten representeren. Voor elke 'demand move' worden kosten gerekend die afhangen van de afstand tussen de betrokken klanten en de hoeveelheid goederen. Een 'branch-price-and-cut' oplossingsmethode wordt voorgesteld om de IRPDM op te lossen. De resultaten tonen dat substantiële kostenbesparingen kunnen worden bepaald in vergelijking met de IRP. De resultaten tonen verder dat een beperkt aantal 'demand moves' per dag nodig zijn om deze resultaten te behalen. Voor testdata uit de literatuur worden gemiddelde kostenverbeteringen van de IRPDM ten opzichte van de IRP gevonden tot 10% met gemiddeld 3% kostenbesparing.

Hoofdstuk 5 introduceert de 'Vehicle Routing Problem with Partial Outsourcing' (VRPPO), dat wil zeggen, een routeringsprobleem waarin gedeeltelijke uitbesteding van een levering mogelijk is. In de 'Vehicle Routing Problem' moet een aantal klanten die elk een bepaalde vraag naar goederen hebben in één dag bediend worden door vrachtwagens met een beperkte capaciteit. In de VRPPO kan een klant bediend worden door één private vrachtwagen (een vrachtwagen in het bezit van de leverancier), door een externe transporteur (een volledig uitbestede levering) of door zowel één private vrachtwagen als de externe transporteur. Hiermee is dit probleem een variant van de Vehicle Routing Problem with Private Fleet and Common Carrier (VRPPC) waarin een levering ofwel volledig, of helemaal niet uitbesteed kan worden. In hoofdstuk 5 worden twee verschillende formuleringen voor de VRPPO voorgesteld en deze worden opgelost met een 'branch-price-and-cut' oplossingsmethode. Voor elke probleemformulering zijn twee verschillende algoritmes om routes te genereren ontwikkeld. Om de prestaties van de oplossingsmethodes te beoordelen en inzicht te krijgen in de potentiële kostenbesparingen van de VRPPO in vergelijking met de VRPPC, worden experimenten uitgevoerd op twee verschillende datasets met maximaal 100 klanten uit de literatuur. Analyse van de kostenverschillen tussen de VRPPO en de VRPPC toont hogere kostenbesparingen van de VRPPO ten opzichte van de VRPPC als de vraag van de klanten dichtbij de capaciteit van de vrachtwagens ligt. Daarnaast, als de klanten in clusters zijn gevestigd zijn de kostenbesparingen lager dan wanneer de klanten willekeurig over een gebied zijn verspreid. Visualisatie van een aantal oplossingen toont dat oplossingen van de VRPPO compleet andere routes kan bevatten dan de overeenkomstige oplossing van de VRPPC.

Dit proefschrift bestudeert de computationele complexiteit van een klasse van distributieproblemen, modelleert zowel fundamentele als meer praktische distributieproblemen en ontwikkelt exacte oplossingsmethodes voor dit soort problemen. De problemen zijn geïnspireerd door optimalisatieproblemen die zich in de praktijk voordoen bij geldketens, maar zijn breder toepasbaar. De studies bieden inzicht in probleemstructuren en oplossingsaspecten en dragen bij aan de ontwikkeling van oplossingsmethodes.

ABRI DISSERTATION SERIES

1. Drees, J.M. (2013). The polycentricity of expansion strategies: Beyond performance as a main driver.
2. Arzlanian, S. (2014). Social networks and firm performance: Examining the relation between dimensions of social capital, social network perception and firm performance.
3. Fleisher, C. (2014). The contemporary career navigator: Individual and organizational outcomes of self-directed career management.
4. Wruck, S. (2014). Warehouse operations revisited – Novel challenges and methods.
5. Volk-Makarewicz, W. (2014). Advances in derivative estimation: Ranked data, quantiles, and options.
6. Van Anholt, R. (2014). Optimizing logistics processes in cash supply chains.
7. Polat, T. (2015). Active aging in work: Motivating employees to continue working after retirement.
8. Ossenkop, C. (2015). What you see is what you get!? Looking into ethnic diversity and professional careers in Dutch organizations.
9. Engel, Y. (2015). Venturing into the unknown, but not for the first time: An examination of firm-founders' careers & entrepreneurial decision-making under uncertainty.
10. Kolbe, L. (2015). The mindset of the R&D professional: Decision making in innovative context.
11. Pachidi, S. (2015). Crunching the numbers: Studying the enactment of analytics in an organization.
12. El Baroudi, S. (2015). Shading a fresh light on proactivity research: Examining when and how proactive behaviors benefit individuals and their employing organizations.
13. Eijdenberg, E.L. (2016). Small business growth in east African least developed countries: Unravelling the role of the small business owners.
14. Lysova, E.I. (2016). What does your career mean to you? Understanding individual career and work behaviors through the prism of the meaning of career.
15. De Mol, E. (2016). Heart and brain: The influence of affective and rational determinants in new venture teams: An empirical examination.
16. Daubner-Siva, D. (2016). Dealing with dualities: A paradox perspective on the relationship between talent management and diversity management.
17. Berkhout, J. (2016). Topics in Markov chain theory and simulation optimization.
18. Van-Werven, R. (2017). Acquiring resources for a new venture: A study of the micro-level linguistic practices of startup entrepreneurs.
19. Prats Lopez, M. (2017). Managing citizen science in the Humanities: The challenge of ensuring quality.
20. Kaandorp, M.S. (2017). Creating from within: A study on interpersonal networking approaches and intuition in task-related interaction.
21. Van Grinsven, M. (2017). A patient is not a car; Lean in healthcare: Studying agency in the translation of management concepts.
22. Muleta Eyana, S. (2017). Entrepreneurial behavior and firm performance of Ethiopian tour operators.
23. Van Ee, M. (2017). Routing under Uncertainty: Approximation and complexity.
24. Van Dijk, M. (2017). When I give, I give myself: essays on individual contributions to societal goals.

25. Oostervink, N. (2017). Self-Organizing Knowledge: Examining the conditions under which professionals share and integrate knowledge.
26. Mousavi, S. (2017). Managing innovation for sustainability- A dynamic capabilities approach.
27. Cai, W. (2017). Awakening employee creativity in organizations.
28. Kranzbuhler, A.M. (2018). Orchestrating the customer journey: Four essays on how to create meaningful customer experiences.
29. Chabala, M. (2018). Small firm growth in a least developed country: How small firm owners affect the growth of their firms in Zambia.
30. Choongo, P. (2018). Sustainable Entrepreneurship in Zambia: The engagement in and effect of sustainable practices in small and medium-sized enterprises.
31. Hilbolling, S. (2018). Organizing ecosystems for digital innovation.
32. Van der Wal, A.J. (2018). Harnessing ancestral roots to grow a sustainable world.
33. Hofstra, N. (2018). Individual decision-making in operations: A behavioral perspective.
34. Mu, Y. (2019). Management of Service Innovation Quality.
35. Stephenson, K.A. (2019). Paperless professors: A study of changing academic work and workspaces.
36. Hummel, J.T. (2019). Collaboration and innovation between heterogeneous actors.
37. De Jong, G. (2019). National carriers, market power and consumer loyalty: A study of the deregulatory international airline industry.
38. Szabo, A. (2019). The adoption of governance practices in hospitals: The role of multi-level frames of board members in making decisions about practice adaptation.
39. Sagath, D. (2019). Entrepreneurship in the Dutch space sector: The role of institutional logics, legitimacy and business incubation.
40. Günther, W.A. (2019). Data as strategic resources: Studies on how organizations explore the strategic opportunities of data.
41. Laurey, N. R. (2019). Design Meets Business: an ethnographic study of the changingwork and occupations of creatives.
42. Gorbatov, S. (2019). Personal branding: self-presentation in contemporary careers.