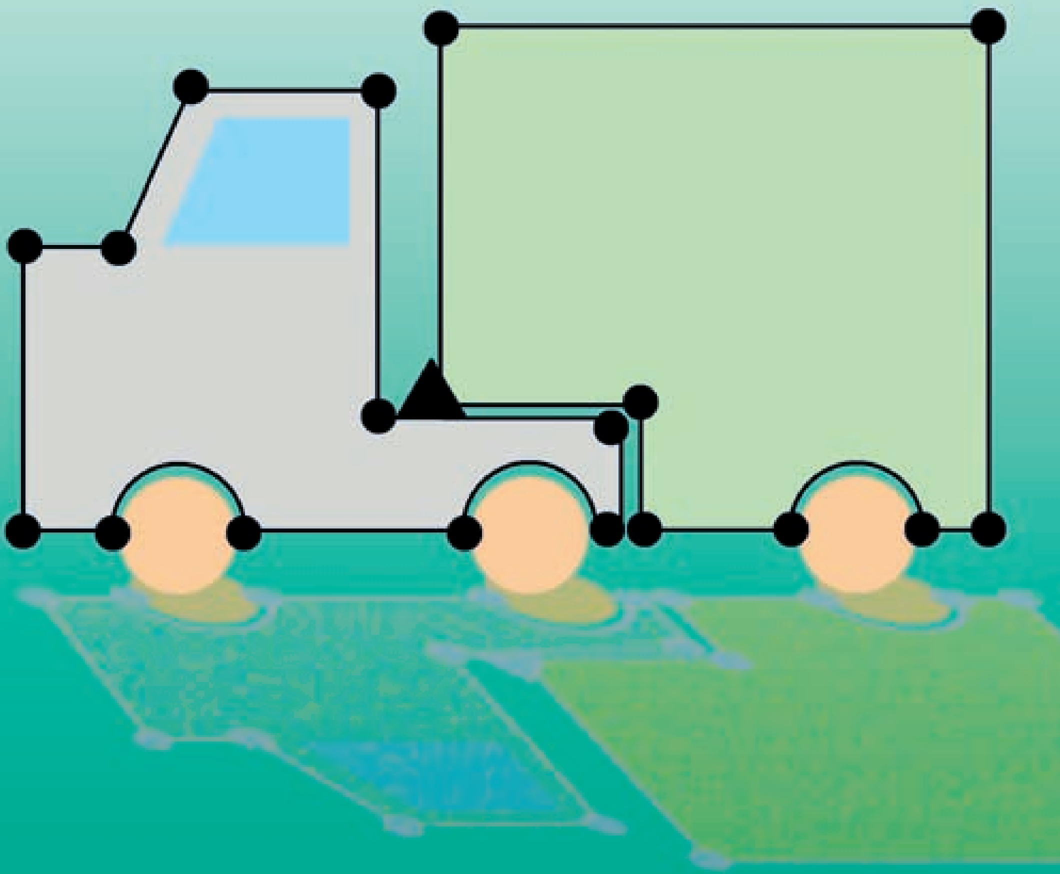
REMY SPLIET

Vehicle Routing with Uncertain Demand



Vehicle Routing with Uncertain Demand

Vehicle Routing with Uncertain Demand

Voertuig routering met onzekere vraag

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof.dr. H.G. Schmidt

and in accordance with the decision of the Doctorate Board

The public defense shall be held on

Friday 18 October 2013 at 13:30 hours

by

Remy Spliet
born in Nieuwegein, the Netherlands.

Erasmus Universiteit Rotterdam

Doctoral Committee

Promotor:	Prof.dr.ir. R. Dekker
Other members:	Prof.dr. G. Desaulniers Dr. D. Huisman Prof.dr. A.P.M. Wagelmans
Copromotor:	Dr. A.F. Gabor

Erasmus Research Institute of Management - ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM PhD Series in Research in Management, 293
ERIM reference number: EPS-2013-293-LIS
ISBN 978-90-5892-340-0

Design: B&T Ontwerp en advies www.b-en-t.nl

This publication (cover and interior) is printed by haveka.nl on recycled paper, Revive®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Flower, FSC, ISO14001.
More info: http://www.haveka.nl/greening

FSC

Acknowledgments

This thesis is about research I have conducted as a PhD candidate together with my supervisors and colleagues. However, the thesis has a different meaning to me than merely being the result of doing research for the past couple of years. In fact, I consider this thesis a byproduct of my studies. Instead, having written this thesis feels like a proper way to end all my activities as a PhD candidate. These activities include more than doing research. For me, being a PhD candidate is about developing myself as an academic, which roughly means developing research skills as well as teaching skills. As a PhD candidate I have been supported by many people which I would like to thank. It feels right to thank everyone that has supported me during my time as a PhD candidate, not just those who contributed directly to this thesis. I am very happy that this thesis allows me to publicly give credit to the people that truly deserve it.

First, I would like to thank my promotor Rommert Dekker. Even as his bachelor and master student, Rommert showed great confidence in me. He continued to give me this feeling throughout my time as a PhD candidate. Furthermore, his creativity in research is very inspiring, with Rommert there is no such thing as 'getting stuck'. Even though Rommert is infamous for being a very busy man, when I need him he is always there for me. Despite his full schedule, whenever I want to talk to Rommert, an appointment can always be scheduled on the same day or the next. I am sure this always means small sacrifices in his research time or personal time, so I am very grateful for this. I can always go to Rommert for advice, research related or otherwise, and his advice has always been useful to me. Most importantly, I have had great fun with Rommert during our many meetings and the trips we went on together.

Next, I would like to thank my copromotor Adriana F. Gabor. If I have to name only one person that taught me how to be an operations researcher, it is Adriana. Most importantly, she taught me how to write a scientific paper. She has always been tireless in providing highly detailed feedback on everything I wrote. I remember a week in which we sat down together at her desk and we went through one of our papers word by word, editing until we were satisfied. This makes me feel very special, I have never heard of a supervisor that is so committed as Adriana. Furthermore, she also invested in me with respect to developing teaching skills. She let me join her in supervising students writing their thesis, this gave me the opportunity to learn how she did it. I really admire her didactic skills, she always seems to know what the student is thinking and how to make the student discover things on its own. Also the door of Adriana has always been open when I needed advice. Moreover, I really enjoyed the many meetings we have had so far, Adriana always made sure there was the perfect mix of hard work and laughter.

The third person that directly contributed to this thesis is Guy Desaulniers. I am very grateful to him for hosting me on a research visit at GERAD and for making time to do research with me. Listening to Guy and working with him has given me a clear vision of the current scientific advances in vehicle routing. Moreover, I know Guy for being a warm person. He made me feel very comfortable in Montréal, for instance by introducing me to his wife and daughters, and inviting me along to see beautiful autumn leaves in a nice forest outside of the city. Finally, I thank Guy for agreeing to be a member of my inner committee.

I want to thank Guy Drop for writing a master thesis that is directly related to this thesis. I really appreciate the hard work he has done, and the insights his research has given me. Together we visited store managers, transport managers and dispatchers of four major Dutch retail companies, whom I also want to thank for spending time on me and educating me on current practice in vehicle routing. Furthermore, I would like to thank Ritish Oemraw and Tim Lamballais Tessensohn for writing their bachelor and master thesis respectively on the TWAVRP. As their co supervisor, both Ritish and Tim inspired me with their creativity.

Jan Brinkhuis has not directly been involved with the writing of this thesis. However, he has had a big influence on my development of teaching skills. As his teaching assistant, he showed me the art of motivating students. But more importantly, when I was the main teacher of a course for the first time, Jan was there to guide me. I could not have done this without him, and I think it is to his credit that I have learned so much from this experience as I did. Unrelated, the reading courses that Jan organized with great enthusiasm have also been a great source of inspiration.

The second time I got to teach a course, Albert Wagelmans was my go-to guy. Albert showed great involvement and it felt as if we were giving the course together. This made it a very relaxed experience. He showed me the administrative side of teaching, which has also been very educational to me. Furthermore, I want to also thank Albert for agreeing to be a member of my inner committee.

Working together with Dennis Huisman has also been a real pleasure. He showed great support of my research activities. I enjoyed the many trips Dennis and I have made together. I also want to mention that I am grateful to Dennis for introducing me to Guy Desaulniers, as this resulted in a good research collaboration. Finally, I want to thank Dennis not only for agreeing to be a member of my inner committee but also for acting as secretary of this committee.

My thanks also go to Goos Kant, Tom van Woensel and René de Koster for agreeing to be the outer committee. Additional thanks go to René for supporting my endeavours together with Guy Drop as his thesis supervisor.

I have also learned a lot from being the teaching assistant of Hans Frenk, Martyn Mulder and Wilco van der Heuvel. Each has a different teaching style which has rubbed off on me. They all trusted me with a great deal of responsibility and freedom in teaching. With Wilco in particular I have formed a bond. Not only because I have been his assistant most often, but also because we have been meeting on social occasions as well, for instance when playing basketball and indoor football. Also, Wilco helped me when I was doing a course in didactics, most notably by letting the course he was teaching be the object of my study.

Kristiaan Glorie has been my officemate for the most part of my time as a PhD candidate. We have grown close and often meet outside of the office. We share most of the same passions and interests, which makes it really easy to be around him all day. We have a lot of fun together. I want to thank Kristiaan for agreeing to be my paranymph.

Twan Dollevoet was of great moral support. I truly enjoy our many profound discussions, and perhaps even more so the less profound ones. I consider Willem van Jaarsveld as the most creative of my colleagues. He always makes me laugh with the astonishing insights that he has on a daily basis. I mention Mathijn Retel Helmrich and Judith Mulder in one go, as they both felt like my officemates from time to time. Their moral support is relentless, and both Mathijn and Judith have an unparalleled sense of humour. Also Lanah made me laugh often enough. Moreover, her competitive nature has always been very inspirational. Moral support in the morning hours has always fallen on the shoulders of Ilse Louwerse, at which she did a great job. Even when I tried, I could never beat Ilse Louwerse at being the first at the office. Zahra Mobini Dehkordi has a truly endearing personality, which always puts a smile on my face. Kristiaan, Twan, Willem, Mathijn, Judith, Lanah, Ilse and Zahra do not only provide me with much needed laughter, they are also always prepared to listen to my ideas and to give valuable comments. That way, they also contributed a lot to this thesis.

I also want to thank Evşen Korkmaz for all her support. Her joyful spirit made me walk to the other side of the campus for a cup of coffee on more than one occasion, even when I was not in the mood for coffee. I am grateful to Evşen for all the fun times we have had together during and after office hours. Also Başak Manders I have gotten to know inside the office and outside. I want to thank Başak as well for all the nice coffee breaks, lunches and social gatherings we experienced together.

My battery remained charged by the presence of my colleagues at the many social events that I was part of such as Friday evening drinks at the Smitse, basketball and indoor football. Also the conferences that I attended have been made a lot of fun by my colleagues. Besides the people that I have already expressed my gratitude to, I also want to thank in no particular order Pieter Schoonees, Harwin de Vries, Paul Bouman, Luuk Veelenturf, Joris Wagenaar, Theresia van Essen, Elaine Rouwet, Tommi Tervonen, Guangyuan Yang, Clint Pennings, Gertjan van den Burg, Sarita Koendjbiharie, Xiao Peng, Anne Opschoor, Wouter Vermeer, Panagiotis Ypsilantis, Evelien van der Hurk and Jelmer van der Gaast.

My stay at GERAD in Montréal was very pleasant. I was impressed by the hospitality of Thibault Lehouillier, Samuel Rosat, Matthieu Delorme, Lê Nguyên Hoang, Marilène Cherkesly, Jean-Bertrand Gauthier, Charles Gauvin and Mélisende Brazeau. I would like to mention the daily game of cards that I enjoyed very much, in particular playing the game of klaverjassen/belote made me feel at home straight away.

Furthermore, I want to thank Carien de Ruiter-Coopmans, Anneke Kop, Marjon van Hees-Gouweleeuw, Ursula David, Elli Hoek van Dijke and Marianne Kroek-Buijs for both their administrative as well as their moral support. Their help with the many seminars we organized together has been invaluable.

For a relatively short period of time, I also shared an office with Sven Weeda. We started our PhD candidacies at roughly the same time and despite being in different fields of research, we really hit it off. I still uphold the tradition we started together of having a basketball hoop in the office.

Ioannis Attalialis is an admirable friend. I have been crediting the people that helped me develop research and teaching skills; Ioannis is my shining example of social skills. There is never a dull moment when he is around. I hope we will have good times together for all the years to come, just like we have had in the past years. I want to thank Ioannis for agreeing to be my paranymph.

I want to thank my friends that take my mind off work at appropriate times. They help me attain a good balance between work and pleasure, which I truly believe increases my productivity. In particular, my thanks go to Michelle Lieuw On, Diederik de Ree, Vân Bùi, Ioanna Panagiotopoulou and Ka Wang Man.

I have received a lot of moral support from my family. I want to thank my father Jan, my mother Marja, my brothers Roy, Luke and Tom, and my sister Lois. They are the source of my ambition. Everything I do, I do to make them proud. I hope I am succeeding.

Finally, I want to thank Priscilla Kuhl. Priscilla gives me a reason to stop working at the end of every day. Whenever I need to vent my enthusiasm or sometimes even my frustration, she is there to take the brunt of my rants. Her support and affection is unwavering. This has kept me going during my time as a PhD candidate. Thank you Priscilla, for everything.

Remy Spliet
Rotterdam, May 18, 2013

Table of Contents

Chapter 1

Introduction

1.1 The vehicle routing problem

The vehicle routing problem, VRP, is the problem of designing minimum cost routes for vehicles with limited capacity, to transport goods from a distribution center to a set of customers. Since its introduction in the scientific literature by Dantzig and Ramzer (1959), this problem has become one of the classic combinatorial optimization problems. Solving the vehicle routing problem has been the research topic for many researchers ever since, see among others Baldacci et al. (2012) and Laporte (2009) for recent advances in algorithms to solve this problem.

In practice, this problem plays an important role in many distribution networks. For instance, many retail organizations store goods at a central depot and transport these to the retailers on a weekly or sometimes even daily basis. Hence, having an efficient delivery schedule is important for achieving low transportation costs.

Moreover, not only the delivery schedule itself is important for transport operations, even the method of designing a delivery schedule is crucial. For instance, package delivery companies often have very limited time in between the moment a customer places an order and the moment a package has to be picked up or delivered. In this case, a delivery schedule has to be made in a couple of hours, sometimes even minutes or seconds. Clearly, constructing efficient delivery schedules in such a short amount of time requires the aid of sophisticated algorithms and significant computing power.

1.2 Constructing delivery schedules

In practice, many delivery schedules are currently still constructed manually by experienced schedulers. They are often the product of small modifications to an existing delivery schedule over time. Only recently are more and more companies using commercial software to aid in the scheduling process. This is made possible by the increase in available computing power and by the development of algorithms to construct good delivery schedules in limited computation time.

Algorithms that solve the VRP to construct a delivery schedule can be divided into two categories: exact algorithms and heuristic algorithms. Exact algorithms are able to solve the VRP to optimality. The most successful exact algorithms in the current scientific literature are branch-and-cut and branch-price-and-cut algorithms, see also Baldacci et al. (2012), Laporte (2009) and Toth and Vigo (2002). Much research has been devoted to finding strong formulations of the vehicle routing problem and separating valid inequalities. These exact algorithms are used to solve instances of the VRP with tens of customers and specific instances of over a hundred customers, in reasonable computation time, see for instance Baldacci et al. (2004).

Industrial scale instances of the VRP may consist of hundreds or thousands of customers and have many additional side constraints such as time window constraints or driver assignment constraints, see for instance Groër et al. (2009). Therefore, many researchers have also devoted their studies to designing heuristic algorithms to find good solutions for large VRP instances in limited computation time. For an overview of such algorithms see for instance Laporte (2009), Bräysy and Gendreau (2005a,b) and Toth and Vigo (2002).

1.3 Fixed schedules

In distribution networks where deliveries are made regularly to the same customers, a fixed delivery schedule is usually maintained. Such a schedule allows the packing of the vehicles at the distribution center to be planned easily, and it allows the customer to roster the delivery handling personnel. Also, inventory control and sales management benefit greatly from knowing the delivery schedule in advance. Furthermore, it is often even beneficial for business to have the same driver visit a customer. For instance, Groër et al. (2009) indicate that because drivers at UPS form a real bond with customers, they generate additional sales with a volume of over 60 million packages per year.

As many business processes in a distribution network are dependent on the delivery schedule, deviating from this schedule can lead to significant cost increases. In a study by Drop (2011), the effects on retailers of deviating from a delivery schedule were investigated for four large Dutch retail chains, each having a distribution network consisting of between 175 and 600 retailers. These retailers indicated that for each hour by which a delivery is late, on average an additional 5.5 man-hours are required. This is caused by the delivery handling crew being idle at the moment that the delivery was scheduled, and having to work overtime when the delivery arrives. Moreover, the retailers indicate that secondary costs such as lost sales, loss of goodwill from the customers and waste as a result of perishable goods not being sold in time, have an even more negative influence on the performance.

A ‘fixed schedule’ can refer to different things; in this thesis the following three types of fixed schedules are considered. First of all, one can simply stick to a delivery schedule as much as possible, i.e. always have the same driver visit the same customers in the same order. A second view is the following. A customer will usually care most about receiving a delivery during a given time window. In such a situation one may choose not to have a fixed delivery schedule, but rather to have fixed time windows for each customer. This way, the time at which a customer receives a delivery is fixed but the routes driven by the vehicles may change all the time. Thirdly, there are situations where the time at which a delivery is made does not matter, but business benefits from having the same driver visit a customer. Now, a fixed schedule is in fact a fixed assignment of customers to drivers. Also in this case the routes driven by the vehicles may change all the time, as long as customers are always visited by the same driver.

1.4 Uncertain demand

A fixed schedule can be reused to make weekly deliveries for a period of a year or longer. Travel time between two locations and the demand of customers might still be uncertain at the moment of scheduling. In the study by Drop (2011) several practitioners stated in interviews that although travel time is usually perceived as uncertain, travel times can be reasonably well foreseen. Almost all significant fluctuations in travel times are caused by traffic jams, and they are quite consistent over the days. Therefore, travel time is time dependent rather than uncertain, and is typically taken into account effectively in a fixed schedule. Contrarily, practitioners state that uncertainty of demand is the main cause of having to deviate from a fixed schedule.

The demand of each customer for each delivery is typically not known at the moment of scheduling. Moreover, demand of one customer is usually not the same for every delivery. Furthermore, in retail chains, the variations in demand are often highly correlated among retailers. This is because of shared advertisement campaigns, similar seasonal effects, and general homogeneity of the customers of the retailers.

In the case of a fixed delivery schedule, when demand becomes known it is often necessary to alter the delivery schedule, i.e. to reschedule. High demand might render the original delivery schedule infeasible due to limited capacity of the vehicles, while low demand might offer a possibility to save on transportation costs by using less vehicles. Finally, note that in retail, demand is usually communicated to the distribution center one day before delivery, leaving little time to adjust the delivery schedule if necessary.

The expected costs of a fixed schedule is determined by the rescheduling procedure used to construct a delivery schedule when demand becomes known. Constructing a fixed schedule when facing uncertain demand has only been considered in a limited number of studies. The rescheduling procedures used in these studies are typically not very sophisticated and may lead to inefficient delivery schedules in certain applications, but usually present computational advantages.

Among the most popular ways of rescheduling in the scientific literature is the strategy suggested by Dror et al. (1989). When rescheduling, the original delivery route is followed until the load of the truck is depleted, and is resumed after a visit to the distribution center to restock. An advantage of this rescheduling procedure is that the expected costs of a vehicle route can be computed efficiently. Studies on this model with uncertain demand include the work by Laporte et al. (2002) who find the fixed delivery schedule with minimum expected transportation costs using an integer L-shaped method and Novoa and Storer (2009) who develop an approximate dynamic programming approach for the single vehicle variant. In cases where demand is known before vehicles are dispatched, more efficient delivery schedules can be constructed than with the rescheduling procedure described above.

1.5 Contribution

In this thesis several different views on a fixed schedule are explored. It addresses the need in practice to incorporate the uncertainty of demand that is experienced in reality in transportation models to increase the efficiency of transport. Innovative vehicle routing models are presented taking uncertain or varying demand into account. New algorithms using state-of-the-art methods are presented, based on these models, to construct fixed schedules and vehicle routes. The algorithms make use of recent scientific advances in mathematical programming, specifically in the domain of vehicle routing. Next we will list the contributions in more detail.

A new advanced rescheduling procedure is introduced that can be used when a fixed delivery schedule is given. In this procedure penalties are incurred when deviations are made from the fixed delivery schedule. An exact algorithm is used to solve the rescheduling problem to optimality. Also a fast heuristic is proposed to find good solutions to the rescheduling problem.

Furthermore, the problem of assigning time windows to customers as a fixed schedule, before demand of the customers is known, is investigated. In this case, rescheduling means constructing a delivery schedule in which the time window constraints are satisfied such that the transportation costs are minimized. Only limited research has been done so far on assigning time windows in this setting. State-of-the-art exact algorithms are presented in this thesis to find time window assignments yielding minimal expected transportation costs.

Also, the problem of assigning customers to drivers before demand is known is introduced. Here, customers need to be visited by the driver to which they are assigned, which has to be taken into account in the rescheduling procedure. A fast heuristic is presented to find driver assignments yielding low expected transportation costs.

1.6 Thesis outline

In this thesis, several different types of fixed schedules are investigated for making frequent deliveries with uncertain and varying demand. In Chapter 2 the case is considered where a fixed delivery schedule is given. Also, the demand of each customer is known. Depending on the demand, rescheduling might be desired when the fixed delivery schedule is very inefficient, or necessary when the fixed delivery schedule is infeasible. In this chapter, a model is proposed in which a penalty is suffered when the new schedule deviates from the master schedule. The vehicle rescheduling problem is the problem of rescheduling such that the total transportation and penalty costs are minimized. This problem is solved to optimality using a branch-and-cut algorithm. Moreover, a fast two-phase heuristic is presented and its performance is analyzed. This chapter is based on joint work with Adriana F. Gabor and Rommert Dekker.

In many distribution networks, the supplier and customer agree on a time window in which the customer receives its delivery. In Chapter 3 the problem of assigning time windows to customers before demand is known is introduced, such that the expected transportation costs are minimized. In this problem, each customer has a wide exogenous time window, for instance the opening hours of a store, in which a small time window has to be selected. A branch-price-and-cut algorithm is developed to solve this problem to optimality. In this chapter, the costs of assigning time windows using a delivery schedule based on average demand, as is typically done in practice, are compared with the costs of the optimal time window assignment. This chapter is based on joint work with Adriana F. Gabor.

The time window assignment problem is extended in Chapter 4. In the problem introduced in this chapter, a time window is not selected from a wide exogenous time window, instead a time window is selected for each customer from a discrete set of candidate time windows. In practice, only a limited number of time windows are sensible. For example, delivery handling shifts might be blocks of two hours starting on the hour. Having a discrete set of time windows also enables the development of a more sophisticated branch-price-and-cut algorithm. The costs of assigning time windows using a delivery schedule based on average demand are also in this chapter compared with the costs of the optimal time window assignment. This chapter is based on joint work with Guy Desaulniers.

Finally, in Chapter 5 the case is considered in which a customer always needs to be visited by the same driver. A model is introduced in which customers are assigned to a driver before demand is known, and when demand is known a delivery schedule has to be constructed such that for every driver at least a fraction α of the customers assigned to that driver are visited by it. This problem is particularly relevant in the case where the drivers are also responsible for placing the delivered goods in the storage facility, and as such require a key or password and needs to know exactly how to place the goods in the storage facility. A cluster first-route second heuristic is proposed to find a solution to this problem and it is used to study the additional costs of adhering to the driver assignments as opposed to not adhering to the driver assignments. This chapter is based on joint work with Rommert Dekker.

Chapters 2 to 5 can be read individually. As a consequence there is some overlap in the introduction of each of these chapters. In Chapter 6 a summary and conclusion is provided.

Chapter 2

The Vehicle Rescheduling Problem

2.1 Introduction

Scheduling and rescheduling

The capacitated vehicle routing problem (CVRP) is a classical problem in operations research. Consider a depot where goods are stored and a set of customers that have nonnegative demand for these goods. A set of homogeneous vehicles of finite capacity is available to transport the goods from the depot to the customers. The vehicles start and end their routes at the depot. Costs are incurred for traveling from one location to another. The CVRP is to find a routing schedule that describes the sequence of locations visited by every vehicle that minimizes the total traveling costs, while the capacity constraints are satisfied. The CVRP is known to be an NP-hard problem.

Many solution methods can be found in the scientific literature for the CVRP. The branch-and-cut scheme of Baldacci et al. (2004) seems presently to be one of the most successful at solving CVRP instances of up to 100 customer locations. For larger problem instances, many heuristic algorithms have also been developed that are able to find good solutions with greater speed. An overview of exact and heuristic algorithms can be found in Fisher (1995), Toth and Vigo (2002), Laporte (1992) and (2007) and Laporte et al. (2000) amongst others.

In the classical CVRP, demand is deterministic and known. A situation that often occurs in practice is that demand only becomes apparent at a late moment. For example, in the retail industry it is very common that the orders of the individual stores are placed only a few days, sometimes even just one day, before delivery. In these situations it is beneficial for operational processes to determine the delivery schedule before the orders are placed. It is for instance very costly, if at all possible, to make a roster for delivery handling personnel shortly before they are needed. A common solution to this problem is to determine a long term schedule, henceforward *master schedule*, that serves as a guiding schedule over a certain period of time in which multiple deliveries are made. For example, such a master schedule would describe the weekly or even daily deliveries for a period of six months. In practice, master schedules are usually constructed by solving deterministic CVRP instances based on average customer demand as predicted for the upcoming period. In this chapter we assume that such a master schedule is given.

A master schedule is thus made before demand realizations become apparent. As a result, when the demand becomes known, the master schedule may not be optimal due to inefficient use of the vehicles, or may even be infeasible due to violation of the capacity constraints. In such cases the master schedule needs to be deviated from. This is often necessary in practice, for example when demand of customers is highly correlated, as typically is the case in a retail chain. The construction of a new schedule when demand realizations become known, will be referred to as *rescheduling*.

Effects of rescheduling

After rescheduling, the new schedule will typically deviate from the master schedule. This can have negative effects on a distribution network. Locations are visited in a different order, by different trucks or by different truck drivers than initially planned. This may cause confusion among drivers and negatively affect the regularization and personalization of service, as is also recognized by Bertsimas and Simchi-Levi (1996) and Li et al. (2007) and (2009). Furthermore, consider the situation where personnel is hired only for handling deliveries and deliveries do not arrive at the agreed upon moment due to a deviation in the schedule. Here, labor costs increase due to the fact that personnel has to work overtime or has to be hired for another shift. When rescheduling is done by constructing a completely new schedule, many deviations may occur resulting in high additional costs.

Our experience with Dutch retail companies has shown that currently rescheduling is often done manually. Dispatchers typically operate under the notion that when a route needs to be deviated from, costs are lower when the first deviation occurs at a later stage in the route. There are several arguments that support this notion. Firstly, when deviating at a late stage in each route large portions of the master schedule remain intact, diminishing the above mentioned negative effects of rescheduling. Secondly, the changes made in this manner are easily communicated through the distribution network. Finally, when changing the first locations of a route, the dispatching times of the truck will be altered. However, changing the working hours of a driver at a late moment is very expensive and often practically infeasible.

The vehicle rescheduling problem

In this chapter, we propose a rescheduling model in which the negative effects of deviating from the master schedule are incorporated. We introduce deviation costs, which are incurred for each route that deviates from the master schedule. Furthermore, the height of the deviation cost per route is dependent on the customer at which the first deviation occurs and the position in the route it has. In this way, we are able to model the above described notion of dispatchers that deviations early in a route are more costly than deviations late in a route.

Given a master schedule and a demand realization, the goal is to find a new schedule that minimizes the total traveling and deviation costs, while satisfying the capacity constraints. This problem will be referred to as the vehicle rescheduling problem (VRSP). This model is of particular interest to for instance retail chains that control both the supply chain and the stores, as they not only incur the transportation costs, but also both the deviation costs at the supply side and at the customer side.

Rescheduling in current literature

In the literature, rescheduling is mainly considered in conjunction with designing a master schedule. Given a rescheduling method, the master schedule is designed before demand is known such that the expected costs incurred after rescheduling are minimized. The rescheduling method proposed by Bertsimas (1992) is maybe the most popular method in the literature. In this method, the master schedule is used until a vehicle arrives at a location where its cargo is depleted. After it has returned to the depot to refill, the vehicle resumes the master schedule from the location where it left off. Under this rescheduling protocol, for specific demand distributions the expected costs of a master schedule can easily be calculated. For this reason, the rescheduling method proposed by Bertsimas has been incorporated in models used to design a master schedule with minimal expected costs. Examples of solution methods to solve these models are the L-Shaped integer method to find the optimal master schedule by Laporte et al. (2002), a tabu search heuristic by Gendreau et al. (1996), a rollout algorithm by Secomandi (2001) and an evolutionary algorithm by Tan et al. (2007).

In a study by Groër et al. (2009) they propose to reschedule in such a way that each customer is always visited by the same driver and within the same time window. They apply this to a setting encountered in the small-package shipping industry in which a customer does not require service on all delivery days. In their paper they focus on generating a master schedule for large instances and do this using a local search heuristic. Similarly, Chen et al. (2009) consider an arc-routing model for small-package delivery in which arcs that do not need service are skipped.

To the best of our knowledge, the literature on rescheduling strategies that take into account deviation costs is scarce. Li et al. (2007) and (2009) consider the problem of reassigning vehicles to trips, when one of the vehicles breaks down. In their model, costs are incurred when trips are delayed. The main application of this model is in passenger transportation, for situations where traveling costs and capacity constraints do not play an important role.

Contribution

In this chapter, we introduce a novel model for the rescheduling problem which can be applied where demand is known shortly before vehicles are dispatched. We provide a mixed integer programming formulation based on a formulation of the CVRP by Baldacci et al. (2004). Using this formulation, the VRSP of moderate size can be solved by general purpose optimization software or slight modifications of existing algorithms for solving the CVRP.

Furthermore, we design a solution approach based on first removing the last locations of routes and rescheduling them. We will refer to this approach as the *two-phase heuristic*. We analyze the performance of this heuristic and derive sufficiency conditions on the value of the deviation costs for which the two-phase heuristic is guaranteed to give the optimal solution to the VRSP. Moreover, numerical experiments indicate that, in general, for low deviation costs the two-phase heuristic often provides optimal or near optimal solutions. Finally, we describe this algorithm in such a way that it can be implemented directly in existing commercial CVRP software available to many dispatchers in large distribution networks.

Outline

In the following section, the VRSP is described in detail and a mixed integer linear programming formulation is presented. In Section 2.3, the two-phase heuristic is presented. It is accompanied by an analysis of its behavior with respect to the deviation costs. Finally, in Section 2.4 the sensitivity of the solution to the VRSP with respect to the value of the deviation costs is investigated and the performance of the two-phase heuristic is studied by comparing its solutions with the optimal solutions of the VRSP.

2.2 The vehicle rescheduling problem

In this section, first the vehicle rescheduling problem is defined. This is followed by a mixed integer programming formulation based on an existing model developed by Baldacci et al. (2004). In Section 2.4, we will use this MIP formulation in computational experiments.

2.2.1 Problem definition

Consider an undirected complete graph $G = (V, E)$. The set of nodes $V = \{0, 1, ..., n+1\}$ corresponds to a starting depot 0, an ending depot $n+1$ and the set of customers $V' = \{1, ..., n\}$. For every edge $(i, j) \in E$, traveling costs $c_{ij} \geq 0$ are given that satisfy the triangle inequality. We suppose that an unlimited number of vehicles of capacity Q are available for supplying goods to the customers. Furthermore, for every location $i \in V'$ the demand q_i is given and satisfies $0 < q_i \leq Q$.

A *route* $r \subset E$ is defined as a path in G including the depot. With every route $r = \{(0, i_1), ..., (i_k, n+1)\}$ we associate the ordered set of vertices $\{i_1, ..., i_k\}$. Throughout this chapter we will use these representations interchangeably. It will be clear from the context whether the edge or the vertex representation is meant. A route r is called *feasible* when the total demand of the locations on r is less than or equal to the capacity of a vehicle, i.e., $\sum_{i \in r} q_i \leq Q$.

A routing schedule S is a collection of edge-disjoint routes such that all customers are included in exactly one route. Hence, $S = \bigcup_{i=1}^{m} r_i$, where for the routes $r_1, ..., r_m$ it holds that $r_i \bigcap r_j = \emptyset$ for $i \neq j$. A schedule S is called *feasible* when all routes in S are feasible. The set of all feasible schedules will be denoted by $\mathbb{S}$.

The classical CVRP, a closely related problem to the VRSP, can now be defined as finding a feasible schedule that minimizes the total traveling costs and can be formulated as:

$$\text{(CVRP)} \min_{S \in \$} \sum_{(i,j) \in S} c_{ij}.$$

Next we define the VRSP. Assume that a master schedule S_M is available. Note that this master schedule need not be feasible for all demand realizations, as capacity restrictions might be violated. The VRSP is to create a new feasible schedule S^* that minimizes both the traveling costs and the costs of deviating from the master schedule. Next we formally define a deviation and the accompanying deviation costs.

Consider a route $r_M = \{i_1, ..., i_j, ..., i_k\}$ in the master schedule S_M. When for a new schedule S there is a route $r \in S$ such that the first locations are $\{i_1, ..., i_{j-1}\}$ and the following location, if any, is not i_j, then r *deviates from location* i_j *onwards in schedule* S. In other words, route r originates from route r_M and it has remained the same up to location i_j. Whenever a route deviates from location $i \in V'$ onwards, costs $u_i \geq 0$ are incurred. These costs are in practice often dependent not only on the location at which the master schedule is deviated from, but also on the position of that location in the route in the master schedule. However, as the master schedule is given and the position of each location in the route in the master schedule is fixed, introducing deviation costs per location is sufficient for our purposes. Thus, for a master schedule S_M and a schedule S, the *deviation costs* for a route from location i onwards are defined as:

$$U(S_M, S, i) = \begin{cases} u_i, & \text{if a routes deviates from location } i \text{ onwards in } S \text{ with respect to } S_M \\ 0, & \text{otherwise.} \end{cases};$$

Throughout this chapter we will assume that for each route in the master schedule that the deviation costs associated with the locations on that route are decreasing in the positions on that route.

For $r_M = \{i_1, ...i_j, ..., i_k\}$ in the master schedule, if a route r deviates from location i_j onwards, we will refer to any i_v in route r_M for $j \leq v$ as a *rescheduled location.*

It is now possible to fully define the VRSP as finding a feasible schedule S^* such that it minimizes the total traveling and deviation costs for a given master schedule S_M:

$$\text{(VRSP)} \min_{S \in \$} [\sum_{(i,j) \in S} c_{ij} + \sum_{i \in V'} U(S_M, S, i)]$$

Note that if in the VRSP $u_i = 0 \ \forall i \in V'$ one obtains the classical CVRP. As the latter is NP-hard, the VRSP is also NP-hard.

2.2.2 Mixed integer programming formulation

Next we provide a mixed integer programming formulation of the VRSP which is a modification of an existing formulation of the CVRP. As stated in Laporte (2007), one of the most successfully used formulations of the CVRP is the two commodity flow formulation introduced by Baldacci et al. (2004). Due to the polynomial number of constraints and variables, a direct implementation of this formulation in general purpose mixed integer programming software is sufficient to find a solution to the CVRP for moderately sized instances. Moreover, any cutting-plane designed for the CVRP, like for instance generalized capacity constraints (2004), can be applied here as well.

Next, we briefly discuss the parts inherited from the CVRP model of Baldacci et al. and refer the interested reader to their paper for more details. Furthermore, we elaborate on the addition of deviation costs.

For all $(i, j) \in E$, let ξ_{ij} indicate whether edge (i, j) is included in the new schedule. Next, let the variables $x_{ij} \in \mathbb{R}_+$, for $i, j \in V$, be flow variables. When traveling from i to j, x_{ij} might be interpreted as the load of a vehicle and x_{ji} the remaining capacity.

To model the deviation costs, the variable y_i is introduced for each location $i \in V'$, which indicates whether a route is deviating from location i onwards. Observe that for any route r, $\sum_{j \in r} u_j y_j = \sum_{j \in r} U(S_M, S, j)$.

In the following formulation we will use $j_{\preceq}$ to denote the set containing j and all locations which are visited prior to j on the same route in the master schedule. For a given master schedule S_M, the mixed integer programming formulation of the VRSP is:

$$\text{(CF)} \ \min \sum_{(i,j)\in E} c_{ij}\xi_{ij} + \sum_{i\in V'} u_i y_i \tag{2.1}$$

$$\sum_{j\in V}(x_{ji} - x_{ij}) = 2q_i \qquad \forall i \in V' \tag{2.2}$$

$$\sum_{i\in V'} x_{0i} = \sum_{i\in V'} q_i \tag{2.3}$$

$$\sum_{i\in V'} x_{i0} = Q\sum_{i\in V'} \xi_{0i} - \sum_{i\in V'} q_i \tag{2.4}$$

$$\sum_{i\in V'} x_{n+1i} = Q\sum_{i\in V'} \xi_{0i} \tag{2.5}$$

$$x_{ij} + x_{ji} = Q\xi_{ij} \qquad \forall (i,j) \in E \tag{2.6}$$

$$\sum_{j\in V, i<j} \xi_{ij} + \sum_{j\in V, i>j} \xi_{ji} = 2 \qquad \forall i \in V' \tag{2.7}$$

$$1 - \xi_{ij} \leq \sum_{k\in j_{\preceq}} y_k \qquad \forall (i,j) \in S_M, i<j, j\in V' \tag{2.8}$$

$$1 - \xi_{ij} \leq \sum_{k\in i_{\preceq}} y_k \qquad \forall (j,i) \in S_M, i<j, j\in V' \tag{2.9}$$

$$\xi_{ij} \in \{0,1\} \qquad \forall (i,j) \in E \tag{2.10}$$

$$x_{ij} \in \mathbb{R}_+ \qquad \forall i,j \in V' \tag{2.11}$$

$$y_i \in \{0,1\} \qquad \forall i \in V' \tag{2.12}$$

The set of constraints (2.2)-(2.5) and (2.11) ensure that x provides a correct flow pattern between the depots 0 and $n+1$. By constraints (2.2) the difference between inflow and outflow at a customer location of both the vehicle load and the remaining vehicle capacity is equal to the demand. Constraint (2.3) ensures that the outflow of the starting depot is equal to the total demand of all customer locations and constraint (2.5) ensures that the inflow at the ending depot, is the total capacity of the used trucks. Note that $\sum_{j\in V'} \xi_{0j}$ represents the number of vehicles that are used. Constraint (2.4) sets the remaining capacity of trucks leaving the depot.

Constraints (2.6) ensure that there is either no flow through edge (i,j) when this edge does not belong to any route or that the total load and empty space defined for this edge is exactly Q otherwise. Constraints (2.7) ensure that exactly two edges incident to any customer are used. The objective function $\sum_{(i,j)\in E} c_{ij}\xi_{ij}$ together with constraints (2.2)-(2.7) and (2.10)-(2.11), give a correct formulation of the CVRP, as indicated by Baldacci et al. (2004).

Finally, the VRSP formulation is completed as (2.8), (2.9) and (2.12) force y_i to take value 1 whenever a route deviates from location i onward. Moreover, note that as $u_i \geq 0$ for all $i \in V'$ and the formulation describes a minimization problem, for each route at most one location will have value $y_i = 1$.

Note that when using the above formulation, an optimal solution may exist including paths from 0 to 0 and from $n+1$ to $n+1$. The solution can in this case be transformed into a solution solely with paths from 0 to $n+1$, without an increase in costs. For paths from 0 to 0 or from $n+1$ to $n+1$ the corresponding flow variables x can not be interpreted as the load of a vehicle or remaining capacity at an edge.

Note that the integrality of y can be relaxed without compromising the validity of the two commodity flow formulation.

2.3 Two-phase rescheduling heuristic

In this section we propose and analyze a two-phase heuristic. The main idea behind the two-phase heuristic is to start with the possibly infeasible master schedule S_M and modify it to make it feasible, resulting in a new schedule S^{TP}. In the first phase of the heuristic, a specific set of edges is removed from the master schedule and in the second phase new edges are added such that the obtained schedule is feasible and has low deviation costs. Next we describe the two phases in more detail.

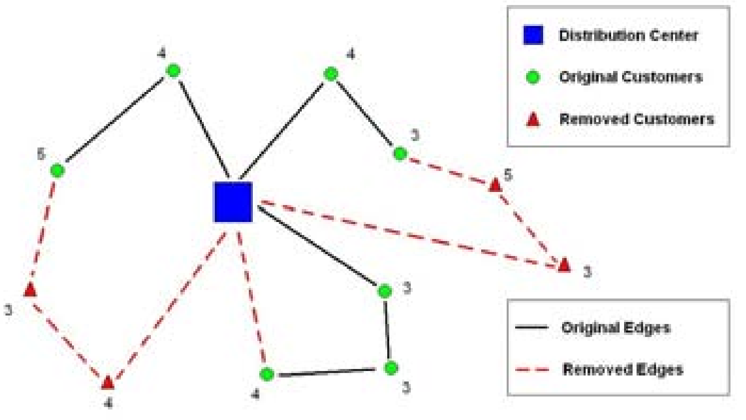
2.3.1 Phase 1: Removing edges

When choosing the edges that will be removed from S_M, the main criterion is to limit the total deviation costs that are incurred in the resulting schedule S^{TP}. For any route $r \in S_M$, the final edge is removed. Next, for each route the last edge is removed and this is repeated iteratively until the total demand of the remaining locations in r does not exceed Q. Denote by $\hat{V}$ the set of resulting isolated locations, these will be the rescheduled locations in S^{TP}.

The result of Phase 1 is a rooted tree S_1 with root node 0 and vertex set $V' \backslash \hat{V}$, representing the set of incomplete routes. The total demand of the locations on any path from the root to a leaf is at most Q.

Figure 2.1 shows an example with a network of a single depot and several customers. The solid and dashed lines combined show the original schedule. The numbers next to the customers correspond to a realization of demand. The vehicles have a capacity of 10 units of demand. The dashed lines correspond to edges that are removed during the first phase.

Figure 2.1: Example phase 1: removed edges from the master schedule



2.3.2 Phase 2: Adding edges

In this phase, edges are added to the incomplete schedule S_1 such that it becomes a feasible schedule S^{TP}. This is done at minimal additional traveling costs. The problem that needs to be solved can therefore be defined as:

$$S^{TP} = \arg \min_{S \in \mathbb{S} | S_1 \subset S} \sum_{(i,j) \in S} c_{ij} \tag{2.13}$$

This is an instance of the CVRP in which certain edges are fixed. In some standard CVRP software, it may not be possible to prescribe the use of certain edges in the generated solution. In these cases, (2.13) can be reformulated as a CVRP without fixing edges by using artificial customer locations as follows. Contract each path in S_1 from root to a leaf into a node. Let V_{CR} be the set of these contracted nodes. The costs of using edges connecting any two vertices in $\hat{V} \bigcup \{0, n+1\}$ remain unchanged. For $i \in V_{CR}$ let c_{0i} be equal to the costs of traversing the edge starting at the depot and ending at the first location on the path contracted into i. Similarly, for $j \in \hat{V} \bigcup \{0\}$ let c_{ij} be the costs of traversing the edge starting at the last location on the path contracted into i and ending at location j. Furthermore, for $i \neq 0$ and $j \in V_{CR}$ let $c_{ij} = \infty$.

For each $i \in V_{CR}$, let q_i be the total demand of the locations on the path contracted into i. Clearly, after the first phase of the heuristic, the demand q_i for $i \in V_{CR}$ does not exceed the vehicle capacity Q. The demand for the locations in $\hat{V}$ does not change.

Consider a solution to the CVRP problem defined on the complete graph with the set of customer locations $V_{CR} \bigcup \hat{V}$ with demand and costs as defined above. As the costs of using an edge between any location in $\hat{V}$ and any vertex in V_{CR} is infinite, in an optimal schedule any node in V_{CR} will be preceded only by the depot. A feasible schedule to the VRSP is now found by expanding back all the contracted nodes.

The problem that has to be solved in the second phase of the heuristic is obviously an NP-hard problem as the CVRP can be reduced to it. Fortunately, in most practical cases, the size of this CVRP is small. This is due to the fact that the number of nodes is equal to the number of routes in the master schedule (these are the artificial nodes) plus the number of isolated nodes, which is relatively low.

Next we present some properties of the solution obtained by the two phase heuristic.

2.3.3 Properties of the two-phase heuristic

Consider the problem of finding a feasible schedule that minimizes the total deviation costs when a master schedule S_M is given:

$$U^* = \min_{S \in \$} \sum_{i \in V'} U(S_M, S, i) \tag{2.14}$$

In the next proposition it is shown that when the deviation costs are non-increasing on each route, i.e. $u_i \geq u_j$ for $(i, j) \in S_M$, the deviation costs of the schedule obtained by the two-phase heuristic are equal to U^* and that the number of rescheduled locations in this schedule is minimal.

Proposition 2.1. *If the deviation costs are decreasing on each route, the two-phase heuristic produces a feasible schedule S^{TP} such that the number of rescheduled locations and the total deviation costs are minimized.*

Proof. For a feasible schedule S, denote by V_S the set of rescheduled locations. We will next show that the minimum number of rescheduled locations is $|\hat{V}|$, by proving that $\hat{V} \subseteq V_S$ for any feasible schedule S. Consider a location $j \in \hat{V}$ and the route $r = \{i_1, ..., i_j, ..., i_k\}$ in S. If $i_j \notin V_S$, none of the locations $i_1, ..., i_j$ are rescheduled and it must hold that $\sum_{l=1}^{j} q_{i_l} \leq Q$. However, this contradicts the construction of $\hat{V}$, hence

$\hat{V} \subseteq V_S$. As $\hat{V}$ is the set of rescheduled locations in S^{TP}, $|\hat{V}|$ is the minimum number of rescheduled locations.

Since $\hat{V} \subseteq V_S$, any schedule S deviates from the same locations onward as the schedule S^{TP} or from earlier locations. As the deviation costs are decreasing on each route, the two-phase heuristic provides a schedule such that the deviation costs are minimized.

□

Note that that in order to determine the minimum number of rescheduled locations and the minimum deviation costs, one can apply the procedure described in the first phase of the heuristic, which does not require the construction of a schedule.

For certain values of the parameters, the schedule that minimizes the total deviation costs achieves the optimal value for the VRSP. This is in particular the case when the costs of deviating are very large relative to the traveling costs. In such a case the two-phase heuristic produces the optimal schedule for the VRSP, as stated in the next proposition. For notational convenience, let $u_{n+1} = 0$ and $u_i^\delta = u_i - u_j$ for all (i, j) in the master schedule.

Proposition 2.2. *Let* $u_{\min} = \min_{i \in V'} u_i^\delta$, $c_{\min} = \min_{(i,j)\in E} c_{ij}$ *and let* S^{TP} *be the schedule produced by the two-phase heuristic. If*

$$u_{min} \geq \sum_{i \in V'} [c_{0i} + c_{i,n+1}] - (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil) c_{\min},$$

then the schedule S^{TP} *is optimal.*

Proof. For any schedule S that is a feasible solution to the VRSP, let $Z_S = \sum_{(i,j)\in S} c_{ij}$ denote the traveling costs and $U_S = \sum_{i \in V'} U(S_M, S, i)$ the deviation costs.

Note that $Z_{S^{TP}} \leq \sum_{i \in V'} [c_{0i} + c_{i,n+1}]$. Since $\lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil$ is a lower bound on the number of vehicles that are needed, for every feasible solution S to the VRSP it holds that $Z_S \geq (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil) c_{\min}$. Therefore,

$$Z_{S^{TP}} \leq Z_S + \sum_{i \in V'} [c_{0i} + c_{i,n+1}] - (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil) c_{\min}.$$

Furthermore, $U_{S^{TP}} = U^*$. If S has at least one more rescheduled location than $\hat{V}$, then $U^* + u_{\min} \leq U_S$. Hence, for $u_{min} \geq \sum_{i \in V'} [c_{0i} + c_{i,n+1}] - (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil) c_{\min}$, it follows that

$$Z_{S^{TP}} + U^* \leq Z_S + U_S. \tag{2.15}$$

If S and S^{TP} deviate from the same locations onward, (2.15) is always satisfied. This proves the optimality of S^{TP} for all instances with $u_{min} \geq \sum_{i \in V'}[c_{0i} + c_{i,n+1}] - (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil)c_{\min}$.

□

Tight Example:

To show that the bound on $u_{\min}$ cannot be improved, consider the following example. Let $V = \{0, 1, 2\}$ and $c_{ij} = c$, for all $i, j \in V$. The master schedule S_M consists of a return trip to location 1 and a separate return trip to location 2. The demand realizations are such that $q_1 + q_2 \leq Q$ and therefore the two routes might feasibly be merged. Furthermore, let $u_1 = u_2 = u_{\min}$. There are only three feasible solutions to this VRSP. The master schedule can be used as a solution to the rescheduling problem and yields a total cost of $4c$. The other two solutions visit nodes 1 and 2 on one route and both have costs $3c + u_{min}$. The two-phase heuristic will produce S_M. The resulting schedule will be optimal if and only if $u_{min} \geq c = \sum_{i \in V'}[c_{0i} + c_{i,n+1}] - (n + \lceil \frac{\sum_{i \in V'} q_i}{Q} \rceil)c_{\min}$.

For specific problem instances the relative difference between the traveling and the deviation costs need not be high for the two-phase heuristic to produce the optimal solution. However, we are not able to provide a general guarantee for small relative differences. In section 2.4, we analyze the impact of the numerical values of the deviation costs on the optimality of the solution obtained by the two-phase heuristic.

In the next proposition a worst case bound is provided on the ratio of the solution value of the solution provided by the two-phase heuristic and the optimum.

Proposition 2.3. *The costs of using the routing schedule S^{TP} produced by the two-phase heuristic is at most $\min\{\frac{Q}{q_{\min}}, \frac{2Qc_{\max}}{(Q+\bar{q})c_{\min}} + 1\}$ times the costs of the optimal schedule S^* for the VRSP, where $q_{\min} = \min_{j \in V'} q_j$, $\bar{q} = \frac{\sum_{i \in V'} q_i}{n}$, $c_{\max} = \max_{(i,j) \in E} c_{ij}$ and $c_{\min} = \min_{(i,j) \in E} c_{ij}$.*

Proof. Let the traveling costs and deviation costs of S^{TP} be given by $Z_{S^{TP}}$ and U^* respectively. Similarly, let the traveling and deviation costs of S^* be given by Z_{S^*} and U_{S^*}. Furthermore let $Z^* = \min_{S \in \mathbb{S}} \sum_{(i,j) \in S} c_{ij}$. To prove the theorem, it is first shown that $\frac{Z_{S^{TP}} + U^*}{Z_{S^*} + U_{S^*}} \leq \frac{Q}{q_{\min}}$ and secondly that $\frac{Z_{S^{TP}} + U^*}{Z_{S^*} + U_{S^*}} \leq \frac{2Qc_{\max}}{(Q+\bar{q})c_{\min}} + 1$. For ease of notation, assume $c_{0i} = c_{0,n+1}$.

In Simchi-Levi et al. (2005, page 220) it is proven that for the CVRP it holds that $2\sum_{j \in V'} c_{0j} q_j \leq QZ^*$. Now observe that:

$$Z_{STP} \leq 2\sum_{i\in V'} c_{0i} \leq \frac{2}{q_{\min}}\sum_{i\in V'} c_{0i}q_i \leq \frac{Q}{q_{\min}}Z^*,$$

which implies that:

$$\frac{Z_{STP}+U^*}{Z_{S^*}+U_{S^*}} \leq \frac{\frac{Q}{q_{\min}}(Z^*+U^*)}{Z^*+U^*} \leq \frac{Q}{q_{\min}}.$$

Next, as $\lceil\frac{\sum_{i\in V'} q_i}{Q}\rceil$ is a lower bound on the number of vehicles that are used, it follows that:

$$\frac{Z_{STP}+U^*}{Z_{S^*}+U_{S^*}} \leq \frac{Z_{STP}+U^*}{Z^*+U^*} < \frac{2nc_{\max}}{(n+\lceil\frac{\sum_{i\in V'} q_i}{Q}\rceil)c_{\min}}+1 \leq \frac{2nc_{\max}}{(n+\frac{\sum_{i\in V'} q_i}{Q})c_{\min}}+1 = \frac{2Qc_{\max}}{(Q+\bar{q})c_{\min}}+1$$

Here the strict inequality follows from $\frac{a+b}{c+b} < \frac{a}{c}+1$ for $a, b, c > 0$. This concludes the proof.

□

Tight Example:

In this example we show that the bound provided in Proposition 2.3 can not be improved upon. Consider a problem instance of n locations and let an arbitrary master schedule be given. Now let demand be given by $q_i = Q$ for all $i \in V'$. Obviously there is only one feasible schedule, hence $\frac{Z_{STP}+U^*}{Z_{S^*}+U_{S^*}} = 1 = \frac{Q}{q_{\min}} = \min\{\frac{Q}{q_{\min}}, \frac{2Qc_{\max}}{(Q+\bar{q})c_{\min}}+1\}$.

2.4 Computational results

In this section, results of numerical experiments are presented to provide insight into the sensitivity of the model with respect to different values of the deviation costs. Furthermore, the performance of the two-phase heuristic is evaluated empirically by applying it to several test cases.

The following settings are used for the generation of individual problem instances:

- n customer locations are randomly generated according to a uniform distribution over a square with sides of length 20 units. The depot is situated in the center of the square.
- The traveling costs between two locations are equal to the Euclidean distance between them.

- All vehicles have a capacity of 60 units.
- Presumed demand is normally distributed with mean 5 and standard deviation 1.5, truncated from below to 1 and from above to 60.
- Actual demand per location is normally distributed with standard deviation 1.5 and the demand average equal to 1.5 times the realization of the presumed demand, also truncated from below to 1 and from above to 60

For each experiment, we indicate the corresponding deviation costs. For every problem instance, first a master schedule S_M is generated by solving a CVRP using presumed demand. This is done either exactly or heuristically depending on the experiment at hand. Next a demand realization is generated to represent actual demand. The deviation costs will be specified for every individual experiment. As actual demand will typically be higher than presumed demand in our experiments, deviations from the master schedule will most often be necessary. These instances are inspired by a practical case in a retail chain with recurrent sales actions.

We have implemented the branch-and-cut algorithm by Baldacci et al. (2004) to solve the CVRP to optimality. It uses their two-commodity flow formulation to find lower bounds. These are strengthened by separating capacity cuts using a greedy randomized algorithm. This algorithm is used to generate the master schedule in some instances and to solve the CVRP in the second phase of the two-phase heuristic. We use the same algorithm to solve the VRSP to optimality by adding constraints (2.8), (2.9) and (2.12) to the formulation. For the instances where the master schedule is found by solving a CVRP heuristically, we use the savings algorithm by Clarke and Wright (1964).

All experiments are performed on an Pentium(R) Dual-Core CPU, E5800, 3.2GHz with 4.00GB of RAM. The branch-and-cut algorithm makes use of ILOG CPLEX 12.3 to solve the linear programming relaxations.

2.4.1 Impact of deviation costs

Recall that, by Proposition 2.2, for large values of $u_{\min}$ with respect to the traveling costs, the two-phase heuristic gives the optimal solution. It is, thus, interesting to study whether optimality is also obtained for lower values of u_{min}. We will refer to the lowest value of u_{min} for which the two-phase heuristic produces the optimal schedule as the *critical level* and we will denote it by $u_{critical}$.

Next we will argue that in order to analyze $u_{critical}$ it is sufficient to look at the number of rescheduled locations in an optimal schedule. Note that the minimal number of rescheduled locations can easily be determined by applying the first phase of the rescheduling heuristic and, by Proposition 2.1, the solution of the two phase heuristic has the minimal number of rescheduled locations.

If an optimal solution has a minimal number of rescheduled locations, the locations that deviate in both the optimal solution and the solution provided by the two-phase heuristic are identical. Since the two-phase heuristic inserts the deviating locations such that the traveling costs are minimized, the schedule produced by the two-phase heuristic must be optimal. Hence, in order to assess the optimality of the schedule generated by the two-phase heuristic, it is sufficient to look at the number of deviations in the optimal schedule.

For the numerical experiments in this paragraph we use the following deviation costs. For each route r in the master schedule and costs u, we assign deviation costs $(|r|+1-i)u$ to the ith location on r, where $|r|$ indicates the number of locations visited by r. Hence, $u_{min} = u$. We generate an instance with $u = 0$ and solve it. Next, we repeatedly modify the deviation costs by increasing the value of u by 0.125, and solve the modified instance. We repeat this until the two-phase heuristic provides the optimal solution for a modified instance.

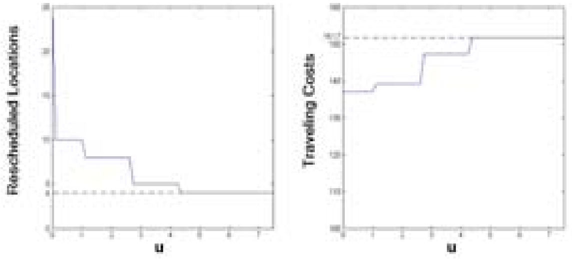
Let us first look at the value of $u_{critical}$ for an example. Consider a single randomly generated instance of 25 customer locations. For this example, the upper bound on $u_{critical}$ given in Proposition 2.2 is equal to 349.65. As remarked in Section 2.3.3, the minimum number of rescheduled locations can be found beforehand by applying the first phase of the two-phase heuristic. The optimal schedules are found using a direct implementation of the two commodity flow formulation of the VRSP.

In Figure 2.2 the number of rescheduled locations and traveling costs in the optimal solution are depicted, for different values of u. Notice that when $u = 0$, all locations are rescheduled. However, as u grows slightly above 0, a new schedule is found with less rescheduled locations but with equal traveling costs.

Figure 2.2 illustrates the fact that the number of rescheduled locations in the optimal schedule decreases as u grows. In this particular instance, the minimum number of rescheduled locations is 4 and the critical value is 4.25, a much lower value than the theoretical upper bound. However, the value of $u_{critical}$ is meaningless unless related to the traveling costs. Let $\bar{c}_M$ be the average of the traveling costs over the edges used in the master schedule. For the example depicted in Figure 2.2, $\bar{c}_M = 4.13$, which can be considered very close to the critical level u of 4.25.

We have repeated this experiment for 100 randomly generated instances. In 29 of the cases, $u_{critical}$ lies below $0.5\bar{c}_M$, in 61 cases below $\bar{c}_M$ and in 82 cases below $1.5\bar{c}_M$. Observe

Figure 2.2: Number of rescheduled locations and traveling costs



that 53 of the critical levels do not differ more than 50% of the value of $\bar{c}_M$. When we calculate the bound derived in Proposition 2.2, the two-phase heuristic could only have been guaranteed to generate the optimal schedule for $u \geq 102.4\bar{c}_M$ on average.

Finally, we discuss the tradeoff between transportation costs and the number of rescheduled locations. The schedules with minimal traveling costs in the first example, have a traveling cost equal to 137.1. Among these schedules the best in terms of number of rescheduled locations, is a solution with 10 rescheduled locations. The schedule with minimal number of rescheduled locations, 4, has a traveling costs equal to 151.7. Observe that in this case, the number of rescheduled locations decreases by 6 while the traveling costs increase by 10.6%. For the 100 generated cases, the average increase in traveling costs between the schedule with minimal traveling costs and the schedule with minimal number of rescheduled locations is 10.8%, with standard deviation 5.1.

2.4.2 Algorithm performance

The performance of the two-phase heuristic is evaluated on multiple test instances. For these cases, deviation costs decreasing in locations per route are obtained by generating a positive cost decrease u_i^δ for each customer $i \in V'$. We use a normal distribution with mean equal to either $0.25\bar{c}_M$ or $0.75\bar{c}_M$ and a standard deviation of $0.5\bar{c}_M$. The cost decreases are truncated from below at 0. These parameters were chosen such that it is unlikely that the generated instances either revert to standard CVRP because all u are near or equal to 0, or that they are sufficiently high so that the two-phase heuristic is guaranteed optimal.

The performance of the two-phase heuristic is compared to solving the VRSP to optimality using the branch-and-cut algorithm. For each instance a time limit of one hour is maintained for both the heuristic and the exact algorithm.

In Tables 2.1 and 2.2 the results of computational experiments for instances of different sizes are presented. The master schedule is generated by solving a CVRP to optimality. For each value of n, representing the number of customer locations, 50 instances were generated. The instances used for Table 1 have average deviation cost decreases equal to $0.25\bar{c}_M$ and the ones used for Table 2.2 have average cost decrease equal to $0.75\bar{c}_M$.

Column $BL2.2$ shows the average value of the theoretical bound on $u_{\min}$ in Proposition 2.2, and the standard deviation in between brackets. Column $BL2.3$ presents the average worst case bound as described in Proposition 2.3, expressed in percentages, and the standard deviation in between brackets. Column CD_{TP} shows, in percentages, the average difference between the cost of the schedule produced with the two-phase heuristic and the cost of the optimal schedule, and the standard deviation in between brackets. Note that these costs include both the traveling costs and the deviation costs. Finally, the values in T_{opt} and T_{TP} represent average running times in seconds of the exact algorithm and the two-phase heuristic respectively. As a master schedule was assumed to be given, the time needed to generate it is not incorporated. The column OPT not found indicates the number of instances out of 50, for which the optimal solution was not found within a one hour time limit. These instances were not considered in the presented averages.

Table 2.1: Deviation costs average $0.25\bar{c}_M$

n	$BL2.2$	$BL2.3$	CD_{TP}	T_{opt}	T_{TP}	OPT not found
10	141.4(20.5)	2022(1230)	2.9(4.4)	0.13	0.01	0
20	294.2(28.0)	2795(1413)	2.4(2.7)	7.92	0.05	0
30	440.5(28.9)	3835(1724)	3.0(2.8)	256.83	0.37	2
40	587.5(39.6)	3960(1637)	4.9(3.9)	1483.44	1.84	37

Table 2.2: Deviation costs average $0.75\bar{c}_M$

n	$BL2.2$	$BL2.3$	CD_{TP}	T_{opt}	T_{TP}	OPT not found
10	141.4(20.5)	2022(1230)	0.9(2.1)	0.235	0.01	0
20	294.2(28.0)	2795(1413)	1.0(1.6)	17.636	0.06	0
30	440.5(28.9)	3835(1724)	1.2(1.5)	481.763	0.43	4
40	587.5(39.6)	3960(1637)	1.0(0.7)	2436.45	1.16	45

In these experiments, the total costs of the schedules generated by the two-phase heuristic are on average not more than 2.9% above the optimum in Table 2.1 and not more than 1.0% in Table 2.2. As can be seen in column $BL2.3$, this differs significantly from the theoretical performance bound in Proposition 2.3. As expected, the solutions produced by the two-phase heuristic are on average closer to the optimum for the instances with average cost decreases of $0.75\bar{c}_M$ than for the instances with average cost decreases of $0.25\bar{c}_M$.

From the columns indicating running times it can be concluded that using the two-phase heuristic reduces the solution time significantly with respect to solving it to optimality using the branch-and-cut algorithm.

Out of the 39 instances with low deviation costs for which the optimal solution is not found, a feasible solution was found for 14 instances using the branch-and-cut algorithm. The average difference in solution value with respect to the two-phase heuristic is 4.4%. The two-phase heuristic solved these instances using an average computation time of 262.17 seconds. Out of the 49 instances with high deviation costs for which the optimal solution is not found, a feasible solution was found for 23 instances using the branch-and-cut algorithm. The average difference in solution value with respect to the two-phase heuristic is −0.7%. The two-phase heuristic solved these instances using an average computation time of 3.32 seconds.

2.4.3 Impact of the master schedule

Using an inefficient master schedule with respect to traveling costs, might affect the performance of the two-phase heuristic. When rescheduling using an inefficient master schedule, the traveling costs might be considerably reduced at the expense of deviating early in a route. In such cases, the two-phase heuristic will not perform well as it never generates unnecessarily early deviations. To investigate the effect of the quality of the master schedule on the performance of the two-phase heuristic, the experiment is repeated using instances where the master schedule is obtained by solving a CVRP heuristically using the savings algorithm by Clarke and Wright (1964). Tables 2.3 and 2.4 show the results of these experiments.

Table 2.3: Deviation costs average $0.25\bar{c}_M$, heuristic master schedule

n	$BL2.2$	$BL2.3$	CD_{TP}	T_{opt}	T_{TP}	OPT not found
10	137.3(18.5)	1863(745)	3.4(6.3)	0.142	0.011	0
20	290.5(27.4)	2645(1143)	2.7(3.4)	2.686	0.056	0
30	443.1(32.7)	3925(1778)	4.5(3.5)	232.103	0.425	0
40	598.1(42.5)	4135(1807)	4.6(3.3)	939.925	3.567	22

Table 2.4: Deviation costs average $0.75\bar{c}_M$, heuristic master schedule

n	$BL2.2$	$BL2.3$	CD_{TP}	T_{opt}	T_{TP}	OPT not found
10	137.3(18.5)	1863(745)	1.7(4.5)	0.198	0.011	0
20	290.5(27.4)	2645(1143)	1.3(2.3)	3.768	0.049	0
30	443.1(32.7)	3925(1778)	1.8(2.4)	493.469	0.489	3
40	598.1(42.5)	4135(1807)	1.5(1.5)	1187.79	2.814	30

For the instances used in Tables 2.3 and 2.4, more are solved within the one hour time limit than for the instances in Tables 2.1 and 2.2. Moreover, the average computation time of the exact method is lower. Nevertheless, the computation time of the two-phase heuristic is slightly higher for the instances where the master schedule is generated heuristically. The difference between the optimal solution value and the value of the solution produced by the two-phase heuristic is not significantly different for the instances where the master schedule is generated heuristically.

2.4.4 Number of deviating locations

The minimum number of deviating locations is an important factor for the speed of the two-phase heuristic. In the second phase a CVRP with the number of customers roughly equal to the minimum number of deviating locations should be solved. It depends on the instance at hand what the minimum number of deviations is. The average minimum number of deviating locations of the instances used in section 4.2 can be found in Table 2.5.

Table 2.5: Minimum number of deviating locations

n	Exact CVRP master schedule	Heuristic CVRP master schedule
10	2.4(1.0)	2.0(1.1)
20	4.6(1.4)	5.0(1.4)
30	8.8(2.0)	8.2(2.0)
40	14.7(2.4)	12.0(3.1)

In our implementation of the two-phase heuristic, we use an exact method to solve the reduced CVRP in the second phase. Many heuristic algorithms exist that can be used to deal with for instance large scale CVRP instances. When the size of the second phase CVRP grows too large for exact algorithms to be useful, such a heuristic can be employed. As the performance of existing exact algorithms and heuristics directly translate to our setting, we refer the interested reader to Laporte (2007). For a software library of heuristic methods for the CVRP that can be used in the second phase of the heuristic see for instance Groër et al. (2010).

2.5 Conclusion

In this chapter, the negative effects of deviating from a master schedule have been incorporated in a vehicle routing model, hence introducing the VRSP. Insight has been obtained into the behavior of the optimal solution of the VRSP for different values of the deviation costs relative to the traveling costs. We have formulated this problem that allows existing CVRP algorithms to be used to solve the VRSP after slight modifications.

Furthermore, we propose a two-phase heuristic that is capable of finding good solutions within a small amount of computation time. We have proven that this algorithm generates an optimal schedule when deviation costs are sufficiently high. Even when the deviation costs are not as high as required by our proposition, numerical experiments show that solutions of the two-phase heuristic are on average close to optimal. Moreover, for general problem instances, an analytical bound on the difference between the solution generated by the two-phase heuristic and the optimum is presented. Numerical results indicate, however, that this analytical upper bound is extremely far from the actual difference. As in the second phase of this heuristic an instance of the CVRP, an NP-hard problem, has to be solved for the locations that need to be rescheduled, the computation time heavily depends on the number of isolated vertices after the first phase of the heuristic.

Chapter 3

The Time Window Assignment Vehicle Routing Problem

3.1 Introduction

In many distribution networks deliveries are made at regular intervals and take place within a scheduled time window. Typically, these time windows are endogenously imposed. The supplier and customer might for instance agree on a specific time window for delivery. These endogenous time windows are long term decisions in certain industries. In retail it is very common that deliveries at a store are always made within a specific time interval on the same day of the week for an entire year. This is crucial for many operational processes like inventory management and the scheduling of personnel. Distribution networks are often also faced with exogenous time windows. For example, an exogenous time window might be imposed by a local government which forces trucks to make deliveries in a populated area during day time only. Hence an endogenous time window can only be chosen within the exogenous time window.

Demand is usually unknown at the moment that endogenous time windows are assigned and most often fluctuates per delivery. When demand of the customers becomes known, a vehicle routing schedule has to be determined for making the deliveries within the endogenous time windows. This problem is known as the vehicle routing problem with time windows, VRPTW, a well studied problem in the scientific literature, see for instance the surveys by Baldacci et al. (2012) and Kallehauge et al. (2005).

In this chapter a model is presented to assign time windows before demand is known. The problem of assigning time windows will be referred to as the time window assignment vehicle routing problem, TWAVRP. A finite number of scenarios is given, each scenario describing a realization of demand for each location. Furthermore, the probability with which each scenario occurs is known at the moment of scheduling. The TWAVRP consists of assigning a time window to each customer and constructing a vehicle routing schedule for each scenario satisfying these time windows, such that the expected costs are minimized. The TWAVRP is NP-hard as for one scenario it is the VRPTW.

The TWAVRP is similar to the consistent vehicle routing problem, ConVRP, introduced by Groër et al. (2009). The TWAVRP and ConVRP differ in the following requirements. In the ConVRP, a customer does not necessarily require service in each scenario. Also, each customer needs to be visited by the same driver in all scenarios in which it requires service. Finally, there are restrictions on the total driving time of each driver. The ConVRP has applications in small-package shipping. Our experience with specific Dutch retail chains suggests that the ConVRP is too stringent for application in their case. Here, every customer requires goods on each day of delivery. Furthermore, personalization of service is not an issue, as both the supplier and customers are part of the same retail chain, hence, the same-driver condition is not necessary. Finally, delivery routes will (almost) never exceed the maximum allowed driving time as capacity constraints in many retail settings prohibit long routes, unlike in the small package shipping industry to which Groër et al. (2009) applied the ConVRP. In the paper by Groër et al. (2009) computational experiments are provided with ConVRP instances with up to 12 customers and 3 scenarios. They solve these instances to optimality using a commercial mixed integer program solver. They report computation times of up to several days. Furthermore, they develop a local search heuristic to find solutions to instances of up to 3715 customers and 25 scenarios.

A closely related problem to the TWAVRP is the vehicle routing problem with stochastic demand, SVRP. In this problem, vehicle routes are determined before demand is known. When demand becomes known, routes may be infeasible due to the limited capacity of each truck in which case a recourse action is required. Among the most popular recourse actions is the strategy suggested by Dror et al. (1989), in which the original vehicle route is followed until the load of the truck is depleted, and is resumed after a visit to the depot to restock. An advantage of this recourse action is that the expected costs of a vehicle route can be computed efficiently. Studies on this model include the work by Laporte et al. (2002) who solve the SVRP to optimality using an integer L-shaped method and Novoa and Storer (2009) who develop an approximate dynamic programming approach for the single vehicle variant. In this model, however, the time of service at the customer is not taken into account.

Another closely related problem is considered by Jabali et al. (2010). In their paper, demand is assumed to be known at the moment of assigning the time windows and travel time is stochastic. They develop a tabu search algorithm to find good solutions for this problem. Furthermore, Agatz et al. (2011) consider the problem of deciding which time slot to offer to customers in different zip code areas for a web store offering home deliveries. They propose a local search heuristic.

In this chapter, we propose a relevant new problem which we have encountered in practice, the TWAVRP. We develop a column generation algorithm to find lower bounds on the optimal solution value of the TWAVRP. We apply route relaxation to allow cyclic routes and apply the algorithm by Ioachim et al. (2008) to solve the pricing problem. Furthermore, to strengthen the lower bound we eliminate routes containing 2-cycles and modify the algorithm by Ioachim et al.(2008) accordingly. We incorporate this column generation algorithm into a branch-price-and-cut algorithm to find optimal integer solutions to the TWAVRP. We show by means of computational experiments that this algorithm is capable of solving instances with up to 25 customers and 3 scenarios to optimality within one hour of computation time. Finally, as is frequently done in practice, we construct a solution by solving a VRPTW using average (historic) demand and using the arrival times at each customer as reference points for the endogenous time windows. We compare the solutions obtained in this fashion with the optimal solution value of the TWAVRP to offer insight in the value of an exact approach for the TWAVRP.

This chapter is organized as follows. A formal definition of the TWAVRP is given in Section 3.2. In Section 3.3, the branch-price-and-cut algorithm is presented. The results of our computational experiments are provided in Section 3.4. The chapter ends with a short conclusion.

3.2 Problem definition

Consider a complete graph $G = (V, A)$, where $V = \{0, ..., n + 1\}$ is a set of locations such that 0 represents the starting depot, $n + 1$ the ending depot and $V' = \{1, ..., n\}$ are the customers. Let $c_{ij} \geq 0$ be the cost to travel along arc (i, j) and let $t_{ij} \geq 0$ be the corresponding travel time. Both the travel costs and travel times satisfy the triangle inequality. Furthermore, an unlimited number of vehicles of equal capacity Q is available.

Let Ω be a set of scenarios, where each scenario represents a realization of demand. The probability that scenario ω occurs is p_ω. Let demand at location v in scenario $\omega \in \Omega$ be given by d_v^ω where $0 < d_v^\omega \leq Q$. For ease of notation, let $d_0^\omega = d_{n+1}^\omega = 0$.

Associated with each location $v \in V$ is the exogenous time window $[s_v, e_v]$, which is not to be confused with the endogenous time window.

In this chapter we will use the term route to refer to a pair (P, t) where P is a path in G starting at 0 and ending at $n+1$ and t is a vector containing arrival times at each location on the path. Let a_r^v be the number of times customer $v \in V'$ is visited by route r. Furthermore, let t_r^v be the cumulative time of service of customer $v \in V'$, i.e., if location v is not visited $t_r^v = 0$, if it is visited once, t_r^v is the time of service, and if customer v is visited multiple times t_r^v is the sum of the times of service. To each route r with arcs $\{r_1, ..., r_k\}$ we assign costs $c_r = \sum_{i=1}^{k} c_{r_i}$.

A route is considered feasible for scenario ω if i) the capacity constraint in scenario ω is satisfied, ii) the exogenous time window constraints are satisfied, and iii) the service time at location j is not before the service time at location i plus the travel time t_{ij} if location j is visited directly after i. Note that waiting at a customer is allowed. Let $R(\omega)$ be the set of all feasible routes for scenario ω.

An endogenous time window of width w_v has to be assigned to each customer $v \in V'$ within which it will receive its delivery. The assignment is made before the realization of demand is learned. Prior to the dispatching of the vehicles, demand becomes known and an optimal routing schedule will be designed to make the deliveries within the assigned time windows. The TWAVRP is to assign time windows before demand is known and selecting feasible routes in each scenario $\omega \in \Omega$ that satisfy these time windows. The objective is to minimize the expected traveling costs.

Next we provide a mixed integer linear programming formulation for the TWAVRP. Let the time window variable y_v be the start time of the endogenous time window at each location $v \in V'$. Note that $y_v \in [s_v,\, e_v - w_v]$. We will assume $s_v \leq e_v - w_v$. Let the binary route variable x_r^ω indicate whether route r is used for scenario ω. The TWAVRP can be formulated using the following mixed integer linear program.

$$\min \sum_{\omega\in\Omega} p_\omega \sum_{r\in R(\omega)} c_r x_r^\omega \tag{3.1}$$

$$\sum_{r\in R(\omega)} a_r^v x_r^\omega = 1 \qquad \forall v \in V', \forall \omega \in \Omega \tag{3.2}$$

$$\sum_{r\in R(\omega)} t_r^v x_r^\omega \geq y_v \qquad \forall v \in V', \forall \omega \in \Omega \tag{3.3}$$

$$\sum_{r\in R(\omega)} t_r^v x_r^\omega \leq y_v + w_v \qquad \forall v \in V', \forall \omega \in \Omega \tag{3.4}$$

$$x_r^\omega \in \{0,1\} \qquad \forall \omega \in \Omega, \forall r \in R(\omega) \tag{3.5}$$

$$y_v \in [s_v, e_v - w_v] \qquad \forall v \in V' \tag{3.6}$$

Here (3.1) are the expected total costs of a time window assignment. Constraints (3.2) ensure that every location is visited exactly once and constraints (3.3) and (3.4) ensure that all locations are visited within the assigned time windows. Finally, note that as time is continuous the number of routes in $R(\omega)$ for all $\omega \in \Omega$, and therefore also the number of variables, is infinite, unless $w_v = 0$ and $s_v = e_v$ for all $v \in V'$.

3.3 Solution method

In this section we propose a branch-price-and-cut algorithm to solve the TWAVRP. First, we present a column generation algorithm to find lower bounds by solving the LP relaxation of the TWAVRP formulated in (3.1)-(3.6). We consider two route relaxations, allowing the path of a route to be nonelementary. Moreover, we discuss the algorithm to solve the pricing problem, an acceleration strategy and the addition of valid inequalities. Finally we discuss the branch-price-and-cut algorithm.

3.3.1 Column generation algorithm

We propose using a column generation algorithm to solve the LP relaxation of (3.1)-(3.6), referred to as the master problem. We consider the master problem where only a subset of routes are included, also known as the restricted master problem. At each iteration of the column generation algorithm a restricted master problem is solved, followed by solving a pricing problem to identify feasible routes with negative reduced costs. Routes with negative reduced costs are added to the restricted master problem. If no such route exists, the current solution to the restricted master problem is optimal for the master problem.

We decompose the pricing problem into several problems, one for each scenario. For scenario ω, the pricing problem is to find a feasible route (P, t) such that P is elementary, with minimum reduced costs. Let us denote the dual variables corresponding to (3.2)-(3.4) by λ, μ and ν respectively. For ease of notation, let $\pi = \nu - \mu$. Observe that both λ and π are unrestricted. The reduced costs corresponding to route variable x_r^ω are given by

$$p_\omega c_r - \sum_{v \in V'} \lambda_v^\omega a_r^v - \sum_{v \in V'} \pi_v^\omega t_r^v. \tag{3.7}$$

We model the pricing problem for scenario ω using graph G. With each node $v \in V'$ we associate demand d_v^ω, time window $[s_v, e_v]$, and the cost coefficient $-\pi_v^\omega$. Furthermore, with each arc $(i, j) \in E$ we associate the travel time t_{ij}, and costs $p_\omega c_{ij} - \lambda_j^\omega$ if $j \in V'$ and $p_\omega c_{ij}$ otherwise. For each route (P, t) we calculate the corresponding reduced costs in scenario ω as the sum of the costs of the arcs on path P and the costs at each node v. These costs are linear in the arrival time t_v with coefficient $-\pi_v^\omega$. The pricing problem is solved by finding an elementary shortest path in G with a capacity constraint, time window constraints and linear node costs.

We consider this pricing problem to be very difficult to solve exactly. To the best of our knowledge, no algorithm is described in the current scientific literature to solve this problem. Instead, we suggest using route relaxations, i.e., allowing nonelementary routes, yielding a less complex pricing problem.

Observe that the optimal integer solution of the TWAVRP does not change when cyclic routes are used in the formulation. However, the LP-value will decrease. Route relaxation has been successfully used by for instance Desrochers et al. (1992) to solve the VRPTW, which is a closely related problem to the TWAVRP. They solve a pricing problem in which they allow cyclic routes. Moreover, they eliminate 2-cycles, i.e., cycles of the form $i - j - i$ to strengthen the LP bound with respect to allowing all cycles, at the cost of increased computational complexity. Other examples of route relaxations that provide stronger LP-bounds than allowing all cycles at the cost of increased computational complexity are $k-$cycle elimination for $k \geq 3$ as described by Irnich and Villeneuve (2006) and the ng-route relaxation as introduced by Baldacci et al. (2011). In this chapter, we consider allowing all cyclic routes and allowing all cyclic routes not including 2-cycles. Next, we present the algorithms used to solve the corresponding pricing problems.

3.3.2 Pricing problem with all cyclic routes

When all cyclic routes are allowed, the pricing problem is a shortest path problem with a capacity constraint, time window constraints and linear node costs. In order to solve this problem, we introduce an auxiliary acyclic graph $\hat{G}^\omega$ for each scenario ω in which only the paths in G with a total load less or equal to Q are represented. This allows us to solve the pricing problem of scenario ω by applying the algorithm by Ioachim et al. (1998) to the auxiliary graph $\hat{G}^\omega$. In this section, we formally define the auxiliary graph and discuss the labeling algorithm.

Let $\hat{G}^\omega = (\hat{V}^\omega, \hat{A}^\omega)$ be the auxiliary graph in scenario ω. The set $\hat{V}^\omega$ includes a node for i) the starting depot 0, ii) each triple (v, m, q) such that $v \in V'$, there exists a $(0, v)-$path in G visiting exactly m locations and with a total load of $q \leq Q$ in scenario ω, and iii) each pair $(n+1, q)$ such that there exists a $(0, n+1)-$path in G with a total load of $q \leq Q$. By construction, to each node $u \in \hat{V}^\omega$ there is a corresponding node in V denoted by $o(u)$. We refer to $o(u)$ as the original node corresponding to u. We also associate to each node u such that $o(u) = i$, demand $d_u^\omega = d_i^\omega$, time window $[\hat{s}_u, \hat{e}_u] = [s_i, e_i]$, and the linear node cost coefficient $\hat{c}_u$ which is $-\pi_i^\omega$ if $i \in V'$ and 0 otherwise.

The set $\hat{A}^\omega$ includes a) the arcs $(0, v)$ for every node $v \in \hat{V}^\omega$ representing a triple $(o(v), 1, d_{o(v)}^\omega)$, b) the arcs (v, u) where $v \in \hat{V}^\omega$ represents the triple $(o(v), m, q)$ and $u \in \hat{V}^\omega$ represents the triple $(o(u), m+1, q+d_{o(u)}^\omega)$, such that $(o(v), o(u)) \in A$, and c) the arcs (v, u), where $v \in \hat{V}^\omega$ represents the triple (v, m, q) and $u \in \hat{V}^\omega$ represents the pair $(n + 1, q)$. With each arc $(v, u) \in \hat{A}^\omega$ where $o(v) = i$ and $o(u) = j$, we associate the travel time $\hat{t}_{vu} = t_{ij}$ and costs $\hat{c}_{vu} = c_{ij} - \lambda_j^\omega$ if $j \neq n + 1$ and costs $\hat{c}_{vu} = c_{ij}$ if $j = n + 1$.

With each pair (P, t), where P is a path in $\hat{G}^\omega$ and t are service times at each node visited on P, we associate costs. These costs are computed as the sum of the costs of each arc traversed by P and the costs at each node $v \in \hat{V}^\omega$ visited by P equal to the service time t_v multiplied with the linear node cost coefficient $\hat{c}_v$.

Observe that there exists a bijection between the paths in G and $\hat{G}^\omega$. Moreover, the costs of corresponding paths in G and $\hat{G}^\omega$ are equal when the times of service coincide. Hence, when allowing all cycles, the pricing problem can be solved by finding a shortest path in $\hat{G}^\omega$, with a capacity constraint, time window constraints and linear node costs.

To solve the pricing problem we apply the algorithm by Ioachim et al. (1998) to $\hat{G}^\omega$. We describe this algorithm next. First note that even when a path P in $\hat{G}^\omega$ is given, as the linear node costs $\hat{c}_v$ of each node $v \in \hat{V}^\omega$ on the path can be positive or negative, determining the optimal times of service is an optimization problem in itself. To deal with this, Ioachim et al. (1998) introduce a node cost function $g^P(T)$ that provides the minimum costs of using path P where service at the last node in P is performed before time T. They show how to construct this function and prove that it is piecewise linear, convex and contains at most $|P|$ line pieces.

Consider the set of partial paths Π_v starting at the depot and ending at node $v \in \hat{V}^\omega$. Define the dominance function $D_v(T) = \min\{g^P(T)|P \in \Pi_v\}$ which provides the minimum costs of servicing node v before time T. Ioachim et al. (1998) prove that D_v is piecewise linear, non increasing but not necessarily convex or continuous. Next, we describe the dynamic programming algorithm they propose to construct this function.

Let f_v be the number of line pieces of D_v restricted to the interval $[\hat{s}_v, \hat{e}_v]$. We refer to the start and end points of these line pieces as the breakpoints $b_v^1, ..., b_v^{f_v+1}$. Line piece l_k, $1 \leq k \leq f_v$, of the dominance function can be represented by

$$l_v^k = (b_v^k, D_v(b_v^k), h_v^k), \tag{3.8}$$

where b_v^k is the start of the line piece, $D_v(b_v^k)$ is the value of the dominance function at the start of the line piece and h_v^k is the slope. The dominance function can be described using the set of line pieces $\{l_v^k|1 \leq k \leq f_v\}$. Note that $D_v(T)$ is not defined for $T < b_v^1$, and

$$D_v(T) = D_v(b_v^{f_v}) + h_v^{f_v}(b_v^{f_v+1} - b_v^{f_v}), \text{ if } T > b_v^{f_v}. \tag{3.9}$$

Every line piece corresponds to a label in the labeling algorithm. The label extension operator that is used to extend a label from node v to u, is defined as follows:

$$\begin{aligned} \text{EXTEND}_{vu}(l_v^k) = (\, & \max\{\hat{s}_u, b_v^k + \hat{t}_{vu}\}, \\ & D_v(b_v^k) + \hat{c}_{vu} + \hat{c}_u \max\{\hat{s}_u, b_v^k + \hat{t}_{vu}\}, \\ & \min\{0, h_v^k + \hat{c}_u\}) \end{aligned} \tag{3.10}$$

Note that extended labels with $\max\{\hat{s}_u, b_v^k + \hat{t}_{vu}\} > \min\{\hat{e}_u, b_v^{k+1} + \hat{t}_{vu}\}$ are removed, as the path corresponding to such a line piece does not satisfy the time window constraint.

We denote by $\text{EXTEND}_{vu}(D_v)$ the extension operator that provides the set of extended labels for each label l_v^k for $1 \leq k \leq f_v$, and an additional label if $b_v^{f_v+1} + t_{vu} < \hat{e}_u$. This additional label represents a line piece that provides the minimum costs for service commencing before T, for $T \in [b_v^{f_v} + \hat{t}_{vu}, e_u]$ and is defined by

$$l'_u = \begin{cases} (b_v^{f_v+1} + \hat{t}_{vu}, & \\ D_u(b_v^{f_v+1}) + \hat{c}_{vu} + \hat{c}_u(b_v^{f_v+1} + \hat{t}_{vu}), & \\ 0) & \text{if } b_v^{f_v+1} + t_{vu} < \hat{e}_u; \\ \emptyset & \text{otherwise.} \end{cases} \tag{3.11}$$

The extension operator on a dominance function is defined as

$$\text{EXTEND}_{vu}(D_v) = \{\text{EXTEND}_{vu}(l_v^k) | 1 \leq k \leq f_v\} \bigcup \{l'_u\}. \tag{3.12}$$

The set of labels $\text{EXTEND}_{vu}(D_v)$ describes a piecewise linear function. Let F denote the operator that finds the minimum of a set of piecewise linear functions represented by a set of labels, which we use to construct the dominance functions. Furthermore, let $\hat{V}^\omega$ be ordered as follows. First 0, next the nodes representing triples (v, m, q) in increasing order of m and ordered lexicographically in v and q, and finally the nodes representing the pairs $(n+1, q)$ ordered with respect to q. The labeling algorithm is summarized in Algorithm 3.1.

Algorithm 3.1 Labeling algorithm to solve the pricing problem

```
Initialize L_v = ∅ for all v ∈ V̂^ω.
Initialize l_0^1 = (ŝ_0, 0, 0), and f_0 = 1.
Initialize L_0 = {l_0^1}.
for all v ∈ V̂^ω do
  D_v = F(L_v).
  for all (v, u) ∈ Â^ω do
    Add EXTEND_vu(D_v) to L_u.
  end for
end for
```

This labeling procedure yields the dominance functions D_v for all $v \in \hat{V}^\omega$. Backtracking allows us to find the shortest paths corresponding to the labels of the dominance functions D_v for $o(v) = n+1$. In our experiments, we add all found routes with negative reduced costs. Hence, at each iteration of the column generation algorithm, multiple routes might be added to the restricted master problem for one scenario.

3.3.3 Pricing problem with 2-cycle elimination

To improve the LP-bound obtained when allowing all cyclic routes, we propose to eliminate 2-cycles. A 2-cycle $i-j-i$ in G is represented in $\hat{G}^\omega$ by a partial path $\hat{v}-\hat{v}'-\hat{v}''$ where $o(\hat{v}) = o(\hat{v}'')$. Next we discuss the modifications to the labeling algorithm to eliminate 2-cycles.

Let $\text{Pred}(l_v^k)$ be the predecessor of v on the path in $\hat{G}^\omega$ corresponding to line piece l_v^k. Similarly, let $\text{Pred}(P)$ be the node preceding the last node on path P in $\hat{G}^\omega$. For $T \in [b_v^k, b_v^{k+1}]$, define $D'_v(T) = \min\{g^P(T) | P \in \Pi_v, o(\text{Pred}(P)) \neq o(\text{Pred}(l_v^k))\}$, providing the minimum costs of servicing node v before time T, considering only the paths with a different original previous customer than $o(\text{Pred}(l_v^k))$. The main idea of the modified algorithm is to extend the path corresponding to $D'_v(T)$ at time T instead of the path corresponding to $D_v(T)$, when extending the latter would yield a 2-cycle.

For the labeling algorithm with 2-cycle elimination, we redefine the extension operator. We associate with every line piece l_v^k of D_v the line piece $l_v^{k'}$, $k' < k$, as the last line piece such that $b_v^{k'} \leq b_v^k$ with a different original predecessor customer, i.e., $o(\text{Pred}(l_v^k)) \neq o(\text{Pred}(l_v^{k'}))$. When extension of the path corresponding to l_v^k yields a 2-cycle $l_v^{k'}$ is used instead. In this case, the extended line piece represents service at v at time $b_v^{k'+1}$ and waiting to service u. Even though the resulting label will never be part of D_u, it might be part of D'_u. The new extension operator is defined as follows, using the same notation for line pieces of D'_v.

$$\text{EXTEND}'_{vu}(l_v^k) = \begin{cases} \text{EXTEND}_{vu}(l_v^k) & \text{if } o(\text{Pred}(l_v^k)) \neq o(u); \\ (\max\{\hat{s}_u, b_v^k + \hat{t}_{vu}\}, & \\ D_v(b_v^{k'+1}) + \hat{c}_{vu} + \hat{c}_u \max\{\hat{s}_u, b_v^k + \hat{t}_{vu}\}, & \\ \min\{0, \hat{c}_u\}) & \text{otherwise.} \end{cases} \tag{3.13}$$

Let F' denote the operator that finds D'_v. The labeling algorithm is summarized in Algorithm 3.2.

Algorithm 3.2 Labeling algorithm to solve the pricing problem with 2-cycle elimination

Initialize $L_v = \emptyset$ for all $v \in \hat{V}^\omega$.
Initialize $l_0^1 = (\hat{s}_0, 0, 0)$, and $f_0 = 1$.
Initialize $L_0 = \{l_0^1\}$.
for all $v \in \hat{V}^\omega$ **do**
    $D_v = F(L_v)$.
    $D'_v = F'(L_v)$.
    for all $(v, u) \in \hat{A}^\omega$ **do**
        Add $\text{EXTEND}'_{vu}(D_v)$ to L_u.
        Add $\text{EXTEND}'_{vu}(D'_v)$ to L_u.
    end for
end for

3.3.4 Acceleration strategy

The column generation algorithm requires solving a pricing problem for each scenario $\omega \in \Omega$ at every iteration. These pricing problems differ only in the values of the dual variables, the demand of each customer and the scenario probabilities that are part of the reduced costs. Therefore, we propose the following acceleration strategy. Whenever a route is found as a solution to the pricing problem of some scenario ω, it is also added to the restricted master problem in scenario ω' if it is feasible and has negative reduced costs in that scenario as well. In this case, the pricing problem of scenario ω' is not solved during this iteration. This procedure potentially reduces the number of pricing problems that have to be solved. The column generation algorithm is summarized in Algorithm 3.3.

Algorithm 3.3 Column Generation Algorithm, Reusing Routes

```
Initialize R(ω) for all ω.
repeat
  Solve the restricted master problem using the routes R(ω) for scenario ω.
  Set Ω̃ = Ω.
  while Ω̃ ≠ ∅ do
    Choose ω ∈ Ω̃ and remove it from Ω̃.
    Solve the pricing problem for scenario ω, to find a set of routes R.
    Add all routes in R that have negative reduced costs for scenario ω to R(ω).
    for All ω̃ ∈ Ω̃ do
      Let R̃ ⊆ R be all routes that are feasible and have negative reduced costs for
      scenario ω̃.
      if R̃ ≠ ∅ then
        Add all routes in R̃ to R_ω̃ and remove ω̃ from Ω̃.
      end if
    end for
  end while
until No new routes are added to the master problem.
```

3.3.5 Valid inequalities

To improve the LP-bound of the TWAVRP we add valid inequalities. In particular we consider inequalities that are valid for the vehicle routing problem, as they are also valid for each scenario in the TWAVRP. These inequalities include capacity, comb, hypotour and multistar inequalities (Lysgaard et al. (2004)). We have experimented with adding these inequalities using the separation routines of Lysgaard (2003). Preliminary experiments showed that adding only capacity inequalities yields the lowest computation time.

Next, we briefly discuss the capacity inequalities. Let z_{ij}^{ω} be the arc flow in G on arc (i,j) in scenario ω. Let $b(S)$ be the minimum number of vehicles needed to visit all customers in $S \subseteq V'$. The capacity inequalities are

$$\sum_{i \in S, j \notin S} z_{ij}^{\omega} \geq b(S) \qquad \forall S \subseteq V', \forall \omega \in \Omega. \tag{3.14}$$

As is common, we replace $b(S)$ by the lower bound $\left\lceil \frac{\sum_{i \in S} d_i^{\omega}}{Q} \right\rceil$. These constraints can be reformulated using the route variables x_r^{ω}. When capacity inequalities are added, the pricing problem remains a shortest path problem with a capacity constraint, time window constraints and linear node costs. However, the costs on each arc are modified as follows. Let σ_S^{ω} be the dual variable associated with the capacity inequalities for subset S in scenario ω. We subtract σ_S^{ω} from the initial costs of each arc $(i,j) \in \hat{A}$ such that $i \in S$ and $j \notin S$.

Other valid inequalities for the VRP and VRPTW which might also be applied here are the following. The $k-$path inequalities introduced by Kohl et al. (1999) and extended to generalized $k-$path inequalities by Desaulniers et al. (2008) have been used to solve the VRPTW successfully. These inequalities are strongest when capacity and time window constraints are tight. Since we focus on instances with wide exogenous time windows, we have chosen not to include these inequalities. Also, the subset row inequalities introduced by Jepsen et al. (2008) have been used to solve the VRPTW using a branch-price-and-cut algorithm. However, the pricing problem changes substantially when adding these inequalities, making it more difficult to solve. Therefore we have chosen not to include these inequalities.

3.3.6 Branch-price-and-cut

Next we describe the branch-price-and-cut algorithm we propose to solve the TWAVRP. Lower bounds are obtained by using the column generation algorithm to solve the LP-relaxation of (3.1)-(3.6) and adding capacity inequalities. In our implementation, capacity inequalities are only separated during the iterations of the column generation algorithm where no new routes with negative reduced costs are found.

With each feasible solution to the LP-relaxation of (3.1)-(3.6) we associate an arc flow in G for each scenario ω. Observe that an integer arc flow in each scenario corresponds to an integer solution of the TWAVRP, even when the route variables are fractional. In our branch-price-and-cut algorithm we perform special ordered subset (SOS) branching on the arcs as follows.

For scenario ω and customer v, let $\delta^-_\omega(v)$ and $\delta^+_\omega(v)$ be the sets of in and out arcs respectively. Next, a customer v', a scenario ω' and an arc type $o' \in \{-,+\}$ is selected with the highest number of arcs a in $\delta^{o'}_{\omega'}(v')$ for which $z^{\omega'}_a > 0$. Let $\delta^{o'}_{\omega'}(v') = \{a_1, ..., a_k\}$ be ordered such that $z^{\omega'}_{a_i} \geq z^{\omega'}_{a_j}$ if $i < j$. The arcs are divided into two groups, S and its complement $\bar{S}$, where $S = \{a_1, ..., a_i\}$ is such that $\sum_{a \in S} z^{\omega'}_a \geq 0.5$ and $\sum_{a \in S \setminus \{a_i\}} z^{\omega'}_a < 0.5$. In one branch we disallow the use of the arcs in S and in the other we disallow the use of the arcs in $\bar{S}$. Observe that the pricing problem remains a shortest path problem with a capacity constraint, time window constraints and linear node costs. However, less arcs are included in the graph.

Upper bounds are obtained when a solution with integer arc flow in each scenario is found to the LP-relaxation. At each iteration of the branch-price-and-cut algorithm, the node with the lowest lower bound is selected.

3.4 Computational results

In this section we present the results of numerical experiments using our algorithms. First, we discuss the test instances that we have generated. Next, we show results of solving the LP-relaxation for these instances, obtained by using the column generation algorithm. This is followed by the results of using the branch-price-and-cut algorithm. Finally, we compare the optimal solution value of the TWAVRP to the value of the solution found by solving a VRPTW with average demand, as is often done in practice. In all experiments, a one hour time limit is used.

All algorithms are coded in C++ and ILOG CPLEX 12.4 is used to solve the restricted master problem at each iteration of the column generation algorithm. The experiments were performed on an Intel(R) Core(TM) i5-2450M CPU 2.5 GHz processor.

3.4.1 Test instances

We have generated a total of 40 instances, consisting of 10 instances with 10, 15, 20 and 25 customers respectively[1]. These instances are inspired by Dutch retail chains.

Customer locations are generated using a uniform distribution over a square with sides of length 5. The depot is located in the center of the square. Both the travel costs and times are equal to the Euclidean distance between locations, rounded to two

[1] All instances are available on request.

digits. The depot has the exogenous time window $[6, 22]$. Each customer is given one of three exogenous time windows, each assigned with a fixed frequency. The exogenous time window $[10, 16]$ is given to 10% of the customers, $[8, 18]$ to 60% and $[7, 21]$ to 30%. The endogenous time window width is set to 2 for all customers. The vehicle capacity is 30.

For every instance, 3 scenarios are generated, each occurring with equal probability. To vary demand throughout the scenarios for each customer, we generate it by computing $d_v^\omega = \lceil u_v^\omega d_v \rceil$. Here, d_v is drawn from a normal distribution with an expectation of 5 and a variance of 1.5. Furthermore, for each $\omega \in \Omega$ the multiplier u_v^ω is drawn from a uniform distribution on the interval $[0.7, 0.8]$, $[0.95, 1.05]$ or $[1.2, 1.3]$, to generate scenarios with low, medium or high demand respectively.

3.4.2 Column generation results

Next, we provide the results of solving the LP-relaxation of (3.1)-(3.6) using the proposed column generation algorithm. We compare the two route relaxations considered in this chapter, allowing all cyclic routes and 2-cycle elimination.

Preliminary experiments suggest that the column generation algorithm employing the acceleration strategy of reusing routes, as summarized in Algorithm 3.3, is faster than without this acceleration strategy. Therefore, we only present results obtained using this algorithm.

The column generation algorithm is initialized by including single customer routes, i.e., routes of the form $((0, v, n+1), (t - t_{0v}, t, t + t_{v,n+1}))$, for each customer $v \in V'$ in each scenario, for different values of t in the exogenous time window. More precisely, we use the values $t = s'_v, s'_v + w_v, ..., s'_v + kw_v$, where $s'_v = \max\{s_v, s_0 + t_{0v}\}$ is the earliest possible arrival times at customer v, $e'_v = \min\{e_v, e_{n+1} - t_{v,n+1}\}$ is the latest possible arrival times at customer v, and $k = \lfloor \frac{e'_v - s'_v}{w_v} \rfloor$. This way, for every endogenous time window assignment, feasible routes are included in the restricted master problem in each scenario satisfying the assigned time windows.

Table 3.1 shows the results of the experiments using the column generation algorithm. In the first two columns, the instance and the number of customers in that instance are indicated. For each instance, we report the results obtained when allowing all cycles and when 2-cycles are eliminated. In the columns T.Time, the total computation time in seconds is reported. The columns P.Time report the total time in seconds spent on solving the pricing problems. The columns Iter. indicate the total number of iterations before termination. Finally, the columns LP contain the value of the LP-relaxation per instance for each route relaxation.

Table 3.1: Column generation results

		All cycles allowed				2-cycle elimination			
Inst.	$\|V'\|$	T.Time	P.Time	Iter.	LP	T.Time	P.Time	Iter.	LP
1	10	0.44	0.20	13	14.22	0.69	0.67	15	17.64
2	10	0.39	0.37	15	13.48	1.05	1.03	12	14.27
3	10	0.92	0.86	17	14.66	2.72	2.70	18	16.63
4	10	0.16	0.16	13	16.51	0.70	0.66	17	18.48
5	10	0.30	0.30	15	14.15	0.75	0.75	11	14.84
6	10	0.37	0.34	19	17.27	0.89	0.89	12	18.00
7	10	0.48	0.48	14	13.34	1.53	1.51	15	16.48
8	10	0.16	0.14	12	19.61	0.58	0.58	12	22.65
9	10	0.47	0.47	15	17.38	1.06	1.03	13	19.74
10	10	0.61	0.61	17	14.69	2.00	2.00	15	15.58
11	15	3.40	3.38	29	14.96	13.67	13.62	28	17.28
12	15	1.23	1.17	20	22.73	3.73	3.68	21	24.33
13	15	0.80	0.78	25	25.83	2.53	2.41	27	27.65
14	15	1.11	1.11	25	18.83	4.49	4.45	22	22.34
15	15	1.25	1.23	24	21.30	3.84	3.79	24	22.91
16	15	1.58	1.57	30	19.71	3.68	3.62	27	20.20
17	15	1.09	1.05	24	19.44	3.45	3.40	21	21.30
18	15	1.44	1.39	26	19.98	4.32	4.29	20	20.97
19	15	1.54	1.50	24	24.40	5.29	5.27	33	25.59
20	15	1.89	1.86	28	20.32	5.45	5.40	23	21.42
21	20	2.32	2.29	37	26.44	8.61	8.47	40	27.80
22	20	3.03	2.95	34	27.29	9.98	9.89	37	29.56
23	20	3.10	3.04	30	26.39	10.23	10.16	34	28.20
24	20	4.62	4.53	36	22.69	15.82	15.69	33	23.44
25	20	2.17	2.03	35	28.00	7.41	7.27	35	28.83
26	20	4.26	4.13	35	26.94	13.99	13.93	37	28.38
27	20	3.28	3.18	36	23.50	9.22	9.11	36	25.58
28	20	3.24	3.16	35	24.30	9.31	9.25	29	25.60
29	20	3.20	3.11	35	23.69	13.23	13.09	36	25.83
30	20	5.77	5.68	40	24.49	17.71	17.56	40	25.26
31	25	6.47	6.40	49	28.04	23.24	23.03	42	30.59
32	25	9.08	8.92	48	28.31	40.90	40.62	57	29.37
33	25	8.56	8.39	47	30.91	37.05	36.74	51	32.42
34	25	5.21	5.01	47	31.61	16.88	16.74	51	32.57
35	25	6.80	6.63	49	26.97	26.36	26.12	51	27.71
36	25	4.51	4.38	36	28.47	20.66	20.50	46	29.91
37	25	15.58	15.38	45	25.48	73.32	72.94	68	27.40
38	25	4.90	4.78	42	31.85	19.70	19.48	48	34.56
39	25	4.77	4.68	37	28.88	17.02	16.81	42	31.36
40	25	4.06	3.87	49	29.08	14.49	14.26	52	30.80

First observe that the computation time increases with the number of customers. Furthermore, almost all the computation time is spent on solving the pricing problems. When comparing the two route relaxations, Table 3.1 indicates that the column generation algorithm is significantly faster when all cycles are allowed. However, the values of the LP-relaxations are higher when 2-cycles are eliminated. In the next section we discuss how the branch-price-and-cut algorithm is affected by the increase of both the LP-value and computation time in the case of 2-cycle elimination as opposed to allowing all cycles.

3.4.3 Branch-price-and-cut results

In this section we present the results of the computational experiments performed with the branch-price-and-cut algorithm. In this algorithm, lower bounds are obtained by using Algorithm 3.3 and by adding capacity inequalities in iterations where no new routes are found. We compare the branch-price-and-cut algorithms using the two route relaxations, allowing all cycles and 2-cycle elimination.

Table 3.2: Branch-price-and-cut results, allowing all cycles

Inst.	$\lvert V' \rvert$	Tot.Time	Opt.Gap	LP Gap	Root Gap	Nodes	CI
1	10	2.28	0	19.46	0	6	41
2	10	86.25	0	13.39	0.19	543	54
3	10	1.28	0	15.86	0	1	43
4	10	21.92	0	10.83	0.14	241	65
5	10	5.19	0	11.93	0.29	19	53
6	10	1.44	0	4.04	0	4	41
7	10	3.87	0	21.65	0	5	34
8	10	4.20	0	17.90	0.85	62	50
9	10	4.03	0	14.42	0	17	41
10	10	9.22	0	9.94	0	21	45
11	15	35.01	0	15.85	0	12	76
12	15	3600.00	-	-	-	3932	212
13	15	3600.00	-	-	-	3634	304
14	15	47.46	0	18.74	0.03	53	123
15	15	22.82	0	11.81	0.08	42	112
16	15	64.41	0	6.29	0.13	151	86
17	15	29.70	0	11.80	0	42	97
18	15	709.16	0	10.43	0.42	1107	142
19	15	232.58	0	7.99	0.71	318	206
20	15	24.52	0	8.11	0	31	101
21	20	3600.00	-	-	-	1824	382
22	20	237.68	0	8.40	0.12	150	239
23	20	104.68	0	12.91	0.11	78	171
24	20	161.57	0	6.10	0.30	111	199
25	20	3600.00	-	-	-	2040	361
26	20	100.45	0	9.36	0	80	177
27	20	92.35	0	11.27	0	53	209
28	20	120.96	0	7.05	0.04	85	240
29	20	94.50	0	10.96	0	37	261
30	20	110.90	0	7.10	0	54	176
31	25	3600.00	-	-	-	1161	433
32	25	704.47	0	7.82	0.22	192	293
33	25	3600.00	-	-	-	22	353
34	25	3600.00	-	-	-	778	479
35	25	3600.00	-	-	-	827	460
36	25	3600.00	-	-	-	1422	504
37	25	3600.00	-	-	-	288	388
38	25	3600.00	-	-	-	765	537
39	25	3600.00	-	-	-	1343	506
40	25	3600.00	-	-	-	1030	461

Tables 3.2 and 3.3 show the results of applying the branch-price-and-cut algorithm to the test instances when allowing all cycles and eliminating 2-cycles respectively. The column Opt.Gap provides the percentage difference between the best obtained upper and lower bounds after termination of the algorithm. The columns LP Gap and Root Gap show the percentage difference between the value of the LP relaxation and the best found upper bound, before and after adding capacity inequalities respectively. The column Nodes provides the number of nodes processed in the search tree and the column CI gives the number of added capacity inequalities.

Table 3.2 shows that when allowing all cycles, two 15-customer instances, two 20-customer instances and nine 25-customer instances remain unsolved within the one hour time limit. For the other instances, the LP gap ranges from 4.04% to 21.65%. After adding capacity inequalities, these gaps are all tightened to less than 0.85%, and the gap is even completely closed for thirteen instances.

The results of the same experiment, but when eliminating 2-cycles, are shown in Table 3.3. Out of the thirteen unsolved instances when allowing all cycles, four 25-customer instances are now solved. Moreover, for the previously unsolved instances 13

Table 3.3: Branch-price-and-cut results, 2-cycle elimination

Inst.	$\|V'\|$	Tot.Time	Opt.Gap	LP Gap	Root Gap	Nodes	CI
1	10	0.67	0	0.05	0	1	2
2	10	127.53	0	8.29	0.17	483	30
3	10	4.15	0	4.53	0	1	13
4	10	31.17	0	0.14	0.14	193	17
5	10	2.64	0	7.66	0	2	11
6	10	1.72	0	0	0	2	0
7	10	5.76	0	3.16	0	4	4
8	10	3.93	0	5.19	0.65	29	11
9	10	3.45	0	2.82	0	7	6
10	10	6.88	0	4.51	0	5	32
11	15	102.48	0	2.79	0	22	34
12	15	3600.00	-	-	-	1070	83
13	15	3600.00	0.25	5.87	1.11	4391	172
14	15	68.11	0	3.60	0	45	51
15	15	34.59	0	5.10	0	36	56
16	15	108.58	0	3.96	0.10	98	28
17	15	26.94	0	3.35	0	15	24
18	15	123.86	0	5.96	0.20	98	53
19	15	157.45	0	3.49	0.56	133	65
20	15	46.44	0	3.11	0	25	59
21	20	3600.00	-	-	-	864	103
22	20	196.17	0	0.80	0.03	62	74
23	20	159.71	0	6.92	0	65	101
24	20	142.87	0	2.98	0.03	27	68
25	20	3600.00	0.25	3.59	0.83	2130	184
26	20	80.79	0	4.51	0	16	106
27	20	101.20	0	3.390	0	22	98
28	20	146.66	0	2.09	0	47	108
29	20	59.52	0	2.92	0	10	87
30	20	57.10	0	4.17	0	4	112
31	25	312.75	0	2.67	0.13	39	199
32	25	1371.48	0	4.37	0.07	232	144
33	25	3600.00	-	-	-	880	286
34	25	138.33	0	2.31	0	18	285
35	25	389.74	0	4.59	0	58	253
36	25	3600.00	-	-	-	1333	266
37	25	3600.00	-	-	-	232	179
38	25	3600.00	-	-	-	978	294
39	25	3600.00	-	-	-	2083	279
40	25	1749.58	0	4.17	0.30	539	286

and 25, an integer solution is found and the optimality gap is closed to 0.25% within one hour. The other seven previously unsolved instances, remain unsolved with 2-cycle elimination.

As can be seen from Tables 3.2 and 3.3, the LP gaps are significantly smaller when 2-cycles are eliminated. After adding capacity inequalities, the gap is completely closed for the thirteen previously closed instances, as well as for five other instances. For the remaining instances for which the optimum is found, the root gap is smaller when 2-cycles are eliminated. Note that a tighter root gap is not guaranteed as a heuristic procedure is used to separate capacity inequalities.

Out of the twenty-seven instances that are solved when all cycles are allowed, thirteen instances are solved faster when 2-cycles are eliminated while fourteen instances are solved slower.

3.4.4 Comparison with current practice

In practice, a solution to the TWAVRP is commonly found by assigning endogenous time windows using the following procedure. A VRPTW is solved using average demand over all scenarios and using the exogenous time windows as time windows. The arrival time at each customer is used as a point of reference for the time window. For t the arrival time at customer v, the endogenous time window $[y_v, y_v + w_v]$ is computed as

$$[y_v, y_v + w_v] = \begin{cases} [s_v, s_v + w_v] & \text{if } t - \frac{w_v}{2} \leq s_v; \\ [e_v - w_v, e_v] & \text{if } t + \frac{w_v}{2} \geq e_v; \\ \left[t - \frac{w_v}{2}, t + \frac{w_v}{2}\right] & \text{otherwise.} \end{cases} \tag{3.15}$$

We have implemented this procedure and used it to solve the test instances. To evaluate the expected costs of the endogenous time window assignment obtained by this procedure, a VRPTW is solved for each scenario using the endogenous time windows as the time windows. The expected costs are now computed by taking the (weighted) average of the solution values.

Table 3.4 shows the results of using this procedure. The column Value shows the value of the solution based on solving a VRPTW with average demand, and the column Opt. gives the optimal value of each instance. The column Gap provides the percentage difference between the solution value and the optimum. Only instances that have been solved to optimality are included in Table 3.4.

As can be seen from Table 3.4, the heuristic procedure provides the optimal solution for five 10-customer instances and two 20-customer instances. For the other instances, the differences are up to 5.42%. The average difference over all instances is 1.85%. After one hour of computation time, the branch-price-and-cut algorithm with 2-cycle elimination has found a solution for instance 13 with value 29.37 and for instance 25 with value 29.066. The VRPTW with average demand based solution values of instance 13 and 25 are 29.545 and 30.04 respectively. The differences between the solutions obtained by these procedures are 0,57% and 0,45%.

3.5 Conclusion

In this chapter we introduce the time window assignment vehicle routing problem, the TWAVRP, which models the problem of assigning time windows to customers before demand is known. In this model, demand realizations occur according to a predefined set of scenarios with known probability distribution. After demand becomes known,

Table 3.4: VRPTW with average demand based solutions

Inst.	$\|V'\|$	Value	Opt.	Gap
1	10	17.65	17.65	0
2	10	16.17	15.56	3.92
3	10	17.42	17.42	0
4	10	18.51	18.51	0
5	10	16.15	16.07	0.52
6	10	18.00	18.00	0
7	10	17.02	17.02	0
8	10	23.97	23.89	0.33
9	10	21.41	20.31	5.42
10	10	16.54	16.31	1.41
			Average gap:	1.16
11	15	18.05	17.78	1.54
14	15	24.05	23.18	3.77
15	15	24.87	24.15	3.00
16	15	21.11	21.03	0.36
17	15	23.22	22.04	5.35
18	15	23.03	22.30	3.27
19	15	26.66	26.52	0.54
20	15	22.73	22.11	2.80
			Average gap:	2.58
22	20	30.47	29.80	2.26
23	20	30.92	30.30	2.05
24	20	24.30	24.16	0.57
26	20	29.72	29.72	0
27	20	27.48	26.48	3.78
28	20	27.05	26.14	3.47
29	20	27.16	26.61	2.08
30	20	26.36	26.36	0
			Average gap:	1.77
31	25	31.82	31.43	1.25
32	25	31.86	30.71	3.74
34	25	34.54	33.34	3.59
35	25	29.66	29.05	2.11
40	25	32.22	32.14	0.26
			Average gap:	2.19

an optimal vehicle routing schedule is made adhering to the assigned time windows. The problem is to assign time windows such that the expected total traveling costs are minimized.

We propose a branch-price-and-cut algorithm to solve the TWAVRP. We have considered two route relaxations, allowing all cycles and eliminating 2-cycles. Moreover, we strengthen the LP-bound by adding capacity inequalities. Computational experiments show that the proposed branch-price-and-cut algorithm is capable of solving instances of the TWAVRP of up to 25 customers and 3 scenarios. Using 2-cycle elimination in the branch-price-and-cut algorithm increased the number of instances that were solved to optimality. However, neither route relaxation yields a branch-price-and-cut algorithm that is superior with respect to running times.

We compared the optimal solution to a solution obtained by solving a VRPTW with average demand as is commonly done in practice. In our experiments, the solutions based on solving a VRPTW with average demand have costs that are up to 5.42% higher than the optimum, and are on average 1.85% higher.

Chapter 4

The Discrete Time Window Assignment Vehicle Routing Problem

4.1 Introduction

In distribution networks, it is common for a supplier and a customer to agree on a time window in which a delivery will be made. This time window is often used repeatedly within some period of time in which multiple deliveries are made at regular intervals. At the moment of choosing a time window for a customer, its demand is usually unknown and may fluctuate for different deliveries. When demand for all customers becomes known for a given day, a vehicle routing problem with time windows (VRPTW) must be solved to construct a delivery schedule within the agreed time windows.

The time window assignment vehicle routing problem (TWAVRP) is introduced in Chapter 3. Given a set of customers to be visited on the same day, it consists of assigning a time window to each customer before demand is known, and constructing vehicle routes that satisfy the assigned time windows when demand becomes known. The assigned time windows have a predetermined width and can start at any time within an exogenous time window that can be customer-dependent. The objective of the TWAVRP is to minimize the expected total transportation costs. Uncertain demand is represented by a set of scenarios each occurring with a certain probability. For instance, different scenarios can be used for low, normal and high demands.

In this chapter, we study the discrete TWAVRP (DTWAVRP) which differs from the TWAVRP by considering for each customer a discrete set of candidate time windows from which one has to be selected. For example, a customer might divide the day in blocks of two hours commencing on the hour and require one of these blocks to be the assigned time window. We have encountered such time window assignment problems (discrete or not) while collaborating with Dutch retail chains, and believe they are common in this industry. Here, the retailers (customers) are heavily dependent on the time window to be fixed in advance and kept for some time. For instance, a retailer might receive all its deliveries on the same day of the week and more or less the same hour of the day for an entire year. This is crucial for many operational purposes like inventory management and the scheduling of personnel. Considering a discrete set of time windows is often more practical for the retailers, especially to ease the personnel scheduling process which must take into account various regulations. Furthermore, it can give them the opportunity to express preferences for the time windows. Maximizing the satisfaction of these preferences might be taken into account as a secondary objective during the optimization process, an option that is not considered in this chapter.

The DTWAVRP is NP-hard as in the case of one scenario and one candidate time window per customer it reduces to the VRPTW. When it involves several scenarios, the DTWAVRP corresponds to solving several VRPTWs (one per scenario) that are linked together by the choice of the time windows. The VRPTW is a well-studied problem for which many exact and heuristic algorithms have been developed (see, e.g., the surveys of Baldacci et al., 2012, Kallehauge et al., 2005, and Bräysy and Gendreau, 2005a,b). We believe that in the scientific literature, the problem of assigning time windows before knowing demand has been largely overlooked so far. In Chapter 3 an exact branch-and-price algorithm is presented for the TWAVRP that can solve instances with up to 25 customers and 3 scenarios.

Introduced by Groër et al. (2009), the consistent vehicle routing problem (ConVRP) is similar to the DTWVARP. In this problem each customer must be visited on different days of a given horizon (not all customers must be serviced each day) following a consistent schedule, that is, the arrival times at a customer from one day to another cannot differ by much than a limited amount of time. Moreover, it is required that each customer is always visited by the same driver. Groër et al. (2009) found optimal solutions to ConVRP instances involving up to 12 customers and 3 scenarios using a commercial mixed integer programming solver. They reported computation times of up to several days. Furthermore, they developed a local search heuristic to solve instances with over 3700 customers.

Jabali et al. (2010) considered another related problem that involves the assignment of time windows in a vehicle routing problem with stochastic travel times and deterministic demands. They developed a tabu search algorithm for solving it. Also, Agatz et al. (2011) studied a problem faced by e-tailers providing home delivery that consists of selecting which time slots to offer per zip code for making deliveries. They developed a local search heuristic.

The main contributions of this chapter are as follows. First, we propose a new problem, the DTWAVRP. Second, we develop a state-of-the-art exact branch-price-and-cut algorithm to solve it and report computational results obtained on randomly generated instances to evaluate the effectiveness of some of its components. Finally, we compare the gains yielded by exact solutions over current practice, where time window assignment is typically based on the solution of a vehicle routing problem with average historic demand.

In the next section, we provide a formal definition of the DTWAVRP and present an integer programming formulation for it. In Section 4.3, we describe the proposed branch-price-and-cut algorithm. In Section 4.4, we report the results of the numerical experiments that we conducted with our algorithm. Finally, conclusions are drawn in Section 4.5.

4.2 Problem definition

Consider a complete graph $G = (V, A)$, where $V = \{0, ..., n + 1\}$ is a set of locations such that 0 represents the starting depot, $n + 1$ the ending depot and $V' = \{1, ..., n\}$ are the customers. Let $c_{ij} \geq 0$ be the cost to travel along arc (i, j) and $t_{ij} \geq 0$ the corresponding travel time (including, if any, the service time at i). Both the travel costs and travel times satisfy the triangle inequality. Furthermore, an unlimited number of vehicles of equal capacity Q is available.

Let Ω be a set of scenarios, where each scenario is characterized by a realization of demand. Let d_v^ω be the demand at customer v in scenario $\omega \in \Omega$ such that $0 < d_v^\omega \leq Q$. The probability that scenario ω occurs is p^ω.

Associate with each customer v a set W_v of candidate time windows that may or may not overlap. One time window $w = [\underline{w}, \overline{w}] \in W_v$ must be selected for each customer such that in each scenario service at customer v starts between $\underline{w}$ and $\overline{w}$. For the starting and ending depot only one time window exists. Note that waiting at a location is allowed, i.e., a vehicle can arrive prior to the start of a time window and start service later.

Using the set of candidate time windows for each customer, we can construct an auxiliary graph $\hat{G} = (\hat{V}, \hat{A})$, where $\hat{V} = \{(v, w) \mid w \in W_v, v \in V\}$ contains a copy of each customer node v for each of its possible time windows $w \in W_v$. Moreover, $\hat{A}$ contains an arc between two nodes (v, w) and (v', w'), $v \neq v'$, if and only if $\underline{w} + t_{vv'} \leq \overline{w'}$.

We use the term route to refer to a pair $(\hat{P}, t)$ where $\hat{P}$ is a path in $\hat{G}$ starting at 0 and ending at $n+1$ and t is a vector containing the time of service at each location on the path. Associated with each route r is the parameter a_{vwr} indicating the number of times customer v is visited within time window $w \in W_v$ on route r. To each route r whose path contains the arcs $\{r_1, ..., r_k\}$ we assign the cost $c_r = \sum_{i=1}^{k} c_{r_i}$.

Let $R(\omega)$ be the set of all feasible routes for scenario ω. A route $(\hat{P}, t)$ is considered feasible if *i*) it satisfies the capacity constraint, *ii*) t is such that, for each customer v on the route, service commences within a time window in W_v, and *iii*) if location j is visited directly after i on route r then $t_i + t_{ij} \leq t_j$.

The DTWAVRP is the problem of assigning one time window to each customer and selecting feasible routes for each scenario such that, for each scenario, each customer is visited exactly once and is serviced within its assigned time window. The expected travel costs must be minimized.

Next, we present an integer programming formulation for the DTWAVRP. Let the variables x_r^ω, for $\omega \in \Omega$ and $r \in R(\omega)$, be route variables indicating whether route r is selected in scenario ω. Furthermore, let y_{vw}, for $v \in V'$ and $w \in W_v$, be time window assignment variables indicating whether time window w is selected for customer v. The DTWAVRP can be formulated as the following mixed integer linear program.

$$\min \sum_{\omega \in \Omega} p^\omega \sum_{r \in R(\omega)} c_r x_r^\omega \tag{4.1}$$

$$\text{s.t.} \quad \sum_{w \in W_v} y_{vw} = 1 \qquad \forall v \in V' \tag{4.2}$$

$$\sum_{r \in R(\omega)} a_{vwr} x_r^\omega = y_{vw} \qquad \forall v \in V', \forall w \in W_v, \forall \omega \in \Omega \tag{4.3}$$

$$x_r^\omega \in \{0, 1\} \qquad \forall \omega \in \Omega, \forall r \in R(\omega) \tag{4.4}$$

$$y_{vw} \in \{0, 1\} \qquad \forall v \in V', \forall w \in W_v \tag{4.5}$$

The objective function (4.1) aims at minimizing the expected total costs resulting from a time window assignment. Constraints (4.2) ensure that exactly one time window is selected for each customer. Constraints (4.3) impose that each customer is visited exactly once in each scenario and within the selected time window. The integrality requirements on the x and y variables are provided by (4.4) and (4.5)

Next, let us discuss how to reformulate these integrality requirements. Consider the linear programming (LP) relaxation of formulation (4.1)-(4.5) where the integrality requirements on the x and y variables are relaxed continuously. For each scenario, let the arc flow in $\hat{G}$ be the value by which each arc $a \in \hat{A}$ is selected in a solution to this LP relaxation. It is straightforward that when the arc flow in $\hat{G}$ is integer for every scenario, it also provides an optimal integer solution to the DTWAVRP.

Moreover, a solution to the LP relaxation also corresponds to an arc flow in G for each scenario. Observe that when the arc flow in $\hat{G}$ is integer, so is the arc flow in G. However, when the arc flow in G is integer, the arc flow in $\hat{G}$ might not be. Nevertheless, Proposition 4.1 states that an optimal integer solution to the problem can always be derived in this case.

Proposition 4.1. *Let (x, y) be an optimal solution to the LP relaxation of formulation* (4.1)-(4.5)*. When the corresponding arc flow in G is integer for every scenario, there exists an optimal solution (x^*, y^*) to the DTWAVRP of equal value.*

Proof. For each customer v, let $w(v, y) \in \arg\min\{\underline{w} \mid w \in W_v, y_{vw} > 0\}$ be the candidate time window with the earliest start time among the ones selected in solution (x, y).

Let F^ω be the integer arc flow in G for scenario ω, corresponding to solution (x, y). This arc flow can be represented as a set of $(0, n+1)-$paths in G, $F^\omega = \{P_1, ..., P_{k(\omega)}\}$. Furthermore, denote by F^ω_a the flow on arc $a \in A$ for scenario ω.

For any path $P \in F^\omega$ visiting the customers $\{v_1, ..., v_l\}$, consider the path $\hat{P}$ in $\hat{G}$ visiting the nodes $\{(v_1, w(v_1, y)), ..., (v_l, w(v_l, y))\}$. Using path $\hat{P}$ for all $P \in F^\omega$, $\omega \in \Omega$, and the time windows $w(v, y)$ for each $v \in V'$, yields a solution whose value is equal to that of (x, y). To complete the proof, we need to show that this solution is feasible. Because a path $\hat{P}$ visits the same customers as its parent path P, the capacity constraints are satisfied by the routes in the new solution. Hence, all that remains to be shown is that the time window constraints are also satisfied along those routes.

Consider the graph $\hat{G}(F^\omega, y) = (\hat{V}(y), \hat{A}(F^\omega, y))$, where $\hat{V}(y) = \{(v, w) \in \hat{V} \mid y_{vw} > 0\}$ $\bigcup$ $\{(0, w_0), (n+1, w_{n+1})\}$ contains the combinations of locations and time windows that are selected in solution (x, y), and $\hat{A}(F^\omega, y) = \{((v, w), (v', w')) \in \hat{A} \mid (v, w), (v', w') \in \hat{V}(y), F^\omega_{(v,v')} > 0\}$. Observe that all paths from $(0, w_0)$ to $(n+1, w_{n+1})$ in $\hat{G}(F^\omega, y)$ can be represented in $\hat{G}$. Moreover, any such path visits the same customers as some path $P \in F^\omega$ and in the same order.

Let $\underline{t}^\omega_{vw}$ be the earliest possible start of service time in node (v, w) by any path in $\hat{G}(F^\omega, y)$ starting at node $(0, w_0)$. Let $\underline{t}^\omega_{0w_0} = \underline{w}_0$. Observe that as $y_{vw} > 0$ for $(v, w) \in \hat{V}(y)$, constraints (4.3) ensure that there is a route $r \in R(\omega)$ such that $x_r > 0$ for all $\omega \in \Omega$. Hence, $\underline{t}^\omega_{vw}$ exists for all $(v, w) \in \hat{V}(y)$ and all $\omega \in \Omega$.

Let $W_v(y) = \{w | w \in W_v, y_{vw} > 0\}$. Next, let $\underline{t}^\omega_v = \min_{w \in W_v(y)}\{\underline{t}^\omega_{vw}\}$ be the earliest start of service time at customer v in $\hat{G}(F^\omega, y)$. Observe that $\underline{t}^\omega_0 = \underline{t}^\omega_{0w_0}$. For every pair

(v, v') such that $F^{\omega}_{(v,v')} > 0$, it holds that

$$\underline{t}^{\omega}_{v'w(v',y)} = \max\{\underline{w(v',y)}, \underline{t}^{\omega}_{v} + t_{vv'}\} \leq \max\{\underline{w}, \underline{t}^{\omega}_{v} + t_{vv'}\} = \underline{t}^{\omega}_{v'w} \qquad \forall w \in W_{v'}(y).$$

Therefore, $\underline{t}^{\omega}_{v} = \underline{t}^{\omega}_{vw(v,y)}$ and it follows that $\underline{t}^{\omega}_{v'w(v',y)} \geq \underline{t}^{\omega}_{vw(v,y)} + t_{vv'}$. This shows that using path $\hat{P}$ and service times $\underline{t}^{\omega}_{vw(v,y)}$ for the nodes $(v, w(v, y))$ visited on this path provides a feasible route for each $P \in F^{\omega}$ and each scenario $\omega \in \Omega$.

□

The proof of Proposition 4.1 also provides a method to derive an optimal solution. In the rest of this chapter, we relax the integrality conditions (4.4) and (4.5), and instead impose integrality on the arc flow in G.

4.3 Solution method

In this section, we first describe the column generation algorithm that we use to solve the LP relaxation of (4.1)-(4.5). In particular, we present the ng-route relaxation and discuss acceleration strategies to speed up the pricing algorithm. Next, we suggest valid inequalities to strengthen the LP bound. Finally, we describe the branch-price-and-cut algorithm.

4.3.1 Column generation algorithm

In practice, the LP relaxation of (4.1)-(4.5), also called the master problem, contains a very large number of variables. To overcome this difficulty, we solve the master problem using a column generation algorithm that was first proposed by Dantzig and Wolfe (1960). This algorithm iteratively solves a restricted master problem (RMP) and a pricing problem. The RMP is the master problem where only a subset of the routes are included. It is solved using the simplex algorithm, providing a feasible primal solution and the values of the dual multipliers associated with constraints (4.2) and (4.3). The pricing problem is solved to identify route variables with negative reduced costs that have not yet been added to the RMP. When a route with a negative reduced cost is identified, it is added to the RMP and the procedure is repeated. If no route with a negative reduced cost exists, the current solution to the RMP is also optimal for the master problem.

For the DTWAVRP, the pricing problem can be decoupled into several problems, one for each scenario. The pricing problem for scenario ω aims at finding a feasible route for scenario ω with the least reduced cost. Let λ be the vector of unrestricted dual multipliers associated with constraints (4.3). The reduced cost of a route $r \in R(\omega)$ is given by

$$p^{\omega} c_r - \sum_{v \in V'} \sum_{w \in W_v} \lambda^{\omega}_{vw} a_{vwr}. \tag{4.6}$$

This pricing problem can be modeled as an elementary shortest path problem with resource constraints defined on network $\hat{G}$. To do so, associate with each node $(v, w) \in \hat{V}$ the demand d^{ω}_v and with each arc $((v, w), (v', w')) \in \hat{A}$ the reduced cost $p^{\omega} c_{vv'} - \lambda^{\omega}_{v'w'}$ and the travel time $t_{v,v'}$. The pricing problem consists of finding a shortest elementary $(0, n+1)$-path in $\hat{G}$ that respects time windows and vehicle capacity (the resource constraints). Note, however, that elementarity is required for the customers. This means that for each customer $v \in V'$ at most one node $(v, w) \in \hat{V}$ can be included in an elementary path.

To solve the pricing problem, we use the labeling algorithm proposed by Feillet et al. (2004) which we modify to consider elementarity of the customers instead of the nodes in network $\hat{G}$. In this algorithm, constructed partial paths are represented by labels. Let l be a label representing a partial path from the starting depot to a specific node $(v, w) \in \hat{V}$. Let $c(l)$ be the total reduced cost of the partial path represented by label l, $t(l)$ its earliest service time at customer v in time window w, and $q(l)$ its total load. Finally, let $f_u(l)$, $u \in V'$, be a binary parameter equal to 1 if customer u has already been visited in the partial path associated with label l or if this path cannot be feasibly extended to reach any node representing customer u as this would violate capacity or time window constraints. In this respect, we define the function $U^{\omega}_u(l)$ that takes value 1 if $q(l) + d^{\omega}_u > Q$ or $t(l) + t_{vu} > \overline{w}$ for all $w \in W_u$, indicating whether l can be extended to u.

The labeling algorithm starts with a single label associated with depot node 0. Next, labels are extended along the arcs in $\hat{G}$. A label l associated with a node (v, w) can be extended to a node (v', w') only if $((v, w), (v', w')) \in \hat{A}$ and $f_{v'}(l) = 0$. To perform this extension and create a label l', we use the following extension functions:

$$c(l') = c(l) + p^{\omega} c_{vv'} - \lambda^{\omega}_{v'w'} \tag{4.7}$$

$$t(l') = \max\{t(l) + t_{vv'}, \underline{w}'\} \tag{4.8}$$

$$q(l') = q(l) + d^{\omega}_{v'} \tag{4.9}$$

$$f_u(l') = \begin{cases} 1 & \text{if } u = v' \\ \max\{f_u(l), U^{\omega}_u(l')\} & \text{otherwise} \end{cases} \qquad \forall u \in V'. \tag{4.10}$$

Label l' is deemed feasible if $t(l') \leq \overline{w'}$. Otherwise, it is discarded. Note that it is not necessary to check if $q(l') \leq Q$ because $f_{v'}(l) = 0$.

In order to avoid the enumeration of all partial paths, a dominance procedure is applied. The aim of this procedure is to remove all non-Pareto optimal labels. A label that is not Pareto optimal is said to be dominated. Label l' is dominated if there exists a label l associated with the same customer and $c(l) \leq c(l')$, $t(l) \leq t(l')$, $q(l) \leq q(l')$ and $f_u(l) \leq f_u(l')$ for all $u \in V'$. We want to emphasize the fact that we check dominance for labels at the same customer instead of at the same node as we require elementarity of customers and not nodes. This increases the number of dominated labels.

This labeling algorithm might provide multiple routes with negative reduced costs. In our implementation of the column generation algorithm, we add all routes with a negative reduced cost to the RMP at each iteration.

4.3.2 Route relaxations

As solving the elementary shortest path problem with resource constraints is computationally expensive, it is common to relax elementarity. Generating routes containing cycles and adding them to the formulation does not alter the optimal integer solution as each customer is visited exactly once. However, the LP bound becomes weaker. For the VRPTW, Desrochers et al. (1992) were the first to suggest a branch-and-price algorithm using a non-elementary shortest path problem as the pricing problem. They eliminate 2-cycles, i.e., cycles of the form $i-j-i$, to strengthen the LP bound at the expense of limited additional computation time. Irnich and Villeneuve (2006) extended this approach by providing an algorithm to solve the shortest path problem with resource constraints and $k-$cycle elimination, for arbitrary values of k.

Recently, Baldacci et al. (2011) proposed the ng-route relaxation. For each customer $v \in V'$ a neighbourhood $N_v \subseteq V'$ with $v \in N_v$ is introduced. An ng-path is not necessarily an elementary path. Indeed, cycles starting and ending at a customer v are allowed if this cycle contains a customer v' such that $v \notin N_{v'}$. Similar to, e.g., Baldacci et al. (2011) and Ribeiro et al. (2012), we construct neighbourhoods of a fixed size Δ_{ng} for each customer $v \in V'$. They contain the closest customers with respect to travel costs, including v itself. This way, any cycle in an ng-path will be relatively long or expensive.

In our branch-price-and-cut algorithm we use the ng-route relaxation. We adjust the labeling algorithm for the elementary shortest path problem with resource constraints to solve a shortest ng-path problem with resource constraints by modifying the extension functions for the customer resources. When extending label l from a node (v, w) to (v', w')

to create label l', customer resource $f_u(l')$ is set to zero if $u \notin N_{v'}$ even though $f_u(l) = 1$. Hence, expression (4.10) is replaced by the following:

$$f_u(l') = \begin{cases} 1 & \text{if } u = v' \\ \max\{f_u(l), U_u^{\omega}(l')\} & \text{if } u \in N_{v'} \setminus \{v'\} \\ 0 & \text{otherwise} \end{cases} \qquad \forall u \in V'. \tag{4.11}$$

In this case, label l' is declared feasible if $t(l') \leq \overline{w'}$ and $q(l') \leq Q$. Here the latter condition must be checked because it can be violated even if $f_{v'}(l) = 0$.

During the label dominance check at a node (v, w), $v \in V'$, only the customer resources for $u \in N_v$ need to be considered, that is, the dominance rule involves only the conditions $f_u(l) \leq f_u(l')$, $\forall u \in N_v$, for the customer resources. This is sufficient because $f_u(l) = f_u(l') = 0$, $\forall u \in V' \setminus N_v$. Using ng-paths typically increases the number of dominated labels and, thus, speeds up the labeling algorithm. Low values of Δ_{ng} yield a fast labeling algorithm at the expense of a decreased LP bound, whereas high values slow down the labeling algorithm but increase the value of the LP bound. Observe that all cycles are allowed in an ng-path when $\Delta_{ng} = 1$, and only elementary paths are allowed when $\Delta_{ng} = n$.

4.3.3 Acceleration strategies

It is well known that, in a column generation algorithm, there is no need to solve the pricing problems to optimality at each iteration. As long as negative reduced cost columns are found, the pricing problems can be solved heuristically and it is even possible to skip some pricing problems. The algorithm remains exact if the pricing problems are solved to optimality in the last column generation iteration when solving a linear relaxation. Below, we discuss two strategies to potentially generate negative reduced cost columns in fast computation times.

Reusing routes

At each iteration of the column generation algorithm, a pricing problem is solved for each scenario. Because these pricing problems are very similar, solutions to the pricing problem of one scenario might also be feasible for another. Reusing a solution in this way potentially decreases the number of pricing problems that have to be solved at each iteration. Therefore, we propose the column generation algorithm described in Algorithm 4.1, in which solutions are reused for other scenarios when they are feasible and have a negative reduced cost.

Algorithm 4.1 Column Generation Algorithm, Reusing Routes

Initialize $R(\omega)$ for all $\omega \in \Omega$
repeat
  Solve the RMP using the routes in $R(\omega)$ for scenario $\omega \in \Omega$
  Set $\tilde{\Omega} = \Omega$
  while $\tilde{\Omega} \neq \emptyset$ **do**
    Choose $\omega \in \tilde{\Omega}$ and remove it from $\tilde{\Omega}$
    Solve the pricing problem for scenario ω to find a set R of routes with negative reduced costs
    Add all routes in R to $R(\omega)$
    for all $\tilde{\omega} \in \tilde{\Omega}$ **do**
      Let $\tilde{R} \subseteq R$ be the subset of routes that are feasible and have a negative reduced cost for scenario $\tilde{\omega}$
      if $\tilde{R} \neq \emptyset$ **then**
        Add the routes in $\tilde{R}$ to $R(\tilde{\omega})$ and remove $\tilde{\omega}$ from $\tilde{\Omega}$
      end if
    end for
  end while
until No new routes are added to the RMP

Note that the order in which the scenarios are solved at each iteration might affect the performance of the algorithm. However, our preliminary experiments showed no significant differences for several strategies of ordering the scenarios. The computational results presented in Section 4.4 were obtained by using a fixed order of the scenarios over all iterations.

Heuristic column generation: tabu search

The column generation algorithm can be further accelerated by using a heuristic to solve the pricing problem. A heuristic might be able to identify feasible routes with negative reduced costs in less time than it takes to solve the pricing problem exactly. When using a heuristic at each iteration of the column generation algorithm, the exact algorithm is only used to find new routes or prove optimality when the heuristic fails.

As done by, e.g., Desaulniers et al. (2008), we developed a tabu search algorithm to solve the pricing problem. In this algorithm an initial route is considered, which is iteratively replaced by a neighbouring route. The neighbourhood of each route contains all feasible elementary routes that can be obtained by performing one move. We consider two types of moves: adding a single node at any position in the route and removing a single node from the route.

At each iteration, the best neighbour in terms of reduced cost is selected as the new route. Note that this might yield a route with a higher reduced cost than that of the previous route. To avoid cycling, selecting the inverse of the move used to obtain the current route is tabu for TS_{tabu} iterations. If the reduced cost of the new route is negative, it is added to the RMP. To diversify the search, at every TS_{It} iterations, it is restarted using a completely new route. The initial route and those used to restart the search corresponds to the routes selected in the current solution to the RMP for the scenario associated with the pricing problem. When such a route is not elementary, the first visit to each customer is maintained and all other visits to the same customer are removed from the route. The algorithm stops when all selected routes have been used to restart, or a total of TS^{max} new routes have been added to the RMP during the current search.

4.3.4 Valid inequalities

For the vehicle routing problem, many valid inequalities have been studied: for example, capacity, comb, hypotour and multistar inequalities (Lysgaard et al. 2004), k-path inequalities (Kohl et al. 1999) and subset row inequalties (Jepsen et al. 2008). These inequalities are also applicable for each scenario in the DTWAVRP.

We have tested all the above mentioned valid inequalities in our algorithm. However, preliminary experiments showed that adding capacity inequalities and subset row inequalities provide the lowest computation time. Below, we describe these inequalities in more detail.

Let z_{ij}^{ω} be the arc flow in G on arc (i, j) in scenario ω. Let $b(S)$ be the minimum number of vehicles needed to visit all customers in $S \subseteq V'$. The capacity inequalities are as follows:

$$\sum_{i \in S, j \notin S} z_{ij}^{\omega} \geq b(S) \qquad \forall S \subseteq V', \forall \omega \in \Omega \tag{4.12}$$

and can be expressed in terms of the variables x_r^{ω}. As is common, we replace $b(S)$ by the lower bound $\left\lceil \frac{\sum_{i \in S} d_i^{\omega}}{Q} \right\rceil$. The separation problem of these rounded capacity inequalities is strongly NP-hard. We use the heuristic of Lysgaard et al. (2004) to separate them, more precisely, we use the implementation that can be found in the package by Lysgaard (2003).

When capacity inequalities are added to the master problem, the pricing problems remain the same. However, the reduced cost of a route may be altered. Let μ_S^{ω} be the dual variable associated with the capacity inequality for subset S in scenario ω. We modify the pricing problem for scenario ω by subtracting μ_S^ω from the reduced cost of the arcs $((v,w),(v',w')) \in \hat{A}$ such that $v \in S$ and $v' \notin S$.

The subset row inequalities are a special case of the Chvátal-Gomory rank 1 cuts. They were introduced by Jepsen et al. (2008). Let $a_{vr} = \sum_{w \in W_v} a_{vwr}$ be the number of times v is visited on route r. The subset row inequalites can be expressed as follows:

$$\sum_{r \in R(\omega)} \left\lfloor \frac{1}{k} \sum_{v \in S} a_{vr} \right\rfloor x_r^\omega \leq \left\lfloor \frac{|S|}{k} \right\rfloor \qquad \forall S \subseteq V',\ 2 \leq k \leq |S|,\ \forall \omega \in \Omega. \tag{4.13}$$

The subset row separation problem is NP-complete. As suggested by e.g. Jepsen et al. (2008) and Desaulniers et al. (2008), we separate only subset row inequalities for subsets of size three, using $k = 2$, by enumeration. In this case, the inequalities ensure that for any set of three customers, at most one route can be selected that includes more than one of these customers.

Adding subset row inequalities to the formulation for scenario ω changes the corresponding pricing problem. Let σ_S^ω be the dual variable associated with the subset row inequality for subset S in scenario ω. For every k customers in S visited by a path in the pricing problem, σ_S^ω is subtracted from the reduced cost of that path.

The labeling algorithm is adjusted to include the dual values of the subset row inequalities in the reduced cost of each path. For the pricing problem associated with scenario ω, the labels are modified by incorporating a new resource h_S for every generated subset row inequality associated with a subset S and scenario ω. When extending a label to a customer in S, h_S is increased by one. When this resource reaches k, then σ_S^ω is subtracted from the reduced cost and the resource is reset to 0. Hence, $h_S(l)$ gives the number of times a customer in S was visited by the partial path corresponding to label l since the last time σ_S^ω was subtracted from the reduced cost.

As proposed by Jepsen et al. (2008), the dominance check is modified as follows. When trying to establish whether a label l dominates a label l', instead of checking whether $c(l) \leq c(l')$, we check whether $c(l) - \sum_{S: h_S(l) > h_S(l')} \sigma_S^\omega \leq c(l')$. Note that the subset row resources $h_S(l)$ and $h_S(l')$ are not compared during the dominance check. This way, more labels might be dominated.

As adding subset row inequalities slows down the labeling algorithm, we limit the number of inequalities added simultaneously as proposed by Desaulniers et al. (2008). In each iteration only a maximum number of $SR_v^{\max}$ subset row inequalities might be added for subsets that include customer v. Furthermore, we limit the number of subset row inequalities added at once by $SR_{It}^{\max}$. Finally, we limit the total number of added subset row inequalities to $SR^{\max}$. To ensure that the limited number of subset row inequalities are likely to make an impact on the LP bound, we only add subset row inequalities that are violated by at least $SR^{\min}_{Vio}$.

4.3.5 Branch-price-and-cut

We propose the following branch-price-and-cut algorithm to solve the DTWAVRP to optimality. Lower bounds are found by solving the LP relaxation using column generation (see Algorithm 4.1) and adding valid inequalities. Capacity inequalities are separated in each iteration of the column generation algorithm where no new routes with negative reduced costs are identified. Because of their negative impact on the computation time of the algorithm that solves the pricing problem, subset row inequalities are only generated when no violated capacity constraint can be found. Branching is performed on the arcs in G, as by Proposition 4.1, integer arc flow in G is sufficient to identify an integer optimal solution.

We perform special ordered subset branching on the arcs (SOS branching). More specifically, for scenario ω and customer v, let $\delta^-_\omega(v)$ and $\delta^+_\omega(v)$ be the sets of in and out arcs of a node representing customer v, respectively. Next, a customer v', a scenario ω' and an arc type $o' \in \{-,+\}$ is selected such that the number of arcs a in $\delta^{o'}_{\omega'}(v')$ for which $z^{\omega'}_a > 0$ is the largest set. Let $\delta^{o'}_{\omega'}(v') = \{a_1, ..., a_k\}$ be ordered with respect to the arc flow in G, such that $z^{\omega'}_{a_i} \geq z^{\omega'}_{a_j}$ if $i < j$. The arcs are divided into two groups, S and its complement $\bar{S}$, where $S = \{a_1, ..., a_i\}$ is such that $\sum_{a \in S} z^{\omega'}_a \geq 0.5$ and $\sum_{a \in S \setminus \{a_i\}} z^{\omega'}_a < 0.5$. In one branch we disallow the use of the arcs in S and in the other we disallow the use of the arcs in $\bar{S}$. This does not alter the nature of the pricing problem, in fact the number of arcs in the network decreases.

In our branch-price-and-cut algorithm, upper bounds are obtained when any current LP relaxation has an integer solution. The search tree is explored using a best-first strategy, that is, the node with the lowest lower bound is selected to process next.

4.4 Computational results

In this section we present the results of our computational experiments. First, we elaborate on the instances that were used. Next, we illustrate the performance of the column generation algorithm. Finally, the results of using the branch-price-and-cut algorithm are presented. A time limit of one hour is enforced to solve each instance.

All our tests were performed on an Intel(R) Core(TM) i5-2450M CPU 2.5 GHz processor. The algorithms were coded in C++ and the IBM ILOG Cplex optimizer, version 12.4, was used to solve the RMP in the column generation algorithm.

4.4.1 Test instances

The instances used for our experiments were randomly generated[1]. For each instance, n customers are generated using a uniform distribution over a square with sides of length 5. The depot is located in the center of the square. Travel costs and times are computed as the Euclidean distance between two locations rounded to two digits. Vehicle capacity is 30. The depot has time window [6,20]. We construct three sets of candidate time windows which we randomly assign to each customer, such that each set of candidate time windows is assigned with fixed frequency. We assign the set $\{[10,12],[12,14],[14,16]\}$ to 10% of the customers, the set $\{[8,10],[10,12],[12,14],[14,16],[16,18]\}$ to 60% of the customers, and $\{[7,9],[9,11],[11,13],[13,15],[15,17],[17,19],[19,21]\}$ to 30% of the customers.

For each instance, we generate 3 demand scenarios, each occurring with equal probability. The scenarios are generated such that the first scenario has low demand, the second scenario has medium demand and the final scenario has high demand. We accomplish this by randomly generating a demand realization d_v for all $v \in V'$ according to a normal distribution with expectation 5 and variance 1.5. Next we generate multipliers u_v^1, u_v^2 and u_v^3 for all $v \in V'$ uniformly distributed in $[0.7,0.8]$, $[0.95,1.05]$ and $[1.2,1.3]$, respectively. Finally, we generate the demand for each customer $v \in V'$ and each scenario $\omega \in \{1,2,3\}$ by computing $d_v^\omega = \lceil u_v^\omega d_v \rceil$. Generating scenarios in this way resembles demand behavior that is encountered in the case, for instance, of ice cream vendors. When the weather is exceptionally good or bad, demand for ice cream goes up or down respectively. Moreover, all vendors in the network are affected similarly by the weather, leading to an increase or decrease of demand for all vendors simultaneously.

These settings are inspired by experience with a Dutch retail chain. The time window and capacity constraints ensure that no more than roughly 7 or 8 customers can be visited by a single vehicle in any scenario. We have generated 10 instances for each of the following 4 sizes, namely, 10, 15, 20 and 25 customers, making a total of 40 instances.

4.4.2 Column generation results

In this section we present the results obtained by the column generation algorithm when solving the LP relaxation of (4.1)-(4.5). The algorithm used in these experiments is the

[1] Instances are available on request.

algorithm in which columns of different scenarios are reused, as summarized in Algorithm 4.1. As initial routes in the RMP, we use routes visiting a single node, i.e., routes of the form $(0, w_0) - (v, w) - (n + 1, w_{n+1})$ for all $(v, w) \in \hat{V}$ and for all scenarios $\omega \in \Omega$. We will distinguish between using only the exact algorithm to generate routes, and using the tabu search heuristic to generate routes as well. No valid inequalities are added during these experiments.

Table 4.1 shows the results of using Algorithm 4.1, without the tabu search algorithm, for the case where all cycles are allowed, when only ng-paths are allowed for a neighbourhood size of $\Delta_{ng} = 5$, and when only elementary paths are allowed. Recall that the same implementation of the algorithm can be used for these route relaxations by setting $\Delta_{ng} = 1$, $\Delta_{ng} = 5$ and $\Delta_{ng} = n$, respectively.

The first and second columns of Table 4.1 show the number of the instance and the number of customers in this instance. For each instance and each type of pricing problem, we report the total time in seconds needed by the column generation algorithm to solve the LP relaxation of the instance (T.Time), the time spent on solving pricing problems (P.Time), the number of column generation iterations needed (Iter.), and the LP value of the instance (LP).

Observe that almost all of the computation time is spent on solving the pricing problems. For four of the instances with 25 customers, the time limit is exceeded before solving the LP relaxation, when using only elementary paths.

When comparing the use of elementary paths versus allowing all cycles, we observe that the computation times are in general significantly faster when all cycles are allowed but the LP values are significantly lower. When using ng-paths with $\Delta_{ng} = 5$, the LP values are very close to those obtained when using elementary paths. Moreover, for the largest instances, the computation times are significantly lower than when using elementary shortest paths. Hence, using ng-paths provides bounds that are comparable to those obtained when using elementary paths, in much less time.

Table 4.2 shows the results of using the column generation algorithm in which the tabu search algorithm is used to find routes with negative reduced costs heuristically. We use the settings $TS_{\text{tabu}} = 5$, $TS_{It} = 15$ and $TS^{max} = 150$.

When comparing the results in Tables 4.1 and 4.2, one can observe a significant decrease in computation time when using the tabu search heuristic in the elementary route case. In this case, all instances are now solved within the time limit of one hour. When the ng-route relaxation is used, a smaller decrease in computation time is observed. When all cycles are allowed, using the tabu search algorithm leads to an increase in computation time in many instances. Recall that the tabu search heuristic generates only elementary

Table 4.1: Column generation experiment results, without tabu search

		All cycles allowed				ng-paths with $\Delta_{ng} = 5$				Elementary paths			
Inst.	$\lvert V'\rvert$	T.Time	P.Time	Iter.	LP	T.Time	P.Time	Iter.	LP	T.Time	P.Time	Iter.	LP
1	10	25.43	25.29	18	9.80	6.86	6.57	28	12.78	7.99	7.71	28	12.79
2	10	3.39	3.28	17	15.19	3.04	2.73	32	16.67	3.67	3.26	30	16.67
3	10	1.06	1.03	16	12.02	1.40	1.23	33	16.53	1.76	1.59	32	16.53
4	10	1.73	1.67	21	14.87	4.31	3.93	37	15.70	6.66	6.30	36	15.83
5	10	1.81	1.64	32	17.61	1.97	1.63	48	19.65	2.31	1.98	45	19.65
6	10	0.86	0.81	21	16.09	1.59	1.33	32	18.06	1.83	1.61	33	18.06
7	10	2.01	1.81	28	11.36	2.15	1.81	30	12.17	4.38	4.20	31	12.17
8	10	1.50	1.31	28	15.16	2.29	1.82	38	17.09	2.32	2.04	43	17.09
9	10	1.92	1.79	31	16.91	1.98	1.54	47	19.78	1.83	1.53	39	19.78
10	10	0.89	0.81	20	14.92	1.40	1.09	26	17.17	1.47	1.19	33	17.17
11	15	8.27	7.92	29	20.77	12.61	12.03	33	22.22	44.99	44.45	33	22.23
12	15	4.54	4.35	27	22.12	8.86	8.37	33	24.86	14.27	13.82	34	24.86
13	15	9.24	9.00	22	18.41	12.67	12.03	26	21.36	28.39	27.86	28	21.41
14	15	38.55	38.08	26	15.31	42.85	41.43	40	18.08	134.41	132.99	39	18.08
15	15	8.72	8.44	24	21.34	16.46	15.84	40	24.15	33.68	33.06	32	24.26
16	15	22.93	22.37	29	16.65	29.53	28.37	38	19.11	103.72	102.27	43	19.11
17	15	16.72	16.04	30	20.49	30.65	28.92	52	21.45	141.79	140.29	49	21.53
18	15	16.44	16.18	21	19.56	21.42	20.25	38	22.49	38.74	37.81	44	22.55
19	15	6.77	6.58	28	20.51	8.94	8.38	33	22.58	19.91	19.25	33	22.65
20	15	10.28	10.01	22	17.00	15.44	14.70	35	18.29	69.42	68.50	30	18.30
21	20	15.66	15.21	29	24.78	36.11	34.63	42	27.46	176.25	174.95	32	27.54
22	20	73.96	72.20	37	23.30	223.78	221.14	35	25.23	2061.23	2058.81	39	25.27
23	20	45.19	44.29	35	22.49	153.04	150.59	42	25.90	1162.78	1160.08	47	25.91
24	20	22.42	21.59	27	29.50	38.41	37.25	35	31.13	166.98	165.80	31	31.16
25	20	32.95	32.14	27	25.54	56.43	54.43	44	27.56	145.28	143.21	40	27.70
26	20	17.49	16.79	34	24.07	51.48	48.92	43	26.95	402.75	399.91	43	26.97
27	20	58.47	57.60	25	24.98	156.16	154.70	29	26.76	1334.75	1333.41	34	26.82
28	20	45.41	44.24	34	25.15	63.34	61.01	42	26.22	1029.01	1026.10	46	26.23
29	20	64.93	63.87	26	26.30	94.58	93.30	28	28.61	367.40	366.21	30	28.74
30	20	35.35	34.41	27	21.71	73.54	71.36	39	23.24	1626.60	1624.22	43	23.25
31	25	73.94	71.87	34	33.65	178.12	175.02	40	35.02	2108.34	2105.72	34	35.14
32	25	90.18	87.86	30	29.48	150.37	145.88	47	31.37	1170.66	1165.34	51	31.56
33	25	169.32	166.78	34	27.97	276.76	273.67	31	30.45	3600.00	3600.00	1	-
34	25	39.55	38.91	28	30.56	94.21	92.20	36	33.18	1584.22	1582.39	35	33.27
35	25	93.62	92.38	33	27.86	193.52	189.53	45	30.03	3600.00	3600.00	1	-
36	25	129.92	127.83	30	29.76	236.79	232.68	48	31.62	2823.07	2819.57	40	31.69
37	25	110.81	108.98	30	24.62	250.16	245.17	51	27.17	3600.00	3600.00	1	-
38	25	60.72	58.81	35	32.16	151.21	146.47	49	34.07	942.12	938.66	43	34.14
39	25	124.61	122.74	32	31.69	239.12	235.84	39	33.46	2476.24	2473.37	37	33.51
40	25	121.01	119.37	30	28.09	269.54	266.53	36	29.66	3600.00	3600.00	1	-

routes. Therefore the routes produced by this heuristic may be less useful when cycles are allowed. Note that we also developed a similar tabu search algorithm for generating ng-routes. It was not successful because checking whether a route is an ng-route is computationally expensive.

All results presented in the next sections were obtained using the tabu search heuristic as well as the ng-route relaxation. Moreover, preliminary experiments with various values of Δ_{ng} showed that the algorithm yields its best results for $\Delta_{ng} = 5$.

4.4.3 Branch-price-and-cut results

Next, we present the results of our experiments using the exact branch-price-and-cut algorithm. Table 4.3 reports the results obtained when only capacity inequalities are considered. The column Opt.Gap shows the percentage difference between the best obtained upper and lower bounds after termination of the algorithm. The column LP Gap shows the percentage difference between the value of the LP relaxation, without adding valid inequalities, and the best upper bound found. The column Root Gap specifies the same

Table 4.2: Column generation experiment results, with tabu search

		All cycles allowed				ng-paths with $\Delta_{ng} = 5$				Elementary paths			
Inst.	$\lvert V' \rvert$	T.Time	P.Time	Iter.	LP	T.Time	P.Time	Iter.	LP	T.Time	P.Time	Iter.	LP
1	10	21.01	19.92	103	9.80	2.87	2.53	47	12.78	1.90	1.62	34	12.79
2	10	8.42	7.41	97	15.19	1.31	0.99	46	16.67	1.59	1.26	50	16.67
3	10	4.34	3.67	86	12.02	1.25	0.81	63	16.53	1.23	0.86	50	16.53
4	10	5.40	4.51	94	14.87	2.75	2.12	79	15.70	2.34	1.84	73	15.83
5	10	4.59	3.58	105	17.61	1.61	1.08	71	19.65	1.53	0.98	68	19.65
6	10	4.57	3.84	90	16.09	0.98	0.70	46	18.06	1.00	0.67	46	18.06
7	10	7.00	6.15	124	11.36	1.23	0.95	67	12.17	1.31	1.05	70	12.17
8	10	7.66	6.53	119	15.16	1.83	1.53	79	17.09	1.48	1.11	75	17.09
9	10	6.12	4.93	120	16.91	1.79	1.42	63	19.78	2.18	1.67	73	19.78
10	10	4.29	3.38	99	14.92	0.95	0.47	53	17.17	1.11	0.53	62	17.17
11	15	20.00	17.55	110	20.77	3.35	2.53	29	22.22	4.07	3.17	31	22.23
12	15	13.21	11.24	86	22.12	3.39	2.56	33	24.86	3.46	2.71	33	24.86
13	15	18.49	16.72	103	18.41	3.39	2.69	37	21.36	3.17	2.59	34	21.41
14	15	45.01	40.18	164	15.31	8.39	5.87	79	18.08	8.02	5.64	77	18.08
15	15	26.60	23.46	116	21.34	7.10	5.53	57	24.15	6.32	4.99	50	24.26
16	15	37.03	32.53	143	16.65	9.94	8.32	67	19.11	9.31	7.38	58	19.11
17	15	30.03	26.42	119	20.49	7.16	5.43	52	21.45	7.69	6.42	44	21.53
18	15	24.37	21.83	103	19.56	8.08	6.49	69	22.49	6.96	6.21	49	22.55
19	15	18.44	16.61	106	20.51	3.92	3.15	41	22.58	3.37	2.75	35	22.65
20	15	21.36	18.28	122	17.00	5.87	4.57	51	18.29	6.79	5.48	54	18.30
21	20	35.90	30.64	124	24.78	8.56	6.49	42	27.46	6.27	4.57	33	27.54
22	20	72.59	65.06	133	23.30	46.22	41.32	100	25.23	104.72	101.42	66	25.27
23	20	84.13	74.98	196	22.49	12.45	9.64	50	25.9	16.23	13.56	47	25.91
24	20	35.55	30.53	116	29.50	12.06	9.48	54	31.13	11.36	9.05	47	31.16
25	20	52.34	45.87	134	25.54	9.75	7.49	46	27.56	10.28	8.25	45	27.70
26	20	48.31	41.46	128	24.07	14.71	11.68	55	26.95	29.75	26.68	64	26.97
27	20	56.07	50.81	113	24.98	29.86	27.34	56	26.76	44.48	42.46	43	26.82
28	20	46.50	41.39	117	25.15	14.62	11.58	61	26.22	10.67	8.10	46	26.23
29	20	60.34	53.14	153	26.30	12.86	10.64	45	28.61	8.36	6.77	31	28.74
30	20	67.89	61.36	142	21.71	14.99	12.28	56	23.24	16.10	13.68	47	23.25
31	25	80.04	71.14	143	33.65	42.71	37.88	78	35.02	47.55	44.38	44	35.14
32	25	133.01	116.34	225	29.48	39.72	32.69	91	31.37	64.93	59.54	69	31.56
33	25	135.53	124.32	155	27.97	47.00	41.11	79	30.45	344.21	339.67	67	30.60
34	25	73.83	66.83	154	30.56	15.55	12.67	44	33.18	20.44	18.07	38	33.27
35	25	129.25	120.55	143	27.86	34.23	29.90	62	30.03	41.50	38.37	41	30.04
36	25	110.64	100.48	162	29.76	26.38	21.71	63	31.62	44.77	41.11	50	31.69
37	25	192.65	176.89	194	24.62	62.90	55.74	91	27.17	163.25	157.09	72	27.22
38	25	81.14	71.37	151	32.16	24.82	20.04	70	34.07	21.90	18.46	50	34.14
39	25	136.41	124.78	164	31.69	56.36	50.76	80	33.46	91.06	87.11	56	33.51
40	25	129.25	119.31	186	28.09	28.77	25.09	61	29.66	121.38	118.08	55	29.71

difference but after adding valid inequalities. The column Nodes indicates the number of nodes processed in the search tree and the column CI gives the number of added capacity cuts.

Observe that the total computation time increases rapidly with the number of customers in the instances. Three of the instances with 20 customers could not be solved within one hour and eight of the instances with 25 customers could not be solved. For four 10-customer instances, the LP bound is already tight. For fourteen more instances the gap is completely closed by adding capacity cuts, including the instance with the largest (observed) LP gap.

Table 4.4 shows the results of using the branch-price-and-cut algorithm while adding both the capacity inequalities and the subset row inequalities. Recall that subset row inequalities are only separated when no violated capacity inequalities are identified. We limit the subset row inequalities that we add as described in Section 4.3.4. We use the settings $SR_v^{\max} = 5$, $SR_{It}^{\max} = 10$, $SR^{\max} = 30$ and $SR_{Vio}^{\min} = 0.1$. In this table, the column SRI indicates the number of generated subset row inequalities.

Table 4.3: Branch-price-and-cut experiment results, with capacity inequalities only

Inst.	$\lvert V'\rvert$	Tot.Time	Opt.Gap	LP Gap	Root Gap	Nodes	CI
1	10	4.85	0	0.4	0	2	1
2	10	2.18	0	1.04	0	1	6
3	10	1.29	0	0.47	0	1	3
4	10	20.48	0	1.61	0.58	23	24
5	10	1.59	0	0	0	1	0
6	10	1.54	0	0.38	0	1	6
7	10	1.23	0	0	0	1	0
8	10	1.84	0	0	0	1	0
9	10	3.41	0	1.78	0	4	7
10	10	0.95	0	0	0	1	0
11	15	726.36	0	3.56	0.47	345	59
12	15	6.00	0	1.6	0	1	31
13	15	16.10	0	3.44	0	2	27
14	15	78.40	0	2.06	0	10	4
15	15	701.79	0	2.93	1.01	329	29
16	15	82.74	0	3.59	0.09	7	16
17	15	344.62	0	2.36	0.52	69	55
18	15	66.48	0	1.9	0	19	12
19	15	33.64	0	2.39	0	8	44
20	15	35.44	0	2.9	0	7	26
21	20	32.94	0	1.87	0	2	29
22	20	1438.67	0	1.59	0.34	125	56
23	20	3080.33	0	2.39	0.13	216	87
24	20	894.09	0	3.8	0.36	223	106
25	20	105.59	0	4.44	0	3	54
26	20	150.07	0	0.12	0.11	21	4
27	20	3600.00	0.06	2.87	0.33	233	81
28	20	3600.00	0.02	1.16	0.41	346	114
29	20	654.39	0	2.97	0.29	46	112
30	20	3600.00	0.57	1.69	0.93	307	118
31	25	1460.91	0	1.26	0.1	44	184
32	25	3600.00	-	-	-	346	110
33	25	3600.00	-	-	-	163	138
34	25	3600.00	-	-	-	504	118
35	25	3600.00	-	-	-	97	132
36	25	3600.00	-	-	-	140	648
37	25	3600.00	-	-	-	98	135
38	25	204.20	0	2.16	0	5	88
39	25	3600.00	-	-	-	110	178
40	25	3600.00	-	-	-	152	166

Three instances (28, 30 and 40) that were previously unsolved are now solved by adding subset row inequalities. Out of the twenty other instances in which subset row inequalities were added, seven instances were solved faster than without adding them, eight remain unsolved, while the others required more computation time. The LP gap of one additional instance is closed after adding subset row inequalities.

Adding subset row inequalities improves the lower bounds that are obtained and ensures that less nodes have to be evaluated in the branching tree. However, the additional time spent on solving the pricing problems as a result of adding these inequalities often outweighs the gains of these improved bounds.

4.4.4 Comparison with current practice

In practice, the DTWAVRP is often heuristically solved as follows. A vehicle routing problem with multiple time windows is solved, where the time windows for each customer are its candidate time windows and demand is the average over all scenarios. Note that when there is no time between subsequent candidate time windows, the problem reverts

Table 4.4: Branch-price-and-cut experiment results, with capacity and subset row inequalities

Inst.	$\|V'\|$	Tot.Time	Opt.Gap	LP Gap	Root Gap	Nodes	CI	SRI
1	10	4.88	0	0.4	0	2	1	0
2	10	2.02	0	1.04	0	1	6	0
3	10	1.32	0	0.47	0	1	3	0
4	10	26.31	0	1.61	0.58	16	24	17
5	10	1.59	0	0	0	1	0	0
6	10	1.56	0	0.38	0	1	6	0
7	10	1.23	0	0	0	1	0	0
8	10	1.82	0	0	0	1	0	0
9	10	3.44	0	1.78	0	4	7	0
10	10	0.96	0	0	0	1	0	0
11	15	507.00	0	3.56	0.27	119	41	30
12	15	5.91	0	1.6	0	1	31	0
13	15	15.92	0	3.44	0	2	27	0
14	15	78.51	0	2.06	0	10	4	0
15	15	1364.49	0	2.93	1.01	221	28	30
16	15	70.60	0	3.59	0.08	3	16	10
17	15	501.76	0	2.36	0.27	17	31	30
18	15	66.83	0	1.9	0	21	12	5
19	15	32.93	0	2.39	0	8	44	0
20	15	35.03	0	2.9	0	7	26	0
21	20	32.94	0	1.87	0	2	29	0
22	20	1225.18	0	1.59	0.12	19	43	26
23	20	3079.46	0	2.39	0.13	216	87	0
24	20	594.39	0	3.8	0.21	86	105	25
25	20	105.65	0	4.44	0	3	54	0
26	20	138.12	0	0.12	0	3	1	10
27	20	3600.00	-	-	-	7	75	27
28	20	3284.31	0	1.16	0.2	91	61	30
29	20	343.38	0	2.97	0.07	11	84	20
30	20	2425.43	0	1.33	0.33	147	96	30
31	25	820.01	0	1.26	0.01	7	162	17
32	25	3600.00	-	-	-	157	98	30
33	25	3600.00	-	-	-	30	118	30
34	25	3600.00	-	-	-	174	78	30
35	25	3600.00	-	-	-	46	113	24
36	25	3600.00	-	-	-	68	110	30
37	25	3600.00	-	-	-	25	101	30
38	25	214.90	0	2.16	0	5	88	2
39	25	3600.00	-	-	-	81	172	27
40	25	489.06	0	3.49	0	4	101	10

to a vehicle routing problem with a single time window. The time window within which service commences at each customer is selected for the time window assignment.

We have implemented this procedure and used it to obtain solutions for our instances. To evaluate the expected costs of the time window assignment that is obtained, a VRPTW is solved for each scenario using the assigned time windows.

Table 4.5 shows the difference between the quality of the solutions obtained by this procedure and that of the optimal solutions obtained using the branch-price-and-cut algorithm. The column Value gives the expected costs of using the heuristic procedure, Opt. gives the optimal expected costs of the instance, and Gap provides the percentage difference between the heuristic solution value and the optimal one. Note that only the instances for which an optimal solution was found are included in Table 4.5.

Only for one instance does the heuristic find an optimal time window assignment. The difference between the optimal solution value and the solution value found by the heuristic is up to 7.01% for these instances with an average difference of 3.32%. Table 4.5 suggests that the difference increases with the number of customers.

Table 4.5: Current practice experiment results

Inst.	$\|V'\|$	Value	Opt.	Gap
1	10	12.87	12.83	0.31
2	10	17.53	16.84	4.10
3	10	16.72	16.60	0.72
4	10	15.96	15.96	0
5	10	20.11	19.65	2.34
6	10	18.38	18.13	1.28
7	10	12.35	12.17	1.48
8	10	17.44	17.09	2.05
9	10	20.54	20.14	1.99
10	10	17.54	17.17	2.15
			Average gap:	1.65
11	15	23.47	23.04	1.87
12	15	25.73	24.27	6.02
13	15	23.15	22.12	4.66
14	15	18.73	18.46	1.46
15	15	25.47	24.87	2.41
16	15	20.73	19.82	4.59
17	15	22.80	21.96	3.83
18	15	23.40	22.93	2.05
19	15	24.15	23.14	4.36
20	15	19.16	18.84	1.70
			Average gap:	3.29
21	20	29.54	27.99	5.54
22	20	26.97	25.63	5.23
23	20	27.25	26.53	2.71
24	20	33.79	32.36	4.42
25	20	30.85	28.84	6.97
26	20	27.97	26.99	3.63
28	20	28.39	26.53	7.01
29	20	30.15	29.49	2.24
30	20	24.53	23.55	4.16
			Average gap:	4.65
31	25	37.02	35.46	4.40
38	25	36.80	34.83	5.57
40	25	32.30	30.73	5.11
			Average gap:	5.05

4.5 Conclusion

In this chapter, we have introduced a new problem, the DTWAVRP. We have developed an exact branch-price-and-cut algorithm to solve it. The column generation algorithm exploits the fact that columns for one scenario can be reused in another scenario. Furthermore, we use an ng-route relaxation to speed up the pricing problem while limiting the decrease of the LP value and we also generate columns using a tabu search heuristic. Finally, the branch-price-and-cut algorithm incorporates valid inequalities that are known from vehicle routing, namely, capacity and subset row inequalities.

We are able to solve instances of up to 25 customers and 3 scenarios. Moreover, the experiments show that using the exact algorithm for the instances presented in this chapter, provides a decrease of up to 7.01% in expected costs with respect to current practice.

In the future, various research directions ensuing from this work can be explored. One of them would be to consider customer preferences on the candidate time windows that can be assigned to them and to include in the DTWAVRP a secondary objective consisting of maximizing the customer preference satisfaction. Another line of research would be to enhance the proposed method or develop a new one to tackle instances involving a large number of scenarios. Finally, it would be interesting to devise a heuristic based on the proposed branch-price-and-cut algorithm that would be able to solve instances involving more than 25 customers. In particular, one may think about branching directly and only on the time window variables as the assigned time windows are the only decisions imposed following the solution of the DTWAVRP.

Chapter 5

The Driver Assignment Vehicle Routing problem

5.1 Introduction

The capacitated vehicle routing problem, CVRP, is the problem of designing routes for vehicles with limited capacity to deliver goods to customers in a distribution network, such that the total transportation costs are minimized. This is a well studied problem in the scientific literature, see Baldacci et al. (2012), Laporte (2009) and Toth and Vigo (2002) amongst others for a survey on exact and heuristic methods to solve the CVRP.

In distribution networks where each customer frequently receives a delivery, it is often desired that the same driver makes these deliveries. The quality of service benefits from regularity and personalization by having the same driver visit a customer, as is suggested by Bertsimas and Simchi-Levi (1996). Moreover, Groër et al. (2009) indicate that because drivers at UPS form a real bond with customers they generate additional sales with a volume of over 60 million packages per year. In this chapter we focus on distribution networks in which the driver is also responsible for unloading the shipment and placing them in the storage facility of the customer, e.g. as is the case for the service provided by TNT Innight. This requires the driver to carry a key or password to enter the storage facility, which increases the need of a customer to be visited by the same driver. Moreover, security screening of drivers in this case, further increases this need.

In this chapter we study the problem of assigning customers to drivers before the quantity to be delivered to these customers is known. We consider a set of demand scenarios, and for each scenario a delivery schedule has to be made which minimizes the transportation costs while satisfying the vehicle capacity constraints. Furthermore, the delivery schedules per scenario should be such that at least a fraction α of the customers that are assigned to a driver is actually visited by that driver, where α is provided by the decision maker. The driver assignment vehicle routing problem, DAVRP, is to assign customers to drivers such that the expected transportation costs over all scenarios are minimized. The DAVRP is NP-hard as in the case of one scenario it reduces to the CVRP.

The DAVRP is similar to the consistent vehicle routing problem, ConVRP, introduced by Groër et al. (2009). In the ConVRP each customer must also always be visited by the same driver. However, it is additionally required that the time of delivery for a single customer cannot differ by more than a limited amount of time per scenario. In the DAVRP we do not consider the timing of deliveries as this is not relevant in the application on which we focus. Moreover, the decision maker is allowed more flexibility by setting an appropriate α. Groër et al. (2009) report finding optimal solutions to the ConVRP of instances with up to 12 customers and 3 scenarios using a commercial mixed integer programming solver, and they design a local search heuristic which they use to solve instances with over 3700 customers.

In another related study, Li et al. (2009) consider the rescheduling of bus trips in case of a disruption. In their model, they incorporate a penalty for assigning drivers to a trip they are unfamiliar with. They design a Lagrangian heuristic to solve their problem.

The main contributions of this chapter are the following. We propose a new and relevant problem, the DAVRP. Secondly, we design a cluster first-route second heuristic and use it to find good solutions to the DAVRP for instances with up to 100 customers and instances with up to 100 scenarios. Thirdly, in our computational experiments we study the costs of adhering to the driver assignments. We compare the costs of always having a customer visited by the same driver, with the costs of relaxing this requirement entirely. Such an analysis aids a policy maker in determining whether it is worthwhile to require customers to be visited by the same driver. Furthermore, using two variants of the cluster first-route second algorithm, we study the increase in transportation costs of only constructing new routes with customers that cannot be visited by their assigned drivers, instead of trying to assign them to different drivers.

The outline of this chapter is the following. In the next section, the DAVRP is formally defined. In Section 5.3, the cluster first-route second heuristic is presented. We provide the results of our computational experiments in Section 5.4, and we end with our conclusions in Section 5.5.

5.2 Problem definition

Consider a complete graph $G = (V, E)$, where $V = \{0, ..., n\}$ is a set of locations such that 0 represents the depot and $V' = \{1, ..., n\}$ are the customers. A *route* is a path in G starting and ending at the depot. A *routing schedule* is a collection of routes such that each customer is visited exactly once.

Let $c_{ij} \geq 0$ be the cost to travel along edge (i, j). Hence, the costs of a routing schedule is the sum of the edges that are used on the routes. The travel costs satisfy the triangle inequality.

Let K be the set of available vehicles, each having a capacity of Q. Without loss of generality, let the set of vehicles be ordered, $K = \{k_1, ..., k_{|K|}\}$. In our model there is no distinction between a driver and a vehicle. Therefore, we will use the term *driver* and *vehicle* interchangeably throughout this chapter. Each driver will drive at most one route. Moreover, in this chapter we consider $|K| = n$.

Furthermore, a set Ω of scenarios is given, where each scenario is characterized by a realization of demand. Let demand at location i in scenario $\omega \in \Omega$ be given by the integer q_i^ω such that $1 \leq q_i^\omega \leq Q$. Let the probability that scenario ω occurs be p^ω.

A *driver assignment* is an assignment of every customer to a driver. Note that not every driver necessarily has a customer assigned to it. Given a driver assignment, a routing schedule is considered feasible for scenario ω if for every driver at least a fraction α of the customers assigned to it is visited by that driver and additionally when every route satisfies the vehicle capacity constraint. A driver assignment is considered feasible if for every scenario there exists at least one feasible routing schedule. The driver assignment vehicle routing problem, DAVRP, is to find a feasible driver assignment and a feasible routing schedule for every scenario such that the expected traveling costs over all scenarios are minimized.

Next, we provide a mixed integer linear programming formulation of the DAVRP. Let a_{ik}, for all $i \in V'$ and $k \in K$, indicate whether customer i is assigned to driver k. Let x_{ijk}^ω, for all $i, j \in V$, $k \in K$ and $\omega \in \Omega$, indicate whether driver k travels from customer i to j in scenario ω. Furthermore, let f_{ijk}^ω, for all $i, j \in V$, $k \in K$ and $\omega \in \Omega$, be the commodity flow between customer i and j on vehicle k in scenario ω. Finally, let d_{ik}^ω, for all $i \in V'$, $k \in K$ and $\omega \in \Omega$, indicate whether customer i is assigned to vehicle k but is visited by another vehicle in scenario ω. The DAVRP can be formulated as follows.

$$\min \sum_{\omega\in\Omega, i,j\in V, k\in K} p^{\omega} c_{ij} x^{\omega}_{ijk} \tag{5.1}$$

$$\sum_{k\in K} a_{ik} = 1 \qquad \forall i \in V' \tag{5.2}$$

$$\sum_{j\in V, k\in K} x^{\omega}_{ijk} = 1 \qquad \forall i \in V', \forall \omega \in \Omega \tag{5.3}$$

$$\sum_{j\in V'} x^{\omega}_{0jk} \leq 1 \qquad \forall k \in K, \forall \omega \in \Omega \tag{5.4}$$

$$\sum_{j\in V} x^{\omega}_{ijk} = \sum_{j\in V} x^{\omega}_{jik} \qquad \forall i \in V, \forall k \in K, \forall \omega \in \Omega \tag{5.5}$$

$$\sum_{j\in V} (f^{\omega}_{jik} - f^{\omega}_{ijk}) = q^{\omega}_i \sum_{j\in V} x^{\omega}_{ijk} \qquad \forall i \in V', \forall k \in K, \forall \omega \in \Omega \tag{5.6}$$

$$f^{\omega}_{ijk} \leq (Q - q^{\omega}_i) x^{\omega}_{ijk} \qquad \forall i,j \in V, \forall k \in K, \forall \omega \in \Omega \tag{5.7}$$

$$f^{\omega}_{ijk} \geq q^{\omega}_j x^{\omega}_{ijk} \qquad \forall i,j \in V, \forall k \in K, \forall \omega \in \Omega \tag{5.8}$$

$$a_{ik} - \sum_{j\in V} x^{\omega}_{ijk} \leq d^{\omega}_{ik} \qquad \forall i \in V', \forall k \in K, \forall \omega \in \Omega \tag{5.9}$$

$$\sum_{i\in V'} d^{\omega}_{ik} \leq (1-\alpha) \sum_{i\in V'} a_{ik} \qquad \forall k \in K, \forall \omega \in \Omega \tag{5.10}$$

$$\sum_{i\in V', k\in K} x^{\omega}_{0ik} \geq \left\lceil \frac{\sum_{i\in V'} q^{\omega}_i}{Q} \right\rceil \qquad \forall \omega \in \Omega \tag{5.11}$$

$$\sum_{i\in V'} a_{ik_l} \geq \sum_{i\in V'} a_{ik_{l+1}} \qquad \forall k_l \in K \setminus \left\{k_{|K|}\right\} \tag{5.12}$$

$$\sum_{i\in V} x^{\omega}_{0ik_l} \geq \sum_{i\in V} x^{\omega}_{0ik_{l+1}} \qquad \forall k_l \in K \setminus \left\{k_{|K|}\right\}, \forall \omega \in \Omega \tag{5.13}$$

$$a_{ik} \in \{0,1\} \qquad \forall i \in V', \forall k \in K \tag{5.14}$$

$$x^{\omega}_{ijk} \in \{0,1\} \qquad \forall i,j \in V, \forall k \in K, \forall \omega \in \Omega \tag{5.15}$$

$$f^{\omega}_{ijk} \in [0,Q] \qquad \forall i,j \in V, \forall k \in K, \forall \omega \in \Omega \tag{5.16}$$

$$d^{\omega}_{ik} \in \{0,1\} \qquad \forall i \in V', \forall k \in K, \forall \omega \in \Omega \tag{5.17}$$

The objective function is given by (5.1). Constraints (5.2) ensure that each customer is assigned to a driver.

Constraints (5.3)-(5.8) are the vehicle routing constraints that make sure that any integer solution represent a routing schedule for each scenario. These constraints are based on a formulation of the heterogenous vehicle routing problem, HVRP. The HVRP is a vehicle routing problem in which a heterogenous fleet of vehicles is available. In this problem customers have to be assigned to specific vehicles among a collection of vehicles with different capacities. The HVRP is to construct a routing schedule that minimizes the total traveling and vehicle usage costs, while adhering to this assignment and the capacity constraints. Six formulations of the HVRP are provided and compared by Yaman (2006). Constraints (5.3)-(5.8) are based on $HVRP6$, which is the strongest formulation of these six.

Constraints (5.3) make sure that each customer is departed from exactly once in every scenario and constraints (5.4) make sure that each driver drives at most one route. Each driver departs from a location as often as it arrives there due to (5.5).

Constraints (5.6)-(5.8) are the commodity flow constraints. They prohibit the violation of the capacity constraints in any scenario. Moreover, they ensure that in each scenario no subtours are allowed in any integer solution. Constraints (5.6) ensure that for each scenario ω and driver k the flow into location i is exactly its demand q_i^ω if driver k is used in scenario ω to visit customer i. The vehicle capacity is never violated by the flow from node i to j due to constraints (5.7). Moreover, constraints (5.8) ensure that if driver k travels from i to j in scenario ω, the flow is at least the demand of j.

Constraints (5.9) ensure that when customer i is assigned to driver k but is not visited by that driver in scenario ω then $d_{ik}^\omega = 1$. Constraints (5.10) ensure that at most a fraction $1 - \alpha$ of the customers assigned to driver k may be visited in scenario ω by another driver. This is of course equivalent with the requirement that at least a fraction α of the customers assigned to driver k are visited in scenario ω by that driver.

Constraints (5.11)-(5.13) are not required for the validity of the formulation. Constraints (5.11) are valid inequalities known for vehicle routing problems which strengthen the LP bound. They force a minimum number of vehicles to be used in each scenario. Constraints (5.12) and (5.13) are symmetry breaking constraints that might speed up a branching procedure. Constraints (5.12) ensure that a driver always gets assigned at least the same amount of customers as the next driver. Constraints (5.13) ensure that in each scenario ω, a driver can only be used whenever the previous driver is also in use.

The variable domains are specified by (5.14)-(5.17). Note that the formulation remains valid when we relax integrality on d_{ik}^ω.

In Section 5.4, we present results of computational experiments in which the DAVRP is solved using formulation (5.1)-(5.17) and a commercial mixed integer programming solver. Next, we describe a cluster first-route second heuristic to quickly find good solutions to the DAVRP.

5.3 Solution method

To quickly find solutions to the DAVRP with a large number of customers and scenarios, we propose a heuristic. In this heuristic we decouple the driver assignment and the routing in each scenario. It is a two-phase approach that is similar to cluster first-route second heuristics, which are a well known family of heuristics for vehicle routing problems. In the first phase of cluster first-route second heuristics for the vehicle routing problem, customers are clustered and in the second phase a routing schedule is constructed based on these clusters. For the CVRP, one typically ensures for every cluster in the first phase that the total demand of the customers in a cluster does not exceed the capacity of a vehicle. This way, a feasible routing schedule can be obtained in the second phase by simply constructing a route for each cluster. Well known examples of cluster first-route second algorithms for the CVRP are provided by Fisher and Jaikumar (1981) and Bramel and Simchi-Levi (1995).

In this section we describe a cluster first-route second algorithm for the DAVRP. In Section 3.1 we describe an algorithm used in the first phase to construct clusters. In Section 3.2 we describe two algorithms that are used in the second phase to construct a routing schedule based on the clusters obtained in the first phase. In the first algorithm for the second phase, we allow customers that are not visited by their assigned driver to be assigned to another driver that already has customers assigned to it. In the second algorithm for the second phase, customers that are not visited by their assigned driver are used to construct routes for drivers that do not have any customers assigned to them.

5.3.1 Cluster first

In the first phase we construct clusters of customers. We require of every cluster that in each scenario at least one subset of customers, containing at least a fraction α of all customers in that cluster, has a total demand less or equal to the vehicle capacity. This allows us to use the clusters of customers as driver assignments, i.e. a feasible driver assignment is obtained by assigning all customers in one cluster (and no other customers) to a single driver. In the second phase we construct a routing schedule using the driver assignment obtained in the first phase.

The clustering problem

Next, we introduce the *clustering problem* which we solve to construct clusters. Consider a set of potential cluster centers, we will use the set of customers V' for this. When a cluster center is in use, costs are incurred equal to the traveling costs from the depot to a cluster center plus some penalty costs $\beta \geq 0$. Furthermore, all customers are assigned to a cluster. Note that a customer location can be used as a cluster center, while that customer itself is assigned to a different cluster. In each scenario a decision is made whether a customer is skipped. If a location is not skipped, costs are incurred equal to the traveling costs from that customer to its assigned cluster center, otherwise traveling costs to the depot are incurred. In every scenario, at least a fraction α of the customers in a cluster must not be skipped. Furthermore, in each scenario the capacity constraints must be satisfied by the locations in a cluster that are not skipped. The clustering problem is to select clusters, assign each customer to one of the selected clusters and select which customers to skip in each scenario, such that the total costs are minimized.

A solution of the clustering problem can directly be used as a feasible driver assignment. Moreover, note that the corresponding solution value (times 2) provides an upper bound on the solution value of any feasible solution to the DAVRP using this driver assignment. The optimal solution to the clustering problem minimizes this upper bound. Furthermore, β is added to the costs of using a cluster center to discourage the use of too many cluster centers.

Next we provide an integer linear programming formulation for the clustering problem. Let y_j, for all $j \in V'$, indicate whether potential cluster center j is selected. Let z_{ij}, for all $i, j \in V'$ indicate whether customer i is assigned to cluster center j. Finally, let z_{ij}^{ω}, for all $i, j \in V'$ and $\omega \in \Omega$, indicate whether location i is assigned to cluster center j and is not skipped in scenario ω. The clustering problem can be formulated as follows.

$$\min \sum_{j \in V'} (c_{0j} + \beta) y_j + \sum_{i,j \in V', \omega \in \Omega} p_{\omega} \left[c_{ij} z_{ij}^{\omega} + c_{0i}(z_{ij} - z_{ij}^{\omega}) \right] \tag{5.18}$$

$$\sum_{j \in V'} z_{ij} = 1 \qquad \forall i \in V' \tag{5.19}$$

$$z_{ij} \leq y_j \qquad \forall i, j \in V' \tag{5.20}$$

$$z_{ij}^{\omega} \leq z_{ij} \qquad \forall i, j \in V', \forall \omega \in \Omega \tag{5.21}$$

$$\sum_{i \in V'} q_i^{\omega} z_{ij}^{\omega} \leq Q y_j \qquad \forall j \in V', \forall \omega \in \Omega \tag{5.22}$$

$$\alpha \sum_{i \in V'} z_{ij} \leq \sum_{i \in V'} z_{ij}^{\omega} \qquad \forall j \in V', \forall \omega \in \Omega \tag{5.23}$$

$$y_j,\, z_{ij},\, z_{ij}^{\omega} \in \{0, 1\} \qquad \forall i, j \in V', \forall \omega \in \Omega \tag{5.24}$$

The objective function is given by (5.18). Constraints (5.19) ensure that each location is assigned to a cluster center. Furthermore, constraints (5.20) model the requirement that a location can only be assigned to a cluster center that is in use. It is ensured in (5.21) that a location cannot be assigned to a cluster in a specific scenario, if it not assigned to that cluster in general. Constraints (5.22) are the capacity constraints per scenario. Finally, constraints (5.23) ensure that at least $\alpha\%$ of the customers in a cluster are not skipped per scenario. Note that constraints (5.20) are not necessary for the validity of the formulation but serve to strengthen it.

Lower bounds for the clustering problem

To find a lower bound on the solution value of the clustering problem, we apply Lagrangian relaxation. Let λ and μ be the dual multipliers associated with constraints (5.21) and (5.23) respectively. Consider the Lagrangian relaxation of the clustering problem obtained by relaxing (5.21) and (5.23). In this relaxed problem, for each $j \in V'$ and $\omega \in \Omega$, if $y_j = 0$ then $z^{\omega}_{ij} = 0$ for all $i \in V'$ otherwise the optimal values of z^{ω}_{ij} can be found by solving the following knapsack problem.

$$v^{\omega}_j(\lambda,\mu) = \min \sum_{i\in V'} (p_{\omega}(c_{ij} - c_{0i}) + \lambda^{\omega}_{ij} - \mu^{\omega}_j) z^{\omega}_{ij} \tag{5.25}$$

$$\sum_{i\in V'} q^{\omega}_i z^{\omega}_{ij} \leq Q \tag{5.26}$$

$$z^{\omega}_{ij} \in \{0,1\} \qquad \forall i \in V' \tag{5.27}$$

For specific values of λ and μ, let $\theta(\lambda,\mu)$ be the value of the Lagrangian relaxation. It is obtained by solving the following uncapacitated facility location problem.

$$\begin{aligned}\theta(\lambda,\mu) = \min \sum_{j\in V'} \left(c_{0j} + \beta + \sum_{\omega\in\Omega} v^{\omega}_j(\lambda,\mu) \right) y_j + \\ + \sum_{i,j\in V'} (c_{0i} - \sum_{\omega\in\Omega} \lambda^{\omega}_{ij} + \alpha \sum_{\omega\in\Omega} \mu^{\omega}_j) z_{ij}\end{aligned} \tag{5.28}$$

$$\sum_{j\in V'} z_{ij} = 1 \qquad \forall i \in V' \tag{5.29}$$

$$z_{ij} \leq y_j \qquad \forall i,j \in V' \tag{5.30}$$

$$y_j,\, z_{ij} \in \{0,1\} \qquad \forall i,j \in V' \tag{5.31}$$

Computing $\theta(\lambda,\mu)$ entails solving $|V'|\cdot|\Omega|$ knapsack problems and one uncapacitated facility location problem. In our implementation we solve each knapsack problem using a standard dynamic programming algorithm, see also Keller et al. (2004). We solve the uncapacitated facility location problem using a commercial mixed integer programming solver. For more specialized solution procedures see for instance Erlenkotter (1978) and Körkel (1989).

To optimize the lower bound provided by $\theta(\lambda,\mu)$ we apply a subgradient optimization procedure. This procedure makes use of some upper bound to the clustering problem UB and a parameter γ. For some initial multipliers $\lambda(0)$ and $\mu(0)$, the lower bound $\theta(\lambda(0),\mu(0))$ is calculated. Next, the multipliers are updated and used to calculate a new lower bound. The multipliers are updated as follows.

$$\lambda_{ij}^{\omega}(t+1) = \max\left\{\lambda_{ij}^{\omega}(t) + \gamma\frac{\theta(\lambda(t),\mu(t)) - UB}{\sum_{i,j\in V',\omega\in\Omega}\left(z_{ij}-z_{ij}^{\omega}\right)^2}\left(z_{ij}-z_{ij}^{\omega}\right), 0\right\}$$
$$\mu_{j}^{\omega}(t+1) = \max\left\{\mu_{j}^{\omega}(t) + \gamma\frac{\theta(\lambda(t),\mu(t)) - UB}{\sum_{j\in V',\omega\in\Omega}\left(\sum_{i\in V'}\left(z_{ij}^{\omega}-\alpha z_{ij}\right)\right)^2}\left(\sum_{i\in V'}\left(z_{ij}^{\omega}-\alpha z_{ij}\right)\right), 0\right\}$$

This procedure is repeated iteratively. If the lower bound does not improve for γ_i iterations, then γ is decreased by a factor γ_f. The procedure terminates when either the relaxed solution is feasible for the clustering problem, or a preset optimality gap is obtained, or γ decreases below a certain threshold γ_t.

Note that this updating scheme of the multipliers does not guarantee convergence to optimality. However, it performs well in practice.

Upper bounds for the clustering problem

At every iteration of the subgradient optimization algorithm, a solution to the Lagrangian relaxation is obtained. When this relaxed solution is not feasible for the clustering problem, a heuristic is used to find a feasible solution by modifying the relaxed solution, yielding an upper bound. If the new upper bound is lower than UB, UB is replaced. Next, we describe the heuristic to obtain this upper bound.

Observe that initially constraints (5.21) and (5.23) might be violated by the relaxed solution. We attempt to repair this using a greedy procedure. First, we skip all customers that are not skipped in some cluster, but which are also not assigned to that cluster, i.e. if $z_{ij}^{\omega} = 1$ and $z_{ij} = 0$ then we set $z_{ij}^{\omega} = 0$. The current solution now satisfies constraints (5.21).

If (5.23) is violated, then there exists a customer i that is assigned to some cluster center j and is skipped in some scenario ω, i.e. $z_{ij} = 1$ and $z_{ij}^{\omega} = 0$. Next, customer i will be assigned to a different cluster center. Find all cluster centers that are in use and where customer i can be assigned to without violating (5.22) and (5.23), and select the cheapest cluster center j'. If cluster center i is not in use, compare the costs of assigning customer i to cluster center j' with the costs of using cluster center i. The cheapest option is executed. If location i cannot be assigned to a different cluster center, the heuristic fails to find a feasible solution. This step is repeated until (5.23) is no longer violated. The heuristic is summarized in Algorithm 5.1.

Algorithm 5.1 Heuristic to find a feasible solution to the clustering problem.

Require: A feasible solution to the Lagrangian relaxation (5.28)-(5.31).

for all $i, j \in V'$ and $\omega \in \Omega$ such that $z_{ij}^{\omega} = 1$ and $z_{ij} = 0$ **do**
  Set $z_{ij}^{\omega} = 0$.
end for

for all $j \in V'$ and $\omega \in \Omega$ such that (5.23) is violated **do**
  Select an $i \in V'$ such that $z_{ij} = 1$ and $z_{ij}^{\omega} = 0$.
  Set $z_{ij} = 0$ and $z_{ij}^{\omega} = 0$ for all $\omega \in \Omega$, i.e. remove location i from cluster j.
  Let j' be the cheapest cluster center where location i can feasibly be added to, in particular without violating (5.22) and (5.23).
  if no such j' exists or the costs of using location i as a cluster center are lower **then**
    if $y_i = 0$ **then**
      Create a new cluster center by setting $y_i = 1$, $z_{ii} = 1$ and $z_{ii}^{\omega} = 1$ for all $\omega \in \Omega$.
    else
      The algorithm fails to identify a feasible solution
    end if
  else
    Add location i to cluster j'.
  end if
end for

In our implementation, for all $j \in V'$ and $\omega \in \Omega$ such that (5.23) is violated we select location $i \in V'$ such that $z_{ij} = 1$ and $z_{ij}^{\omega} = 0$ as follows. Select the customer that is skipped most in cluster center j among all scenarios in which (5.23) is violated. Also note that in searching for a new cluster center to assign a customer to, it is sufficient for our purposes to merely try to assign the customer to that cluster center in different scenarios, without considering skipping customers that are currently not skipped. This does, however, increase the likeliness of the heuristic failing to identify a feasible solution.

The clustering algorithm

To solve the clustering problem we use a branch-and-bound algorithm, which we refer to as the *clustering algorithm*. Lower bounds are found using the subgradient optimization algorithm and upper bounds are found by using Algorithm 5.1 at each iteration of the subgradient optimization algorithm. Branching is done in such a way that for each node that is added to the branching tree one of the constraints (5.21) or (5.23), which is violated by a current solution to the Lagrangian relaxation, is no longer violated. Next, we explain this in more detail.

First, select a customer i, a cluster center j and scenario ω for which (5.21) is violated. Next, we add two new nodes to the search tree. In one node we set $z_{ij} = 0$ and $z_{ij}^{\omega'} = 0$ for all $\omega' \in \Omega$. In the other node we set $z_{ij} = 1$, $y_j = 1$ and $z_{ij'} = z_{ij'}^{\omega'} = 0$ for all $j' \in V' \backslash \{j\}$ and $\omega' \in \Omega \backslash \{\omega\}$. In both newly added nodes any solution to the Lagrangian relaxation now satisfies (5.21) for i, j and ω.

If constraints (5.21) are not violated in the current solution, we select a cluster center j and scenario ω for which (5.23) is violated. For every customer i such that $z_{ij} = 1$, a new node is added to the branching tree in which we set $z_{ij} = 0$ and $z_{ij}^{\omega'} = 0$ for all $\omega' \in \Omega$. Furthermore, for every customer i such that $z_{ij}^{\omega} = 0$, a new node is added to the branching tree in which we set $z_{ij}^{\omega} = 1$, $z_{ij} = 1$, $y_j = 1$ and $z_{ij'} = z_{ij'}^{\omega'} = 0$ for all $j' \in V' \backslash \{j\}$ and $\omega' \in \Omega \backslash \{\omega\}$.

In each iteration of the branch-and-bound algorithm, the node with the lowest lower bound is selected to be processed next.

5.3.2 Route second

The clustering algorithm as described in Section 5.3.1 provides clusters of customers. By assigning every customer in one cluster (and no other customers) to a single driver, a feasible driver assignment is obtained. In the routing phase of the cluster first-route second algorithm, this driver assignment is used to construct a feasible routing schedule for every scenario.

The *routing problem* is the problem of, given a feasible driver assignment, finding a feasible routing schedule for scenario ω that minimizes the traveling costs. Next, we provide an algorithm to solve the routing problem to optimality, referred to as the *exact* *routing algorithm*. We also present a heuristic algorithm to find a solution to the routing problem, referred to as the *heuristic routing algorithm*. In the heuristic routing algorithm, customers that are not visited by their assigned driver are used to construct new routes instead of trying to add them to drivers that already have customers assigned to it.

The exact routing algorithm

We provide a mixed integer linear programming formulation of the routing problem. It is obtained by considering the parts of (5.1)-(5.17) that pertain to scenario ω. Moreover, to make the formulation more compact, we introduce the set of drivers $\hat{K}$ containing all drivers $k \in K$ that are part of the driver assignment and containing an artificial driver $\hat{k}$. This artificial driver represents all drivers that do not have a customer assigned to it in the driver assignment. As such, the artificial driver may drive multiple routes. For ease of notation, let $k(i) \in \hat{K}$ be the driver to which customer i is assigned. Finally, let A_k be the number of customers assigned to driver k. The formulation is the following.

$$\min p^\omega \sum_{i,j\in V,k\in\hat{K}} c_{ij}x^\omega_{ijk} \tag{5.32}$$

$$\sum_{j\in V,k\in\hat{K}} x^\omega_{ijk} = 1 \qquad \forall i \in V' \tag{5.33}$$

$$\sum_{j\in V} x^\omega_{ijk} = \sum_{j\in V} x^\omega_{jik} \qquad \forall i \in V,\, \forall k \in \hat{K} \tag{5.34}$$

$$\sum_{j\in V'} x^\omega_{0jk} \leq 1 \qquad \forall k \in \hat{K}\backslash\{\hat{k}\} \tag{5.35}$$

$$\sum_{j\in V}(f^\omega_{jik} - f^\omega_{ijk}) = q^\omega_i \sum_{j\in V} x^\omega_{ijk} \qquad \forall i \in V',\, \forall k \in \hat{K} \tag{5.36}$$

$$f^\omega_{ijk} \leq (Q - q^\omega_i)x^\omega_{ijk} \qquad \forall i,j \in V,\, \forall k \in \hat{K} \tag{5.37}$$

$$f^\omega_{ijk} \geq q^\omega_j x^\omega_{ijk} \qquad \forall i,j \in V,\, \forall k \in \hat{K} \tag{5.38}$$

$$1 - \sum_{j\in V} x^\omega_{ijk(i)} \leq d^\omega_{ik(i)} \qquad \forall i \in V' \tag{5.39}$$

$$\sum_{i\in V':k(i)=k} d^\omega_{ik} \leq (1-\alpha)A_k \qquad \forall k \in \hat{K}\backslash\{\hat{k}\} \tag{5.40}$$

$$\sum_{i\in V',k\in K^M} x^\omega_{0ik} \geq \left\lceil \frac{\sum'_{i\in V} q^\omega_i}{Q} \right\rceil \tag{5.41}$$

$$x^\omega_{ijk} \in \{0,1\} \qquad \forall i,j \in V,\, \forall k \in \hat{K} \tag{5.42}$$

$$d^\omega_{ik(i)} \in \{0,1\} \qquad \forall i \in V \tag{5.43}$$

$$f^\omega_{ijk} \in [0,Q] \qquad \forall i \in V,\, \forall k \in \hat{K} \tag{5.44}$$

The interpretation of (5.32)-(5.44) is analogous to that of (5.1), (5.3)-(5.11) and (5.15)-(5.17).

Formulation (5.32)-(5.44) can be strengthened by adding valid inequalities known for the CVRP. In particular, we consider adding capacity inequalities, which are defined as follows.

$$\sum_{i \in S, j \notin S} x^{\omega}_{ijk} \geq b(S) \qquad \forall S \subseteq V', \forall k \in \hat{K} \tag{5.45}$$

Here $b(S)$ is the minimum number of vehicles needed to visit all customers in S. Computing $b(S)$ requires solving a bin packing problem. As is common, we replace $b(S)$ by the lower bound $\left\lceil \frac{\sum_{i \in S} q^{\omega}_i}{Q} \right\rceil$ instead.

The exact routing algorithm is a branch-and-cut algorithm using formulation (5.32)-(5.44) to solve the routing problem for every scenario $\omega \in \Omega$. Violated capacity inequalities are separated only in the rootnode using the heuristic of Lysgaard et al.(2004). We use the implementation of the separation algorithm provided in the package by Lysgaard (2003). Furthermore, we use a commercial mixed integer programming solver to solve the LP relaxation and to construct the search tree.

The heuristic routing algorithm

The heuristic routing algorithm makes use of the solution to the clustering problem as found by the clustering algorithm. In every scenario, a route is constructed for each cluster center using the customers that are not skipped. This is done by solving a traveling salesman problem, TSP. Furthermore, a CVRP is solved using the skipped customers of every cluster center. The routes obtained by solving the TSP for each cluster and solving the CVRP, together form a feasible routing schedule.

To solve the TSP, we use a branch-and-cut algorithm and a formulation containing only degree constraints. We add subtour elimination constraints that are separated by solving a max flow problem. We perform special ordered set branching and perform depth first search until an integer solution is found and switch to best node first search next. For an overview of algorithms to solve the TSP see Applegate et al.(2006).

To solve the CVRP, we use our implementation of the exact routing algorithm, where we remove the customers that are not skipped in the solution to the clustering problem, and where we set $\hat{K} = \{\hat{k}\}$. This yields a branch-and-cut algorithm for the CVRP, where violated capacity inequalities are separated at the rootnode only.

5.4 Computational results

In this section, we present results of computational experiments in which instances of the DAVRP are solved. The procedure used to generate instances is described in Section 5.4.1. In Section 5.4.2 we illustrate the computational complexity of the DAVRP empirically by solving instances using a commercial mixed integer programming solver and formulation (5.1)-(5.17). The computational results in Section 5.4.3 show how the optimality gap of the solution to the clustering problem affects the quality of the solution to the DAVRP produced by the cluster first-route second algorithm. In Section 5.4.4, the performance of the cluster first-route second heuristic using the exact routing algorithm and using the heuristic routing algorithm is compared. In particular, this gives insight in the increase in transportation costs from constructing new routes with skipped customers as is done in the heuristic routing algorithm, instead of trying to assign them to drivers that already have customers assigned to it. We illustrate the limitations of the cluster-first route-second algorithm in Section 5.4.5 by solving instances with a large number of customers and a large number of scenarios. Finally, computational results are presented in Section 5.4.6 to show the effect of the value of α on the costs of the routing schedule produced by our algorithm. The same experiments also provide a bound on the quality of the solutions produced by the cluster first-route second algorithm using the exact routing algorithm.

Preliminary experiments with the clustering algorithm indicate that choosing $\beta = 7.1$ (an upper bound on the traveling costs in our experiments), initially setting $\gamma = 2$, and setting $\gamma_i = 100$, $\gamma_f = 2$, $\gamma_t = 0.0001$ produces good results. These settings are used in all experiments of which results are presented in this chapter. Also we set a time limit of 60 seconds on the running time of the clustering algorithm. In our experiments the quality of the solution produced by the cluster first-route second algorithm never improves from maintaining a higher time limit. The time limit on the running time of the cluster first-route second algorithm is set to 1 hour.

All experiments are performed on an Intel(R) Core(TM) i5-2450M CPU 2.5 GHz processor. The algorithms were coded in C++ and the commercial mixed integer programming solver IBM ILOG Cplex optimizer, version 12.4, is used.

5.4.1 Test instances

We generate the instances[1] used in our computational experiments as follows. First, n customers are randomly generated, uniformly distributed over a square with sides 5. The depot is located in the center of the square. The travel costs are computed as the

[1]Instances are available on request.

Table 5.1: Solving the DAVRP to optimality

Inst.	$\|V'\|$	Gap	Time
1	10	0	220.08
2	10	0	577.03
3	10	0	2903.97
4	10	0	491.97
5	10	0	638.78
6	15	0.09	3600.00
7	15	0.06	3600.00
8	15	0.06	3600.00
9	15	0	3259.74
10	15	0	2991.34
11	20	-	3600.00
12	20	-	3600.00
13	20	-	3600.00
14	20	-	3600.00
15	20	-	3600.00

Euclidean distance between two locations. The vehicle capacity is set to 50 and, unless stated otherwise, we set $\alpha = 0.75$.

With the exception of the experiments presented in Section 5.4.5, 3 demand scenarios are generated, each occurring with equal probability. Let the demand scenarios be $\Omega = \{1, 2, 3\}$. Demand of customer $i \in V'$ in scenario $\omega \in \Omega$ is computed as $d_i u_i^\omega$, where d_i is generated using a normal distribution with expectation 5 and variance 1.5, and where u_v^1, u_v^2 and u_v^3 are generated using a uniform distribution on $[0.7, 0.8]$, $[0.95, 1.05]$ and $[1.2, 1.3]$, respectively. Generating demand in this fashion ensures that the scenarios resemble low, medium and high demand respectively for all customers.

5.4.2 Results on solving the DAVRP to optimality

Next, we present the results of an experiment in which the DAVRP is solved using a commercial mixed integer programming solver and formulation (5.1)-(5.17). Table 5.1 shows the results of solving instances with 10, 15 and 20 customers. The column Gap provides the optimality gap after termination, a dash indicates that no integer solution has been found within one hour. The column Time shows computation time in seconds.

Optimality is not proved for three out of the five instances with 15 customers within one hour. Furthermore, no integer solution is found for any of the instances with 20 customers within one hour. This illustrates that standard branch-and-bound procedures using formulation (5.1)-(5.17) will not be sufficient to solve the DAVRP in practice.

Table 5.2: Cluster first-route second, different optimality gaps in the clustering algorithm

	Gap 1%			Gap 5%			Gap 10%		
Inst.	Value	Time Ph.1	Time Ph.2	Value	Time Ph.1	Time Ph.2	Value	Time Ph.1	Time Ph.2
16	29.69	60.00	425.88	29.06	2.01	92.84	29.02	1.62	163.21
17	24.96	30.39	424.11	25.08	10.37	182.57	27.30	4.99	415.98
18	30.36	60.00	1477.08	30.21	10.14	488.73	31.20	4.01	1125.69
19	27.66	4.10	171.55	27.62	1.19	232.41	27.34	0.91	137.37
20	27.42	60.00	302.61	27.22	24.24	242.40	27.16	9.47	175.17
21	25.92	60.00	667.50	25.92	30.21	607.39	25.95	6.94	439.07
22	29.26	60.00	877.75	28.88	5.99	293.67	29.35	4.96	385.37
23	28.48	60.00	587.53	28.48	19.00	1517.42	28.20	6.79	164.21
24	26.18	60.00	169.56	26.33	5.41	68.02	26.36	3.00	30.94
25	27.43	4.38	29.61	27.45	1.19	18.60	28.61	0.47	382.41

5.4.3 Analysis of the optimality gap in the clustering algorithm

The optimal solution to the clustering problem is not guaranteed to provide the optimal driver assignment. Moreover, spending computation time on proving optimality might not be necessary in the first phase of the cluster first-route second algorithm. Therefore, the clustering algorithm is terminated when the optimality gap of its current solution is below some specified level. In Table 5.2 the results are shown of using the cluster first-route second algorithm, using the exact routing algorithm, to solve ten instances with 25 customers. Each instance is solved three times where the clustering algorithm is terminated when an optimality gap of respectively 1%, 5% and 10% is attained. The columns Value show the value of the solution to the DAVRP produced by the cluster first-route second algorithm. The columns Time Ph.1 present the computation time in seconds of the clustering algorithm and the columns Time Ph.2 present the computation time in seconds of the exact routing algorithm.

Requiring an optimality of 1% in the clustering algorithm allows termination within the time limit of 60 seconds in three out of ten instances. On average, the solution produced when requiring an optimality gap of 1% are 0.4% more expensive than requiring an optimality gap of 5%. Requiring an optimality gap of 10% yields solutions that are on average 1.6% more expensive than requiring an optimality gap of 5%. Closing the optimality gap in the clustering algorithm does not seem to significantly decrease the costs of the solution produced by the cluster first-route second algorithm.

In our experiments, terminating the clustering algorithm when an optimality gap of 5% is attained provides good solutions in relatively little computation time. In all the experiments presented in the remainder of this section, the clustering algorithm is terminated when an optimality gap of 5% is attained.

Table 5.3: Cluster first-route second, exact and heuristic routing

Inst.	$\|V'\|$	Time Ph.1	Time Ph.2 Exact	Time Ph.2 Heuristic	Diff.
26	20	4.38	56.41	3.65	20.97
27	20	6.30	30.20	6.41	19.13
28	20	60.00	132.88	60.14	15.47
29	20	4.21	28.91	4.37	11.97
30	20	60.00	121.96	60.13	23.72
31	20	0.61	49.94	0.69	14.77
32	20	60.00	161.62	60.29	9.60
33	20	0.61	29.33	0.72	16.13
34	20	60.00	77.98	60.14	24.06
35	20	60.00	88.11	60.12	12.06
36	30	21.47	1078.58	22.01	9.24
37	30	13.31	350.60	13.53	10.97
38	30	0.05	34.09	0.28	15.92
39	30	60.00	1879.49	60.47	17.64
40	30	23.82	482.76	24.11	9.50
41	30	5.01	1039.21	5.24	11.87
42	30	4.43	664.74	4.76	15.19
43	30	11.97	322.36	12.39	18.03
44	30	5.79	814.52	6.26	15.25
45	30	1.98	561.73	2.39	17.94
46	40	48.19	3540.00	48.61	-
47	40	20.69	3540.00	21.53	-
48	40	17.38	3540.00	17.82	-
49	40	52.96	3540.00	54.20	-
50	40	18.72	2851.80	19.36	12.44
51	40	33.70	3540.00	34.87	-
52	40	20.00	3540.00	22.12	-
53	40	60.00	3540.00	61.03	-
54	40	1.84	3540.00	3.51	-
55	40	45.88	3540.00	43.82	-

5.4.4 Comparison of the exact and heuristic routing algorithm

Table 5.3 shows the results of solving ten instances with 20, 30 and 40 customers. Each instance is solved twice using the cluster first-route second algorithm with the exact routing algorithm and with the heuristic routing algorithm. The column Time Ph.1 shows the computation time in seconds of the clustering algorithm. The columns Time Ph.2 Exact and Time Ph.2 Heuristic show the computation time in seconds of the exact and heuristic routing algorithm respectively. Finally, the column Diff. shows the percentage difference between the solutions obtained by using the exact and heuristic routing algorithm.

The cluster first-route second algorithm does not find a feasible integer solution within the time limit for nine out of ten instances with 40 customers. The computation time of the exact routing algorithm is much larger than that of the heuristic routing algorithm. Furthermore, for the instances where a solution is found by both algorithms, the solutions produced by the heuristic routing algorithm are on average 15.3% more expensive. This shows that transportation costs increase significantly by constructing new routes for customers that are skipped by their assigned driver. Transportation costs can be much

Table 5.4: Cluster first-route second, large instances

	10 Scenarios		50 Scenarios		100 Scenarios	
$\|V'\|$	Time	St.dev.	Time	St.dev.	Time	St.dev.
30	88.67	40.92	119.36	19.01	135.14	4.12
50	126.42	0.67	179.84	8.87	247.49	18.45
70	141.80	2.92	350.39	38.92	571.57	97.27
100	201.61	10.90	-	-	-	-

lower when these skipped customers are assigned to a different driver that already has customers assigned to it.

5.4.5 Results for large instances

The cluster first-route second heuristic using the exact routing algorithm is not able to solve all instances with 40 customers and 3 scenarios within one hour of computation time. Next, we illustrate the limitations of the cluster first-route second algorithm when using the heuristic routing algorithm.

The scenarios of the instances in this experiment are generated in the following way. Demand of customer $i \in V'$ in scenario $\omega \in \Omega$ is generated using a normal distribution with expectation d_v and variance 1.5, rounding to the nearest integer, rounding up to 1 if demand is below that and rounding down to Q if demand is above that. We generate d_v using a normal distribution with demand 5 and variance 1.5.

In Table 5.4, results are presented for instances with 30, 50, 70 and 100 customers. For each of these numbers of customers, ten instances are generated. For every instance three variants are constructed, with 10, 50 and 100 scenarios. For every ten instances with n customers and $|\Omega|$ scenarios, the columns Time and St.dev. provide the average computation time in seconds and its standard deviation of the cluster first-route second algorithm using the heuristic routing algorithm.

No instances with 100 customers and 50 and 100 scenarios are solved. Also no instances with 200 scenarios are solved. This is due to memory requirements of our implementation of the cluster first-route second algorithm. The computation time seems to increase less than linearly with the number of scenarios and roughly linearly with the number of customers.

Table 5.5: Cluster first-route second, different values of α

Inst.	$\alpha = 0$	$\alpha = 0.75$	$\alpha = 1$
56	27.40	28.45	32.20
57	27.15	27.51	29.79
58	26.72	27.29	28.34
59	27.09	27.65	30.24
60	25.89	27.02	31.32
61	28.68	30.18	32.16
62	27.78	28.39	30.53
63	25.18	25.79	28.00
64	25.77	26.09	29.93
65	26.09	26.60	31.54
66	27.47	28.42	29.36
67	26.51	27.01	29.34
68	25.19	25.47	29.26
69	29.36	30.04	35.02
70	25.20	25.84	27.54
71	28.62	29.36	32.11
72	29.71	30.35	34.63
73	27.38	28.58	29.23
74	25.08	25.84	27.24
75	25.85	27.86	28.57

5.4.6 The effect of α

Adhering to the driver assignments may be beneficial for business, but it decreases flexibility in transportation. Hence, the expected transportation costs increase. Next, we investigate the increase of expected transportation costs.

In Table 5.5 the results are presented of an experiment in which twenty instances with 25 customers are solved. For each instance three variants are considered in which we set $\alpha = 0$, $\alpha = 0.75$ and $\alpha = 1$. For $\alpha = 0$, the instances are solved by solving a CVRP to optimality for every scenario to construct a routing schedule. It is equivalent to not imposing any driver assignment constraints. Hence, the lowest possible expected transportation costs are obtained for $\alpha = 0$. For $\alpha = 0.75$ and $\alpha = 1$ the instances are solved using the cluster first-route second heuristic using the exact routing algorithm. Table 5.5 shows the solution values of the obtained solutions.

Next, we report the average percentage difference of the solution values obtained for the instances with different values of α. For these instances, adhering to the driver assignments with $\alpha = 1$ increases the expected transportation costs with 12.7%. The highest increase of 21.0% is obtained for instance 60. Adhering to the driver assignments with $\alpha = 0.75$ increases the expected transportation costs with 2.9%. This can be considered a moderate increase. Having driver assignment requirements but allowing a little flexibility, i.e. using $\alpha = 0.75$ instead of $\alpha = 1$, decreases the expected transportation costs significantly.

Finally, note that the value of the optimal solution to the DAVRP for instances with a specific value of α, is in between the solution value obtained for instances with $\alpha = 0$ and those obtained using the cluster first-route second algorithm. This allows us to deduce that for instances with $\alpha = 1$ the cluster first-route second algorithm produces solutions that are on average at most 12% more expensive than the optimal solutions. Moreover, this shows that for the instances with $\alpha = 0.75$ the cluster first-route second heuristic produces solutions that are on average at most 2.9% more expensive than the optimal solution.

5.5 Conclusion

In this chapter the DAVRP is introduced. We developed a cluster first-route second heuristic to find solutions to the DAVRP. In the first phase, a solution to the clustering problem is found using a branch-and-bound algorithm based on a Lagrangian relaxation. In the second phase, a solution to the DAVRP is constructed based on the clusters constructed in the first phase, which are used as a driver assignment. For the routing problem that is solved in this phase we designed an exact and a heuristic algorithm. In the latter, customers that are not visited by their assigned driver are used to construct new routes instead of trying to add them to drivers that already have customers assigned to them.

Using our formulation of the DAVRP and a commercial mixed integer programming solver, we are able to solve instances with 10 customers and 3 scenarios to optimality within one hour of computation time. Computational experiments show that the cluster first-route second heuristic is able to solve instances with up to three times more customers when the exact routing algorithm is used. Furthermore, when using the heuristic routing algorithm, instances with up to 100 customers and up to 100 scenarios can be solved well within one hour.

We present an experiment in which the cluster first-route second algorithm produces on average 15.3% more expensive solutions when the heuristic routing algorithm is used instead of the exact routing algorithm. This quantifies the increase in transportation costs from constructing new routes with skipped customers instead of assigning them to drivers that already have customers assigned to them.

From experiments where we solve instances of the DAVRP with the cluster first-route second algorithm using the exact routing algorithm, we conclude that adhering to the driver assignments can lead to an increase in expected transportation costs of up to 21.0%. When setting $\alpha = 1$ the increase is on average 12.7%. However, when adhering to the driver constraints but allowing a little flexibility by using $\alpha = 0.75$, the increase in expected transportation costs is on average only 2.9%. These experiments also allow us to conclude that, even though we do not solve these instances to optimality, the solution value of the solutions produced by the cluster first-route second algorithm for instances with $\alpha = 0.75$ are on average at most 2.9% more expensive than the optimal solutions.

Chapter 6

Summary and conclusion

In this thesis vehicle routing with uncertain demand is considered. Several different views on a fixed schedule are explored and sophisticated rescheduling procedures are used. It accommodates the growing need to achieve efficiency in transport, and the need to incorporate the uncertainty experienced in reality into transportation models. Next, each chapter is summarized and findings are presented.

In Chapter 2 the vehicle rescheduling problem, VRSP, is introduced. It is the problem of finding a new schedule that not only minimizes the total traveling costs but also minimizes the incurred penalty costs for deviating from a given fixed schedule. A branch-and-cut algorithm is used to solve the VRSP to optimality. Moreover, a fast two-phase heuristic is presented. Sufficiency conditions are provided which state that when the penalty costs for deviating are high enough with respect to the traveling costs, the heuristic produces an optimal solution. These sufficiency conditions are unlikely to be met in most real world problem instances. However, computational experiments show that the heuristic often produces an optimal solution even when the penalty costs for deviating are close to the average traveling costs between two locations in the delivery schedule.

Furthermore, in Chapter 3 the time window assignment vehicle routing problem, TWAVRP, is introduced. In this problem time windows have to be assigned before demand is known. Next, a realization of demand is revealed and a vehicle routing schedule is made that satisfies the assigned time windows. The objective is to minimize the expected traveling costs. In the TWAVRP, time windows of fixed width are chosen for each customer from an exogenous time window. A branch-price-and-cut algorithm is presented to solve the TWAVRP to optimality. This algorithm is used to solve instances with up to 25 customers and 3 scenarios to optimality within one hour of computation time. Finally, the value of an exact approach for the TWAVRP is investigated by comparing the optimal solution to the solution found by assigning time windows based on solving a VRPTW with average demand, as is commonly done in practice. In the presented experiments, the solutions obtained with the exact algorithm yield a decrease in costs of up to 5.42% with respect to current practice, and an average decrease of 1.85%.

Introduced in Chapter 4 is the discrete time window assignment vehicle routing problem, DTWAVRP. This problem is similar to the TWAVRP. The main difference is that in the DTWAVRP a time window is not chosen from an exogenous time window, but selected from a discrete set of candidate time windows. Selecting time windows from a set of candidates does not only make sense from a practical point of view. It also allows more sophisticated techniques, that are successfully used to solve classical vehicle routing problems, to be incorporated into a branch-price-and-cut algorithm to solve the DTWAVRP to optimality. The branch-price-and-cut algorithm is used to solve instances with up to 25 customers and 3 scenarios to optimality within one hour of computation time. Moreover, computational experiments show that using the exact algorithm for the instances presented in this chapter, provides a decrease of up to 7.01% in expected costs with respect to current practice, and an average decrease of 3.32%.

When comparing the TWAVRP with the DTWAVRP, observe that modeling a real world problem as a TWAVRP allows more flexibility in selecting time windows than modeling the same problem as a DTWAVRP. As such, implementing TWAVRP solutions yield lower costs. However, the freedom to select time windows as done in the TWAVRP might not always be desirable or possible. Finally, note that experiments show that selecting time windows as is done in current practice produces solutions that are further away from the optimum in the DTWAVRP case than in the TWAVRP case.

Finally, the driver assignment vehicle routing problem, DAVRP, is introduced in Chapter 5. In this problem customers have to be assigned to drivers before demand is known, and after demand is known a routing schedule has to be made such that every driver visits at least a fraction α of its assigned customers. A cluster first-route second heuristic is designed to find good solutions to this problem. Computational experiments show that adhering to driver assignments can lead to an increase of the expected transportation costs of up to 21.0%, and on average 12.7%. Allowing a little flexibility, by choosing $\alpha = 0.75$, leads to an average increase in transportation costs of only 2.9% with respect to not adhering to the driver assignments. Finally, for instances with $\alpha = 0.75$ we compare the expected transportation costs from constructing new routes with customers that can not be visited by their assigned driver, to the costs from trying to assign them to drivers that already have customers assigned to it. The former leads to an average increase in expected transportation costs of 15.3%.

This thesis shows that significant decreases in transportation cost can be attained with respect to current practice by taking into account uncertain demand. Many organizations operate with a fixed schedule that is often designed before demand of customers is known. A fixed delivery schedule can almost never be maintained and rescheduling will always be necessary or desired due to demand uncertainty and variations in demand. Different types of fixed schedules, and accompanying rescheduling procedures, call for different methods of designing a fixed schedule.

Furthermore, not only the direct transportation costs should be taken into account when constructing a schedule. Also indirect effects that transportation has on a network as a whole, such as costs for the customer when deliveries arrive late, should be integrated into a procedure to construct delivery schedules. This holds in particular for the many industries in which the supplier and customer are part of a single organization.

The methods presented in this thesis can be used to construct fixed schedules and to do rescheduling in different settings. They decrease total costs in distribution networks by primarily taking into account uncertain demand, and also by taking into account the indirect effects of transportation on the customers and supplier.

Nederlandse Samenvatting (Summary in Dutch)

Dit proefschrift gaat over het transport van goederen vanuit een distributiecentrum naar klanten in een distributienetwerk door middel van voertuigen met beperkte capaciteit. Het voertuig routering probleem, VRP, is het probleem om een rittenplanning te maken voor de voertuigen om goederen af te leveren bij klanten, zodanig dat de transportkosten minimaal zijn. Dit is een klassiek probleem binnen de combinatorische optimalisering en wordt al bestudeerd sinds het geïntroduceerd werd in de wetenschappelijke literatuur door Dantzig en Ramer in 1959. In dit probleem wordt verondersteld dat de hoeveelheid goederen die aan iedere klant geleverd moet worden bekend is.

Echter, in de praktijk wordt een planning vaak gemaakt voordat de vraag van klanten bekend is. Voor ketens van detailhandelaren is het meer regel dan uitzondering dat een rittenplanning gemaakt wordt die gebruikt wordt voor wekelijks of zelfs dagelijks transport voor een periode van een jaar. Op het moment dat een dergelijke langetermijnplanning gemaakt wordt is normaal gesproken de vraag van de klanten gedurende deze planningsperiode nog niet bekend. Een langetermijnplanning wordt dan ook vaak gebaseerd op gemiddelde historische vraag.

In het geval dat een rittenplanning voor langetermijn is vastgesteld, wordt deze rittenplanning aangepast zodra de hoeveelheid goederen die klanten vragen bekend is. Immers, als de vraag van klanten, die volgens de planning in dezelfde rit bezocht moeten worden door één voertuig, uitzonderlijk hoog is, dan past het niet meer in het voertuig. Ook als de vraag van klanten laag is kan herplannen gewenst zijn. Het combineren van de vracht van verschillende voertuigen kan tot kostenbesparing leiden doordat minder voertuigen gebruikt hoeven te worden of minder afstand afgelegd hoeft te worden. Als in de oorspronkelijke planning deze onzekerheid van de vraag niet goed is meegenomen, kan dit leiden tot hoge transportkosten na het herplannen.

Het herplannen heeft verder ook indirecte gevolgen voor zowel de leverancier als de klanten binnen een distributienetwerk. Onderzoek onder Nederlandse detailhandelaren door Drop (2011) wees uit dat de kosten die een winkelier maakt als gevolg van een te late levering, dusdanig hoog zijn dat het loont om hiermee rekening te houden bij het herplannen. Een vrachtwagen om laten rijden kan goedkoper zijn dan te laat leveren bijvoorbeeld. In dit proefschrift wordt onder andere een nieuwe procedure voor herplannen gepresenteerd waarmee een rittenplanning gemaakt wordt die niet alleen de kosten van transport maar ook de kosten voor het afwijken van de oorspronkelijke rittenplanning minimaliseert.

Er zijn verschillende typen langetermijnplanningen mogelijk die helpen de directe en indirecte kosten van transport laag te houden. In plaats van het maken van een rittenplanning is het niet ongebruikelijk dat de leverancier met zijn klant een tijdsvenster afspreekt waarbinnen de klant bezocht moet worden. In veel ketens van detailhandelaren, waar winkeliers bijvoorbeeld wekelijks een levering ontvangen, wordt een dergelijke toewijzing van tijdsvensters vastgezet voor periodes van vaak meer van een jaar. Tijdens het toewijzen van de tijdsvensters weet men vaak niet wat de vraaghoeveelheid is van elke klant gedurende de planningsperiode. Daarnaast zal deze ook variëren voor de afzonderlijke leveringen. In de praktijk worden deze tijdsvensters meestal gekozen rondom de aankomsttijd bij een klant in een standaard rittenplanning gebaseerd op gemiddelde historische vraag. Zodra de vraag van de klant bekend wordt, wordt er een nieuwe rittenplanning gemaakt waarin deze tijdsvensters niet geschonden worden.

In dit proefschrift introduceren we twee nieuwe modellen voor het toewijzen van tijdsvensters voordat de vraag bekend is. In het ene model worden tijdsvensters aan klanten toegewezen binnen een exogeen tijdsvenster, bijvoorbeeld een tijdsvenster van twee uur gedurende de openingstijden van de winkel. In het andere model moet voor iedere klant een tijdvenster gekozen worden uit een beperkt aantal kandidaten. In dit proefschrift wordt voor beide modellen een algoritme gepresenteerd dat tijdsvensters toewijst aan klanten zodanig dat de verwachte transportkosten wordt geminimaliseerd. In numerieke experimenten leveren de tijdvensters die gevonden worden met deze algoritmes kostenbesparingen op tot 7.01% ten opzichte van de tijdsvensters zoals deze in de praktijk toegewezen worden.

Een ander type langetermijnplanning is het volgende. Een klant kan voor lange tijd toegewezen worden aan een chauffeur. Deze chauffeur zal alle leveringen doen aan de klant gedurende de planningsperiode. Dit is bijvoorbeeld gewenst als de chauffeur een sleutel of wachtwoord van een opslagruimte nodig heeft om een levering daarin te plaatsen. Dit is onder andere het geval bij de dienst aangeboden door TNT Innight. In een artikel geschreven door Groër et al. (2009) wordt bovendien gesteld dat doordat de chauffeurs van UPS een persoonlijke band met hun klanten vormen er jaarlijks 60 miljoen extra pakketten verzonden worden via UPS.

Voor het probleem van het toewijzen van klanten aan chauffeurs wordt in dit proefschrift een nieuw model geïntroduceerd. Hierin worden klanten toegewezen aan chauffeurs nog voor de vraag van de klanten bekend is. Zodra de vraag van de klant bekend is wordt er een rittenplanning gemaakt waarbij klanten bezocht moeten worden door de chauffeurs aan wie ze toegewezen zijn, zodanig dat de transportkosten geminimaliseerd worden. Voor dit probleem wordt een heuristiek gepresenteerd die snel goede toewijzingen van klanten aan chauffeurs construeert. Numerieke experimenten geven inzicht in de additionele kosten die gemaakt worden voor het vasthouden aan de toewijzingen. Zo kan een goede afweging gemaakt worden tussen de baten van het vasthouden aan de toewijzing van klanten aan chauffeurs, en de extra transportkosten die dit met zich meebrengt.

In dit proefschrift wordt laten zien dat aanzienlijke kostenbesparingen gemaakt kunnen worden door de onzekerheid van de vraaghoeveelheid van klanten in ogenschouw te nemen tijdens het maken van een langetermijnplanning. Verschillende types van langetermijnplanningen worden onderzocht, te weten een vaste rittenplanning, vaste tijdsvensters, of vaste chauffeurs voor klanten. Voor elk van dit type langetermijnplanning is een andere procedure voor herplannen nodig en een andere methode om de langetermijnplanning te genereren.

Verder moeten de secundaire kosten van transport niet vergeten worden bij het maken van een planning, zoals kosten van de klant als een levering te laat is. Deze secundaire kosten dienen geïntegreerd te worden in een methode om langetermijnplanningen en rittenplanningen te construeren. Dit geldt in het bijzonder voor de vele distributienetwerken waarin de leverancier en klant onderdeel zijn van hetzelfde bedrijf.

De methodes die gepresenteerd worden in dit proefschrift kunnen gebruikt worden om vaste planningen te maken en om te herplannen. Zij zorgen voor verlaagde kosten van transport in distributienetwerken door de onzekerheid van de vraag in te calculeren, en ook door de indirecte effecten van transport op de leverancier en klanten mee te nemen.

References

Agatz, N., Campbell, A., Fleischmann, M. and Savelsbergh M. 2011, 'Time Slot Management in Attended Home Delivery', *Tranportation Science*, Vol. 45, No. 3, pp. 435-449.

Applegate, D.L., Bixby, R.E., Chvátal, V. and Cook, W.J. 2006, 'The Traveling Salesman problem', Princeton University Press, Princeton New Jersey.

Baldacci, R., Hadjiconstantinou, E. and Mingozzi, A. 2004, 'An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation', *Operations Research*, Vol. 52, No. 5, pp. 723-738.

Baldacci, R., Mingozzi, A. and Roberti, R. 2011, 'New route relaxation and pricing strategies for the vehicle routing problem', *Operations Research*, Vol. 59, No. 5, pp. 1269-1283.

Baldacci, R., Mingozzi, A. and Roberti, R. 2012, 'Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints', *European Journal of Operational Research*, Vol. 218, No. 1, pp. 1-6.

Bertsimas, D.J. 1992, 'A vehicle routing problem with stochastic demand', *Operations Research*, Vol. 40, No. 3, pp. 574-585.

Bertsimas, D.J. and Levi, D.S. 1996, 'A New Generation of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty', *Operations Research*, Vol. 44, No. 2, pp. 286-304.

Bramel, J. and Simchi-Levi, D. 1995, 'A Location Based Heuristic for General Routing Problems', *Operations Research*, Vol. 43, No. 4, pp. 649-660.

Bräysy, O. and Gendreau, M. 2005a, 'Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms', *Transportation Science*, Vol. 39, No. 1, pp. 104-118.

Bräysy, O. and Gendreau, M. 2005b, 'Vehicle Routing Problem with Time Windows, Part II: Metaheuristics', *Transportation Science*, Vol. 39, No. 1, pp. 119-139.

Chen, S., Golden, B., Whong, R. and Zhong, H. 2009, 'Arc-Routing Models for Small-Package Local Routing', *Transportation Science*, Vol. 43, No. 1, pp. 43-55.

Clarke, G. and Wright, J.W. 1964, 'Scheduling of Vehicles from a Central Depot to a Number of Delivery Points', *Operations Research*, Vol. 12, No. 4, pp. 568-581.

Dantzig, G.B. and Ramzer, J.H. 1959, 'The Truck Dispatching Problem', *Management Science*, Vol. 6, No. 1, pp. 80-91.

Dantzig, G.B. and Wolfe, P. 1960, 'Decomposition principle for linear programs', *Operations Research*, Vol. 8, No. 1, pp. 101-11.

Desaulniers, G., Lessard, F. and Hadjar, A. 2008, 'Tabu Search, Partial Elementarity, and Generalized k-Path Inequalities for the Vehicle Routing Problem with Time Windows', *Transportation Science*, Vol. 42, No. 3, pp. 387-404.

Desrochers, M., Desrosiers, J. and Solomon, M. 1992, 'A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows', *Operations Research*, Vol. 40, No. 2, pp. 342-354.

Drop, G. 2011, 'Empirical Research on Additional Costs Generated at Store Level as a Result of Deviations in the Delivery Schedule', *unpublished master's thesis*, Rotterdam School of Management, Erasmus University Rotterdam.

Dror, M., Laporte, G. and Trudeau, P. 1989, 'Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks', *Transportation Science*, Vol. 23., No. 3, pp. 166-176.

Erlenkotter, D. 1978, 'A Dual-Based Procedure for Uncapacitated Facility Location', *Operations Research*, Vol. 26, No. 6, pp. 992-1009.

Feillet, D., Dejax, P., Gendreau, M. and Gueguen, C. 2004, 'An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to some Vehicle Routing Problems', *Networks*, Vol. 44, No. 3, pp. 216-229.

Fisher, M.L. and Jaikumar, R. 1981, 'A generalized assignment heuristic for vehicle routing', *Networks*, Vol. 11, No. 2, pp. 109-124.

Fisher, M. 1995, 'Vehicle Routing', *the handbooks in operations research and management science*, ed. Ball, M.O., Magnanti, T.L., Monma, C.L. and Nemhauser, G.L., Elsevier Science B.V., Vol. 8.

Gendreau, M., Laporte, G. and Séguin, R. 1996, 'A tabu search heuristic for the vehicle routing problem with stochastic demands and customers', *Operations Research*, Vol. 44, No. 3, pp. 469-477.

Groër, C., Golden, B. and Wasil, E. 2009, 'The Consistent Vehicle Routing Problem', *Manufacturing & Service Operations Management*, Vol. 11, No. 4, pp. 630-643.

Groër, C., Golden, B. and Wasil, E. 2010, 'A library of local search heuristics for the vehicle routing problem', *Mathematical Programming Computation*, Vol. 2, No. 2, pp. 19-101.

Ioachim, I., Gélinas, S., Soumis, F. and Desrosiers, J. 1998, 'A Dynamic Programming Algorithm for the Shortest Path Problem with Time Windows and Linear Node Costs', *Networks*, Vol. 31, No. 3, pp. 193-204.

Irnich, S. and Villeneuve, D. 2006,'The Shortest Path Problem with Resource Constraints and k-Cycle Elimination for $k \geq 3$', *INFORMS Journal on Computing*, Vol. 18, No. 3, pp. 391-406.

Jabali, O., Leus, R., van Woensel, T. and de Kok, A.G. 2010, 'Self-imposed time windows in vehicle routing', *Working Paper* 1028, Katholieke Universiteit Leuven, Leuven, Belgium.

Jepsen, M., Petersen, B., Spoorendonk, S. and Pisinger, D. 2008, 'Subset-Row Inequalities Applied to the Vehicle-Routing Problem with Time Windows', *Operations Research*, Vol. 56, No. 2, pp. 497-511.

Kallehauge, B., Larsen, J., Madsen, O.B.G., Solomon, M.M. 2005, 'Vehicle routing problem with time windows', Desaulniers, G., Desrosiers, J., Solomon, M. M., eds. Column Generation. Springer, New York, pp. 67-98.

Kellerer, H., Pferschy, U., Pisinger, D. 2004, 'Knapsack Problems', Springer, Berlin.

Kohl, N., Desrosiers, J., Madsen, O.B.G., Solomon, M.M. and Soumis, F. 1999, '2-Path Cuts for the Vehicle Routing Problem with Time Windows', *Transportation Science*, Vol. 33, No. 1, pp. 101-116.

Körkel, M. 1989, 'On the exact solution of large-scale simple plant location problems', *European Journal of Operational Research*, Vol. 39, No. 2, pp. 157-173.

Laporte, G. 1992, 'The Vehicle Routing Problem: An overview of exact and approximate algorithms', *European Journal of Operational Research*, Vol. 59, No. 3, pp. 345-358.

Laporte, G., Gendreau, M., Potvin, J. and Semet, F. 2000, 'Classical and modern heuristics for the vehicle routing problem', *International Transactions in Operational Research*, Vol. 7, No. 4-5, pp. 285-300.

Laporte, G., Louveaux, F.V. and Hamme, L. van 2002, 'An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands', *Operations Research*, Vol. 50, No. 3, pp. 415-423.

Laporte, G. 2007, 'What you should know about the vehicle routing problem', *Naval research Logistics*, Vol. 54, No. 8, pp. 811-819.

Laporte, G. 2009, 'Fifty Years of Vehicle Routing', *Transportation Science*, Vol. 43, No. 4, pp. 408-416.

Li, J., Mirchandani, P.B. and Borenstein, D. 2007, 'The Vehicle Rescheduling Problem: Model and Algorithms', *Networks*, Vol. 50, No. 3, pp. 211-229.

Li, J., Mirchandani, P.B. and Borenstein, D. 2009, 'A Lagrangian heuristic for the real-time vehicle rescheduling problem', *Transportation Research Part E*, Vol. 45, No. 3, pp. 419-433.

Lysgaard, J. 2003. 'CVRPSEP: A package of separation routines for the capacitated vehicle routing problem.', *Working Paper 03-04*, Department of Management Science and Logistics, Aarhus School of Business, Aarhus, Denmark.

Lysgaard, J., Letchford, A.N. and Eglese, R.W. 2004, 'A new branch-and-cut algorithm for the capacitated vehicle routing problem' *Mathematical Programming*, Vol. 100, No. 2, pp. 423-445.

Novoa, C. and Storer, R. 2009, 'An approximate dynamic programming approach for the vehicle routing problem with stochastic demands', *European Journal of Operational research*, Vol. 196, No. 2, pp 509-515.

Ribeiro, G.M., Desaulniers, G. and Desrosiers, J. 2012, 'A branch-price-and-cut algorithm for the workover rig routing problem', *Computers & Operations Research*, Vol. 39, No. 12, pp. 3305-3315.

Secomandi, N. 2001, 'A Rollout Policy for the Vehicle Routing Problem with Stochastic Demands', *Operations Research*, Vol. 49, No. 5, pp. 796-802.

Simchi-Levi, D., Chen, X., Bramel, J. 2005, *The Logic of Logistics. Theory, Algorithms, and Applications for Logistics and Supply Chain Management*, Springer Series in Operations Research, second edition, Springer Science+Business Media Inc., New York.

Tan, K.C., Cheong, C.Y. and Goh, C.K. 2007, 'Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation', *European Journal of Operational Research*, Vol. 177, No. 2, pp. 813-839.

Toth, P. and Vigo, D. 2002, 'Models, relaxations and exact approaches for the capacitated vehicle routing problem', *Discrete Applied Mathematics*, Vol. 123, No 1-3, pp. 487-512.

Yaman, H. 2006, 'Formulations and Valid Inequalities for the Heterogenous Vehicle Routing Problem', *Mathematical Programming*, Vol. 106, No. 2, pp. 365-390.

About the author

Remy Spliet was born on April 4, 1986, in Nieuwegein, the Netherlands. He received his bachelor's degree in Econometrie & Besliskunde (Econometrics and Management Science) in 2007 at the Erasmus University in Rotterdam. In 2008, he received his master's degree in Econometrics & Management Science, the Operations Research and Quantitative Logistics programme, at the Erasmus University in Rotterdam with distinction. Following this, he started his PhD research at the Econometric Institute at the Rotterdam University in Rotterdam.

As a student, Remy worked as a non-traveling salesman at a local liquor store in his hometown IJsselstein. There, he underwent the hardships caused by deliveries arriving too late or too early. Quite often, he had to wait for a truck that arrived late, followed by working overtime to handle the late delivery. Also customers had to be disappointed when luxurious bottles of wine or whiskey were not delivered in time. Missed sales frustrated Remy, but not as much as being called in the morning, while still sleeping, to come to work because the truck had arrived early. He took it upon himself to do something about it. He devoted his studies for both his bachelor's as well as his master's thesis to planning methods to make trucks be on time. Eventually, he became a PhD student to take the design of delivery schedules to a higher level.

As a PhD student, Remy became passionate about combinatorial optimization, and in particular mathematical programming. His interests are in analyzing different formulations, finding valid inequalities and accompanying separation algorithms, experimenting with different types of relaxations and designing algorithms to solve these, and of course implementing branch-and-bound procedures.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

DISSERTATIONS LAST FIVE YEARS

Acciaro, M., *Bundling Strategies in Global Supply Chains*. Promoter(s): Prof.dr. H.E. Haralambides, EPS-2010-197-LIS, http://hdl.handle.net/1765/19742

Agatz, N.A.H., *Demand Management in E-Fulfillment*. Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, http://hdl.handle.net/1765/15425

Alexiev, A., *Exploratory Innovation: The Role of Organizational and Top Management Team Social Capital*. Promoter(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-208-STR, http://hdl.handle.net/1765/20632

Asperen, E. van, *Essays on Port, Container, and Bulk Chemical Logistics Optimization*. Promoter(s): Prof.dr.ir. R. Dekker, EPS-2009-181-LIS, http://hdl.handle.net/1765/17626

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-Frequency Data*, Promoter(s): Prof.dr.D.J.C. van Dijk, EPS-2013-273-F&A, http://hdl.handle.net/1765/38240

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2011-241-MKT, http://hdl.handle.net/1765/23670

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2013-278-S&E, http://hdl.handle.net/1765/ 39128

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promoter(s): Prof.dr. B. Krug, EPS-2012-262-ORG, http://hdl.handle.net/1765/32345

Bezemer, P.J., *Diffusion of Corporate Governance Beliefs: Board Independence and the Emergence of a Shareholder Value Orientation in the Netherlands*. Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-192-STR, http://hdl.handle.net/1765/18458

Binken, J.L.G., *System Markets: Indirect Network Effects in Action, or Inaction,* Promoter(s): Prof.dr. S. Stremersch, EPS-2010-213-MKT, http://hdl.handle.net/1765/21186

Blitz, D.C., *Benchmarking Benchmarks,* Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, http://hdl.handle.net/1765/22624

Borst, W.A.M., *Understanding Crowdsourcing: Effects of Motivation and Rewards on Participation and Performance in Voluntary Online Activities,* Promoter(s): Prof.dr.ir. J.C.M. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2010-221-LIS, http://hdl.handle.net/1765/ 21914

Budiono, D.P., *The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds,* Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2010-185-F&A, http://hdl.handle.net/1765/18126

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promoter(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.R. Commandeur, EPS-2011-243-ORG, http://hdl.handle.net/1765/26178

Camacho, N.M., *Health and Marketing; Essays on Physician and Patient Decision-making*, Promoter(s): Prof.dr. S. Stremersch, EPS-2011-237-MKT, http://hdl.handle.net/1765/23604

Carvalho, L., *Knowledge Locations in Cities; Emergence and Development Dynamics*, Promoter(s): Prof.dr. L. van den Berg, EPS-2013-274-S&E, http://hdl.handle.net/1765/ 38449

Carvalho de Mesquita Ferreira, L., *Attention Mosaics: Studies of Organizational Attention*, Promoter(s): Prof.dr. P.M.A.R. Heugens & Prof.dr. J. van Oosterhout, EPS-2010-205-ORG, http://hdl.handle.net/1765/19882

Chen, C.M., *Evaluation and Design of Supply Chain Operations Using DEA*, Promoter(s): Prof.dr. J.A.E.E. van Nunen, EPS-2009-172-LIS, http://hdl.handle.net/1765/16181

Cox, R.H.G.M., *To Own, To Finance, and to Insure; Residential Real Estate Reealed*, Promoter(s): Prof.dr. D. Brounen, EPS-2013-290-F&A, http://hdl.handle.net/1765/1

Defilippi Angeldonis, E.F., *Access Regulation for Naturally Monopolistic Port Terminals: Lessons from Regulated Network Industries,* Promoter(s): Prof.dr. H.E. Haralambides, EPS-2010-204-LIS, http://hdl.handle.net/1765/19881

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2012-255-ORG, http://hdl.handle.net/1765/ 31174

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promoter(s): Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2011-232-ORG, http://hdl.handle.net/1765/23268

Diepen, M. van, *Dynamics and Competition in Charitable Giving*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, http://hdl.handle.net/1765/14526

Dietvorst, R.C., *Neural Mechanisms Underlying Social Intelligence and Their Relationship with the Performance of Sales Managers,* Promoter(s): Prof.dr. W.J.M.I. Verbeke, EPS-2010-215-MKT, http://hdl.handle.net/1765/21188

Dietz, H.M.S., *Managing (Sales)People towards Performance: HR Strategy, Leadership & Teamwork*, Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2009-168-ORG, http://hdl.handle.net/1765/16081

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2013-272-LIS, http://hdl.handle.net/1765/38241

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promoter(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-258-STR, http://hdl.handle.net/1765/32166

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory*, Promoter(s): Prof.dr. W.F.C. Verschoor, EPS-2012-257-F&A, http://hdl.handle.net/1765/31914

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promoter(s): Prof.dr. A. de Jong, EPS-2011-240-F&A, http://hdl.handle.net/1765/26041

Duursema, H., *Strategic Leadership; Moving Beyond the Leader-follower Dyad*, Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2013-279-ORG, http://hdl.handle.net/1765/ 39129

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2011-247-LIS, http://hdl.handle.net/1765/26509

Eijk, A.R. van der, *Behind Networks: Knowledge Transfer, Favor Exchange and Performance*, Promoter(s): Prof.dr. S.L. van de Velde & Prof.dr.drs. W.A. Dolfsma, EPS-2009-161-LIS, http://hdl.handle.net/1765/14613

Essen, M. van, *An Institution-Based View of Ownership*, Promoter(s): Prof.dr. J. van Oosterhout & Prof.dr. G.M.H. Mertens, EPS-2011-226-ORG, http://hdl.handle.net/1765/22643

Feng, L., *Motivation, Coordination and Cognition in Cooperatives,* Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2010-220-ORG, http://hdl.handle.net/1765/21680

Gertsen, H.F.M., *Riding a Tiger without Being Eaten: How Companies and Analysts Tame Financial Restatements and Influence Corporate Reputation*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2009-171-ORG, http://hdl.handle.net/1765/16098

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promoter(s): Prof.dr.ir. M.B.M. de Koster, EPS-2012-269-LIS, http://hdl.handle.net/1765/ 37779

Gijsbers, G.W., *Agricultural Innovation in Asia: Drivers, Paradigms and Performance,* Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2009-156-ORG, http://hdl.handle.net/1765/14524

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2012-270-ORG, http://hdl.handle.net/1765/ 38028

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-256-MKT, http://hdl.handle.net/1765/31913

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promoter(s): Prof.dr. G.M.H. Mertens & Prof.dr. E. Peek, EPS-2012-264-F&A, http://hdl.handle.net/1765/37170

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, Promoter(s): Prof.dr. M.B.M. de Koster & Prof.dr. S.L. van de Velde, EPS-2009-180-LIS, http://hdl.handle.net/1765/16724

Greeven, M.J., *Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China*, Promoter(s): Prof.dr. B. Krug, EPS-2009-164-ORG, http://hdl.handle.net/1765/15426

Hakimi, N.A, *Leader Empowering Behaviour: The Leader's Perspective: Understanding the Motivation behind Leader Empowering Behaviour*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2010-184-ORG, http://hdl.handle.net/1765/17701

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promoter(s): Prof.dr. A. Jolink & Prof.dr. S.J. Magala, EPS-2010-193-ORG, http://hdl.handle.net/1765/19494

Hernandez Mireles, C., *Marketing Modeling for New Products,* Promoter(s): Prof.dr. P.H. Franses, EPS-2010-202-MKT, http://hdl.handle.net/1765/19878

Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promoter(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-259-STR, http://hdl.handle.net/1765/32167

Hoever, I.J., *Diversity and Creativity: In Search of Synergy*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, http://hdl.handle.net/1765/37392

Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet*, Promoter(s): Prof.dr. H.P.G. Pennings & Prof.dr. A.R. Thurik, EPS-2011-246-STR, http://hdl.handle.net/1765/26447

Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promoter(s): Prof.dr. D. De Cremer & Dr. M. van Dijke, EPS-2011-244-ORG, http://hdl.handle.net/1765/26228

Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesign,* Promoter(s): Prof.dr. M.J.C.M. Verbeek & Prof.dr. R.J. Mahieu, EPS-2010-196-F&A, http://hdl.handle.net/1765/19674

Hytönen, K.A. *Context Effects in Valuation, Judgment and Choice*, Promoter(s): Prof.dr.ir. A. Smidts, EPS-2011-252-MKT, http://hdl.handle.net/1765/30668

Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2013-288-LIS, http://hdl.handle.net/1765/ 39933

Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics,* Promoter(s): Prof.dr. L.G. Kroon, EPS-2011-222-LIS, http://hdl.handle.net/1765/22156

Jaspers, F.P.H., *Organizing Systemic Innovation*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, http://hdl.handle.net/1765/14974

Jiang, T., *Capital Structure Determinants and Governance Structure Variety in Franchising*, Promoter(s): Prof.dr. G. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-F&A, http://hdl.handle.net/1765/14975

Jiao, T., *Essays in Financial Accounting*, Promoter(s): Prof.dr. G.M.H. Mertens, EPS-2009-176-F&A, http://hdl.handle.net/1765/16097

Kaa, G. van, *Standard Battles for Complex Systems: Empirical Research on the Home Network,* Promoter(s): Prof.dr.ir. J. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-ORG, http://hdl.handle.net/1765/16011

Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems,* Promoter(s): Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, http://hdl.handle.net/1765/19532

Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promoter(s): Prof.dr. S. Stremersch, EPS-2011-239-MKT, http://hdl.handle.net/1765/23610

Karreman, B., *Financial Services and Emerging Markets,* Promoter(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.P.G. Pennings, EPS-2011-223-ORG, http://hdl.handle.net/1765/ 22280

Kwee, Z., *Investigating Three Key Principles of Sustained Strategic Renewal: A Longitudinal Study of Long-Lived Firms,* Promoter(s): Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-174-STR, http://hdl.handle.net/1765/16207

Lam, K.Y., *Reliability and Rankings*, Promoter(s): Prof.dr. P.H.B.F. Franses, EPS-2011-230-MKT, http://hdl.handle.net/1765/22977

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promoter(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, http://hdl.handle.net/1765/30682

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promoter(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, http://hdl.handle.net/1765/23504

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models,* Promoter(s): Prof.dr.ir. J. Dul & Prof.dr. M.B.M. de Koster, EPS-2010-217-LIS, http://hdl.handle.net/1765/21527

Li, T., *Informedness and Customer-Centric Revenue Management,* Promoter(s): Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-146-LIS, http://hdl.handle.net/1765/14525

Liang, Q., *Governance, CEO Indentity, and Quality Provision of Farmer Cooperatives*m Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2013-281-ORG, http://hdl.handle.net/1765/1

Loos, M.J.H.M. van der, *Molecular Genetics and Hormones; New Frontiers in Entrepreneurship Research*, Promoter(s): Prof.dr. A.R. Thurik, Prof.dr. P.J.F. Groenen & Prof.dr. A. Hofman, EPS-2013-287-S&E, http://hdl.handle.net/1765/ 40081

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promoter(s): Prof.dr. J. Spronk & Prof.dr.ir. U. Kaymak, EPS-2011-229-F&A, http://hdl.handle.net/1765/ 22814

Maas, K.E.G., *Corporate Social Performance: From Output Measurement to Impact Measurement,* Promoter(s): Prof.dr. H.R. Commandeur, EPS-2009-182-STR, http://hdl.handle.net/1765/17627

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promoter(s): Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, http://hdl.handle.net/1765/22744

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, http://hdl.handle.net/1765/34930

Meuer, J., *Configurations of Inter-Firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promoter(s): Prof.dr. B. Krug, EPS-2011-228-ORG, http://hdl.handle.net/1765/22745

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promoter(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, http://hdl.handle.net/1765/32343

Milea, V., *New Analytics for Financial Decision Support*, Promoter(s): Prof.dr.ir. U. Kaymak, EPS-2013-275-LIS, http://hdl.handle.net/1765/ 38673

Moonen, J.M., *Multi-Agent Systems for Transportation Planning and Coordination*, Promoter(s): Prof.dr. J. van Hillegersberg & Prof.dr. S.L. van de Velde, EPS-2009-177-LIS, http://hdl.handle.net/1765/16208

Nederveen Pieterse, A., *Goal Orientation in Teams: The Role of Diversity*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-162-ORG, http://hdl.handle.net/1765/15240

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in Short-term Planning and in Disruption Management*, Promoter(s): Prof.dr. L.G. Kroon, EPS-2011-224-LIS, http://hdl.handle.net/1765/22444

Niesten, E.M.M.I., *Regulation, Governance and Adaptation: Governance Transformations in the Dutch and French Liberalizing Electricity Industries*, Promoter(s): Prof.dr. A. Jolink & Prof.dr. J.P.M. Groenewegen, EPS-2009-170-ORG, http://hdl.handle.net/1765/16096

Nijdam, M.H., *Leader Firms: The Value of Companies for the Competitiveness of the Rotterdam Seaport Cluster,* Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2010-216-ORG, http://hdl.handle.net/1765/21405

Noordegraaf-Eelens, L.H.J., *Contested Communication: A Critical Analysis of Central Bank Speech,* Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2010-209-MKT, http://hdl.handle.net/1765/21061

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promoter(s): Prof.dr. G. van der Pijl & Prof.dr. H. Commandeur & Prof.dr. M. Keil, EPS-2012-263-S&E, http://hdl.handle.net/1765/34928

Nuijten, I., *Servant Leadership: Paradox or Diamond in the Rough? A Multidimensional Measure and Empirical Evidence*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-183-ORG, http://hdl.handle.net/1765/21405

Oosterhout, M., van, *Business Agility and Information Technology in Service Organizations,* Promoter(s): Prof,dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, http://hdl.handle.net/1765/19805

Oostrum, J.M., van, *Applying Mathematical Models to Surgical Patient Planning*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2009-179-LIS, http://hdl.handle.net/1765/16728

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on Bureaucracy and Formal Rules*, Promoter(s): Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, http://hdl.handle.net/1765/23250

Otgaar, A.H.J., *Industrial Tourism: Where the Public Meets the Private,* Promoter(s): Prof.dr. L. van den Berg, EPS-2010-219-ORG, http://hdl.handle.net/1765/21585

Ozdemir, M.N., *Project-level Governance, Monetary Incentives and Performance in Strategic R&D Alliances*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, http://hdl.handle.net/1765/23550

Peers, Y., *Econometric Advances in Diffusion Models*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-251-MKT, http://hdl.handle.net/1765/ 30586

Pince, C., *Advances in Inventory Management: Dynamic Models,* Promoter(s): Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, http://hdl.handle.net/1765/19867

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, http://hdl.handle.net/1765/30848

Potthoff, D., *Railway Crew Rescheduling: Novel Approaches and Extensions,* Promoter(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. L.G. Kroon, EPS-2010-210-LIS, http://hdl.handle.net/1765/21084

Poruthiyil, P.V., *Steering Through: How Organizations Negotiate Permanent Uncertainty and Unresolvable Choices*, Promoter(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. S. Magala, EPS-2011-245-ORG, http://hdl.handle.net/1765/26392

Pourakbar, M. *End-of-Life Inventory Decisions of Service Parts*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, http://hdl.handle.net/1765/30584

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promoter(s): Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2013-282-S&E, http://hdl.handle.net/1765/1

Rijsenbilt, J.A., *CEO Narcissism; Measurement and Impact*, Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, http://hdl.handle.net/1765/ 23554

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promoter(s): Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, http://hdl.handle.net/1765/18013

Rosmalen, J. van, *Segmentation and Dimension Reduction: Exploratory and Model-Based Approaches*, Promoter(s): Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, http://hdl.handle.net/1765/15536

Roza, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of Innovation, Absorptive Capacity and Firm Size,* Promoter(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, http://hdl.handle.net/1765/22155

Rubbaniy, G., *Investment Behavior of Institutional Investors*, Promoter(s): Prof.dr. W.F.C. Vershoor, EPS-2013-284-F&A, http://hdl.handle.net/1765/ 40068

Rus, D., *The Dark Side of Leadership: Exploring the Psychology of Leader Self-serving Behavior*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-178-ORG, http://hdl.handle.net/1765/16726

Schellekens, G.A.C., *Language Abstraction in Word of Mouth,* Promoter(s): Prof.dr.ir. A. Smidts, EPS-2010-218-MKT, http://hdl.handle.net/1765/21580

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promoter(s): Prof.dr. G.M.H. Mertens, EPS-2013-283-F&A, http://hdl.handle.net/1765/39655

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization,* Promoter(s): Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, http://hdl.handle.net/1765/19714

Srour, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, http://hdl.handle.net/1765/18231

Stallen, M., *Social Context Effects on Decision-Making; A Neurobiological Approach*, Promoter(s): Prof.dr.ir. A. Smidts, EPS-2013-285-MKT, http://hdl.handle.net/1765/ 39931

Sweldens, S.T.L.R., *Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands,* Promoter(s): Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, http://hdl.handle.net/1765/16012

Tarakci, M., *Behavioral Strategy; Strategic Consensus, Power and Networks*, Promoter(s): Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-280-ORG, http://hdl.handle.net/1765/ 39130

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promoter(s): Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, http://hdl.handle.net/1765/37265

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes,* Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, http://hdl.handle.net/1765/18457

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments,* Promoter(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, http://hdl.handle.net/1765/19868

Tröster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, http://hdl.handle.net/1765/23298

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision Making*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, http://hdl.handle.net/1765/ 37542

Tzioti, S., *Let Me Give You a Piece of Advice: Empirical Papers about Advice Taking in Marketing,* Promoter(s): Prof.dr. S.M.J. van Osselaer & Prof.dr.ir. B. Wierenga, EPS-2010-211-MKT, hdl.handle.net/1765/21149

Vaccaro, I.G., *Management Innovation: Studies on the Role of Internal Change Agents,* Promoter(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-212-STR, hdl.handle.net/1765/21150

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains,* Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, http://hdl.handle.net/1765/19594

Venus, M., *Demystifying Visionary Leadership; In Search of the Essence of Effective Vision Communication*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2013-289-ORG, http://hdl.handle.net/1765/ 40079

Verwijmeren, P., *Empirical Essays on Debt, Equity, and Convertible Securities*, Promoter(s): Prof.dr. A. de Jong & Prof.dr. M.J.C.M. Verbeek, EPS-2009-154-F&A, http://hdl.handle.net/1765/14312

Visser, V., *Leader Affect and Leader Effectiveness; How Leader Affective Displays Influence Follower Outcomes*, Promoter(s): Prof.dr. D. van Knippenberg, EPS-2013-286-ORG, http://hdl.handle.net/1765/40076

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, http://hdl.handle.net/1765/30585

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, http://hdl.handle.net/1765/18012

Wall, R.S., *Netscape: Cities and Global Corporate Networks,* Promoter(s): Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, http://hdl.handle.net/1765/16013

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promoter(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, http://hdl.handle.net/1765/26564

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2011-242-F&A, http://hdl.handle.net/1765/26066

Wang, Y., *Corporate Reputation Management; Reaching Out to Find Stakeholders*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2013-271-ORG, http://hdl.handle.net/1765/ 38675

Weerdt, N.P. van der, *Organizational Flexibility for Hypercompetitive Markets: Empirical Evidence of the Composition and Context Specificity of Dynamic Capabilities and Organization Design Parameters,* Promoter(s): Prof.dr. H.W. Volberda, EPS-2009-173-STR, http://hdl.handle.net/1765/16182

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value of Dutch Firms*, Promoter(s): Prof.dr. A. de Jong, EPS-2013-277-F&A, http://hdl.handle.net/1765/ 39127

Wubben, M.J.J., *Social Functions of Emotions in Social Dilemmas*, Promoter(s): Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2009-187-ORG, http://hdl.handle.net/1765/18228

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, http://hdl.handle.net/1765/18125

Yang, J., *Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics*, Promoter(s): Prof.dr. H.E. Harlambides, EPS-2009-157-LIS, http://hdl.handle.net/1765/14527

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promoter(s): Prof.dr. M.B.M. de Koster, EPS-2013-276-LIS, http://hdl.handle.net/1765/1

Zhang, D., *Essays in Executive Compensation*, Promoter(s): Prof.dr. I. Dittmann, EPS-2012-261-F&A, http://hdl.handle.net/1765/32344

Zhang, X., *Scheduling with Time Lags,* Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, http://hdl.handle.net/1765/19928

Zhou, H., *Knowledge, Entrepreneurship and Performance: Evidence from Country-level and Firm-level Studies,* Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. L.M. Uhlaner, EPS-2010-207-ORG, http://hdl.handle.net/1765/20634

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, http://hdl.handle.net/1765/23422

Erasmus Research Institute of Management - ERiM

VEHICLE ROUTING WITH UNCERTAIN DEMAND

In distribution networks a supplier transports goods from a distribution center to customers by means of vehicles with limited capacity. Drivers will drive routes on which they visit multiple customers to make deliveries. Typically, deliveries are made regularly and a fixed schedule is maintained. A fixed schedule is beneficial for many operational purposes, as it for instance allows for easy planning of the packing of the vehicles at the distribution center, or it allows the customer to roster the delivery handling personnel. A fixed schedule is often reused to make weekly deliveries for a period of a year or longer.

However, at the moment of designing a schedule, the demand of the customers is usually unknown. Moreover, in most cases, demand of a customer will be different for each delivery. Therefore, it will be necessary to construct or adapt vehicle routes for each day of delivery, without deviating too much from the fixed schedule.

In this thesis several different views on a fixed schedule are explored. It addresses the need from practice to incorporate the uncertainty of demand in transportation models to increase the efficiency of transport. Innovative vehicle routing models are presented taking uncertain or varying demand into account. New algorithms using state-of-the-art methods are presented based on these models, to construct fixed schedules and vehicle routes. The algorithms make use of recent scientific advances in mathematical programming, specifically in the domain of vehicle routing.

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM PhD Series

Research in Management

Erasmus Research Institute of Management - ERiM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel.	+31 10 408 11 82
Fax	+31 10 408 96 40
E-mail	info@erim.eur.nl
Internet	www.erim.eur.nl

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)