

Congestion Avoidance and Break Scheduling within Vehicle Routing

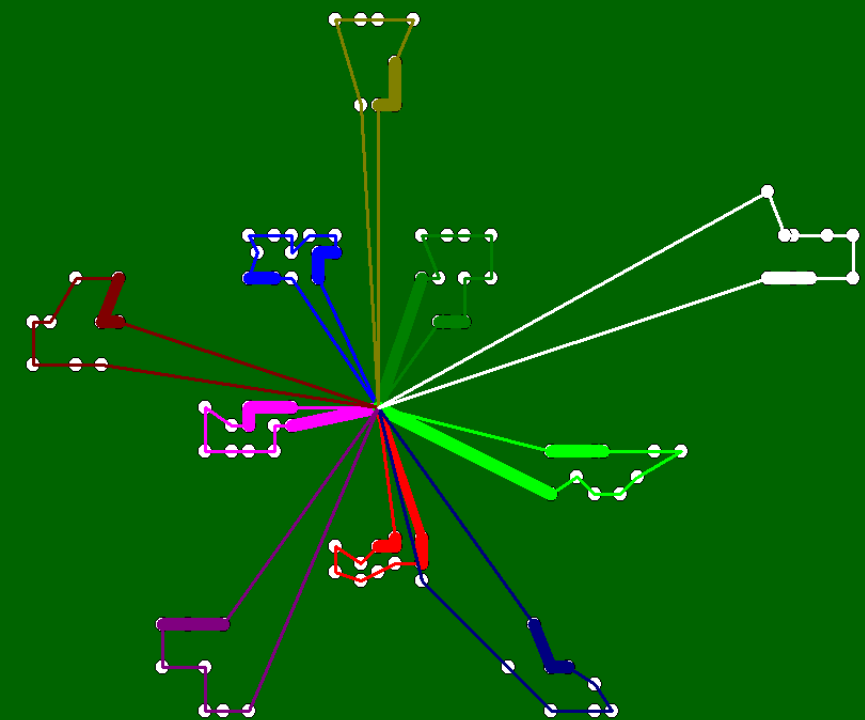
Vehicle routing is a complex daily task for businesses such as logistic service providers and distribution firms. Planners have to assign many orders to many vehicles and, for each vehicle, assign a delivery sequence. The objective is to minimize total transport costs. These costs typically include the number of vehicles used and the total travel distance or time. Two general timing restrictions make vehicle routing particularly difficult: traffic congestion and driving hours regulations. As a result of traffic congestion, travel times depend on the time of departure. Therefore, vehicle routing also involves the subtask of optimizing each vehicle's departure times (both from the depot and from the customers). Driving hours regulations - which pose restrictions on driving and working times (between breaks) - have to be taken into account, making departure time optimization particularly difficult.

In this research, we study the Vehicle Routing Problem under time-dependent travel times and driving hours regulations. We propose a generic solution method for Vehicle Routing Problems that can handle various restrictions, such as vehicle capacities and service time windows. Furthermore, we demonstrate that this method performs very well on problems which include driving hours regulations.

Test results on Vehicle Routing Problems with traffic congestion are also very promising. Most delays caused by traffic congestion can be avoided by considering them when developing vehicle route plans. This is done by avoiding predictably busy areas during problematic hours.

The solution methods proposed in this thesis are not limited to the problems they were initially designed for. We illustrate how they can be used in other studies, such as policy making, by analyzing vehicle routing from a distributed decision making perspective. In conclusion, there are various applications of the solution methods proposed in this thesis and they may allow for substantial improvements in practice.

Congestion Avoidance and Break Scheduling within Vehicle Routing



Congestion Avoidance and Break Scheduling within Vehicle Routing

Leendert Kok

Dissertation committee

Chairman / Secretary	Prof. dr. P.J.J.M. van Loon
Promotor	Prof. dr. W.H.M. Zijm
Assistant Promotors	Dr. ir. J.M.J. Schutten Dr. ir. E.W. Hans
Members	Prof. dr. ir. E.C. van Berkum Prof. dr. B. Fleischmann Prof. dr. J.L. Hurink Prof. dr.-ing. H. Kopfer



This thesis is number D130 of the thesis series of the Beta Research School for Operations Management and Logistics. The Beta Research School is a joint effort of the departments of Technology Management, and Mathematics and Computing Science at the Eindhoven University of Technology and the Centre for Telematics and Information Technology at the University of Twente. Beta is the largest research centre in the Netherlands in the field of operations management in technology-intensive environments. The mission of Beta is to carry out fundamental and applied research on the analysis, design, and control of operational processes.

This research has been partly funded by Transumo. Transumo (TRANSition SUSTainable MOBility) is a Dutch platform for companies, governments and knowledge institutes that cooperate in the development of knowledge regarding sustainable mobility.



Ph.D. thesis, University of Twente, Enschede, the Netherlands

Printed by Wöhrmann Print Service

© A.L. Kok, Enschede, 2010

All rights reserved. No part of this publication may be reproduced without the prior written permission of the author.

ISBN 978-90-365-2990-7

CONGESTION AVOIDANCE AND
BREAK SCHEDULING WITHIN
VEHICLE ROUTING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 9 april 2010 om 15:00 uur

door

Adrianus Leendert Kok
geboren op 22 augustus 1982
te Stolwijk

Dit proefschrift is goedgekeurd door de promotor:

prof. dr. W.H.M. Zijm

en de assistent-promotoren:

dr. ir. J.M.J. Schutten

dr. ir. E.W. Hans

Acknowledgements

Writing this thesis has been a great experience for me, but also a very challenging one. During this process, I received a lot of support from several people. I thank all of them, the ones below in particular.

First, I express my great gratitude to the people who supervised me the last four years. My supervisors Marco Schutten and Erwin Hans have been of great support to me right from the beginning. Their professional advice and critical view on my work has substantially improved the quality of it. I also thank my promoter Henk Zijm for his support. I am particularly grateful that he, despite being only in our department in the final year, put great effort and time in helping to improve my thesis.

Next, I thank my colleagues from the OMPL department for the working atmosphere. I really enjoyed the discussions and laughters we had during the coffee and lunch breaks; I will miss them. I thank my roommate Rob for his pleasant company and for his help on LaTeX and programming related problems.

Furthermore, I would like to thank the people with who I collaborated with over the past four years. I express my gratitude to Herbert Kopfer and Manuel Meyer for hosting me for two months at their department at the University of Bremen. It was a great and productive time, resulting in three scientific papers. I also enjoyed the beer and bratwurst we had at the Schlachte. Also many thanks to Joaquim Gromicho and Jelke van Hoorn from ORTEC. Our collaboration not only resulted in a scientific paper, but their extensive expertise from practice has also had a fundamental impact on my thesis.

I thank Johann Hurink for advising me to do this PhD project. He also gave various helpful comments on my research. I am also grateful for the financial support of Transumo.

Finally, I thank my family for their belief in me, and in particular my parents for always supporting me. I foremost thank my beloved wife Simona, who unconditionally supported me during the last four years and who always put faith in me, especially at the moments I needed it the most.

Leendert Kok
Enschede, April 2010

Contents

1	Introduction	1
1.1	Research motivation	1
1.2	Related literature	7
1.3	Outline of the thesis	16
2	Scheduling departure times	19
2.1	Introduction	19
2.2	The departure time scheduling problem	20
2.3	ILP formulation	22
2.4	Computational experiments	25
2.5	Model extensions	30
2.6	Conclusions	33
3	Dynamic programming for the vehicle routing problem	35
3.1	Introduction	35
3.2	Dynamic programming	37
3.3	Restricting the state space	40
3.4	The flexibility of our solution approach	41
3.5	Computational experiments	42
3.6	Conclusions	50
4	Vehicle routing with driving hours regulations	51
4.1	Introduction	51
4.2	EC social legislation	52
4.3	Solution method	55
4.4	Computational experiments	63
4.5	Conclusion	67
5	The impact of congestion avoidance	69
5.1	Introduction	69
5.2	Strategies	71
5.3	Solution methods	73
5.4	Speed model	73
5.5	Computational experiments	77

5.6	Conclusions	84
6	Vehicle routing with traffic congestion and break scheduling	87
6.1	Introduction	87
6.2	Problem description of the TDVRP-EC	88
6.3	Waiting time assumptions	90
6.4	Solution approach	91
6.5	Computational experiments	102
6.6	Conclusions	107
7	An alternative: distributed decision making	109
7.1	Introduction	109
7.2	Problem description	111
7.3	Framework for distributed decision making	112
7.4	Computational experiments	117
7.5	Conclusions	123
8	Conclusions and Recommendations	125
8.1	Conclusions	125
8.2	Recommendations for further research	130
	Bibliography	133
	Appendices	143
A	Complexity theory	143
B	Maximum number of breaks per day	145
C	Glossary of symbols	147
	Index	151
	Samenvatting	153
	About the author	157

Chapter 1

Introduction

Vehicle routing problems have been extensively studied over the past decades. On the one hand, efficient vehicle route plans lead to substantial cost savings for businesses such as logistic service providers and distribution firms. On the other hand, constructing efficient vehicle route plans is a complex daily task. Therefore, vehicle routing is a challenging problem in practice, and has drawn a lot of attention from the scientific community. In the last decades, vehicle routing models have evolved towards more and more realistic ones. Two common real life restrictions, however, have been generally ignored: 1) traffic congestion, and 2) driving hours regulations. In this thesis, we consider vehicle routing problems with these two timing restrictions.

This chapter is organized as follows. Section 1.1 motivates this research. In Section 1.2, we provide an overview of related literature on this topic and identify gaps between practice and literature. Section 1.3 provides the outline of this thesis.

1.1 Research motivation

Each year 1,721 billion ton kilometers of goods are transported over the European road networks. The total turnover of these transports varies between 0.8% and 7.9% of the national turnovers of the various countries in the European Union. In 2008, the turnover of the entire road transport sector in the Netherlands was 23 billion euro. Furthermore, transport costs constitute 4 to 10% of a product selling price (Coyle et al., 1996). Therefore, efficient vehicle route plans that reduce travel distances and travel times to a minimum have a large impact on the profitability of businesses in the transport sector, and has a substantial impact on national economies.

Vehicle routing is a complex daily planning problem for businesses such as logistic service providers and distribution firms. In practice, planners have to deal with many vehicles and have to assign large numbers of customer requests to these vehicles. When making these assignments, various restrictions have to be taken into consideration, such as vehicle capacities and compatibilities. Their objective within this task is generally to minimize the number of vehicles used to serve all customer requests, or to assign as many requests as possible to the available vehicles. Next, planners have to assign a delivery sequence for each vehicle. Their objective is then generally to minimize the total travel distance. Other restrictions such as time windows for customer service and precedence relations between customer visits have to be taken into consideration. This problem is generally known as the Vehicle Routing Problem (VRP).

In practice, two real life timing restrictions have a large impact on the quality of vehicle route plans: *time-dependent travel times* and *driving hours regulations*. Over the past decades, the problem of traffic congestion has been growing considerably. For example, in the USA the annual travel delay has grown from 2.5 billion delay hours in 1995 to 4.2 billion delay hours in 2005 (Schrank and Lomax, 2007). Another example is the loss of travel times¹ on highways in the Netherlands, which has grown with 53% over the period 2000-2007 (Jorritsma et al., 2008). Due to traffic congestion, travel times between customers depend on the time of departure. If traffic congestion is not accounted for in the vehicle route plans, vehicles may arrive too late at customers and truck drivers' hiring costs may very well become larger than expected. The Dutch Organization for Transport and Logistics (TLN) estimated that the total direct traffic congestion costs for the Dutch transport sector in 2002 amounts to 1.2 billion euro. TLN estimated that over 10% of the truck drivers' working hours are lost due to delays as a result of traffic congestion. To increase delivery reliability and avoid large truck drivers' hiring costs due to congestion delays, vehicle route plans must account for time-dependent travel times.

Within the European Union, there are about 1.5 million road accidents a year with over 40 thousand fatalities. Driver fatigue is considered as a major cause of such road accidents. To avoid driver fatigue, drivers should regularly take breaks and rest periods. Therefore, the European Union introduced Regulation (EC) No 561/2006² on driving hours for people working in road transport (European Union, 2006). This regulation poses restrictions on the amount of driving and non-rest times before breaks or rests of sufficient length must be taken. The regulation, which is valid for all member countries of the European Union, has to be taken into account by schedulers when establishing

¹The loss of travel time is measured against a reference speed of 100 km/h, which is considered as the average free-flow travel speed on highways in the Netherlands.

²The European Community (EC) social legislation on driving and working hours for people working in road transport basically comprises two legal acts: 1) Regulation (EC) No 561/2006, which poses restrictions on truck drivers' driving hours, and 2) Directive 2002/15/16, which poses restrictions on drivers' working hours. For an extensive description of the rules in the EC social legislation we refer to Chapter 4.

vehicle tours. Since their negligence can be fined severely, and not only drivers but also their employees are held responsible for violations, vehicle route plans must account for these driving hours regulations.

Time-dependent travel times and driving hours regulations have a large impact on the VRP models and proposed solution methods. The main reason is that, next to the assignment and sequencing problem, also a scheduling problem appears: the scheduling of all departure times for each vehicle. These departure times have a large impact on the quality of the vehicle route in terms of truck drivers' hiring costs and the times vehicles are unavailable for other services. This quality can be measured in terms of truck drivers' *duty times* (which is, in practice, a better quality measure than the, in the literature, commonly used quality measure *travel distance*). Duty time is defined as the time a truck driver is on duty, i.e., the total time from the moment he starts working until he completes his work. In addition to the quality of vehicle routes, the chosen departure times also determine the feasibility of vehicle routes with respect to driving hours regulations.

The practical applicability of scheduling departure times is emphasized by the fact that time-dependent travel times can now be better estimated, because more information is available on historical travel speeds during each time of the day. This information is already used by several route-planners on the Internet to provide travel time estimations depending on travel date and time of the day. An example is the route planner of the Dutch Motorists' Organization (ANWB). This planner provides travel time estimations based on historical information on time and location dependent travel speeds using a travel time estimator developed by the Netherlands Organization for Applied Scientific Research (TNO). Another example that demonstrates the applicability in practice of using historical travel time data to construct vehicle routes is of Eglese et al. (2006). For their analysis, they use a so-called Road TimetableTM produced by the UK road networking system ITIS Floating Vehicle Data. This Road TimetableTM contains information on time-dependent travel times for a road network based on a record of past road conditions so that travel times can be related to time of the day, day of the week, and season of the year.

In Chapter 2, we demonstrate that existing vehicle routing methods fail in case either time-dependent travel times or driving hours regulations are ignored. Therefore, we design a new solution method for the VRP that accounts for both timing restrictions. In the remainder of this section, we define the scope of this thesis (Section 1.1.1) and we state our research objective (Section 1.1.2) from which we extract our research questions (Section 1.1.3).

1.1.1 Scope

In this thesis, we focus on *off-line* vehicle route planning. This means that all relevant information for constructing the vehicle routes is already known in advance and not dynamically revealed during planning or execution of the vehicle routes. The off-line planning problem is important in practice, since customer requests are often already known at least one day in advance. Moreover, good estimations of traffic congestion delays can often already be made one day in advance (e.g., based upon historical travel time data), since crucial information for such estimations such as good weather forecasts are already available then.

Next, we restrict ourselves to the *deterministic* planning problem, which means that we model all relevant information for the planning problem as deterministic. Although the exact delays as a result of traffic congestion are sometimes hard to predict, the majority of the delays caused by traffic congestion are well-predictable, since they are recurrent because of commuter traffic (Skabardonis et al., 2003). In this thesis, we extend existing deterministic vehicle routing models. Considering stochastic elements such as stochastic demand quantities or stochastic presence of customer requests is beyond the scope of this thesis.

Furthermore, considering time-dependent travel times, we focus in our numerical experiments on delays caused by *peak hour traffic congestion*. As demonstrated by Skabardonis et al. (2003), the major part of traffic congestion delays are recurrent occurrences during the peak hours. Therefore, taking into account time-dependent travel times in off-line vehicle routing has a large impact if peak hour traffic congestion is considered. However, the solution methods proposed in this thesis that account for time-dependent travel times can handle delays at any moment of the day; they are not restricted to peak hour delays.

In addition, considering driving hours regulations, we propose our solution methods for vehicle routing problems taking into account the *EC social legislation*. Since the EC social legislation is more restrictive than the US Hours-Of-Service Regulations (Federal Motor Carrier Safety Administration, 2008) are, our solution methods can also be applied to problems taking into account the US Hours-Of-Service Regulations. Moreover, our solution methods handle restrictions on driving and working hours in a generic way, such that new restrictions can easily be included.

Finally, we focus on solution methods that can solve problem instances of *realistic sizes* within *practical computation times*. This implies that the method should be fast enough to solve large problem instances within practical computation times, but also that the method should be flexible with respect to additional real life restrictions. Since quality of the vehicle route plans is the major concern, we evaluate the solution method to be developed in terms of solution quality, computation time, and flexibility.

1.1.2 Research objective

The research objective of this thesis is

To design an off-line vehicle routing approach that improves delivery reliability and reduces transport costs by avoiding traffic congestion whenever possible taking into account the EC social legislation on driving and working hours.

In Section 1.1.3, we extract a number of research questions from this objective which we use as guidelines throughout this thesis.

As mentioned previously, ignoring traffic congestion causes unreliable route plans and higher costs than expected. If traffic congestion is accounted for in off-line vehicle route plans, then not only the reliability of these plans increases, but also traffic congestion could be avoided by, e.g., visiting customers in a different sequence. Next, if the EC social legislation is ignored, violations of this legislation might appear during the execution of the plans, or vehicles might arrive too late at customers if unscheduled breaks must be taken before service. Therefore, accounting for this legislation in off-line vehicle routing can avoid such costly events during the execution of the plans, resulting in substantial cost savings.

1.1.3 Research questions

We pose a number of research questions to guide us in reaching the research objective. We briefly elucidate each research question and state in which chapters we will study it.

1. *What is the state of the art in the literature on VRPs with time-dependent travel times and driving hours regulations?*

Before designing new solution methods for a new VRP model, we need to get familiarized with the state of the art in the literature on this topic. Section 1.2 discusses the existing literature on the VRP with time-dependent travel times and driving hours regulations. Since the literature on this topic is scarce, we also discuss the literature on the standard VRP. This gives us the necessary background for designing solution methods for a new VRP model.

2. *What impact do traffic congestion and driving hours regulations have on the performance of vehicle routes constructed with state of the art vehicle routing methods?*

In order to motivate a new approach for solving vehicle routing problems, we study the impact of traffic congestion and driving hours regulations on

existing vehicle routing methods. Chapter 2 demonstrates these impacts and the necessity of a new approach.

3. *What type of solution framework is suitable for handling different types of vehicle routing problems and incorporating complex timing restrictions such as time-dependent travel times and driving hours regulations?*

Chapter 3 proposes a general framework for solving vehicle routing problems that 1) can be applied to various VRP types and 2) is a route construction method which is suitable for incorporating complex timing restrictions such as driving hours regulations and time-dependent travel times.

4. *How can driving hours regulations be incorporated in off-line vehicle routing methods?*

We first focus on including driving hours regulations in the framework. Chapter 4 demonstrates how to incorporate the full EC social legislation on driving and working hours within our solution framework.

5. *What impact do different congestion avoidance strategies in off-line vehicle route plans have on the real-time performance of these plans?*

The solution framework proposed in Chapter 3 already incorporates time-dependent travel times. However, there are different ways and aggregation levels for incorporating time-dependent travel times within off-line vehicle routing. Chapter 5 formalizes these ways and aggregation levels in four strategies which contain different levels of congestion avoidance, and quantifies the impact of these strategies on the quality of off-line vehicle route plans in practice. These impacts show to what extent congestion avoidance within off-line vehicle routing can be profitable in practice.

6. *How can we account for both time-dependent travel times and driving hours regulations during the construction of vehicle routes with duty time minimization as the objective function?*

Although Chapter 4 and 5 propose algorithms for VRPs with driving hours regulations and time-dependent travel times, respectively, combining these two timing restrictions in one algorithm is still a difficult task. This is particularly the case when duty time minimization, which is an important objective in practice, is part of the objective function. Duty time minimization is much more involved than travel distance or travel time minimization when both time-dependent travel times and driving hours regulations are considered. Chapter 6 studies the VRP with time-dependent travel times and driving hours regulations and proposes a solution method based on the solution framework proposed in Chapter 3. This solution method has duty time minimization as the objective function.

1.2 Related literature

Vehicle routing problems have received a lot of attention in the literature. The first paper that considered the vehicle routing problem is of Dantzig and Ramser (1959). The VRP can be formally stated as the problem of optimally routing a fleet of vehicles such that all customer demands are satisfied and some objective function is optimized. It is a generalization of the Traveling Salesman Problem (TSP), because the TSP is a VRP with only one vehicle. Various variants of the VRP have been introduced in which different restrictions have to be satisfied and different objectives are stated. For an extensive overview on VRP variants and solution methods, we refer to Toth and Vigo (2002). In this section, we give an overview of the literature on vehicle routing problems and its variants, in which we focus on the VRP with time-dependent travel times and driving hours regulations, and we identify gaps between practice and literature.

Section 1.2.1 describes the basic VRP and the most common extensions of the VRP. Section 1.2.2 gives an overview of proposed solution methods for the VRP. Section 1.2.3 and 1.2.4 discuss the VRP with time-dependent travel times and the VRP with driving hours regulations, respectively, and Section 1.2.5 concludes this literature review.

1.2.1 VRP variants

The most basic variant of the VRP is the capacitated vehicle routing problem (CVRP). Within the CVRP, a homogeneous fleet of vehicles, located at a depot, has to serve a set of customers. Each vehicle has a capacity and each customer has a demand. The problem is to find for each vehicle a tour, starting and ending at the depot, such that the total travel distance is minimal and the total demand in each tour does not exceed the capacity of the vehicle.

The most studied extension of the CVRP is the vehicle routing problem with time windows (VRPTW). Within the VRPTW, each customer is given a time window in which its service must start. In case a vehicle arrives early at a customer, it has to wait until the time window opening time. Furthermore, a vehicle is not allowed to arrive later than the time window closing time. A special case is when soft time windows are considered. In this case, late arrivals are allowed, but they are penalized at certain costs. Different objectives have been considered for the VRPTW, but the general objective is to minimize the number of vehicle routes as the primary objective and the total travel distance as the secondary objective. The resulting problem is to find for each vehicle at most one route, starting and ending at the depot, such that all customers are visited by exactly one vehicle, the total demand in each vehicle route does not exceed the capacity of the vehicle, and each customer service starts within the given time window.

Another type of vehicle routing problem that has received a lot of attention in the literature in the last 30 years is the Pickup and Delivery Problem (PDP). Within the PDP, each customer request is given by a location pair (i, j) where a quantity must be picked up at location i and it must be delivered at location j . Each pickup and delivery pair must be served by the same vehicle and each pickup location must be visited before its corresponding delivery location can be visited. The total quantity present in each vehicle at each moment of time may not exceed the capacity of the vehicle. For an extensive overview of PDP variants and solution methods we refer to Parragh et al. (2008b).

Many other extensions to the basic VRP have been proposed. In case there are multiple depots and each vehicle has to start and end at the depot where it is located, then the Multi-Depot Vehicle Routing Problem (MDVRP) is considered. Another extension is when the vehicle fleet is heterogeneous, e.g., each vehicle k has a capacity Q_k . When a vehicle can perform more than one route, the Multi-Route Vehicle Routing Problem is considered. When customer demands can be split over different vehicles, then the Split-Delivery Vehicle Routing Problem (SDVRP) is considered. Sometimes some customers can only be served by a subset of the available vehicles, which is referred to as the Site-Dependent Vehicle Routing Problem. When vehicles are not required to return to the depot after their last customer visit, we consider the Open Vehicle Routing Problem (OVRP). The Periodic Vehicle Routing Problem (PVRP) considers a VRP with multiple time-periods (e.g., days), and in which each customer has a set of feasible demand schedules for the planning horizon. For example, a customer i may have a demand q_i which should be delivered before the end of the planning horizon, but it should be delivered all in one visit. Then the number of feasible demand schedules equals the number of days in the planning horizon. The resulting problem consists of selecting a demand schedule for each customer, and solving a CVRP for each day in the planning horizon with the demands corresponding to the selected demand schedules.

Within these VRP variants, it is generally assumed that travel times are time-independent and proportional to the travel distances. Therefore, the general secondary objective is to minimize the total travel distance, with minimizing the number of vehicles as the primary objective. For these problems, many solution methods have been developed. Since the literature on the VRP with time-dependent travel times and driving hours regulations is scarce, we first give an overview of solution methods for VRPs in general to obtain a good insight in successful solutions methods for (variants of) the VRP.

1.2.2 Solution methods for the VRP

The solution methods for the VRP can be categorized in exact methods and heuristics. Exact methods are designed to solve the problems to optimality. However, since the VRP is a generalization of the TSP, which is \mathcal{NP} -hard

(Garey and Johnson, 1979)³, only small problem instances can be solved to optimality within practical computation times. Therefore, various heuristics have been designed to solve larger problem instances of the VRP. These heuristics do not guarantee to find the optimal solution, but they are designed to find good - possibly near-optimal - solutions within practical computation times. We first present some well-known exact methods for the VRP. Then we discuss heuristics for the VRP.

Exact methods

Laporte and Nobert (1987) classify the exact methods for the VRP in three categories: branch & bound methods, dynamic programming (DP), and integer linear programming (ILP). These are the three main categories of exact methods for the VRP and its variants. For extensive overviews on these algorithms we refer to Laporte and Nobert (1987), Laporte (1992), and Toth and Vigo (2002).

Branch & bound (Land and Doig, 1960) is a method based on complete enumeration. However, by using clever bounds on (partial) solutions and systematic enumeration, large sets of candidate solutions can be discarded. For example, the cost of any solution is an upper bound for the cost of the optimal solution. If a subset of solutions can be proven to have costs exceeding this upper bound, then the whole subset can be discarded. Branch & bound methods for the VRP are based on sequentially building vehicle routes by means of a branch and bound tree. The first such method for the VRP was proposed by Christofides and Eilon (1969).

Dynamic Programming (Bellman, 1957) for the VRP is also based on sequentially building vehicle routes. However, complete enumeration is avoided by only expanding optimal partial vehicle routes. Dynamic Programming for the TSP was independently developed by Bellman (1962) and Held and Karp (1962). In Chapter 3, we extend this DP formulation to the VRP, and demonstrate that this formulation contains a flexible framework for solving various VRP types. Another DP formulation for the VRP was proposed by Eilon et al. (1974).

Several integer linear programming formulations have been proposed for the VRP. Set partitioning and column generation have proved to be a successful combination in solving various VRP types (Laporte, 1992). The set partitioning formulation is based on defining binary decision variables for each feasible vehicle route. The difficulty of such formulations is that they lead to a huge amount of binary decision variables. However, this is generally resolved by using a clever column generation (Ford and Fulkerson, 1958) scheme. Rao and Zions (1968) were the first to apply column generation to the VRP.

³For readers unfamiliar with the notion of \mathcal{NP} -hardness we refer to Appendix A which gives an introduction to complexity theory.

Heuristics

Since exact methods generally fail in solving realistic problem instances in practical computation times, the focus of solution methods for vehicle routing problems has moved to heuristic approaches. Heuristic algorithms (Pearl, 1984) aim to produce good solutions for different realistic problem instances, but have no guarantee for the solution quality.

Heuristic algorithms for the VRP can be categorized in constructive methods and improvement methods. Constructive methods are greedy approaches that are, in general, very fast and they construct a solution. Improvement methods are usually more sophisticated methods that typically require a solution as input. These improvement methods belong to the class of local search methods (Aarts and Lenstra, 1997). Extensive overviews of heuristics for vehicle routing problems can be found in Toth and Vigo (2002), Cordeau et al. (2002b), and Cordeau et al. (2005). We give a short overview of constructive and improvement methods for the VRP.

Constructive Heuristics

Probably the best known constructive heuristic for the VRP is the nearest neighbor heuristic. Menger (1930) already considered the nearest neighbor heuristic for the TSP. The sequential version of this heuristic constructs vehicle routes sequentially. The heuristic initializes the first route with the customer located nearest to the depot, and it extends this route each time with the nearest of all customers that can feasibly be added to the route. When no such extensions are possible anymore, the next vehicle route is initialized by the nearest of the remaining customers to the depot. There are several variants of the nearest neighbor heuristic, e.g., parallel nearest neighbor, route initiation with the farthest neighbor.

Another well-known greedy method is the savings heuristic of Clarke and Wright (1964). This heuristic is based on the fact that combining two vehicle routes into one route, such that the last customer of the first route is directly succeeded by the first customer of the second route, saves one vehicle and the additional distance for a detour through the depot. The heuristic is initialized by one-customer routes for each customer and it ends when no two routes can be combined anymore.

Next, we have the sweep algorithm of Gillett and Miller (1974). With this method, customer routes are constructed by drawing a straight line originating from the depot and rotating the line around the depot location. Each time the line intersects a customer location, the customer is added to the current vehicle route if there is enough capacity remaining. Otherwise, a new vehicle route is initialized.

Finally, there are the two-phase constructive methods such as the cluster-first route-second and the route-first cluster-second method. The first method, proposed by Fisher and Jaikumar (1981), creates a number of customer-clusters such that customers in the same cluster are located close to each other and the total demand of such customers does not exceed the vehicle's capacity. Next, through each cluster a vehicle route is determined, which yields solving a TSP for each cluster. The route-first cluster-second method was proposed by Beasley (1983) and basically yields solving a TSP through the depot and all customers, and then optimally partitioning the solution in feasible vehicle routes.

Improvement Heuristics

Improvement heuristics are designed to improve existing solutions to VRPs: they require a VRP solution as input. Several improvement strategies have been developed, some already in the early sixties (e.g. Lin, 1965). The improvement strategies iteratively move from one solution A to a new solution B by some (in general small) modification of A. If solution B can be reached in one improvement step from solution A, then we refer to solution B as a neighborhood solution of solution A.

The earliest improvement structures were only applied to single routes, the so-called intra-route improvement methods. These methods were originally designed for the TSP. The methods are based on edge exchanges that change the customer sequence within a route. Examples of these methods are r-opt (Lin, 1965), Or-opt (Or, 1976), and 4-opt* (Renaud et al., 1996). For extensive numerical analysis of these methods we refer to Johnson and McGeoch (1997).

The improvement methods involving different vehicle routes, so-called inter-route improvement methods, enhance a richer class of improvement strategies. Many different edge exchange schemes have been proposed for the VRP such as chain exchanges (Fahrion and Wrede, 1990) and λ -interchange mechanisms (Osman, 1993). We refer to Thompson and Psaraftis (1993), van Breedam (1994), and Kinderwater and Savelsbergh (1997) for extensive overviews and numerical analysis of these improvement methods.

Applying these improvement strategies to some solutions obtained by a constructive method leads to a local optimum, meaning that applying the improvement strategies again does not lead to any further improvement. However, in general, a local optimum does not coincide with the global optimum, i.e., the optimal solution. Therefore, several improvement mechanisms, so-called metaheuristics, have been proposed to escape from local optima and find better overall solutions. We describe the most successful metaheuristics proposed for the VRP.

Simulated annealing is a metaheuristic in which deteriorations of a solution are accepted with a certain probability. By allowing such deteriorations, solu-

tions can escape from local optima. The probability of accepting deteriorations is initially set relatively high and is lowered each time a predetermined number of iterations has passed. Simulated annealing has been applied to the VRP by, amongst others, Robuste et al. (1990), Alfa et al. (1991), and Osman (1993). We refer to van Breedam (1995) for a numerical analysis of several different implementations of simulated annealing for the VRP.

Deterministic annealing is similar to simulated annealing, however, a deterministic threshold is used for accepting solutions. Two variants of deterministic annealing are threshold accepting (Dueck and Scheuer, 1990) and record-to-record travel (Dueck, 1993). In case of threshold accepting, the threshold value consists of some user specified value that is added to the solution value of the last accepted solution. In case of record-to-record travel, the threshold value is a value (in general slightly larger than 1) multiplied with the solution value of the last accepted solution.

Within tabu search, the best neighborhood solution is chosen, where neighborhood solutions are all solutions that can be reached within one step from the last accepted solution using some improvement heuristics. However, to avoid returning to solution structures that actually have just been changed, a tabu list is maintained of a specified number of most recently accepted changes. Each time a new solution is accepted, the part of the old solution that has been changed (e.g., an edge in the old solution that has been removed) is inserted in the first position of the tabu list, while the last position of the tabu list is removed. Tabu search has been widely applied to the VRP. Sophisticated and successful applications of tabu search, amongst others, are the ones of Gendreau et al. (1994), Rochat and Taillard (1995), Cordeau et al. (2001), and Semet and Taillard (1993).

Next to these metaheuristics, several other metaheuristics have been applied to the VRP. Amongst these are genetic algorithms (Reeves, 2003), memetic algorithms (Moscato and Cotta, 2003), ant algorithms (Kawamuro et al., 1998), and neural networks (Ghaziri, 1991). These methods have received only limited attention in the VRP literature.

1.2.3 The VRP with time-dependent travel times

Due to the growing amount of traffic congestion in the past decades, vehicle routing models assuming time-independent travel times fail in many applications. Therefore, Malandraki and Daskin (1992) introduce the time-dependent vehicle routing problem (TDVRP). They propose an ILP-formulation for this problem and discuss a cutting plane method and a nearest-neighbor heuristic to solve the problem. They model the time-dependent travel times with travel time step functions. However, this model results in the unrealistic situation that a vehicle might overtake another vehicle by departing a bit later, but in a

time interval with a smaller travel time. If a model does not allow overtaking then it has the so-called *non-passing property*.

Several travel time models have been proposed to satisfy the non-passing property. The one that has been used most is a speed step function, such that two vehicles traveling along the same route at the same time always drive at the same speed (Hill and Benton, 1992; Ichoua et al., 2003; Eglese et al., 2006; Van Woensel et al., 2008; Donati et al., 2008). Others allow more complex travel time functions, but only if the slope of these functions is never smaller than -1 (Haghani and Jung, 2005; Hashimoto et al., 2006, 2008) (note that speed step functions never allow travel time functions with slopes smaller than -1). A special model is of Fleischmann et al. (2004), who obtain travel time step functions from a large database with historical travel times, and apply a smoothing procedure to these functions, resulting in continuous piecewise linear travel time functions that satisfy the non-passing property.

The solution methods for the TDVRP focus on local search methods. Tabu search has been applied by Ichoua et al. (2003), Eglese et al. (2006), and Van Woensel et al. (2008). Ichoua et al. (2003) propose an adaptation of the tabu search algorithm of Taillard et al. (1997) for the VRP with soft time windows. They develop an estimation function for the cost of neighborhood solutions, such that these costs can be estimated in constant time, instead of determining the exact costs in linear time. This method, however, fails when hard time windows are considered, since then the exact arrival times for each neighborhood solution must be determined in order to determine whether a neighborhood solution is feasible. Eglese et al. (2006) demonstrate how a road time table can be developed based on floating vehicle data to aid vehicle routing in scheduling. They provide a real life case in which several timing improvements (e.g., reductions of the number of time window violations) could be achieved. Van Woensel et al. (2008) demonstrate how a queuing model can be used to derive realistic travel times. They also propose a method to optimize departure times; however, they do not consider time windows.

Also some construction methods for the TDVRP have been proposed. Mandraki and Dial (1996) propose a restricted dynamic programming heuristic for the TSP with time-dependent travel times. The method is a generalization of the nearest-neighbor heuristic, and a restricted version of the DP algorithm for the TSP of Bellman (1962) and Held and Karp (1962). The unrestricted version of the algorithm is an exact approach if the non-passing property is satisfied. Fleischmann et al. (2004) propose adaptations of several savings and insertion algorithms. They also apply the 2-opt method and demonstrate that this method substantially improves the TDVRP solutions. Hsu et al. (2007) propose a nearest neighbor heuristic for the TDVRP with perishable food, in which not only the travel times are time-dependent, but also the amount of fresh perishable food in the vehicle.

Other methods for the TDVRP are the ant colony optimization algorithm

of Donati et al. (2008), the iterated local search methods of Hashimoto et al. (2006, 2008), and the genetic algorithm of Haghani and Jung (2005). Furthermore, Ahn (1991) develops feasibility checks for several improvement methods (customer insertion, concatenating two routes, customer exchange) in case of time-dependent travel times.

Within these methods, the general primary objective is to minimize the number of vehicles used and the general secondary objective is to minimize the total travel time. However, in practice, duty time minimization is often more important than travel time minimization. Especially when hard time windows are considered, duty times become an important cost factor, since large waiting times cause large hiring costs for the truck drivers and make the vehicles lengthy unavailable for other services. Minimizing travel times does not account for waiting times. The only paper we are aware of that considers minimal duty time as objective is of Savelsbergh (1992). We consider minimal duty times as the objective in Chapter 6.

Furthermore, many solution methods are based on local search. However, in a setting with time-dependent travel times, local updates have up- and downstream effects on the routes under consideration, which makes the evaluation of such updates much more computationally expensive than in VRP models without time-dependent travel times. We propose a general solution framework for VRPs in Chapter 3 and demonstrate in Chapter 5 and 6 how time-dependent travel times can be incorporated with only minor impacts on the running time of the algorithm.

Finally, the majority of the models consider customer networks and ignore the underlying road network. In practice, traffic congestion is time- *and* location-dependent. Therefore, determining time-dependent shortest paths may already resolve some of the delays caused by traffic congestion. In Chapter 5, we consider both the time-dependent shortest path problem and the TDVRP in a realistic setting.

1.2.4 The VRP with driving hours regulations

In all member countries of the European Union and in many other countries, driving and working hours of persons engaged in road transportation is legislated. In the European Union, driving hours are restricted by Regulation (EC) No 561/2006. Moreover, Directive 2002/15/EC restricting drivers' working hours has been implemented into national laws in most member countries of the European Union. These legal acts have to be taken into account by schedulers when establishing vehicle tours. As their negligence can be fined severely, these acts have an enormous impact on the design of vehicle tours in practice. The problem that arises here is a problem of combined vehicle routing and break scheduling. In the literature, however, only a few papers on vehicle

routing including breaks and rest periods can be found. All these papers only include parts of the mandatory legislation, which results in vehicle schedules that do not comply with the legal requirements.

Gietz (1994) investigates a VRP with breaks modeled as fictitious customers. Rochat and Semet (1994) use a similar approach. Stumpf (1998) includes driving time restrictions specified by the former Regulation (EEC) No 3820/85 into a tabu search metaheuristic, a great deluge algorithm, and a threshold accepting algorithm. Savelsbergh and Sol (1998) include breaks and daily rest periods into a branch and price algorithm for a pickup and delivery problem. Cordeau et al. (2002a) suggest the use of a multi-stage network for the inclusion of breaks in a VRP. Xu et al. (2003) present a column generation algorithm and some heuristics to solve a pickup and delivery problem which includes restrictions on driving times specified by the US Department of Transportation. They conjecture that finding driver schedules complying with these driving time restrictions is \mathcal{NP} -hard in the presence of multiple time windows. Archetti and Savelsbergh (2009) present a polynomial time algorithm for this problem in the presence of single time windows that runs in $O(n^3)$ time with n the number of customers the driver has to visit. Goel and Kok (2009b) present an algorithm for this problem that runs in $O(n^2)$ time. Goel and Kok (2009a) present a polynomial time algorithm for a similar problem of scheduling team drivers in the European Union that runs in $O(n^2)$ time. Also the case with modified rules on daily driving times, which allows truck drivers to extend their daily driving times for a limited number of times a week, is included in this algorithm. Campbell and Savelsbergh (2004) modify an insertion heuristic in such a way that it considers maximum shift times for drivers. Goel and Gruhn (2006) introduce a large neighborhood search algorithm for a VRP which takes into account maximum driving times according to the former Regulation (EEC) No 3820/85. Goel (2009) considers parts of the current Regulation (EC) No 561/2006 in a large neighborhood search algorithm. He presents computational results for modified problem instances of the Solomon (1987) test instances for the VRPTW. However, Goel (2009) concentrates on a set of basic rules of Regulation (EC) No 561/2006 and neglects some important modifications of these rules, which allow more flexibility. Additionally, Goel ignores the restrictions on working times set by Directive 2002/15/EC. Zäpfel and Bögl (2008) present a mixed-integer model for a combined vehicle routing and crew pairing problem, which considers breaks after 4.5 hours. To solve the model they apply a tabu search metaheuristic and a genetic algorithm. Bartodziej et al. (2009) use a column generation approach and some local search based metaheuristics for solving a combined vehicle and crew scheduling problem which incorporates rest periods for drivers. Kopfer and Meyer (2009) present an integer programming model for a TSP that considers all relevant rules of Regulation (EC) No 561/2006 for a weekly period.

In Chapter 2, 4, and 6 we consider Regulation (EC) no 561/2006. In Chapter 4, we also consider the impacts of Directive 2002/15/EC and the modifica-

tions of the rules in both the regulation and the directive.

1.2.5 Conclusions

Over the past decades, many variants of the VRP have been considered and many solution methods have been proposed. However, the TDVRP and the VRP with driving hours regulations have received only minor attention in the VRP literature. Despite the fact that these restrictions are common (i.e., each company in the European Union has to respect the EC social legislation, and traffic congestion has become a familiar concept in almost every urban area in the world), VRP models considering both these timing restrictions have - to the best of our knowledge - not been proposed so far.

Another observation we draw from the VRP literature is that almost every new variant of the VRP requires the development of a new solution method. Practice, however, is a dynamic environment in which problems and restrictions often change. Therefore, in practice there is a call for VRP models and solution methods that can easily adapt to new or modified problems. An exception seems the powerful pickup and delivery model and solution method of Pisinger and Ropke (2007), which can solve various different VRP types. However, this model does not include time-dependent travel times or driving hours regulations.

Finally, most solution methods for VRPs are based on local search. However, for VRPs with complex timing restrictions (e.g., time-dependent travel times, driving hours regulations), local search methods are less suitable, since neighborhood evaluations require substantially more computational effort under such timing restrictions. In Chapter 3, we propose a solution approach that forms a flexible framework for solving VRPs. This framework covers all variants of the VRP mentioned so far. Moreover, this solution approach is a constructive heuristic that is suitable for incorporating complex timing restrictions such as time-dependent travel times and driving hours regulations. In this thesis, we evaluate this solutions framework in terms of solution quality, computation time, and flexibility.

1.3 Outline of the thesis

As mentioned previously, timing restrictions such as time-dependent travel times and driving hours regulations introduce a new problem within vehicle routing: the departure time scheduling problem. Chapter 2 proposes an ILP model for this problem, which determines for a vehicle route (i.e., a customer visit sequence) whether there exists a feasible departure schedule. We apply

this ILP model to routes obtained by state of the art VRP models and solution methods to investigate the impact of time-dependent travel times and driving hours regulations on these routes. This impact shall motivate the development of new VRP models and solution methods for the vehicle routing problem considered in this thesis.

Chapter 3 proposes a new solution framework for solving VRPs based on restricted dynamic programming. We show that this framework is flexible with respect to solving a number of variants of the VRP. Furthermore, we demonstrate that the heuristic constructs solutions of acceptable quality for these VRP variants within practical computation times. Therefore, the solution framework fulfills all requirements on solution quality, computation time, and flexibility. Moreover, the framework provides a basis for Chapter 4, 5, 6, and 7, in which we incorporate time-dependent travel times and driving hours regulations in the framework.

Chapter 4 proposes a solution method for the VRPTW with driving hours regulations based on the restricted dynamic programming heuristic of Chapter 3. We design a break scheduling algorithm to account for all regulations in the EC social legislation on driving and working hours. This break scheduling algorithm takes a local perspective on scheduling breaks and rest periods. The major advantage of such a local perspective is that the running time complexity of the restricted dynamic programming heuristic is the same for the VRPTW and the VRPTW with driving hours regulations. The planning horizon considered in Chapter 4 is one week, such that the solution method can also handle complex requirements in the EC social legislation on, e.g., night rests. Moreover, we show how the method can be extended to longer planning horizons, and how it can be used in a rolling horizon framework.

As mentioned previously, there are several strategies and aggregation levels for incorporating time-dependent travel times in vehicle routing. These strategies allow different levels of congestion avoidance to reduce transport costs. Chapter 5 proposes four strategies in which different levels of traffic congestion avoidance are adopted by determining (time-dependent) shortest paths and solving (time-dependent) vehicle routing problems. We also propose a time-dependent speed model that we use to obtain a representative set of VRP instances on real road networks. We investigate the impact and profitability of the different strategies on these problem instances. The results show to what extent congestion avoidance within off-line vehicle route plans can be profitable in practice. To solve the time-dependent vehicle routing problems, we apply the restricted dynamic programming heuristic of Chapter 3.

Chapter 6 proposes a solution method for the VRPTW with time-dependent travel times and the EC social legislation. Duty time minimization is used as the secondary objective. In this chapter, we consider one-day planning, since this is the most relevant planning horizon to minimize duty times (e.g., for on-duty night rests other costs apply than for working times). Moreover, reliable

information on customer requests and time-dependent travel times is typically available one day in advance. In particular, time-dependent travel times are less reliable in case of longer planning horizons due to larger uncertainties in, e.g., weather forecasts. Certain rules in the EC social legislation that consider longer planning horizons than one-day planning may still have an impact on one-day planning. An example is the modified rule on the daily driving time, which allows to extend this driving time to 10 hours at most twice a week. The solution method in Chapter 6 is flexible with respect to such rules, since the application of these rules require a small modification of the parameters for the planning of that day.

In practice, vehicle routing and break scheduling often involves a distributed decision making process in which both planners and drivers are responsible for certain parts of the planning process. A centralized planning in which all decisions are taken by the planner may therefore not always be realistic. Chapter 7 analyzes combined vehicle routing and break scheduling from this alternative distributed decision making perspective. With this perspective, planning is decentralized such that decisions concerning customer clustering, routing, and break scheduling is distributed over planners and drivers. We use the restricted DP heuristic of Chapter 4 to solve the different problems encountered in the decision process. We also analyze the realistic setting in which planners and drivers may have conflicting objectives⁴.

Chapter 8 presents the main conclusions of the research in this thesis and poses recommendations for further research.

⁴The material of Chapter 2 to 7 has also appeared, or will appear, as articles in the scientific literature. Although they build on each other, important notations and definitions are sometimes repeated in each chapter, to make them self-contained. In order to facilitate reading, we have decided to maintain that structure throughout the thesis, possibly at the cost of some repetition.

Chapter 2

Scheduling departure times

2.1 Introduction

When time-dependent travel times and driving hours regulations are considered within vehicle routing, a new set of decision variables is introduced to the problem: the departure times of each vehicle. Since travel times in practice depend on the times of departure, and the amount of driving and duty time available to a truck driver is limited by driving hours regulations, the feasibility of a vehicle route depends on the chosen departure times.

As pointed out in Chapter 1, scheduling departure times is applicable in practice, since information is available on historical travel speeds during each time of the day. Furthermore, explicit break scheduling (which follows from explicit departure time scheduling) is required by law, since the vehicle route plans proposed by schedulers to the truck drivers must comply with the EC social legislation on driving and working hours. Violations can be fined, sometimes even if they are within the plans and not (yet) in the execution of the plans. Therefore, there is a strong call from practice for methods that schedule departure times within vehicle routes such that all timing restrictions are satisfied.

Scheduling departure times is difficult in an integrated solution method for the VRP. The reason is that a change in departure time at one customer may have large effects up- and downstream a partial vehicle route. This is caused, on the one hand, by the time-dependency of the travel times, and, on the other hand, by the fact that driving hours regulations restrict the amount of accumulated driving times until a break has to be scheduled. Therefore, we first propose a decomposition approach, in which the departure time scheduling problem is approached as a post-processing step of solving a VRPTW. This implies that departure times are scheduled after the customers are assigned to

vehicles and the customer visit sequences for each vehicle are determined.

We propose an ILP formulation for the departure time scheduling problem, which determines for a given vehicle route whether a feasible set of departure times exists. Note that we restrict ourselves to the feasibility problem; so far we do not handle an optimization problem. However, we propose some extensions of the ILP formulation that somehow ‘quantifies the infeasibility’ of the problem by, e.g., minimizing the number of late arrivals or minimizing the maximum late time. We apply the ILP model to a set of vehicle routes obtained by state of the art vehicle routing methods applied to well-known benchmark instances. These experiments shall demonstrate that algorithms that neglect time-dependent travel times and driving hours regulations construct vehicle routes that cannot be made feasible with respect to time-dependent travel times and driving hours regulations without changing the customer-vehicle assignments or customer visit sequences. Therefore, new solution methods for vehicle routing problems with time-dependent travel times and driving hours regulations have to be developed.

This chapter¹ is organized as follows. Section 2.2 formally introduces the departure time scheduling problem. Next, Section 2.3 proposes an ILP formulation for the departure time scheduling problem and discusses the modeling of the time-dependent travel times in the ILP formulation. We test the ILP formulation in Section 2.4 on vehicle routes obtained by state of the art solution methods from well-known benchmark instances. Section 2.5 shows that our approach is flexible with respect to several practical extensions and Section 2.6 summarizes the main findings in this chapter.

2.2 The departure time scheduling problem

We approach the departure time scheduling problem as a post-processing step of the VRPTW, i.e., the input of the problem is a set of nodes $i = 0, \dots, n + 1$ ², which need to be visited in this order and service must start within given time windows. Nodes 0 and $n + 1$ both represent the depot in this case, while the other nodes represent customer locations. In general, however, all nodes may represent different locations. For now, we assume that all customers have to be served on one day. Next, since in practice breaks are usually scheduled at customers, we assume that breaks can only be taken at customers. There are exceptions, especially in long distance (international) transports, where breaks are also scheduled at parking lots along the routes. In Section 2.5, we show how our ILP formulation can be extended to the case where breaks can also be scheduled at parking lots, and we show how to extend the ILP formulation to multi-day planning.

¹This chapter is based on Kok et al. (2008)

²Appendix C provides a glossary of symbols that are used in this thesis

Each customer i has given a time window $[e_i, l_i]$ in which its service has to start. The service time of each customer is given by s_i . The travel time between two successive customers i and $i + 1$ is given by $c_i(X_i^d)$, where X_i^d is the chosen departure time from customer i . The chosen departure times at the customers are restricted by driving hours regulations.

Since driving hours regulations are country dependent, it might be hard to propose a general formulation covering the driving hours regulations of each country in the world. Since the European driving hours regulations (European Union, 2006) are more restrictive than the North-American ones (Federal Motor Carrier Safety Administration, 2008) and they are valid for all member countries in the European Union, we base our formulation on the European driving hours regulations. Considering one-day planning, these regulations consist of four components:

1. A truck driver is not allowed to drive more than 9 hours (t_{max}) on a day.
2. A period between two breaks of at least 0.75 hours (b_{total}) is called a *driving period*. The accumulated driving time in each driving period may not exceed 4.5 hours (t_{dp}). The break that ends a driving period may be reduced to 0.5 hours (b_{min}^1) if an additional break of at least 0.25 hours (b_{min}^2) is taken anywhere during that driving period. We call a break of at least b_{min}^1 (b_{min}^2) hours a break of type 1 (2). Therefore, each type 1 break is also a type 2 break.
3. The driving hours regulations do not allow service times at customers to be considered as break time. Therefore, if a truck driver takes a break at a customer, he can do that before or after serving the customer, or both. However, each waiting period before and after serving a customer should be checked separately whether it can be considered a break of type 1 or 2.
4. A truck driver is not allowed to be on duty for more than 13 hours (d_{max}).

These regulations apply throughout the entire European Union and they are hard constraints. In order to control the regulations, each vehicle is equipped with a tachograph that records all driving and working activities of the current truck driver. The regulations are so restrictive that companies often need costly solutions to fulfill their appointments with customers, while respecting the driving hours regulations. For example, there are cases in which truck drivers drive by car to a certain location to take over the vehicle of another truck driver who has reached his driving limit for that day or week. The regulations allow for a few modifications, such as an extension of the total driving time to 10 hours or an extension of the duty time to 15 hours. However, these modifications are only allowed for a limited number of times (e.g., the extension to 10 hours of driving time is only allowed 2 times a week). We show in Section 2.5 how to extend our ILP model to also handle these modifications.

2.3 ILP formulation

Since breaks can be taken both before and after serving a customer, we have to decide for every customer i at what time service starts and at what time the vehicle leaves the customer. We introduce the decision variables X_i^s and X_i^d to indicate the time to start service at customer i and the departure time from customer i , respectively. In addition, we introduce the decision variables W_i^s and W_i^d to indicate the waiting time of the vehicle directly before and after serving customer i .

There are two types of breaks: those of at least b_{min}^1 hours and those of at least b_{min}^2 hours. We introduce the decision variables $B_i^{p,l}$ to indicate the break time at customer $i = 1, \dots, n$, before ($p = s$) or after ($p = d$) serving the customer, and of type $l = 1, 2$. To check whether a waiting time can be considered a break, we also introduce binary decision variables $Y_i^{p,l}$. If a realization of W_i^p does not exceed b_{min}^l , then the corresponding variables $Y_i^{p,l}$ and $B_i^{p,l}$ are set to 0. Otherwise, the corresponding variable $Y_i^{p,l}$ can take value 1, allowing $B_i^{p,l}$ to take the value of W_i^p .

Finally, to ensure that enough breaks are taken during and at the end of each driving period, we introduce binary decision variables V_{ij} ($j > i$). If a driving period starts at customer i and ends at customer j , then V_{ij} is set to 1. In that case, the break time at customer j must be at least b_{min}^1 , and the total break time at customers k ($i < k \leq j$) must be at least b_{total} . This results in the following ILP formulation:

$$X_i^s = X_{i-1}^d + c_{i-1}(X_{i-1}^d) + W_i^s \quad (i = 1, \dots, n+1) \quad (2.1)$$

$$X_i^d = X_i^s + s_i + W_i^d \quad (i = 0, \dots, n) \quad (2.2)$$

$$X_i^s \geq e_i \quad (i = 0, \dots, n+1) \quad (2.3)$$

$$X_i^s \leq l_i \quad (i = 0, \dots, n+1) \quad (2.4)$$

$$W_i^p \geq b_{min}^l Y_i^{p,l} \quad (i = 1, \dots, n, l = 1, 2, p = s, d) \quad (2.5)$$

$$B_i^{p,l} \leq M Y_i^{p,l} \quad (i = 1, \dots, n, l = 1, 2, p = s, d) \quad (2.6)$$

$$B_i^{p,l} \leq W_i^p \quad (i = 1, \dots, n, l = 1, 2, p = s, d) \quad (2.7)$$

$$\sum_{k=0}^j c_k(X_k^d) \leq t_{dp} + M \sum_{k=1}^j V_{0k} \quad (j = 1, \dots, n) \quad (2.8)$$

$$\sum_{k=i}^j c_k(X_k^d) \leq t_{dp} + M \left(\sum_{k=i+1}^j V_{ik} + 1 - \sum_{k=0}^{i-1} V_{ki} \right) \quad (2.9)$$

$$(i = 1, \dots, n-1, j = i+1, \dots, n)$$

$$\sum_{j=1}^n V_{0j} \leq 1 \quad (2.10)$$

$$\sum_{j=i+1}^n V_{ij} \leq \sum_{k=0}^{i-1} V_{ki} \quad (i = 1, \dots, n-1) \quad (2.11)$$

$$B_j^{s,1} + B_j^{d,1} \geq b_{min}^1 V_{ij} \quad (i = 0, \dots, n-1, j = i+1, \dots, n) \quad (2.12)$$

$$\sum_{k=i+1}^j (B_k^{s,2} + B_k^{d,2}) \geq b_{total} V_{ij} \quad (i = 0, \dots, n-1, j = i+1, \dots, n) \quad (2.13)$$

$$\sum_{k=0}^n c_k(X_k^d) \leq t_{max} \quad (2.14)$$

$$X_{n+1}^s - X_0^d \leq d_{max} \quad (2.15)$$

$$\text{All variables} \geq 0 \quad (2.16)$$

$$Y_i^{p,l} \in \{0, 1\} \quad (i = 1, \dots, n, l = 1, 2, p = s, d) \quad (2.17)$$

$$V_{ij} \in \{0, 1\} \quad (i = 0, \dots, n-1, j = i+1, \dots, n) \quad (2.18)$$

There is no objective within the ILP formulation, since we focus on the feasibility problem. Section 2.5 describes how more information on the feasibility of the problem instances (e.g., number of late arrivals) can be obtained by setting a certain objective.

Constraints (2.1) and (2.2) define the time to start service at and the departure time from each customer. Constraints (2.3) and (2.4) ensure that service starts in the given time window. Note that within a VRPTW nodes $i = 0$ and $i = n + 1$ represent the depot, such that these constraints also ensure that the vehicles depart and return within the given time horizon. Constraints (2.5) check whether a waiting period is enough to be considered a break. If not, then $Y_i^{p,l}$ is set to 0 and Constraints (2.6) become tight. These constraints are only defined for customers $i = 1, \dots, n$ since taking a break before leaving the depot or after returning at the depot does not make sense. Constraints (2.7) ensure that the break time never exceeds the waiting time. Constraints (2.8) ensure that the first driving period does not exceed t_{dp} . If the total driving time between customers 0 and $j + 1$ exceeds t_{dp} ($\sum_{k=0}^j c_k(X_k^d) > t_{dp}$), then the first driving period must end at a customer $k, 0 < k < j + 1$ ($\sum_{k=1}^j V_{0k} = 1$). Constraints (2.9) ensure that the succeeding driving periods end in time. If a driving period starts at customer i ($\sum_{k=0}^{i-1} V_{ki} = 1$) and the total driving time between customers i and $j + 1$ exceeds t_{dp} ($\sum_{k=i}^j c_k(X_k^d) > t_{dp}$), then this driving period must end at a customer $k, i < k < j + 1$ ($\sum_{k=i+1}^j V_{ik} = 1$). Constraints (2.10) ensure that the first driving period ends at most once and

Constraints (2.11) ensure that each succeeding driving period ends at most once. Constraints (2.12) ensure that a break of at least b_{min}^1 hours is taken at a customer where a driving period ends and Constraints (2.13) ensure that in each driving period the total break time is at least b_{total} . Finally, Constraints (2.14) and (2.15) ensure that the total driving time does not exceed t_{max} and the total duty time (the difference between the arrival at the end node and the departure at the start node) does not exceed d_{max} , respectively. Note that a sufficiently large and tight value for M is $l_{n+1} - e_0$.

So far, we have modeled the travel time function as a general function that depends on the time of departure. However, in general such a function cannot be written in proper ILP form. In Section 2.3.1, we model the time-dependent travel times as a continuous piecewise linear travel time function, and show how to write it in ILP form.

2.3.1 Travel time modeling

Several ways of modeling the time-dependent travel times have been proposed in the literature. Malandraki and Daskin (1992) propose a travel time step function. A disadvantage of this approach is that the non-passing property is not satisfied, i.e., if vehicles A and B traverse the same link in the network, and vehicle B departs later than vehicle A, but with a smaller travel time, then vehicle B could arrive earlier than vehicle A. Haghani and Jung (2005) propose a continuous travel time function in which the slope is always greater than -1. In that case, departing later can never result in an earlier arrival. The disadvantage of an arbitrary continuous travel time function is that it does not need to be (piecewise) linear. Therefore, we choose to follow the approach of Ichoua et al. (2003), who propose a travel speed step function for each link in the network. This approach results in a continuous piecewise linear travel time function. Since two vehicles traversing the same link drive with the same speed at any moment of time, the non-passing property is satisfied. Figure 2.1 shows an example of a speed function; Figure 2.2 presents the resulting travel time function.

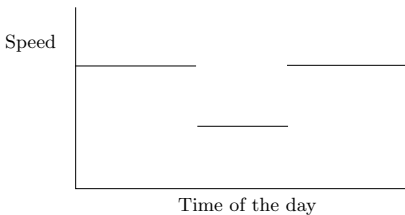


Figure 2.1: Speed function

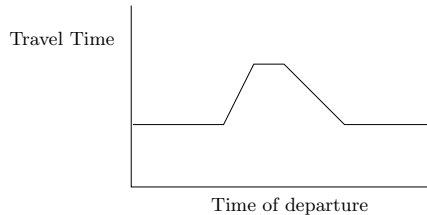


Figure 2.2: Travel time function

Since the travel time function is piecewise linear, we can write it as m_i

different functions $a_{i,r} + b_{i,r} (X_i^d - g_{i,r})$, where each $g_{i,r}$, $r = 1, \dots, m_i$ indicates a time at which the slope of the travel time function changes. Furthermore, $a_{i,r}$ is the travel time at time $g_{i,r}$ and $b_{i,r}$ is the slope of the r^{th} linear function. To determine in which interval $[g_{i,r}, g_{i,r+1}]$ the chosen departure time X_i^d falls, we introduce binary variables $U_{i,r}$ which can take value 1 only if $g_{i,r} \leq X_i^d \leq g_{i,r+1}$. Next, we introduce variables $X_{i,r}^d$ which take the value of X_i^d if the corresponding variable $U_{i,r}$ is 1, and 0 otherwise. By replacing the function $c_i(X_i^d)$ by the variable C_i we derive the following ILP formulation to determine the travel time for departure time X_i^d :

$$\sum_{r=1}^{m_i} U_{i,r} = 1 \quad (i = 0, \dots, n) \quad (2.19)$$

$$g_{i,r} U_{i,r} \leq X_{i,r}^d \quad (i = 0, \dots, n, r = 1, \dots, m_i) \quad (2.20)$$

$$g_{i,r+1} U_{i,r} \geq X_{i,r}^d \quad (i = 0, \dots, n, r = 1, \dots, m_i) \quad (2.21)$$

$$\sum_{r=1}^{m_i} X_{i,r}^d = X_i^d \quad (i = 0, \dots, n) \quad (2.22)$$

$$C_i \geq a_{i,r} + b_{i,r} (X_i^d - g_{i,r}) + M (U_{i,r} - 1) \quad (i = 0, \dots, n, r = 1, \dots, m_i) \quad (2.23)$$

Constraints (2.19) ensure that exactly one $U_{i,r}$ takes value 1. The $U_{i,r}$ with value 1 and Constraints (2.20) and (2.21) force the corresponding variable $X_{i,r}^d$ to be in the interval $[g_{i,r}, g_{i,r+1}]$, and all other variables $X_{i,r}^d$ to be 0. Constraints (2.22) force the only $X_{i,r}^d$ greater than 0 to equal X_i^d , and therefore $U_{i,r}$ can only take value 1 if $g_{i,r} \leq X_i^d \leq g_{i,r+1}$. Finally, if $U_{i,r}$ equals 1 then Constraints (2.23) present the right travel time functions. If $U_{i,r}$ equals 0 then Constraints (2.23) are non-restrictive.

2.4 Computational experiments

We test our ILP formulation on a set of routes obtained from best known solutions to the well-known Solomon (1987) instances for the VRPTW. The computational experiments demonstrate the necessity of developing a new vehicle routing method that constructs vehicle routes accounting for time-dependent travel times and driving hours regulations. We implemented the ILP formulation in Delphi 7 and solved it using CPLEX 11 on a Pentium 4, 3.40GHz CPU and 1.00 GB of RAM.

We test our ILP formulation on a selection of the 100-customer problem instances developed by Solomon (1987). We use those problem instances for

which best known solutions identified by heuristics can be obtained from the literature. The routes obtained from these solutions form the problem instances for the departure time scheduling problem. Our preference was to test the departure time scheduling problem on routes obtained from good solutions to TDVRP instances, since these routes already account for time-dependent travel times. Unfortunately, the involved authors can no longer provide these routes (Ichoua et al., 2003; Fleischmann et al., 2004; Haghani and Jung, 2005). However, we shall demonstrate that even if one of the timing restrictions - time-dependent travel times or driving hours regulations - is neglected during the construction of the vehicle routes, then in many cases it is not possible to find feasible departure schedules. This implies that the routes are not applicable in practice.

The Solomon problem instances are categorized into 3 types of instances: 1) c-instances, where customer locations are clustered, 2) r-instances, where customers are uniformly randomly located, and 3) rc-instances, where 50 percent of the customers are clustered and 50 percent are uniformly randomly located. Each customer is given a hard time window in which its service must start. The time window at the depot indicates the earliest feasible departure time and the latest feasible return time at the depot. Furthermore, some of the problem instances have a relatively large time window at the depot and vehicles with a relatively large capacity, resulting in large vehicle routes (25 up to 50 customers), while other instances have a relatively small time window at the depot, resulting in small vehicle routes (about 10 customers). Since the number of customers visited in a vehicle route defines the input size of the departure time scheduling problem, we discern small and large vehicle routes. This distinction allows us to investigate the impact of the input size of the problem on the required computation time. The number of customers visited in a vehicle route ranges from 4 to 51 customers. We categorize the problem instances into small (≤ 20 customers) and large (> 20 customers) problem instances.

The travel speed in the networks of the Solomon instances equals 1. As a result, the travel times in the Solomon instances equal the euclidean distance between the customer locations. Since the travel speed is time-independent, we develop speed patterns, such that the average travel speed remains 1. This methodology is similar to the one proposed by Ichoua et al. (2003). We define the time window at the depot from 6:30 am until 7:30 pm (i.e., corresponding to a 13 hour working day which equals the maximum available duty time d_{max} for each driver) and we assume that the morning traffic peak causes congestion from 7:00 am until 9:00 am, and the evening traffic peak from 5:00 pm until 7:00 pm. Furthermore, we distinguish between light, medium, and heavy congestion. These three types of congestion cause speed drops during the peak hours of 25, 50, and 75 percent, respectively. Table 2.1 presents the resulting speed patterns. Note that with these speed patterns, speed drops during the peak hours are the same in the whole network. These speed patterns serve the purpose of the experiments in this section, which is to determine the impact of driving

Type of Congestion\Time	6:30-7	7-9	9-17	17-19	19-19:30
Light	1.08	0.81	1.08	0.81	1.08
Medium	1.18	0.59	1.18	0.59	1.18
Heavy	1.30	0.33	1.33	0.33	1.30

Table 2.1: Speed Patterns

hours regulations and different levels of traffic congestion on the feasibility of individual vehicle routes constructed with time-independent travel times. Chapter 5 proposes a more elaborate speed model for real road networks, in which speed patterns depend both on time and location.

The problem instances are composed of the vehicle routes resulting from the best known solutions to the Solomon instances and the travel speed patterns in Table 2.1. Furthermore, we set $b_{min} = 0.25$, $b_{total} = 0.75$, $t_{dp} = 4.5$, and $t_{max} = 9$, corresponding with the European driving hours regulations (note that $d_{max} = 13$ is already enforced by the depot opening hours). We investigate whether the developed routes allow feasible departure schedules. Since we are testing the impact of two different realistic factors in vehicle routing, we consider two test scenarios: in Scenario 1 we *do not* consider driving hours regulations and in Scenario 2 we *do* consider driving hours regulations. In both scenarios, we apply the ILP model for each of the three speed drops as described before, as well as the case in which there is no speed drop at all. This allows us to investigate the impact of driving hours regulations on vehicle routes in congestion free networks. Table 2.2 and 2.3 present results on computation times and percentage of infeasible VRP routes by scheduling the departure times for Scenarios 1 and 2, respectively.

The computation times are small enough for practical use. The maximum computation time over all instances is less than 2.1 seconds (for Scenario 1 even less than 0.2 seconds) and the average is less than 0.1 seconds. Therefore, our approach to determine feasible departure schedules as a post-processing step of a VRPTW is fast enough for practical use.

The solution methods for the original VRP instances do not account for time-dependent travel times and driving hours regulations, and as a consequence the obtained routes are often too tight with respect to the time windows to schedule mandatory breaks. It generally holds that heavier traffic congestion results in fewer feasible vehicle routes. Therefore, vehicle routing methods should account for time-dependent travel times. However, this is not sufficient to obtain vehicle routes that can be used in practice under driving hours regulations.

For the cases with medium and heavy congestion, Table 2.2 and 2.3 look rather similar in terms of percentage of infeasible VRP routes. This may indicate that these infeasibilities are mainly caused by the time-dependent travel

Problem Size	# Instances	Congestion Type	CPU (ms)	VRP route Infeasible
Small ^a	164	No	8	0.00%
		Light	13	18.90%
		Medium	9	48.78%
		Heavy	6	71.95%
Large ^b	25	No	37	0.00%
		Light	51	16.00%
		Medium	45	44.00%
		Heavy	31	68.00%
Average	189	No	12	0.00%
		Light	18	18.52%
		Medium	14	48.15%
		Heavy	10	71.43%

^aAll routes in the best known solutions of instances r103 and rc106 (Li and Lim, 2003), r104, r107, r109, r111 and rc107 (Shaw, 1997), r108, r110 and rc105 (Berger and Barkaoui, 2004), and rc101, rc102, rc103, rc104 and rc108 (Czech and Czarnas, 2002)

^bAll routes in the best known solutions of instances r211 (Rochat and Taillard, 1995), and rc201, rc204, rc205, rc206, rc202, rc203 and rc207 (Czech and Czarnas, 2002)

Table 2.2: Results Scenario 1: no driving hours regulations

Problem Size	# Instances	Congestion Type	CPU (ms)	VRP route Infeasible
Small	164	No	13	59.76%
		Light	27	63.41%
		Medium	27	76.83%
		Heavy	19	85.98%
Large	25	No	134	24.00%
		Light	179	36.00%
		Medium	310	56.00%
		Heavy	283	68.00%
Average	189	No	29	55.03%
		Light	47	59.79%
		Medium	64	74.07%
		Heavy	53	83.60%

Table 2.3: Results Scenario 2: with driving hours regulations

times and not so much by the driving hours regulations. This raises the question whether existing solution methods for the VRPTW with time-dependent travel times produce routes that can feasibly adopt breaks to comply with the driving hours regulations. To test whether this is the case, we test for the feasible routes in Table 2.2 (i.e., feasible with respect to time-dependent travel times) whether there also exists feasible departure schedules for these routes with respect to driving hours regulations.

Table 2.4 presents the percentage of the vehicle routes that are feasible with respect to time-dependent travel times, but turn out to be infeasible when also driving hours regulations are respected. Therefore, some 30 to 50 percent of the routes that are feasible with respect to time-dependent travel times, but that ignore driving hours regulations, fail in practice. This problem is clearly caused by the methods that build the vehicle routes; it does not affect the applicability of the departure time scheduling problem in practice. As we shall argue in the remainder of this section, it is not straightforward to overcome this problem.

Congestion Type	Reduction of # feasible vehicle routes
No	55.03%
Light	50.65%
Medium	50.00%
Heavy	42.59%

Table 2.4: Relative decrease of # feasible vehicle routes when driving hours regulations have to be respected

First, slack time could be added to the original problem instances, such that time is reserved for scheduling mandatory breaks after the vehicle routes have been developed. To keep the proposed solution methods in the VRP literature directly applicable, this slack time should be spread evenly over the travel times between (or service times at) the customers. We tested this approach by adding one sixth of slack travel time. At least one sixth of slack travel time is required, because the total travel time in a driving period does not exceed 4.5 hours, while 45 minutes of break time needs to be scheduled in this period. Computational experiments show that this approach works well for light congestion (the percentage of infeasible vehicle routes reduces from 59.79% to 0.53%). However, with medium and heavy congestion the percentage of infeasible routes remains rather large (14.29% and 50.79%, respectively). A drawback of this approach is that built-up slack might be lost when truck drivers have to wait at customers before they can start service. This is one of the reasons that many routes remain infeasible in case of medium and heavy congestion. Moreover, slack travel time is often lost such that the minimal number of vehicle routes will generally not be attained.

Second, one might argue that the departure time scheduling problems are infeasible because of the ‘tightness’ of the optimal solutions. We therefore also tested less sophisticated methods to develop the vehicle routes, resulting in worse VRP solutions with respect to the overall objective, but with possibly less tight routes with respect to the time windows. We tested this approach with a straightforward nearest neighbor heuristic. The results show that the percentage of infeasible vehicle routes decreases (from 68.12% to 49.14%, on average), but the number of vehicle routes increases dramatically (from 189 to

263). Moreover, the number of feasible vehicle routes may increase, but the total number of customers in all feasible vehicle routes decreases (from 777 to 724, of the 2300 customers in total).

Therefore, since decomposition methods in which driving hours regulations and time-dependent travel times are only handled in a post-processing step fail, the need arises to develop new vehicle routing methods that account for these two timing restrictions. Chapter 6 proposes a solution method for the VRPTW with time-dependent travel times and driving hours regulations.

2.5 Model extensions

In this section, we describe various model extensions that may better suit particular cases. First, we propose two objective functions to ‘quantify the infeasibility of the problem instance’. Second, we describe how to extend the formulation to multi-day planning. Third, we describe how to account for the modification of the rule on the daily driving times, which allows to extend this driving time to 10 hours twice a week. Fourth, we describe how to incorporate the possibility of taking breaks at parking lots, in addition to taking breaks at customer sites.

2.5.1 Objective functions

Solving the ILP formulation tells us whether a feasible departure schedule exists, and if such a schedule exists it provides us a set of feasible departure times. However, when the problem is infeasible, there is no information on the ‘degree of infeasibility’. For example, in case of infeasibility it might be interesting to know whether removing only one customer visit allows for a feasible departure schedule. In order to get insight in the ‘degree of infeasibility’, we have to allow late arrivals at customers. In order to keep the number of late arrivals as small as possible, we propose the following two objective functions: 1) minimize the number of late arrivals, 2) minimize the maximum late time.

For the first objective, we have to allow late arrivals, which can be controlled by introducing binary decision variables $L_i, i = 0, \dots, n + 1$. L_i takes value 1 if the vehicle arrives late at customer i , and can take 0 otherwise. We adjust Constraints (2.4) and add Objective (2.24) as follows:

$$\text{Min} \sum_{i=0}^{n+1} L_i \quad (2.24)$$

$$X_i^s \leq l_i + ML_i \quad (i = 0, \dots, n + 1) \quad (2.4')$$

In case the objective value is greater than zero, the problem is infeasible with respect to the customer time windows. The L_i with value 1 indicate at which customers it is impossible to arrive in time. Note that late arrivals at nodes 0 and $n + 1$ may not make sense or may even not be allowed (e.g., in case of strict opening hours at the depot). This situation is easily controlled by adding constraints on the values of the corresponding L_i .

For the second objective, we have to account for the amount of late time at customers. We introduce decision variables $A_i \in [0, M], i = 0, \dots, n + 1$, which take the value of the late time at customer i , or take value 0 in case there is no late time. We introduce decision variable A^{max} to account for the maximum late time. Then, we add Objective (2.25) and Constraint (2.26), and adjust Constraints (2.4) as follows:

$$\text{Min } A^{max} \quad (2.25)$$

$$A^{max} \geq A_i \quad (i = 0, \dots, n + 1) \quad (2.26)$$

$$X_i^s \leq l_i + A_i \quad (i = 0, \dots, n + 1) \quad (2.4'')$$

2.5.2 Multi-day planning

The ILP formulation proposed in Section 2.3 assumes one-day planning. There are several practical cases in which it is more convenient to extend the formulation to multi-day planning. We demonstrate that this extension can easily be incorporated in our ILP formulation.

For multi-day planning, some extra restrictions are imposed by the driving hours regulations. Both the European and North American driving hours regulations impose a maximum on the total driving time and the total working time on a day, after which a rest has to be taken. More formally, after driving at most t_{max} hours and being on duty for at most d_{max} hours, a rest of at least 11 hours (t_{rest}) has to be taken. Also, a maximum is imposed on the total driving and working time in an entire week. We illustrate how the ILP formulation of Section 2.3 can be extended to one-week planning.

First, in Constraint (2.14), t_{max} must be replaced by the maximum driving time in a week. Next, to check whether a waiting time at a customer can be considered a rest, we introduce variables $B_i^{p,rest}, p = s, d$ and binary variables $Y_i^{p,rest}$, and we add the following constraints to the ILP formulation:

$$W_i^p \geq t_{rest} Y_i^{p,rest} \quad (i = 1, \dots, n, p = s, d) \quad (2.27)$$

$$B_i^{p,rest} \leq M Y_i^{p,rest} \quad (i = 1, \dots, n, p = s, d) \quad (2.28)$$

$$B_i^{p,rest} \leq W_i^p \quad (i = 1, \dots, n, p = s, d) \quad (2.29)$$

Next, we need to check whether the driving (duty) time does not exceed the maximum driving (duty) time on each day before a night's rest is taken. Therefore, we introduce the notion of daily period which has the following three properties: 1) Each daily period ends with a night's rest, 2) in each daily period the driving and duty time do not exceed the maximum driving and duty time, and 3) each time a daily period ends, a new daily period is initiated. Next, we introduce binary variables V_{ij}^{daily} which are set to 1 if a daily period starts at customer i and ends at customer j . To ensure that the driving time does not exceed the maximum driving time in each daily period, and each daily period ends with a rest of at least t_{rest} hours, we add the following constraints:

$$\sum_{k=0}^j c_k(X_k^d) \leq t_{max} + M \sum_{k=1}^j V_{0k}^{daily} \quad (j = 1, \dots, n) \quad (2.30)$$

$$\sum_{k=i}^j c_k(X_k^d) \leq t_{max} + M \left(\sum_{k=i+1}^j V_{ik}^{daily} + 1 - \sum_{k=0}^{i-1} V_{ki}^{daily} \right) \quad (2.31)$$

$$(i = 1, \dots, n-1, j = i+1, \dots, n)$$

$$\sum_{j=1}^n V_{0j}^{daily} \leq 1 \quad (2.32)$$

$$\sum_{j=i+1}^n V_{ij}^{daily} \leq \sum_{k=0}^{i-1} V_{ki}^{daily} \quad (i = 1, \dots, n-1) \quad (2.33)$$

$$B_j^{s,rest} + B_j^{d,rest} \geq t_{rest} V_{ij}^{daily} \quad (i = 0, \dots, n-1, j = i+1, \dots, n) \quad (2.34)$$

Ensuring that the duty time does not exceed the maximum duty time during each daily period can be done with similar constraints. The only difference is that waiting times and service times also add to the total duty time. Therefore, both the arrival time and the end of service time at each customer is a possible moment for exceeding the total duty time. Since there are two possible moments at each customer for starting (ending) a daily period, the total number of possible daily periods is four times the number of possible daily periods for the case with maximum driving time. Therefore, we need four times the number of binary variables V_{ij}^{daily} to indicate when a daily period starts and when it ends. Similarly, we need two times the constraints of type (2.30) and (2.33), and four times the constraints of type (2.31) and (2.34), to ensure that each daily period ends with a break of t_{rest} , the total duty time in the daily period does not exceed d_{max} , and each time a daily period ends, a new daily period is initiated.

2.5.3 Modification of rule on daily driving times

To account for the possibility of extending the driving time twice a week, we add binary variables $E_i, i = 0, \dots, n$, which take value 1 if a new daily driving period starts at customer i , and the total driving time of this period can be extended to 10 hours. To ensure that the total number of daily driving time extensions does not exceed 2, we add Constraint 2.35. Next, Constraints 2.36 ensure that $E_i, i > 0$ can only take value 1 if a new daily driving period starts at customer i . Finally, to allow for the driving time extensions of 1 hour, we adjust Constraints (2.30) and (2.31).

$$\sum_{i=0}^n E_i \leq 2 \quad (2.35)$$

$$E_i \leq \sum_{k=0}^{i-1} V_{ki}^{daily} \quad (i = 1, \dots, n) \quad (2.36)$$

$$\sum_{k=0}^j c_k(X_k^d) \leq t_{max} + E_0 + M \sum_{k=1}^j V_{0k}^{daily} \quad (j = 1, \dots, n) \quad (2.30')$$

$$\sum_{k=i}^j c_k(X_k^d) \leq t_{max} + E_i + M \left(\sum_{k=i+1}^j V_{ik}^{daily} + 1 - \sum_{k=0}^{i-1} V_{ki}^{rest} \right) \quad (2.31')$$

$(i = 1, \dots, n-1, j = i+1, \dots, n)$

2.5.4 Breaks at parking lots

To incorporate the possibility of taking breaks at parking lots along the route, we can simply model these parking lots as customers with zero service time and maximum time window (i.e., $[e_o, l_{n+1}]$). We assume that these parking lots are along the route and that the time needed to park the vehicle and to leave the parking lot are negligible.

2.6 Conclusions

We introduced the departure time scheduling problem and approached it as a post-processing step of solving a VRPTW. We proposed an ILP formulation for the departure time scheduling problem which is flexible with respect to several practical extensions. This flexibility was demonstrated while writing this thesis, as the European driving hours regulations changed and we were able to quickly adapt the ILP formulation to the new regulations.

The computational experiments show that the departure times can be scheduled within practical computation times. Furthermore, they demonstrate that VRP routes will only be of practical use if driving hours regulations and time-dependent travel times are accounted for during the development of these routes. Existing solution methods fail in practice when these two common timing restrictions are considered. We argued that the most appropriate way to solve this problem is to develop new vehicle routing methods. In Chapter 3, we propose a solution framework which is flexible with respect to various variants of the VRP, and forms a suitable framework for incorporating complex timing restrictions such as time-dependent travel times and driving hours regulations, as we shall demonstrate in the remaining chapters of this thesis.

Chapter 3

Dynamic programming for the vehicle routing problem

3.1 Introduction

As pointed out in Chapter 2, growing traffic congestion problems and strict regulations on drivers' driving and working hours cause existing VRP solution methods to fail in practice. Therefore, new solution methods accounting for these timing restrictions should be developed. In this chapter¹, we propose a solution method for the VRP based on restricted dynamic programming. This solution approach is a constructive heuristic, as opposed to the majority of the VRP literature in which the focus is on route improvement heuristics. The main reasons for relying on constructive methods are the following.

Local search methods have proved to be very successful in solving large vehicle routing problems. They develop high quality solutions for for a wide range of VRP extensions. However, they are not suitable for including complex timing restrictions such as time-dependent travel times and driving hours regulations. When such timing restrictions are considered, neighborhood evaluations are much more involved, since local changes can have major effects up- and downstream the vehicle routes under consideration. As a result, local updates become much more computationally expensive.

In addition, many different extensions of the VRP, besides time-dependent travel times and driving hours regulations, appear in practice. Companies such as logistic service providers and distribution firms often have their own set of restrictions of which a certain part may be general to most companies, but other parts being unique for a specific company. As a consequence, each company

¹This chapter is based on Gromicho et al. (2008)

requires a uniquely adapted solution method. Adapting solution methods is necessary, since extra constraints may turn an initially close to optimal solution into a bad or even infeasible one if these constraints are not accounted for when constructing the vehicle routes, as demonstrated in Chapter 2. Therefore, solution strategies for the VRP that are flexible with respect to the addition of new restrictions and still produce good solutions are highly valuable in practice.

In general, solution methods based on local search lack this flexibility. It is difficult to generalize one local search method to solve many different extensions of the VRP. Adapting local search methods when new restrictions are introduced to the problem is generally a difficult task. If such a restriction is added, then the neighborhood structure and the search strategy need to be carefully redesigned in order to obtain high quality solutions again. Therefore, adding a single restriction requires the development of a new algorithm, as the extensive lists in Parragh et al. (2008a,b) show.

We demonstrate that our solution approach forms a flexible framework for solving large vehicle routing and scheduling problems. This framework covers a wide range of VRP extensions. The constructive way of creating vehicle routes will prove to be very suitable for including difficult timing restrictions such as time-dependent travel times and driving hours regulations, as we shall demonstrate in the remaining chapters of this thesis.

The solution approach is a generalization of the restricted dynamic programming heuristic for the TSP of Malandraki and Dial (1996), which can be seen as a form of beam search (Bisiani, 1987). We apply restricted dynamic programming to the VRP through the giant-tour representation of vehicle routing solutions (Funke et al., 2005). The giant-tour representation allows us to handle single tour and multiple tour problems in a similar way.

We test our solution approach on benchmarks for different VRP extensions. For the Capacitated VRP (CVRP), the VRP with time windows (VRPTW), the pickup and delivery problem (PDP) with time windows (PDPTW), and the multi-depot VRP (MDVRP), our heuristic produces solutions with acceptable quality within practical computation times. Therefore, this new framework constitutes a promising approach for solving VRPs in practice.

This chapter is organized as follows. Section 3.2 first describes dynamic programming for the TSP, then describes the giant-tour representation of VRP solutions, and finally describes our solution approach for the VRP. Section 3.3 describes a way to reduce the state space of this dynamic programming formulation to obtain solutions within practical computation times. Section 3.4 demonstrates that our solution approach forms a flexible framework for solving VRPs in practice by showing how a wide range of VRP extensions can be solved with our solution approach. Section 3.5 presents the results of the computational experiments and Section 3.6 summarizes the main findings in this chapter.

3.2 Dynamic programming

Our solution approach for solving VRPs is based on the restricted dynamic programming (DP) heuristic for the TSP of Malandraki and Dial (1996), and applied to the VRP through the giant-tour representation (GTR) of vehicle routing solutions. Therefore, we first describe the DP for the TSP and the GTR of vehicle routing solutions. Then we describe the DP for the VRP.

3.2.1 Dynamic programming for the TSP

The restricted dynamic programming heuristic for the TSP is based on the exact dynamic programming algorithm for the TSP of Held and Karp (1962) and Bellman (1962). This exact dynamic programming algorithm for the TSP can be described as follows.

The TSP considers the problem of visiting a set $V = \{0, 1, \dots, n-1\}$ of n cities exactly once, starting and ending at city 0, and minimizing the total traveling costs. The costs of traveling between a pair of cities $i, j \in V$ are given by c_{ij} .

A state $(S, j), j \in S, S \subseteq V \setminus 0$ in the DP algorithm represents a path starting at city 0, visiting all cities in S exactly once, and ending in city j . The costs $C(S, j)$ of a state are given by the minimal costs over all such paths. In the first stage, the costs of the states are determined by $C(\{j\}, j) = c_{0j}, \forall j \in V \setminus 0$. Next, in each successive stage the costs of the states are calculated with the recurrence relation $C(S, j) = \min_{i \in S \setminus j} \{C(S \setminus j, i) + c_{ij}\}$. Finally, the costs of the optimal TSP tour are given by $\min_{j \in V \setminus 0} \{C(V \setminus 0, j) + c_{j0}\}$.

Since there are $2^{n-1} - 1$ nonempty subsets $S \subseteq V \setminus 0 = \{1, \dots, n-1\}$ subsets S and each subset S contains $|S| \leq n$ possible end nodes, the total number of states is bounded by $\mathcal{O}(n * 2^n)$. Next, each state is calculated by comparing at most $n - 1$ additions, resulting in an algorithm with a running time of $\mathcal{O}(n^2 * 2^n)$. The optimal TSP tour can be backtracked by saving for each state (S, j) the city $i \in S \setminus j$ that minimizes $C(S \setminus j, i) + c_{ij}$.

Since this approach constructs only one route, it cannot be applied directly to the VRP. We propose to apply it to the VRP through the GTR of vehicle routing solutions.

3.2.2 Giant-tour representation

Funke et al. (2005) introduce the GTR of vehicle routing solutions, because it allows to handle single and multiple route problems in a similar way. Besides, it is a ‘natural’ representation of vehicle routing solutions. We use the GTR for the development of our DP algorithm for solving VRPs. The GTR can be

described as follows.

The basis of any routing problem is a directed graph $G = (V, A)$, in which the node set V consists of request nodes $R \subset V$, origin nodes $O \subset V$, and destination nodes $D \subset V$, and the arc set A represents feasible travels between these nodes. For a VRP, the request nodes R correspond to all customer requests. Furthermore, for each vehicle there is one origin and one destination node, which all may represent the same location. Therefore, if m is the number of available vehicles, we get $|O| = |D| = m$. If we order the vehicle routes $k = 1, \dots, m$ in a routing solution, then the GTR of this solution is a cycle in the graph G in which each route end node d^k is connected to the route start node of the next vehicle route o^{k+1} . Finally, the cycle is closed by connecting d^m with o^1 . In Figure 3.1, we present an example of a vehicle routing solution with three vehicles, two depots (A and B) and nine customers. vehicle 1 starts at depot A and ends at depot B , vehicle 2 starts and ends at depot B , and vehicle 3 starts and ends at depot A . Figure 3.2 presents the same solution with its corresponding GTR.

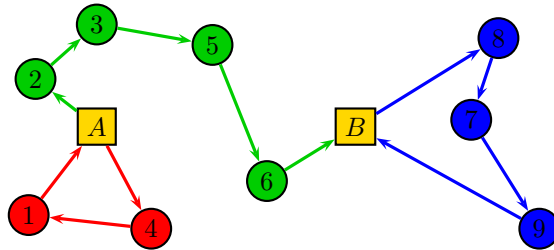


Figure 3.1: Example of a solution to a VRP with three vehicles

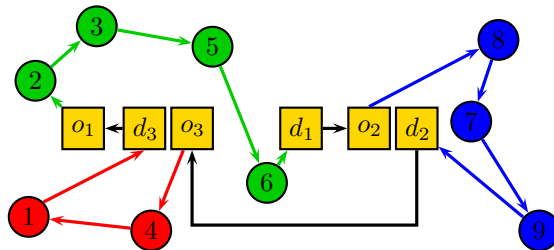


Figure 3.2: The giant-tour representation of a solution to the VRP of Fig. 3.1

3.2.3 Dynamic programming for the VRP

By using the GTR, we transform the VRP into a sequencing problem and we can use the DP formulation for the TSP to solve it. However, we need to ensure that the DP solution is the GTR of a feasible VRP solution. A general way to do this is by checking the feasibility of a partial solution while expanding a state. We call an expansion feasible if it represents the giant-tour of a feasible partial VRP solution. For example, each state (S, d^k) , where the partial solution ends with node d^k , can only be expanded with node o^{k+1} . Furthermore, o^{k+1} can only be an expansion of a state with end node d^k . Therefore, unlike the TSP, not every expansion is feasible for the VRP and we must perform a feasibility check when expanding a state. This seems a downside, but it actually gives us the power to use this algorithm for almost every extension of the VRP, providing a general framework for solving VRPs in practice.

To derive the running time complexity of the DP algorithm for the VRP, observe that we have to add $2m$ nodes to the $|R|$ nodes for the customers to apply the DP to the VRP. This would lead to a total of $2m + |R|$ nodes. However, each end node of a vehicle and start node of the following vehicle can be merged, leaving only m extra nodes in consideration (i_1, i_2, \dots, i_m) : $n = m + |R|$. Furthermore, these nodes have a fixed order in the GTR, which reduces the state space considerably. We first describe this reduction before we derive the running time complexity of the DP algorithm.

The ordering of the vehicles imposes a serial precedence relation of m nodes $i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_m$. For every state (S, j) for which holds that $i_a \notin S$, $i_b \in S$ and $i_a \rightarrow i_b$, there exists no feasible partial solution, because the precedence relation is not satisfied. Therefore, for every (S, j) that does have a feasible partial solution, there exists a k , $1 \leq k \leq m$, such that $i_a \in S$ if $a \leq k$ and $i_a \notin S$ if $a > k$. We first derive the fraction of the total number of subsets that have this property, since this fraction equals the fraction of states that contain feasible partial solutions.

Suppose $|S| = l$ and let $i_1, \dots, i_k \subseteq S$. Then it must hold that $k \leq l$ and $l - k \leq n - m$, with $l - k$ request nodes in S and $n - m$ request nodes in total. The total number of such subsets S equals $\binom{n-m}{l-k}$. If we sum this over all k and l , we get $\sum_{k=0}^m \sum_{l=k}^{n-m+k} \binom{n-m}{l-k} = (m+1)2^{n-m}$ subsets. If we divide this by the total number of subsets $S \subseteq V$, we get $\frac{(m+1)2^{n-m}}{2^n} = \frac{m+1}{2^m}$, which is the fraction of states that can contain feasible partial solutions given the precedence relations $i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_m$. This implies that each independent precedence relation $i \rightarrow j$ reduces the state space by $\frac{1}{4}$ (in this case $m = 2$ such that the fraction of states that contain feasible partial solutions is $\frac{3}{4}$).

Finally, observe that any given state can possibly end in $|R| + 1 = n$ nodes and be expanded to $|R| + 1$ nodes (because of the serial precedence relation).

Therefore, the running time of the DP algorithm for the VRP is

$$\mathcal{O}\left((|R|+1)(|R|+1)\frac{m+1}{2^m}2^{|R|+m}\right) = \mathcal{O}(n^2m2^{n-m}).$$

3.3 Restricting the state space

Although dynamic programming has the best running time complexity of all exact algorithms for the TSP proposed so far (Woeginger, 2003), it is still not fast enough to solve problems of realistic sizes in practical computation times. Therefore, Malandraki and Dial (1996) propose a restricted version of this algorithm, in which the number of states in each stage is bounded by a parameter H . This bounding procedure works as follows.

In each stage, we select the H states with the smallest costs to expand to the next stage. Since each state represents a partial tour and in each stage all partial tours visit the same number of customers, a state with low costs is more likely to yield the first part of a good TSP solution than a state with high costs. Therefore, continuing the algorithm with only the H states with smallest costs will - although it does not guarantee to find the optimal TSP tour - probably result in a good TSP solution. Next, each of these H states is expanded in all possible ways. If certain states are reached multiple times (e.g., expanding (S, j) with customer k results in the same state as expanding (S, i) with customer k : $(S \cup k, k)$), then, according to the original recurrence relation, only the one with lowest costs is maintained.

Instead of selecting the H states with the smallest costs (with respect to the objective function), we could use other criteria for selecting the H most promising states in each stage. For example, we may include a lower bound for the costs of completing the partial TSP tour represented by a state. In the remainder of this thesis, unless otherwise stated, we select the H most promising states in each stage according to the objective value of the partial solution a state represents.

Malandraki and Dial (1996) show that restricted dynamic programming is a flexible approach for solving TSPs by applying it to the TSP with time-dependent travel times. They also show that increasing the value of H results in better solutions, but also in substantial higher computation times. Note that setting $H = 1$ results in the nearest neighbor heuristic and setting $H = \infty$ results in the exact dynamic programming algorithm for the TSP.

We can restrict the state space even further - using the general idea of beam search (Bisiani, 1987) - by expanding each state (S, j) only to the E nearest unvisited nodes. We think this is reasonable, because edges in the optimal solution will most likely be between two nodes that are near neighbors of each other, as observed by Rego and Glover (2002).

The same principle can be applied to the DP algorithm for the VRP. However, we choose to expand each state (S, j) to at most E *feasible* states. From the feasible expansions, we select the ones that extend to the nearest nodes. We observe that our search through the state space will require a polynomial effort. For each fixed H and E we expand to $\mathcal{O}(EH)$ states in each stage. Since we search for E *feasible* state expansions, the number of state expansions (both feasible and infeasible ones) that are investigated in each stage is $\mathcal{O}(nH)$. Each such investigation can be done in $\mathcal{O}(1)$. However, since we need to select the H best states in each of n stages, we get a running time of $\mathcal{O}(n^2H \log(H))$.

3.4 The flexibility of our solution approach

To demonstrate the flexibility of the DP algorithm, we show how it can solve the CVRP and the VRPTW by adding more state dimensions, and the PDP by adding precedence relations. Furthermore, we show how other realistic constraints such as multiple depots, a heterogeneous vehicle fleet, the open vehicle routing problem, and multiple routes per vehicle can be incorporated within the framework. Note that any conceivable combination of these problems is equally suited. Our generic approach is unique in comparison with the large variety of approaches found in the literature (see Parragh et al., 2008a,b), all of which are diverse and specific for some variants of the VRP.

3.4.1 Adding more dimensions

For the CVRP, as well as the VRPTW, we add state dimensions on capacity or time. When expanding a state, we perform a feasibility check to ensure that there is enough slack in capacity or time. However, we have to be careful not to lose the optimality guarantee of the (unrestricted) DP algorithm for the VRP. We demonstrate this by the following example. Suppose two states (S, j) and (S, i) can be feasibly expanded with the same node k such that the first expansion results in a partial solution with lower cost than the second expansion, but with less slack in capacity or time. According to the original recurrence relation, the first expansion will be selected. However, it may prove impossible to complete the first expansion to a feasible complete solution, while the second expansion can be completed to a feasible solution. This is resolved by making two copies of this state such that one represents the partial solution with lower cost, while the other represents the partial solution with more slack. This is formalized in dominance rules as in Dumas et al. (1995), which can result in several copies of a single state. However, in practice costs and slack are correlated, such that for all states with the same visited node set S and end node $j \in S$ it is unlikely that no state is dominated by another one. We apply the bounding procedures E and H to all non-dominated states, meaning

that after each stage we have at most H states which differ in the subset S , or in the end node $j \in S$, or in one of the other state dimensions.

3.4.2 Adding precedence relations

To apply our model to the PDP, we add precedence constraints to the pickup and delivery pairs, thereby reducing the state space. Furthermore, we need to add a feasibility check when a vehicle returns to a depot to ensure that the vehicle only returns if it has visited all the deliveries corresponding to the pickups that it has visited. These precedence relations also have a big effect on the running time, because every independent precedence relation of a pickup and delivery reduces the state space by $\frac{1}{4}$ (see Section 3.2.3). Note that the state space reduces even further, since each pickup i introduces a precedence relation $j \rightarrow d^k$ between its corresponding pickup j and the destination node d^k of the vehicle route k that is currently being constructed.

3.4.3 Other realistic constraints

Several other realistic constraints fit within our algorithmic framework. State dimensions and input characteristics allow for various extensions of the VRP. Table 3.1 presents an overview of known VRP extensions and how they fit within our algorithmic framework.

VRP extension	Inclusion within DP framework
Open VRP ^a	we solve this by setting all distances to route end nodes to 0
Heterogeneous fleet	different capacities or times the vehicles are available can be controlled with feasibility checks
Multiple depots	different depot locations for different vehicles are controlled with the input
Multiple routes	can be modeled by separate vehicles for each route and a flexible time window for the second ^b vehicle depending on the return time of the first vehicle

^aIn the open VRP, vehicles do not have to return to the depot.

^bWe assume w.l.o.g. two routes per vehicle.

Table 3.1: Overview of VRP extensions that fit within the DP framework

3.5 Computational experiments

We test our solution approach on well-known benchmark instances. We apply the DP heuristic on benchmark instances for the CVRP, VRPTW, PDPTW,

and MDVRP. As the results shall indicate, the DP heuristic produces good solutions for these benchmark instances. There are tailored solution methods that produce better results for these problem instances, like the adaptive large neighborhood search of Pisinger and Ropke (2007). However, it is not our aim to compete with these tailored solution methods, but to apply our framework to more realistic extensions of the VRP, such as the VRP with time-dependent travel times and driving hours regulations, without losing solution quality. Local search based solution methods are not well suitable for such extensions. We implemented the DP heuristic in Delphi 7 and ran our experiments on a Pentium 4, 3.40GHz CPU and 1.00 GB of RAM.

3.5.1 CVRP

We first test the performance of the DP heuristic on a set of benchmark instances for the CVRP developed by Augerat (1995). The CVRP is the least restricted variant of the VRP and, therefore, has the largest solution space. Additional restrictions reduce this solution space. The objective within the CVRP is to minimize the total travel distance, while the number of vehicles that can be used is limited. Note that when we have a homogeneous fleet and impose no restrictions at all, the solution to a general VRP will be equal to a TSP, because returning to the depot and starting with a new vehicle makes no sense. Furthermore, even if it is required to use all vehicles, while the objective is still to minimize the total travel distance, it can be reduced to a TSP. This is a well-known reduction from the m-TSP to the TSP, which amounts to replicating the depot with as many copies as the number of salesman and forbidding direct transitions from copy to copy by setting the inter-depot costs to infinity. We demonstrate that our restricted dynamic programming heuristic produces good results for the CVRP benchmark instances.

Within the basic DP formulation, the cost of each state (S, j) is defined as the distance of a shortest path visiting each node in S , ending in node $j \in S$. For the CVRP, this cost definition may cause some problems when we restrict the state space. For example, in the second stage, the state that represents the path of visiting first the node nearest to the depot and then directly returning to the depot is likely to have a quite small cost compared to the other states in the same stage. However, since the number of vehicles within the CVRP is limited, returning low filled trucks to the depot may lead to infeasible solutions. We avoid this problem by only allowing a vehicle to return to the depot if the percentage of demand served so far is at least as much as the percentage of vehicles used so far. This extra restriction does not exclude the optimal solution, since for each solution there always exists an ordering of the vehicles, such that the average demand served by the first k vehicles is at least k times the average demand over all vehicles ($k = 1, \dots, m$).

If states A and B visit the same customer set S and end in the same

node $j \in S$, then state A only dominates state B if its cost is not larger than the cost of state B and its remaining capacity is not smaller than the remaining capacity in state B . This dominance criterion ensures that we do not exclude the optimal solution. If in a certain stage the number of states exceeds its maximum H , then only the H states with the lowest costs are maintained. Possible ties are broken by selecting the state with the highest remaining capacity. Finally, we first set E to its maximum value n . Recall that H and E apply to all non-dominated states, so there are at most H states after each stage, which differ in at least one of the state dimensions (in this case S , $j \in S$, and the remaining capacity).

Table 3.2 presents the average gaps of the solutions found by the DP heuristic with the optimal solutions (all problem instances and optimal solutions are available at Ralphs, 2003). The Augerat instances, which are standard benchmark instances for the CVRP and for which all instances the provably optimal solution has been found, consist of three problem sets: A, B, and P. Customer locations and demands are random in the A-instances, the customer locations in the B-instances are clustered, and the P-instances are modified instances from the literature. There are 27 A-instances, 23 B-instances, and 24 P-instances. The number of customers in the A-, B-, and P-instances varies from 32 to 80, 31 to 78, and 16 to 101, respectively. In the column ‘All’, we also report, between brackets, the maximum gaps over all problem instances. The last column presents the average computation times over all problem instances in seconds.

$H \setminus$ Problem Set	A	B	P	All	cpu(s)
10	23.96	20.80	18.78	21.40 (43.53)	0.063
100	14.71	16.73	10.14	13.95 (36.43)	0.171
1,000	10.11	10.56	6.12	9.01 (21.59)	1.29
10,000	6.79	6.99	4.54	6.16 (21.15)	12.9
100,000	4.59	5.68	3.41	4.58 (17.68)	151
1,000,000	3.11	3.77	2.55	3.15 (15.86)	1727

Table 3.2: Gaps (%) with optimal solutions

The results show that if we set $H = 1,000$, we get results within 10% of the optimum, on average. For this value of H , computation times are still very small (1 second on average). If we increase H to 100,000, then the average gaps drop below 5 percent with computation times around 2.5 minutes.

The maximum gaps are rather large even for large values of H . However, the number of instances with a gap larger than 10% reduces rapidly when H increases, as we can see in Table 3.3. If $H = 1,000,000$, then still 2 instances have a gap larger than 10%. However, although these instances are very challenging from a theoretical perspective, their structure may not be realistic for practice. For example, customers are randomly located in the 2 instances for which the optimality gaps remain larger than 10%. We propose the DP heuristic for

solving realistic extensions of the VRP, such as the VRP with time-dependent travel times and driving hours regulations. Therefore, it is promising that the DP heuristic finds good solutions, on average, even if it has some difficulties with solving some extreme problem instances. Table 3.3 also presents the number of problem instances for which the optimal solution is found. As can be seen, this number increases substantially with H .

H	Gap > 10%		Optimal	
	# Instances	Percentage	# Instances	Percentage
10	63	91.3%	0	0.00%
100	54	73.0%	2	2.70%
1,000	30	40.6%	4	5.41%
10,000	10	13.5%	7	9.46%
100,000	5	6.76%	8	10.81%
1,000,000	2	2.70%	9	12.16%

Table 3.3: Instances with gap larger than 10% and optimal solutions

Table 3.4 and 3.5 present the impact of different E values, limiting the number of expansions of each state, on the performance of the DP heuristic. We set the values of E to n , $0.5n$, $0.25n$, $0.125n$, and $0.05n$ (we round in case of fractional values) and we present average gaps and average computation times over all problem instances. Since $E = 0.05n$ results in an E value of 1 for problem instances with less than 30 customers, yielding the nearest neighbor heuristic, we set the minimum of E to 2. The results indicate that it is possible to decrease E such that computation times also decrease, while the solution quality is maintained. However, when E is getting smaller than $0.125n$, the solution quality starts to reduce substantially, especially for large H . For example, in case $E = 0.05n$ and $H = 1,000,000$ the average gap is larger than when $E = 0.25n$ and $H = 100,000$, while computation times are substantially smaller in the latter case (533 seconds versus 127 seconds).

$H \setminus E$	n	$0.5n$	$0.25n$	$0.125n$	$0.05n^a$
10	21.40	21.40	21.48	21.49	21.89
100	13.95	14.11	14.13	13.86	12.86
1,000	9.01	9.22	9.03	9.23	9.47
10,000	6.16	6.21	6.16	6.85	6.96
100,000	4.58	4.68	4.67	5.03	5.80
1,000,000	3.15	3.12	3.19	3.59	4.83

^afor one instance no feasible solution could be found with $E = 0.05n$ and all H values

Table 3.4: Impact of different E values: average gaps (%)

We conclude that the DP heuristic solves the CVRP benchmark instances to an acceptable quality within practical computation times. The number of

$H \setminus E$	n	$0.5n$	$0.25n$	$0.125n$	$0.05n$
10	0.063	0.063	0.079	0.060	0.066
100	0.171	0.187	0.178	0.133	0.107
1,000	1.29	1.19	1.02	0.874	0.574
10,000	12.9	12.5	11.1	9.08	5.26
100,000	151	145	127	103	56.2
1,000,000	1727	1636	1409	1076	533

Table 3.5: Impact of different E values: cpu (s)

problem instances with a gap larger than 10% reduces substantially when H increases. These results are particularly interesting, since restricted dynamic programming is a construction heuristic, such that no other methods are needed to get some initial solutions. Finally, increasing H will eventually result in an optimal algorithm for the CVRP, implying a guarantee for finding better solutions if H is raised enough.

3.5.2 VRPTW

We test the performance of the DP heuristic on the VRP with time windows by solving the well-known set of benchmark instances proposed by Solomon (1987). In order to comply with the time window constraints, we add a state dimension t indicating the departure time from the last visited customer. We add to the dominance criterion of the CVRP that the departure time of state A is not larger than the departure time of state B (the dominated state).

We set $E = n$ and $H = 1,000,000$. Within the VRPTW, the generally used objective is to minimize the number of vehicles as primary objective and to minimize the total travel distance is secondary objective. Therefore, we select the H best states using the following hierarchical criteria: 1) number of vehicles used, 2) total distance traveled.

Table 3.6 presents the results for the six Solomon problem sets. The first column indicates the different problem sets and, between brackets, the number of problem instances. The three columns on the DP heuristic present the average number of vehicles used, the average travel distance, and the cpu time in seconds, respectively. The last two columns present the objective values for the best known solutions identified by heuristics.

On average, the DP heuristic results in 21.7% more vehicle routes than the best known solutions for the VRPTW. The total travel distance is on average 8.83% larger. The results on the c-instances are better than the results on the r- and rc-instances. For these problem instances, the DP heuristic finds solutions with the same number of vehicle routes as the best known solutions, and the average difference in travel distance is only 1.8%. This is of particular

Problem Set	DP Heuristic			Best Known Solutions	
	# Veh.	Dist.	cpu (s)	# Veh.	Dist.
c1 (9)	10.00	843.35	4730	10.00	828.38
c2 (8)	3.00	600.29	5019	3.00	589.86
r1 (12)	13.92	1326.73	4829	11.92	1205.39
r2 (11)	4.00	1095.94	7174	2.73	951.91
rc1 (8)	14.13	1500.04	8417	11.50	1384.16
rc2 (8)	4.25	1249.46	8690	3.25	1119.35

Table 3.6: results for the VRPTW

interest for practice, since customer locations are often clustered.

The average computation time per problem instance is 6365 seconds, which is almost 4 times more than with the same parameter settings for the CVRP instances. This increase in computation time can be explained as follows. Within the VRPTW instances, the dominance criteria are stronger, hence less states are dominated than within the CVRP instances. Therefore, it happens more often that a new state has to be inserted in the list of H smallest cost states, requiring $\mathcal{O}(\log H)$ time. This is much more than when a state is dominated, which can be checked in constant time.

Increasing H results in substantial reductions of the gaps with the best known solutions. For example, increasing H from 100,000 to 1,000,000 reduces the gaps in number of vehicles and total travel distance from 23.84% and 11.18%, respectively, to 21.7% and 8.83%, respectively.

3.5.3 PDPTW

To apply the DP heuristic to the PDPTW, we ensure that delivery nodes can only be selected when the corresponding pickup node has already been selected. Next, we only allow a vehicle to return to the depot when it is empty, i.e., when all deliveries corresponding to its pickups have been carried out. Furthermore, we add a feasibility check to ensure that each separate delivery that the vehicle still contains can be executed in the next stage. This obviously does not guarantee that all deliveries can still be executed within the given time windows (to guarantee this we need to solve a TSP with time windows on the remaining deliveries), but it is a quick check for removing infeasible expansions. We test the DP on the set of modified Solomon instances proposed by Li and Lim (2001). The primary objective is to minimize the number of vehicles used and the secondary objective is to minimize the total travel distance. Table 3.7 presents our results and the best known solutions for these problem instances.

On average, the DP heuristic requires 0.5 more vehicles than the best known solutions, and the travel distance is on average 6.98% larger. The precedence

Problem Set	DP Heuristic			Best Known Solutions	
	# Veh.	Dist.	cpu (s)	# Veh.	Dist.
lc1	10.00	849.79	8340	9.67	874.12
lc2	3.00	612.97	11551	3.00	589.16
lr1	12.42	1271.28	7908	11.92	1219.62
lr2 ^a	3.80	1183.90	14949	2.80	994.44
lrc1	12.00	1436.09	7279	11.50	1386.74
lrc2	3.75	1240.23	12899	3.25	1133.12

^aOne problem instance did not return a feasible solution. We resolved this by setting $E = 5$ for this problem instance.

Table 3.7: results for the PDPTW

relations in the PDPTW substantially reduce the state space, which results in smaller gaps with the best known solutions than for the VRPTW, especially considering the primary objective (the gap reduces from 1.2 to 0.5 vehicles).

3.5.4 MDVRP

In the MDVRP, vehicles are located at different depots. When we apply the unrestricted version of the DP algorithm, the ordering of the vehicles is not of importance. However, when we do restrict the state space different orderings of the vehicles may have substantial impacts on the resulting vehicle routes. As an extreme case, it may be possible that certain vehicles are not required at all to serve customers. With the DP heuristic, it is likely that such vehicles are the last vehicles to be selected. So when fixing the order of the vehicles we could unintentionally determine which vehicles will be used to serve customers and which not. To avoid this, we only fix the order of the vehicles located at the same depot, since these vehicle all have the same characteristics. We do not impose precedence relations between vehicles located at different depots. This implies that the DP heuristic is initialized with a number of states equal to the number of depots, in which the last visited node is the origin node of the first vehicle at the corresponding depot. Next, when a vehicle returns to its depot, i.e., its destination node is added to the state, then, in the next stage, expansions are possible to the origin node of the first unused vehicle (if any) at each depot.

We test the DP heuristic on the set of MDVRP instances proposed by Cordeau et al. (1997). Within these benchmark instances, the number of vehicles per depot is fixed and the objective is to minimize the total travel distance. To ensure that we do not return low-filled trucks to their depots, we add a large vehicle cost M to a state when a vehicle returns, which we multiply with its relative remaining capacity (if the vehicle's capacity is Q and the total load is γ , then the relative remaining capacity is $1 - \frac{\gamma}{Q}$). Some problem instances also

have a limitation L on the duration of each route. For these instances, we increase the cost for returning a vehicle with the vehicle cost M , multiplied with the relative remaining travel time of that vehicle (if the total route duration is λ , then the relative remaining travel time is $1 - \frac{\lambda}{L}$). We set the maximum vehicle cost M to the maximum distance of a customer to its nearest depot. The total vehicle costs (i.e., the sum of the weighted vehicle costs over all returned vehicles) are reduced each stage, such that vehicle usage is the leading criterion in the first stages, but travel distance becomes leading in the final stages. We do this by multiplying the total vehicle costs with the fraction of stages left (if n is the total number of stages and ν the index of the current stage then this fraction equals $\frac{n-\nu}{n}$).

Preliminary tests indicated that some problem instances were hard to solve: some customers located far away from any depot could not be visited. Therefore, we included a bonus for visiting ‘far away’ customers in early stages, which we subtract from the costs of a state. We set this bonus to the distance from the customer to its nearest depot. To reduce the impact of this bonus at later stages, we multiply it with the fraction of stages left.

Since there are less precedence relations than within the single depot problem instances (we do not fix the orderings of vehicles located at different depots), and some problem instances are much larger than within the single depot benchmark instances (the maximum number of customers is 360 versus 101 for the single depot benchmark instances), computation times are substantially larger for the MDVRP than for the single depot benchmark instances. Therefore, we set $H = 100,000$ and we run the problem instances for $E = 5$, $E = 10$, $E = 20$, and $E = n$. Table 3.8 presents the average gaps of the DP solutions with the best known solutions. The computation times are on average 3 hours per problem instance.

E	Gap
5	17.43%
10	18.23%
20	18.38%
n	18.58%
best	15.82%

Table 3.8: results for the MDVRP

Surprisingly, smaller values for E give better results, on average, for these problem instances. These results can be explained as follows. Smaller E -values allow a more diverse search through the state space. This may be particularly beneficial for multi-depot problems. For example, if one depot has some very near customers, starting a vehicle at this depot and going first to these customers results in very promising states in terms of state costs. When setting E high, all expansions of such states will be investigated, while states representing vehicle routes starting at other depots are not considered, because of

the bound H on the stage width. However, it is hard to compare states that represent vehicle routes starting at different depots and, therefore, restricting the number of expansions of a single state may prevent losing promising other states that represent vehicle routes that start at other depots.

Increasing H substantially reduces the gaps with the best known solutions (increasing H from 10,000 to 100,000 reduces the gap from 18.80% to 15.82%). The larger gaps in comparison with the benchmarks of other VRP extensions are caused by the unfixed order of vehicles located at different depots. This causes the state space to increase, because there are less precedence relations imposed. However, it is promising that the DP heuristic finds feasible solutions - even with the limited number of available vehicles - for all problem instances.

3.5.5 Discussion of the results

We showed that the DP heuristic can solve various variants of the VRP and finds solutions of acceptable quality. In general, the solution quality produced by our solution approach improves when more restrictions are introduced. Although the DP heuristic obtains good results for all benchmark instances, it does not produce the formidable results as the adaptive large neighborhood search framework of Ropke and Pisinger (2006) (for their results on the CVRP, VRPTW, and MDVRP, see Pisinger and Ropke, 2007). However, the main strength of our solution approach is that it is a constructive heuristic that can incorporate many more constraints, such as time-dependent travel times and driving hours regulations (see Chapter 4 and 5), which are hard to incorporate in a local search heuristic.

3.6 Conclusions

We proposed a flexible solution approach for solving realistic vehicle routing problems. The solution approach is based on dynamic programming and allows for a trade-off between solution quality and computation time by restricting the state space. We demonstrated the flexibility of the solution approach by describing how to apply it to various practical extensions of the VRP, providing a new algorithmic framework for solving realistic VRPs. Computational experiments on benchmark instances for the CVRP, the VRPTW, the PDPTW, and the MDVRP demonstrated that the solutions obtained by the DP heuristic are of acceptable quality.

An important property of this solution approach is that it is a constructive heuristic that suits much better for including timing restrictions such as time-dependent travel times and driving hours regulations than improvement heuristics. Therefore, the algorithmic framework presented in this chapter forms a basis for the solution algorithms in the remaining chapters in this thesis.

Chapter 4

Vehicle routing with driving hours regulations

4.1 Introduction

This chapter¹ addresses the VRPTW while accounting for driving hours regulations imposed by national or European policy. In the European Union, driving hours are restricted by Regulation (EC) No 561/2006. In addition, Directive 2002/15/EC, which restricts drivers' working hours, has been implemented into national laws in most member countries of the European Union. These legal acts together comprise the EC social legislation on driving and working hours. The EC social legislation has to be taken into account by schedulers when establishing vehicle tours. As neglecting it can be fined severely, these acts have an enormous impact on the design of vehicle tours in practice.

As pointed out in Chapter 1, only a few papers in the literature have considered vehicle routing and break scheduling problems. However, none of them considers Directive 2002/15/EC and none of them considers the *entire* set of rules laid down in Regulation (EC) No 561/2006: only a set of basic rules in this regulation has been considered, while a set of modifications of the basic rules that increase flexibility has been generally neglected. The extension to the complete legal act implies some additional restrictions laid down by Directive 2002/15/EC. However, exploiting the entire set of rules of Regulation (EC) No 561/2006 may also allow for considerable improvements of the resulting vehicle schedules, since the modifications increase the flexibility of the planning.

In this chapter, we propose a solution method for the VRPTW with EC social legislation (VRPTW-EC) in which all legal rules for a weekly planning pe-

¹This chapter is based on Kok et al. (2009b)

riod are considered. The solution method is based on the restricted DP heuristic proposed in Chapter 3. The results generated with our solution method comply with the rules of the EC social legislation. We compare our method with the solution method proposed by Goel (2009), which is the state of the art for this problem. We conduct computational experiments on the modified Solomon benchmark instances that Goel proposes for the VRPTW-EC. The results show that our solution approach, which is a route construction heuristic, substantially improves the results of Goel (2009) with substantially less computational effort, whereas Goel's results are obtained with a large neighborhood search heuristic and an auction method to create some initial solution for the large neighborhood search.

This chapter is organized as follows. Section 4.2 presents all restrictions of the EC social legislation that have an impact on vehicle routing and break scheduling. Section 4.3 describes our restricted DP heuristic for the VRPTW-EC including all legal rules for a weekly planning period. We test the performance of our solution method for the VRPTW-EC on the modified Solomon instances proposed by Goel (2009) in Section 4.4. Finally, Section 4.5 summarizes the main findings in this chapter.

4.2 EC social legislation

The EC social legislation on drivers' driving and working hours mainly comprises two legislative acts, which we describe in Section 4.2.1 and 4.2.2. Regulation (EC) No 561/2006 restricts driving hours of persons engaged in road transportation, and Directive 2002/15/EC gives restrictions on drivers' working hours.

4.2.1 Regulation (EC) No 561/2006 on Driving Hours

Regulation (EC) No 561/2006 lays down rules for maximum driving hours and for the required breaks and rest periods. It postulates that transport undertakings have to organize the work of their drivers in such a way that the drivers are able to adhere to the restrictions set by this regulation. For infringements of the regulation committed by the driver his employer is held responsible, too. Furthermore, the regulation demands that every party involved in the transportation process, i.e. the transport undertakings, consignors, forwarders, tour operators, principal contractors, subcontractors, and even driver employment agencies ensure that the schedules of the drivers comply with the legal requirements. Therefore, the regulation's impact on vehicle routing and scheduling in real life applications is enormous.

Regulation (EC) No 561/2006 covers different but interconnected time horizons: single driving periods, daily driving times, and weekly driving times.

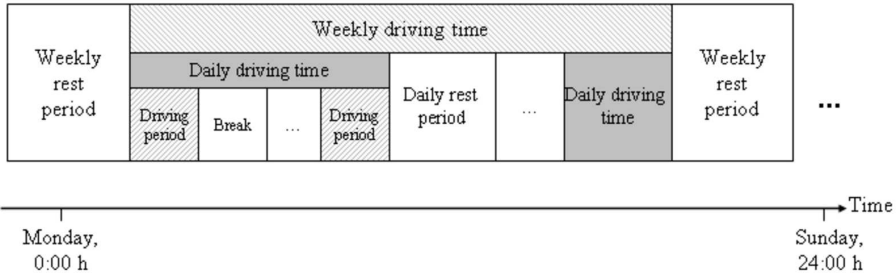


Figure 4.1: Relation of the different time horizons (Kopfer et al., 2007)

These time horizons are ended by breaks, daily rest periods, and weekly rest periods. Figure 4.1 depicts their relationship. We describe the rules on the durations of these time horizons and breaks in detail below. Some modifications of these rules have been introduced to allow for more flexibility for the drivers. We will indicate these cases by the distinction between basic and modified rules.

Driving periods: A driving period may contain at most 4.5 hours of accumulated driving time.

Breaks to end driving periods: A break of at least 45 minutes ends a driving period (basic rule). The duration may be reduced to 30 minutes if an additional break of 15 minutes has been taken anywhere during the same driving period (modified rule). Since the total break time of 45 minutes is now divided into two parts, we refer to this modified rule as splitting breaks. This may be beneficial, for example, if waiting time at a customer site allows a 15 minute break, but not a 45 minute break. If in such a case a 15 minute break is scheduled during the waiting time, then only a 30 minute break is required when the 4.5 hour driving limit is reached.

Daily driving times: The total daily driving time may not exceed 9 hours (basic rule). Twice a week, i.e. twice between Monday 0:00 am and Sunday 23:59 pm, the daily driving time can be extended to 10 hours (modified rule). We refer to this modified rule as extending driving times. A daily driving time ends when a daily or weekly rest period starts.

Daily rest periods: The duration of a daily rest period is at least 11 hours (basic rule). Any daily rest period may be reduced to 9 hours if an additional rest of 3 hours has been taken anywhere after the end of the previous daily rest period (modified rule). We refer to this modified rule as splitting rests. Moreover, drivers are allowed to reduce their daily rest periods to 9 hours without an additional rest of 3 hours up to three times between two weekly rest periods (modified rule). We refer to this

modified rule as reducing rest periods. Within 24 hours after the end of a daily rest period, a new daily rest period must have been taken, allowing a nonrest period (a period between two daily rest periods) to last for at most 13 hours (15 hours in case the nonrest period is ended with a reduced rest period).

Weekly driving times: The total driving time during a week, i.e. from Monday 0:00 am until Sunday 23:59 pm, may not exceed 56 hours. The accumulated driving time in any two consecutive weeks must not exceed 90 hours.

Weekly rest periods: The duration of a weekly rest period is at least 45 hours (basic rule). Drivers are allowed to reduce one weekly rest period to 24 hours in any two consecutive weeks (modified rule). This reduction has to be compensated by an equal extension of another rest period before the end of the third week following the week considered. Within 144 hours (6 days) after the end of a weekly rest period, drivers have to start a new weekly rest period.

4.2.2 Directive 2002/15/EC on Working Hours

Directive 2002/15/EC supplements the restrictions on driving times laid down by Regulation (EC) No 561/2006. As driving times are part of the total working time, these legal acts are interdependent and therefore both have to be considered in vehicle routing and scheduling. Besides driving times, also times for loading and unloading, time to assist passengers while boarding and disembarking from the vehicle, cleaning and maintenance times, and other times in which a driver cannot freely dispose of his time, such as unforeseen waiting times, are included in the working time. Since in the remainder we will address a deterministic vehicle routing problem, only driving and service times are taken into account as working times. Waiting times need not be considered as working times, since in deterministic problems all waiting times are known in advance.

The directive comprises the following restrictions on working periods and weekly working times:

Working period: A working period may contain at most 6 hours of accumulated driving time.

Breaks to end working periods: A break of at least 30 minutes ends a working period (basic rule). If the total working time between two daily rest periods exceeds 9 hours, the total break time in this period has to be extended to at least 45 minutes (basic rule). The total break time can be divided into parts of at least 15 minutes each (modified rule).

Weekly working time: The total working time during a week may not exceed 60 hours. The average weekly working time must not exceed 48 hours over a period of four months.

In order to observe the law, both legal acts have to be respected by drivers. Therefore, each basic rule or its modification must be respected and both are considered of equal importance in practice. In the literature, however, the modified rules have been neglected so far, and Directive 2002/15/EC on working hours has been completely neglected.

4.3 Solution method

We propose a solution method for the VRPTW including the EC social legislation for the planning horizon of one weekly driving period. However, this solution method can be extended to longer planning horizons and a rolling horizon framework (see Section 4.3.3). We use the restricted DP framework of Chapter 3 to design a solution method for the VRPTW-EC.

To determine whether partial solutions represented by a state are feasible, we have to determine whether there exists break schedules that comply with the EC social legislation. Whether efficient algorithms exist for this problem is still an open question. Goel and Kok (2009b) present an $O(n^2)$ algorithm for scheduling rest periods according to the U.S. hours of service regulations for a truck driver visiting n locations. However, these regulations do not require the scheduling of short breaks and have no set of modified rules. Goel and Kok (2009a) present an $O(n^2)$ algorithm for scheduling breaks and rest periods for team drivers according to the EC social legislation. However, in case of team drivers also no short breaks have to be scheduled.

We propose a break scheduling algorithm that decides locally when breaks are scheduled. This means that once we schedule a break, it is fixed and we do not change its duration or start time. When a new customer is added to the end of a partial vehicle route, we may only schedule new breaks along the route to this customer and at this customer. There are two main reasons to use a local view for scheduling breaks and rest periods.

First, it allows us to schedule breaks in constant time. Therefore, the running time complexity of the DP heuristic does not increase. This even holds when the modifications of the basic rules, which are generally ignored in the literature, are incorporated.

Second, a local scheduling algorithm is much easier to implement in practice. The rules we propose for scheduling breaks are intuitive and, therefore, they are both easy to implement, as well as easily acceptable by planners and operations managers. If a global scheduling algorithm is used, then breaks and rests may

be scheduled and extended prematurely, such that the benefits are less clear. For example, it may turn out that we arrive early at a customer j and that the 13 hour non-rest period reaches its maximum before his time window opens, requiring a rest period before serving customer j . If, due to this rest period his time window cannot be met, customer j cannot be added to the vehicle route. However, if there is waiting time at some predecessor i , then it might be possible to schedule an early rest partially during this waiting time without violating any of the time windows between customers i and j . This may decrease the start time of the active non-rest period when arriving at customer j , allowing to serve customer j before having to schedule a new rest period. This global view in which also breaks and rests at predecessors are considered requires at least linear time (instead of constant time in case of a local view), which increases the running time complexity of the restricted DP heuristic. Note that this also implies that the local view for scheduling breaks does not guarantee finding all feasible schedules.

We propose two break scheduling methods: a basic method and an extended method. The basic method is an extension of the naive label setting method proposed by Goel (2009), in which breaks and rests are scheduled as late as possible and as short as possible. We improve this method by increasing local flexibility for customer additions as follows. We minimize the time to start service at the added customer. Next, for this minimal start time, we minimize the accumulated time since the last rest, and the accumulated driving and working time since the last break, by trying to schedule rests or breaks in waiting time caused by hard time windows. We refer to the DP heuristic with the basic break scheduling method as DP^{basic} .

The extended break scheduling method extends the basic method by exploiting the modifications of the basic rules. We apply the same methodology of optimizing local flexibility at the added customer. We refer to the DP heuristic with the extended break scheduling method as DP^{ext} .

4.3.1 Basic break scheduling method

For the basic break scheduling method, we make the simplification that after no more than 6 hours of working time, we schedule a break of 45 minutes (instead of 30 minutes). This ensures that the second requirement of Directive 2002/15/EC on the break length between *working periods*, which states that the total break time on a day should be at least 45 minutes if that day contains more than 9 hours of working time, is also satisfied. Moreover, it fulfills the requirements of Regulation (EC) No 561/2006 on the break length between two *driving periods*, such that also a new driving period is initiated.

To include the legislation on driving and working hours into our DP heuristic, we have to ensure that the partial route represented by each state is feasible with respect to these restrictions. For this purpose, we introduce six state di-

mensions: non-break driving time, non-break working time, daily driving time, non-rest time, weekly driving time, and weekly working time.

t_{nbw} : accumulated non-break working time. Denotes the total amount of working time since the last break of at least 45 minutes.

t_{nbd} : accumulated non-break driving time. Denotes the total amount of driving time since the last break of at least 45 minutes.

t_{nr} : accumulated non-rest time. Denotes the total amount of time passed by since the last rest period of at least 11 hours.

t_{dd} : accumulated daily driving time. Denotes the total amount of driving time since the last rest period of at least 11 hours.

t_{ww} : accumulated weekly working time. Denotes the total amount of working time since the last rest period of at least 45 hours.

t_{wd} : accumulated weekly driving time. Denotes the total amount of driving time since the last rest period of at least 45 hours.

We first consider one-week planning from Monday 0:00 am until Sunday 23:59 pm. Furthermore, we first assume that the planning starts right after a weekly rest period has been taken by all drivers. This results in all state dimensions t_{nbw} , t_{nbd} , t_{nr} , t_{dd} , t_{ww} , and t_{wd} being zero for all vehicles at the start of the planning period. Section 4.3.3 discusses extensions to longer planning horizons and state dimensions not being zero at the start of the planning period, allowing for a rolling horizon framework.

When we start a new vehicle, we check for the first customer to be visited whether it can be reached from the depot. This might not be the case if a vehicle starts from the depot at time zero and requires a break or rest period before starting service, since this might violate the time window. If the customer cannot be served by a vehicle leaving the depot at time zero, we delay the departure time of the vehicle such that the vehicle arrives at the customer node exactly at the start of the time window.

Within our basic approach, we do not consider the modifications of the basic rules. Whenever we want to expand a state (S, i) with a customer j , then we first determine the arrival time a_j at this customer, considering possible breaks and rest periods that have to be scheduled along the travel from i to j . The calculation of a_j is an iterative process, in which we determine in each iteration the maximum driving time until a break or rest must be scheduled, or customer j is reached. The iterative process stops when the latter happens. For this purpose, we first set a_j to the departure time (service completion time) from customer i , and we introduce a variable δ_{ij} , denoting the remaining driving time to customer j . This variable is initially set to the total driving

time c_{ij} from customer i to customer j . We define $\Delta = \min(\delta_{ij}, 6 - t_{nbw}, 4.5 - t_{nbd}, 13 - t_{nr}, 9 - t_{dd})$, which represents the minimal driving time until a break or rest period must be scheduled, or the next customer is reached. Next, we recursively check whether the following holds:

$$\delta_{ij} = \Delta \quad (4.1)$$

If (4.1) does not hold, two cases may appear: 1) we are forced to schedule a break along the route, 2) we are forced to schedule a rest period along the route. Case 1 appears if $\Delta = 6 - t_{nbw}$ or $\Delta = 4.5 - t_{nbd}$, and $9 - t_{dd} > \Delta$ and $13 - (t_{nr} + 0.75) > \Delta$ (note that scheduling a break increases t_{nr} with 0.75, which we have to account for to avoid scheduling a break and a rest period directly after each other). After scheduling the break, we set t_{nbw} and t_{nbd} to zero. Case 2 appears otherwise and after scheduling the rest period we set all values t_{nbw} , t_{nbd} , t_{nr} , and t_{dd} to zero.

After scheduling a rest or break, we update the remaining driving time δ_{ij} , and in case of a break also the values of t_{nr} and t_{dd} , as follows:

$$\begin{aligned} \delta_{ij} &:= \delta_{ij} - \Delta \\ t_{dd} &:= t_{dd} + \Delta \\ t_{nr} &:= t_{nr} + \Delta + 0.75 \end{aligned}$$

Furthermore, in each iteration of (4.1), we update the value of a_j with the driving, break, and rest times that are scheduled. After deriving a_j , we check whether the accumulated non-break working time and the accumulated non-rest time allow to serve the customer without scheduling another break or rest period before service. To check this, we need the service time s_j of customer j and the time window $\{e_j, l_j\}$ in which service must start at customer j . If $t_{nbw} + s_j > 6$, then we schedule a break and update a_j , t_{nr} , t_{nbw} , and t_{nbd} . Next, if $t_{nr} + \max\{0, e_j - a_j\} + s_j > 13$, then we schedule a rest period. However, if both inequalities hold, then we extend the 45 minute break forced by the non-break working time to an 11 hour rest period to avoid scheduling a 45 minute break and an 11 hour rest period directly after each other. Finally, if $a_j \leq l_j$, then we can arrive in time to add customer j to the partial route.

To decide whether the addition of customer j is feasible with respect to all rules of the social legislation, we also need to check whether the vehicle can still return to the depot without violating the restrictions on the weekly driving and working times. In order to avoid including infeasible states, we forbid the expansion if after visiting the customer a return to the depot is infeasible. Consequently, we only allow an expansion if (4.2) and (4.3) are satisfied.

$$c_{ij} + c_{j0} \leq 56 - t_{wd} \quad (4.2)$$

$$c_{ij} + s_j + c_{j0} \leq 60 - t_{ww} \quad (4.3)$$

To improve the basic break scheduling method by increasing the local flexibility at customer j , we introduce a number of scheduling features that reduce the values of t_{nbw} , t_{nbd} , t_{nr} , and t_{dd} without delaying the time to start service at customer j . We give the highest priority to reducing the accumulated non-rest time, since in VRPTWs with large waiting times, this variable reaches its maximum more often than the accumulated daily driving time. Moreover, it requires the scheduling of rest periods, which take much longer than the breaks required by the accumulated non-break working and driving times. We propose the following three hierarchical procedures to increase local flexibility.

Schedule a rest period: If after arriving at customer j there is at least 11 hours of waiting time until the time window opens, then we schedule a rest period in this waiting time. We reset all values t_{nr} , t_{dd} , t_{nbw} , and t_{nbd} to zero, without delaying the time to start service at customer j . If there is more than 11 hours of waiting time, then we extend the rest period until the time window opens, such that t_{nr} is not increased before service starts at customer j .

Extend a rest period: If there is a rest period scheduled along the route to customer j , then we extend this rest by the waiting time at customer j (if any). This reduces the value of t_{nr} at the start of service at customer j without affecting the other state dimensions. This procedure may even reduce the time to start service. This happens if the waiting time that results from not extending the rest period would increase the value of t_{nr} so much that another rest period must be scheduled before service. Since this rest period does not fit in the waiting time at customer j , it delays the time to start service beyond e_j , and may even make the expansion infeasible. We postpone the departure time from the depot for traveling to the first customer in a similar way as we extend a rest along a route between two customers.

Schedule a break: If there is at least 45 minutes of waiting time at customer j , then we schedule a 45 minute break and reset t_{nbw} and t_{nbd} to zero. This increases the flexibility for adding customers afterwards.

4.3.2 Extended break scheduling method

To make the above presented solution method more suitable for realistic planning purposes and to allow for an enlargement of the solution space, we incorporate all modifications of the basic rules of Regulation (EC) No 561/2006 and Directive 2002/15/EC. In line with the DP approach, we only exploit these modifications if they allow for a local improvement of the current partial solution. The general approach for exploiting the modifications is as follows.

When adding a customer, we try to improve local flexibility by: 1) minimizing the time to start service of the customer to be added, 2) minimizing the

values of t_{nbw} , t_{nbd} , t_{nr} , and t_{dd} . Each time a break or rest must be scheduled and each time the arrival time at the customer is derived, we check whether one of the modifications of the basic rules may help to increase local flexibility. Since certain modifications can only be applied a limited number of times, we only apply them if they increase local flexibility. Local flexibility is maximal when service starts at the earliest feasible time to start service and all values t_{nbw} , t_{nbd} , t_{nr} , and t_{dd} are minimal.

Extending Driving Times

Regulation (EC) No 561/2006 allows drivers to extend their daily driving time up to 10 hours twice a week, while the basic rule restricts the daily driving time to no more than 9 hours. When 9 hours of accumulated driving time is reached, an extension to 10 hours may increase local flexibility if the remaining driving time to the next customer is not more than 1 hour. However, the extension may also require an additional 45 minute break, thereby postponing the arrival time. Therefore, we only schedule a driving time extension if it reduces the time to start service at the customer to be added, or if the driving time extension increases the waiting time at this customer, such that a rest period can be scheduled without postponing the time to start service. When the 9 hour driving time limit is reached, we calculate both the arrival time in case of extending the driving time and in case of not extending the driving time to derive whether a driving time extension is profitable.

To account for the fact that only 2 driving time extensions may be scheduled, we add the following state dimension:

n_{dte} : number of driving time extensions taken by the active driver.

The variable n_{dte} is not allowed to exceed 2, and it is reset to 0 each time a new vehicle is used.

Reducing Rest Periods

A rest reduction allows to *decrease* the start time of the *next* non-rest period by 2 hours. Within 24 hours after a rest period the next rest period must have been taken. Therefore, reducing the latter rest period by 2 hours also allows to *increase* the end time of the *current* non-rest period by 2 hours. Note that this is only possible if all other restrictions, such as the daily driving time limit, also allow this extension. Combining a decrease of the start time of the next non-rest period and an increase of the end time of the current non-rest period is also possible, but only if the sum is at most 2 hours. We follow the same methodology as with the driving time extensions to derive whether exploiting a rest reduction increases local flexibility. A special case is when

extending a driving time limit is equally profitable as reducing a rest period. In that case, we choose to extend the driving time, since we can only apply each modification a limited number of times and rest reductions are in general more valuable than driving time extensions. That is, driving time extensions allow additional driving times, while a rest reduction may allow additional driving **and** working times.

After determining the arrival time at a customer, we also check whether it is beneficial to reduce the next rest period. The time to start service may be substantially reduced by extending the current non-rest period to 15 hours (by allowing to serve the customer before having to schedule the next rest period), thereby even making the expansion feasible. Therefore, we reduce the next rest period if this reduces the time to start service.

To account for the fact that only 3 reduced rest periods may be scheduled, we add the following state dimension:

n_{rr} : number of rest reductions taken by the active driver.

This variable is not allowed to exceed 3, and it is reset to 0 each time a new vehicle is used.

Splitting Breaks

Both Regulation (EC) No 561/2006 and Directive 2002/15/EC allow drivers to split breaks. We exploit this rule whenever there is sufficient waiting time at a customer (but not sufficient for a full break), thereby allowing shorter breaks whenever the non-break driving or working time limit is reached. Whenever the non-break *driving time* limit is reached, either a 45 minute break is scheduled, or a 30-minute break in case a break of at least 15 minutes has already been taken since the start of the *driving period*. Whenever the non-break *working time* limit is reached, either a 30 minute break is scheduled, or a 15-minute break in case a break of at least 15 minutes has already been taken since the start of the *working period*.

To account for the rule in Directive 2002/15/EC, stating that at least 45 minutes of break time must be taken if the daily working time exceeds 9 hours, we introduce the state dimension t_{dw} , which denotes the daily working time:

t_{dw} : daily working time of the active driver.

Whenever this state dimension reaches its maximum of 9 hours, and the total break time since the last rest period does not add up to at least 45 minutes yet, then we schedule another break of at least 15 minutes.

Modification	Implementation
Extend driving time	If it reduces the time to start service; if it increases the waiting time, allowing for a rest period before service
Reduce rest period	If it reduces the time to start service; if it increases the waiting time, allowing for a rest period before service
Split breaks	If there is enough waiting time for a 15 minute break
Split rest periods	If there is enough waiting time for a 3 hour rest; if a 3 hour rest makes service without taking a full rest before possible and there is no rest reduction left

Table 4.1: Implementation of the modifications of the basic rules into the break scheduling method

Splitting Rest Periods

Splitting rests may increase local flexibility, since it allows an extension of the non-rest period to 15 hours. Therefore, we schedule a split rest whenever there is sufficient waiting time (but not sufficient for a (reduced) rest period). We also schedule a split rest if there is not sufficient waiting time, but the extension of the non-rest period to 15 hours reduces the time to start service. This case only applies if there are no reduced rest periods available anymore.

Table 4.1 summarizes all implementations of the modifications of the basic rules into the break scheduling method.

4.3.3 Extensions to Other Time Horizons

So far, we considered a weekly planning horizon and a situation in which all drivers have just completed a weekly rest period. However, our algorithm can be extended to longer planning horizons and to a rolling horizon framework.

In case a longer planning horizon is considered, a state dimension (t_{nwr}) must be added to account for the maximum period of 144 hours between two weekly rest periods. When one of the state dimensions t_{ww} , t_{wd} , or t_{nwr} reaches its maximum value, a weekly rest period must be scheduled. Afterwards, all state dimensions are reset to 0. To use the modified rules on the weekly rest period, the weekly driving time, and the weekly working time, we can follow a similar methodology as described with the extended break scheduling method.

In case drivers have *not* just completed a weekly rest period at the start of

the planning period - which is typically the case in a rolling horizon framework - then we set the initial state dimensions for each vehicle to the initial conditions of the corresponding drivers. Therefore, the algorithm can also be used in a rolling horizon framework.

4.4 Computational experiments

We test the two DP heuristics DP^{basic} and DP^{ext} on the modified Solomon instances proposed by Goel (2009) for the VRPTW-EC. We ran our experiments on a Pentium M, 2.00 GHz CPU and 1.0 GB of RAM. We first report on the results of DP^{basic} , in which the modifications of the basic rules are not included. We compare our results with the best results found by Goel (2009). Since the method proposed by Goel does not consider Directive 2002/15/EC on drivers' working hours, we relax our break scheduling method by setting the maximum non-break working period to 13 hours, i.e., the maximum period between two rests with DP^{basic} . Next, we present computational results on the impact of Directive 2002/15/EC. Finally, we present computational results on the impact of the modifications of the basic rules by applying DP^{ext} . Goel (2009) proposes the following modification of the Solomon instances for the VRPTW with the EC social legislation.

Goel considers the depot's opening hours as a period of 144 hours, corresponding to a weekly working period, and to scale the customer time windows accordingly. Next, Goel suggests a driving speed of 5 distance units per hour, and to set all service times to 1 hour. Due to the required breaks and rest periods, it may be impossible to reach certain customers before their closing time, or the vehicle may not be able to return in time to the depot after serving a customer at its opening time. Therefore, Goel suggests to adjust such time windows in such a way that the opening time equals the earliest time the vehicle can reach the customer, and the closing time is such that starting service at this closing time and directly returning to the depot results in a return time at the depot's closing time, respectively.

As described in Chapter 3, the value of H , which restricts the stage width after each iteration of the DP heuristic, has a large impact on computation time and solution quality. We set the value of H to 10,000, because it turned out that this value results in an average computation time of 65 seconds (with a maximum of 107 seconds) per problem instance, which is practically acceptable. Furthermore, we do not restrict the number of state expansions of a single state (we set the maximum number of state expansions E of a single state to n , the number of customers). As in Goel (2009), we minimize the number of vehicles as the primary objective and the total travel distance as the secondary objective. Therefore, state A dominates state B if they visit the same subset S and end in the same node $j \in S$, the costs (in terms of the objective function) of state A

do not exceed the costs of state B , and state A is at least as good as state B in all other state dimensions.

Table 4.2 presents the results of DP^{basic} with the relaxation of Directive 2002/15/EC and the best results found by Goel (2009). Note that in Goel (2009) substantially larger computation times are allowed: these results are the best out of five runs of half an hour each per problem instance.

Table 4.2 clearly shows that our method outperforms the large neighborhood search heuristic proposed by Goel. Only one problem instance (r103) out of 56 requires one more vehicle with our method, while for 47 problem instances a smaller number of vehicles is found. On average over all problem instances, our method finds solutions requiring 18.26% less vehicles.

Also the results on the travel distances show substantial improvements by our solution method. Only for the r1 problem instances no improvement is found, on average. In total, our method reduces the travel distances of 37 problem instances with an average reduction over all problem instances of 5.41%.

Table 4.2: Results DP^{basic} without Directive 2002/15/EC

Problem	DP^{basic}		Best in Goel (2009)		Change	
	vehicles	distance	vehicles	distance	vehicles	distance
c101	11	923.66	13	1,143.32	-15.38%	-19.21%
c102	11	1,097.97	13	1,198.82	-15.38%	-8.41%
c103	10	1,080.04	11	971.11	-9.09%	11.22%
c104	10	1,053.27	10	1,101.42	0.00%	-4.37%
c105	10	839.99	11	908.29	-9.09%	-7.52%
c106	11	900.10	11	1,079.24	0.00%	-16.60%
c107	10	874.03	10	1,023.77	0.00%	-14.63%
c108	10	892.71	10	975.20	0.00%	-8.46%
c109	10	1027.19	11	1,088.87	-9.09%	-5.66%
c1	10.33	965.44	11.11	1,054.45	-7.00%	-8.44%
c201	6	941.60	9	1,064.57	-33.33%	-11.55%
c202	5	866.09	9	990.03	-44.44%	-12.52%
c203	5	810.74	9	982.49	-44.44%	-17.48%
c204	4	768.19	8	873.22	-50.00%	-12.03%
c205	5	711.96	8	973.53	-37.50%	-26.87%
c206	5	677.79	7	838.91	-28.57%	-19.21%
c207	5	709.36	9	966.19	-44.44%	-26.58%
c208	5	677.62	8	948.21	-37.50%	-28.54%

Table 4.2 (Cont'd.)

Problem	DP^{basic}		Best in Goel (2009)		Change	
	vehicles	distance	vehicles	distance	vehicles	distance
c2	5.00	770.42	8.38	954.64	-40.30%	-19.30%
r101	13	1,483.95	15	1,413.43	-13.33%	4.99%
r102	13	1,398.59	13	1,296.16	0.00%	7.90%
r103	11	1,256.53	10	1,251.81	10.00%	0.38%
r104	8	1,023.47	10	1,024.13	-20.00%	-0.06%
r105	11	1,207.87	12	1,276.23	-8.33%	-5.36%
r106	9	1,162.18	11	1,150.95	-18.18%	0.98%
r107	9	1,068.90	10	1,098.62	-10.00%	-2.71%
r108	8	1,011.90	9	1,047.53	-11.11%	-3.40%
r109	9	1,094.14	11	1,058.01	-18.18%	3.42%
r110	8	1,061.92	10	1,062.43	-20.00%	-0.05%
r111	9	1,085.39	10	1,008.31	-10.00%	7.64%
r112	8	973.86	10	1,043.10	-20.00%	-6.64%
r1	9.67	1,152.39	10.92	1,144.23	-11.45%	0.71%
r201	10	1,337.07	13	1,335.17	-23.08%	0.14%
r202	10	1,258.97	12	1,215.88	-16.67%	3.54%
r203	9	1,130.86	10	1,122.58	-10.00%	0.74%
r204	6	913.46	9	1,013.70	-33.33%	-9.89%
r205	8	1,136.25	12	1,183.14	-33.33%	-3.96%
r206	7	1,084.71	9	1,068.91	-22.22%	1.48%
r207	7	1,024.53	11	1,064.22	-36.36%	-3.73%
r208	6	918.88	8	1,088.12	-25.00%	-15.55%
r209	7	1,104.62	10	1,067.09	-30.00%	3.52%
r210	7	1,185.38	10	1,076.23	-30.00%	10.14%
r211	6	1014.32	9	943.45	-33.33%	7.51%
r2	7.55	1,100.83	10.27	1,107.14	-26.55%	-0.57%
rc101	12	1,454.01	13	1,599.01	-7.69%	-9.07%
rc102	11	1,403.06	11	1,434.52	0.00%	-2.19%
rc103	10	1,278.33	11	1,268.81	-9.09%	0.75%
rc104	9	1,188.22	9	1,263.25	0.00%	-5.94%
rc105	12	1,426.29	12	1,405.72	0.00%	1.46%
rc106	10	1,253.11	12	1,297.67	-16.67%	-3.43%
rc107	9	1,189.06	11	1,243.08	-18.18%	-4.35%
rc108	9	1,212.69	10	1,269.90	-10.00%	-4.50%

Table 4.2 (Cont'd.)

Problem	DP^{basic}		Best in Goel (2009)		Change	
	vehicles	distance	vehicles	distance	vehicles	distance
rc1	10.25	1,300.60	11.13	1,347.75	-7.87%	-3.50%
rc201	10	1,554.93	11	1,510.67	-9.09%	2.93%
rc202	9	1,356.14	10	1,415.67	-10.00%	-4.21%
rc203	8	1,295.72	10	1,274.45	-20.00%	1.67%
rc204	6	975.56	9	1,264.73	-33.33%	-22.86%
rc205	9	1,437.07	11	1,521.10	-18.18%	-5.52%
rc206	8	1,220.06	11	1,418.40	-27.27%	-13.98%
rc207	8	1,234.27	10	1,171.94	-20.00%	5.32%
rc208	7	1,059.39	8	1,201.13	-12.50%	-11.80%
rc2	8.13	1,266.64	10.00	1,347.26	-18.75%	-5.98%

This remarkably large improvement with respect to the solutions found by the large neighborhood search heuristic of Goel can be explained as follows. Determining the feasibility of neighborhood solutions which respect Regulation (EC) No 561/2006 requires substantially larger computation times than when this regulation is neglected. Therefore, the number of neighborhood solutions that can be evaluated reduces substantially when respecting this regulation. In contrast, the running time complexity of our DP heuristic does not increase by respecting Regulation (EC) No 561/2006. Therefore, the number of states that can be investigated during a fixed amount of computation time does not substantially decrease when this regulation is respected.

If a practical application allows more computation time, this is beneficial for our method. For example, if H is set to 100,000 then the average computation time increases to 11 minutes (which is still much smaller than the computation times allowed in Goel (2009)), but with an average additional reduction of the number of vehicles and travel distance of 1.46% and 1.90%, respectively.

The restrictions on drivers' working hours that are imposed by Directive 2002/15/EC are generally ignored in the literature. However, they do reduce the solution space and, therefore, may have a substantial impact on the solution quality. We tested this impact by solving the benchmarks of Goel with DP^{basic} . Table 4.3 presents the average results for the six problem sets of DP^{basic} including Directive 2002/15/EC. Columns 4 and 5 present the relative changes in objective value when Directive 2002/15/EC is included. As can be observed, these changes are substantial (3.89% more vehicle routes, on average, and 0.96% more travel distance, on average). Therefore, Directive 2002/15/EC has a substantial impact on the resulting vehicle routes.

Finally, we tested the impact of exploiting the modifications of the basic rules on the quality of the resulting vehicle routes. Table 4.4 reports the average objective values for the six problem sets using DP^{ext} . In columns four and five

Problem	Without Directive		With Directive		Change ^a	
	vehicles	distance	vehicles	distance	vehicles	distance
c1	10.33	965.44	10.33	949.31	0.00%	-1.67%
c2	5.00	770.42	5.75	834.47	15.00%	8.31%
r1	9.67	1,152.39	9.67	1,155.89	0.00%	0.30%
r2	7.55	1,100.83	7.91	1,097.26	4.82%	-0.32%
rc1	10.25	1,300.60	10.25	1,300.14	0.00%	-0.04%
rc2	8.13	1,266.64	8.50	1,264.52	4.62%	-0.17%

^aChange with respect to the results without Directive 2002/15/EC

Table 4.3: Results DP^{basic} including Directive 2002/15/EC

Problem	DP^{basic}		DP^{ext}		Change ^a	
	vehicles	distance	vehicles	distance	vehicles	distance
c1	10.33	949.31	10.11	937.08	-2.15%	-1.29%
c2	5.75	834.47	5.25	773.80	-8.70%	-7.27%
r1	9.67	1,155.89	9.33	1,142.62	-3.45%	-1.15%
r2	7.91	1,097.26	7.36	1,084.70	-6.90%	-1.15%
rc1	10.25	1,300.14	10.00	1,322.41	-2.44%	1.71%
rc2	8.50	1,264.52	8.13	1,247.37	-4.41%	-1.36%

^aChange with respect to the results with DP^{basic}

Table 4.4: Results DP^{ext}

we compare the results with the results of ignoring the modifications of the basic rules (see Table 4.3). These columns indicate the profitability of exploiting the modifications.

On average, the results are improved for all problem sets. There is a substantial reduction in the number of vehicles used (4.28% on average) and in the total distance traveled (1.54% on average). Therefore, the benefits of exploiting the modifications are substantial and they should be accounted for when constructing vehicle routes.

4.5 Conclusion

We proposed a solution method for the VRPTW-EC. The method satisfies both the European legislation on drivers' driving hours and on drivers' working hours, formalized in Regulation (EC) No 561/2006 and Directive 2002/15/EC, respectively. It also considers all modifications of the basic rules in these laws.

We proposed a basic break scheduling method without the modifications of the basic rules, which we embedded in the restricted DP heuristic. The

methodology for scheduling the breaks is to maximize local flexibility at the last visited customer of the partial routes: we minimize the time to start service and we maximize the available driving and working time after service until a rest or a break has to be scheduled. This methodology fits well both in the DP framework and in practice. We extended the basic break scheduling method with exploiting the modifications of the basic rules, such that local flexibility is increased even further.

The computational results show that the basic method outperforms the state of the art local search method for the VRPTW-EC. The average number of vehicle routes is reduced by more than 18% and the average travel distance by more than 5%. On top of that, the computational effort of our approach is much smaller than for the local search method. One reason for this remarkable performance is that complex timing restrictions can be incorporated in the DP heuristic without increasing its running time complexity, as opposed to evaluating moves in approaches based on local search. The running time complexity of the DP heuristic remains the same, because estimating the quality of partial routes is done locally.

The results also show that Directive 2002/15/EC on the drivers' working hours has a substantial impact on the VRPTW-EC solutions and, therefore, cannot be ignored when constructing the vehicle routes. Finally, the results show that the modifications of the basic rules allow substantial cost reductions by reducing the number of vehicles by more than 4%, and the total travel distance by more than 1.5%, on average. Therefore, we recommend to exploit these modifications in practice.

Chapter 5

The impact of congestion avoidance

5.1 Introduction

Due to a growing amount of traffic and a limited capacity of the road network, traffic congestion has become a daily phenomenon. Since traffic congestion causes considerable delays, it is very costly for intensive road users such as logistic service providers and distribution firms. In particular, such delays cause large costs for hiring the truck drivers and the use of extra vehicles, and if they are not accounted for in the route plans, they may cause late arrivals at customers and violations of driving hours regulations. Therefore, accounting for and avoiding traffic congestion has a large potential for cost savings.

Traffic congestion may have several causes. Some of these causes are well-predictable, such as the large amount of commuter traffic during the daily peak hours, and others are less predictable, such as the weather or road accidents. Since the delays caused by peak hour traffic congestion are well-predictable and they constitute a large part (70 to 87%) of all traffic congestion delays (see, e.g., Skabardonis et al., 2003), we focus on avoiding peak hour traffic congestion.

Given a certain realization of the factors causing traffic congestion, peak hour traffic congestion depends on location and time of the day. Therefore, congestion avoidance is all about not being at the wrong place at the wrong time. There are several strategies to achieve this. For example, changing the visit sequence of a vehicle may avoid a large traffic jam. Another example is to remove a customer from one vehicle route and insert it into another. These congestion avoidance strategies can be optimized by solving a VRP with time-dependent travel times (TDVRP).

Another strategy to avoid traffic congestion is to select an alternative route between two customers during problematic hours. This strategy implies that the route between two customers depends on the chosen departure time, which can be optimized by solving a shortest path problem with time-dependent travel times (TDSPP). Orda and Rom (1990) show that a time-dependent shortest path problem for a given departure time can be solved with a modified Dijkstra (1959) search if the non-passing property, which states that overtaking is not possible, is satisfied.

The objective of this chapter is to determine to what extent congestion avoidance in off-line vehicle routing is profitable in practice. For this purpose, we propose four strategies of congestion avoidance to an increasing extent. Each strategy consist of two phases: 1) determining (time-dependent) shortest paths between all customer locations, and 2) solving a (TD)VRP. Phase 1 provides the input for phase 2. In each phase, congestion can be avoided to some extent. The congestion avoidance strategies may be further improved with on-line routing methods using real-time travel time information. However, on-line congestion avoidance strategies are beyond the scope of this thesis.

To test the impact of the congestion avoidance strategies in a realistic setting, we develop a number of VRP instances on real road networks and a speed model that represents peak hour traffic congestion. This speed model reflects the key elements of peak hour traffic congestion, based on observations of the Dutch motorists' organization ANWB (ANWB Reisinformatie, 2010) of peak hour traffic congestion in the Netherlands: large delays in urban areas, large delays on road lanes towards urban areas during the morning peak and in the opposite direction during the evening peak, and large delays on roads with a high speed limit (highways). We assume that these key elements are similar for other road networks with similar densities. We determine for each VRP instance an off-line vehicle route plan for each congestion avoidance strategy. Then, we evaluate the quality of these plans by executing them with the actual speeds in the road network obtained from the speed model. This methodology evaluates the impact of the different levels of congestion avoidance in a realistic setting.

This chapter¹ is organized as follows. Section 5.2 proposes four strategies with different levels of congestion avoidance for solving off-line vehicle routing problems. Section 5.3 proposes a general solution approach to solve the problem instances with these four congestion avoidance strategies. In Section 5.4, we propose a speed model that represents the key elements of peak hour traffic congestion. Section 5.5 presents the impacts of the different congestion avoidance strategies and Section 5.6 summarizes the main findings in this chapter.

¹This chapter is based on Kok et al. (2009a)

5.2 Strategies

In this chapter, we consider realistic vehicle routing problems on real road networks with speed patterns for each arc. Section 5.4 describes a speed model to derive these speed patterns. We solve each VRP in two phases. In phase one, we determine the (time-dependent) shortest paths between all locations. In phase two, we solve a (time-dependent) VRP, using the travel times resulting from phase one. In each phase, traffic congestion can be avoided. We propose four strategies in which congestion is avoided to an increasing extent. The first two strategies are closest to the VRP literature, in which we model the travel times as time-independent. The only difference between these strategies is that Strategy 2 accounts for traffic congestion to some extent, while Strategy 1 completely ignores traffic congestion. The other two strategies consider time-dependent travel times and, therefore, traffic congestion can be avoided. We do this to a greater extent with Strategy 4 than with Strategy 3. We now describe the four strategies in more detail.

Strategy 1 is the base strategy in which traffic congestion is completely ignored. In phase one, we determine all free-flow shortest paths between the customers. For this purpose, we set the speed of each arc in the road network to the maximum in its speed pattern. We calculate the travel times along the paths using these maximum speeds. In phase two, we solve a VRP based on the resulting free-flow travel times between the customers. This strategy does not account for traffic congestion to any extent, and, therefore, congestion is not avoided.

With Strategy 2, we account for traffic congestion by solving a VRP based on average travel times. As with Strategy 1, we determine the free-flow shortest paths in phase one by setting each arc to its maximum speed. We fix these paths and use them if the corresponding origin-destination pair is selected in phase two. Next, we determine the average travel times along these paths by calculating the exact travel times for a large number of different departure times. We set the inter-departure time to 15 minutes, meaning that we set the first departure time equal to the depot opening time, the second departure time 15 minutes later, etc. We continue this process until the departure time exceeds the depot closing time. For each departure time, we calculate the travel times using the time-dependent arc speeds resulting from the speed model. We use the averages over these time-dependent travel times as input for a VRP in phase two. With this strategy, we account for traffic congestion by averaging the delays along a route over the entire day.

With Strategy 3, we avoid traffic congestion by solving a TDVRP. Phase one with this strategy is similar to phase one with Strategy 2. The only difference is that we do not average the time-dependent travel times (with inter-departure times of 15 minutes) in phase one, but we use them as input for the TDVRP in phase two. We use interpolation each time the TDVRP solver requires

the travel time between two customers for a given departure time. Note that with this strategy the route between each pair of customers is fixed at the free-flow shortest path, as with Strategy 1 and 2. This strategy accounts for traffic congestion to a large extent and aims at avoiding traffic congestion by selecting alternative customer visit sequences and alternative customer-vehicle assignments.

Strategy 4 adopts the highest level of congestion avoidance. With this strategy, we determine time-dependent shortest paths for a large number of different departure times in phase one (we again set the inter-departure time to 15 minutes). Next, we determine the time-dependent travel times for all these different departure times along the corresponding time-dependent shortest paths and use them as input for solving a TDVRP in phase two. Again, we use interpolation to estimate the travel time for a given departure time when the TDVRP is being solved. Finally, since the resulting planned departure times generally do not coincide with a departure time for which the time-dependent shortest path has already been determined in phase one, we determine the shortest paths for the planned departure times in a final step. This strategy accounts for traffic congestion to the same extent as Strategy 3, but adopts a higher level of congestion avoidance: at problematic hours we choose alternative routes between customers. Table 5.1 gives an overview of the four strategies.

Strategy	Shortest paths	Travel time input for VRP	Accounting for congestion	Avoiding congestion
1	free-flow	free-flow	no	no
2	free-flow	average	a small extent	no
3	free-flow	time-dep.	yes	yes
4	time-dep.	time-dep.	yes	yes

Table 5.1: Strategy overview

After we solve the VRP instances with each strategy, we evaluate the performances of the resulting vehicle route plans using the speeds resulting from the speed model. Note that with each strategy we use travel time estimations to solve a (TD)VRP. The reason for using travel time estimations with Strategy 3 and 4 is that calculating the exact travel times while solving a TDVRP is too time-consuming (shortest paths generally contain hundreds of arcs - depending, among other things, on the size of the road network - along which the travel time calculation needs to be propagated). In practice, however, the TDVRP often needs to be solved within limited computation times. Therefore, the proposed travel time estimations, which are independent of the size of the road network, are suitable for practice.

5.3 Solution methods

With each strategy, we need to solve a (TD)SPP and a (TD)VRP. To make a fair comparison between the different strategies, we solve the problems with solution methods that do not need to be tailored for each specific problem. This means that the shortest path algorithm can solve both SPPs and TDSPPs, and the VRP solution method can solve both VRPs and TDVRPs. We solve the (TD)VRPs with the restricted dynamic programming heuristic proposed in Chapter 3. This DP heuristic can solve VRPs and TDVRPs within comparable computation times. Therefore, the computation times for the (TD)VRPs are (approximately) the same for each strategy, such that this does not affect the applicability of the different strategies in practice. For the computation times of the shortest path algorithms we can be less restrictive, since shortest path calculations are generally done as a pre-processing in practice.

We solve the (TD)SPPs with a modified Dijkstra (1959) algorithm. The only adaptation we make is that we initiate the searches with a given departure time from the source node, and we keep track of the departure times from each reached node. This is necessary to determine the time-dependent travel times when the labels of the nodes need to be updated. This approach allows us to solve the shortest path problems with Strategy 1, 2, and 3 in (approximately) the same computation times. Only with Strategy 4, the computation time increases, since we have to rerun the algorithm for each possible departure time. However, this is all done in phase one, a pre-processing phase in which computation times play a minor role in practice. Note that to guarantee optimality of Dijkstra's algorithm, the non-passing property - which states that overtaking is not possible - must be satisfied, since violations of this property may allow an optimal path to contain non-optimal subpaths. The speed model in Section 5.4 satisfies this property.

5.4 Speed model

To investigate the impact of the different congestion avoidance strategies in a realistic setting, we propose a speed model for real road networks that reflects the key elements of peak hour traffic congestion, as observed by the Dutch motorists' organization ANWB. The speed model is designed for the road network data used in this chapter, but the methodology can be applied to any other road network data. Note that the speed model is based on the key elements of peak hour traffic congestion; it is not based on real (historical) travel time data. There is no empirical evaluation of the performance of this speed model on real road networks in particular. To apply vehicle routing methods in practice, the speed model should be tailored to the road networks under consideration. This tailoring is beyond the scope of this study: the objective of this study is to get a

good estimation of the performance of different congestion avoidance strategies in a broad and realistic setting.

Our road network data is a selection of the TIGER/Line files (Tiger/line, 2002), which consist of road network data of each of the 50 US states. We select the states Rhode Island, Connecticut, Maryland, Massachusetts, and New Jersey, because they have a high degree of urbanization, resulting in many traffic congestion problems during the peak hours. Moreover, the sizes of these states are comparable to some smaller countries in Europe such as the Netherlands and Belgium, which are also densely populated. Furthermore, we select Kentucky for comparison reasons: in addition to a relatively small urban area it also has large rural areas.

The TIGER/Line data contain geoinformation on nodes in the road network (each node represents an intersection of different roads or a change in average speed on the same road), and distance and road category information on directed arcs connecting these nodes. There are four road categories with the following free-flow time-independent travel speeds: 1 for road category 1, 0.8 for road category 2, 0.6 for road category 3, and 0.4 for road category 4.

We propose a time-dependent speed model that incorporates peak hour traffic congestion. This speed model defines for each arc a travel speed during the peak hours and a travel speed outside the peak hours. We assume the morning peak to last from 6:30AM until 9:30AM and the evening peak from 3:30PM until 7PM, since observations of the Dutch motorists' organization ANWB indicate the peak periods as such. The speed outside the peak hours is set to the speed provided by the TIGER/Line files. We base the speed drops during the peak hours on the key elements of peak hour traffic congestion as follows.

Peak hour traffic congestion is mainly caused by a large amount of commuter traffic. Since commuter traffic needs to be at the same time (at the start of the working day) at the same place (large cities), the most common roads get congested during the peak hours. With respect to peak hour traffic congestion, the following elements are relevant:

1. Degree of urbanization. Within urban areas, there is much more traffic congestion than in rural areas. Therefore, there is a positive correlation between the degree of urbanization and the amount of speed drop during the peak hours.
2. Direction of commuter traffic. During the morning peak, commuter traffic is traveling towards working areas. Therefore, during the morning peak, much more traffic congestion appears on road lanes directed to urban areas than on road lanes in the opposite direction (and during the evening peak vice versa).
3. Speed limit. In general, roads with a high speed limit (highways) are

more heavily used than roads with a lower speed limit (rural roads). Since roads with a high speed limit lie on many (free-flow) shortest paths, there is a positive correlation between a road’s speed limit and the amount of traffic congestion during the peak hours.

We propose the following approach to quantify the speed drops during the peak hours on each arc in the road network, based on the three observations described above. First, we determine the degree of urbanization of the source- and destination-node of the arc under consideration by counting the number of network nodes in the proximity area of each node. We refer to such nodes as proximity nodes. In Section 5.4.1, we explain in detail how this proximity area is defined and how we use it to determine the degree of urbanization of each node. We set the degree of urbanization of each arc to the maximum of the degrees of its source- and destination-node. Next, we determine the direction of the arc, i.e., towards or from an urban area. If the destination-node has a higher number of proximity nodes than the source-node, then the arc is directed towards an urban area. Finally, the speed limit on the arc is given by the road category of the arc under consideration. In this way, all three elements relevant to peak hour traffic congestion are considered.

Using the calculated degree of urbanization, the arc direction, and the road category, we calculate the speed drops as follows. Table 5.2 presents the maximum (relative) speed drops, expressed as a percentage of the free-flow speed, during the morning peak for each road category (for the evening peak the two rows are swapped). These maximum speed drops depend both on the arc direction and on the road category. In Section 5.5.1, we conduct a sensitivity analysis of these speed drops by repeating all computational experiments for a selected number of alternative maximum speed drop patterns. For each arc, we multiply the maximum speed drop with a fraction, depending on the degree of urbanization. Table 5.3 presents these fractions. Degree 1 represents the lowest degree of urbanization. For example, the speed drop during the morning peak on an arc with road category 2, a degree of urbanization of 3, and directed towards an urban area equals $2/3 * 0.65 \approx 43\%$ of the free-flow speed on this arc.

	road cat. 1	road cat. 2	road cat. 3	road cat. 4
Arcs towards urban areas	0.9	0.65	0.4	0.15
Arcs from urban areas	0.3	0.25	0.2	0.15

Table 5.2: Maximum speed drop during the morning peak as a percentage of the free-flow speed

Using this approach, the speed model reflects the three key elements of peak hour traffic congestion. Table 5.2 presents the dependency of the speed

Degree of urbanization	Fraction of speed drop
1	0
2	1/3
3	2/3
4	1

Table 5.3: Degree of urbanization and corresponding speed drop fraction

drops on the road direction and the speed limit, and Table 5.3 shows that arcs located in high degrees of urbanization encounter larger speed drops than arcs in low degrees of urbanization.

5.4.1 Determining the degree of urbanization of a node

We propose the following methodology for determining the degree of urbanization of each node in the road network. In general, nodes that lay in highly urbanized areas contain more proximity nodes than nodes that lay in rural areas. If a node lies in such a highly urbanized area, we say that it has a high degree of urbanization. We define the proximity area of a node to a circle centered at this node with a radius of 10 km. This implies that if we approach a highly urbanized area, we may identify it when we are in a 10 km range of this area. To get an indication of the number of proximity nodes for a node that lies in a highly urbanized area, we determine for each state the maximum number of proximity nodes over all nodes in that state (each state contains at least one highly urbanized area).

State	Max # proximity nodes ($\times 1,000$)
Connecticut	15
Kentucky	19
Rhode Island	24
Maryland	28
New Jersey	32
Massachusetts	38

Table 5.4: Maximum # proximity nodes in each state

Table 5.4 shows that nodes that lay in highly urbanized areas, typically contain 15 thousand or more proximity nodes. Therefore, we set the degree of urbanization of a node to 1 if it contains at least 15 thousand proximity nodes. The numbers of proximity nodes corresponding to the other degrees of urbanization are evenly spread between 0 and 15 thousand. Table 5.5 presents the resulting correspondence between the number of proximity nodes and the degree of urbanization.

# proximity nodes ($\times 1,000$)	degree of urbanization
0 - 5	1
5 - 10	2
10 - 15	3
15+	4

Table 5.5: Number of proximity nodes and degree of urbanization

5.5 Computational experiments

We test the impact of the four congestion avoidance strategies on a large number of VRP instances. These VRP instances are developed on the road networks of the six selected states and the speeds resulting from the speed model. The customer locations are randomly selected from the nodes in the road network. Clustering of the customer locations is a natural result, since the road network is denser in urban areas. Furthermore, the first selected node is considered to be the depot.

We develop 15, 50, and 100 customer problem instances. The 15 customer problem instances are small enough to be solved to optimality in practical computation times (3 to 8 minutes). The 100 customer problem instances are approximately the largest instances for which we can still run a sufficient number of experiments within practical computation times.

We add time windows to 50 percent of the customers indicating the period in which service must start. We set the time window of the depot to $[0, 14]$, indicating a working day of 14 hours from 6AM until 8PM. The morning and evening peak last from 6:30AM until 9:30AM and from 3:30PM until 7PM, respectively. The width of the time windows at the selected customers are randomly drawn from $\{2, 3, 4, 5, 6\}$ quarters of an hour. The customer service times are randomly drawn from $\{1, 2\}$ quarters of an hour and the demands are randomly drawn from $\{1, 2, \dots, 10\}$. If the vehicle capacities are set too low, then the length of vehicle routes are only restricted by these capacities. If they are set too high, then the length of the routes are only restricted by the time windows. Preliminary tests indicated that a capacity of 55 results in a fair trade-off. Therefore, we used this capacity in our experiments. We generate 20 problem instances for each combination of state and number of customers, resulting in 360 VRP instances in total.

The primary objective is set to minimize the total number of vehicle routes, and the secondary objective to minimize the total duty time (sum of travel, service, and waiting times) of the truck drivers. We choose to minimize duty times as a secondary objective instead of travel times or travel distances, since small duty times are generally more important in practice. We choose to dispatch the trucks at time zero. Although postponing the departure times at the depot

may substantially reduce duty times, as we shall demonstrate in Chapter 6, it is beyond the scope of this chapter to also optimize the departure times of each vehicle. Chapter 6 discusses duty time minimization through departure time optimization within VRPs with time-dependent travel times and driving hours regulations.

We implemented the data-structures and solution algorithms in Delphi 7 on a PC with a Core 2 Quad, 2.83 GHz CPU and 4 GB of RAM. Table 5.6 presents the average results for the four strategies over all problem instances (we scale the results of the 15- and 50-customer instances to 100-customer instances), except for the problem instances generated on Kentucky. We scale and average the results, since the results of the 15-, 50-, and 100-customer problem instances are similar. All performance measures are derived by evaluating the developed route plans with each strategy against the speeds resulting from the speed model. Next to the two objectives ‘minimizing number of vehicle routes’ and ‘minimizing duty times’ we also report on the following performance measures: total travel distance, total number of late arrivals at customers, total number of late return times at the depot, maximum late time over all customers, and total late time over all customers. Although we do not optimize on travel distances, we also report the impacts of the strategies on this performance measure. For practice, both measures are relevant, since they imply different transport costs. The last four performance measures indicate the reliability of the route plans. Note that all performance measures present averages over all problem instances, except for ‘maximum late time’, which is the maximum over all problem instances and all vehicle routes.

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.88	11.70%	0.55%	0.36%
Total duty time (hrs)	153.2	3.34%	-6.59%	-7.69%
Total travel distance	58.56	-0.60%	-0.59%	-1.24%
# late arrivals	16.47	-77.37%	-99.86%	-99.72%
# late return times	0.328	-66.10%	-100.00%	-100.00%
Maximum late time (hrs)	3.051	-67.56%	-99.96%	-99.95%
Total late time (hrs)	15.28	-88.58%	-100.00%	-100.00%

^aabsolute figures

^brelative change with Strategy 1

Table 5.6: Main results, aggregated over all problem instances.

The number of vehicle routes is larger with Strategy 2, 3, and 4 than with Strategy 1. This can be explained by the low travel time estimations based on free-flow travel conditions with Strategy 1, which are lower bounds to the travel time estimations with all other strategies. However, the results indicate that avoiding traffic congestion to an increasing extent reduces the number of vehicles again and approaches the number of vehicles required with Strategy 1.

The duty times show a similar pattern, which can be partially explained by the number of vehicle routes. Furthermore, the additional congestion information with Strategy 2, 3, and 4 with respect to Strategy 1 reduces the total duty time. This even results in an overall decrease of total duty time for Strategy 3 and 4 with respect to Strategy 1, despite the larger number of vehicle routes. Note that the *estimated* duty time with Strategy 1 is the best (optimal for the 15-customer instances) for free-flow travel conditions. Therefore, if we subtract the estimated duty times found with Strategy 1 from the real duty times with each strategy, then we obtain estimations of the additional duty times caused by traffic congestion. Table 5.7 presents the average amount of additional duty time for each strategy with respect to the estimated duty time found with Strategy 1. The results show that with Strategy 1 about 8.8% of the total duty time is due to traffic congestion delays. Strategy 2 results in an even larger additional duty time than Strategy 1. Strategy 3 and 4, however, reduce the additional duty time of Strategy 1 substantially by 75% and 87%, respectively.

	Strat. 1	Strat. 2	Strat. 3	Strat. 4
Additional duty time (hrs)	13.49	18.61	3.403	1.711

Table 5.7: Average additional duty time (hours) caused by traffic congestion.

The travel distances are similar with each strategy. Only with Strategy 4 some smaller travel distances are obtained. This can be explained by choosing alternative paths between customer locations at bad hours with this strategy. Such alternative paths typically contain arcs with smaller speed drops than arcs on the free-flow shortest paths due to, e.g., arcs with lower maximum speeds. However, such lower speeds have to be compensated by smaller travel distances.

The reliability of the route plans strongly increases if the level of congestion avoidance increases. All reliability measures show a strong reduction with respect to Strategy 1. This is not surprising, since the strategies account for traffic congestion to an increasing extent. However, the huge improvement of Strategy 2 with respect to Strategy 1 in comparison with the additional improvements of the other two strategies is less obvious. The explanation for this is that an underestimation of a travel time at the start of a vehicle route with Strategy 1 propagates along all arrival times at successive customers of that vehicle route. With Strategy 2, such an underestimation is generally compensated by an overestimation of later travel times.

Table 5.8 presents the results for the sixth state: Kentucky. As mentioned before, Kentucky contains a large rural area compared to the other 5 states. Therefore, the main part of the routes of the problem instances generated on this state do not contain heavy delays caused by traffic congestion. Table 5.8 shows that this has a strong impact on the results. The increase in number

of vehicle routes with Strategy 2 with respect to Strategy 1 is much smaller than for the other states. Moreover, with Strategy 4 almost the same number of vehicle routes is attained as with Strategy 1.

The differences in duty times are also much smaller. With Strategy 1, only about 1 hour of duty time is caused by congestion delays, as opposed to the 13.5 hours for the other states. Congestion avoidance strategies lead to reductions of this additional duty time up to 84% with Strategy 4. The reliability measures show similar results as at the other states. In conclusion, Kentucky leads to much less congestion problems than highly urbanized states, but high levels of congestion avoidance still substantially increase the reliability of the vehicle route plans.

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.48	1.69%	0.93%	0.11%
Total duty time (hrs)	135.4	0.67%	-0.08%	-0.61%
Total travel distance	52.50	0.93%	2.27%	-0.98%
# late arrivals	2.011	-68.51%	-100.00%	-92.82%
# late return times	0.000 ^c	- ^c	- ^c	- ^c
Maximum late time (hrs)	0.349	-80.24%	-100.00%	-99.93%
Total late time (hrs)	0.558	-88.29%	-100.00%	-99.94%
Additional duty time (hrs)	0.992	91.65%	-10.75%	-83.94%

^aabsolute figures

^brelative change with Strategy 1

^cthere were no late return times with each strategy

Table 5.8: Results for Kentucky

5.5.1 Sensitivity analysis of the speed model

The actual speed drops on specific road networks depends on several factors, such as landscape (hilly or flat), urban organization (companies centered at one city or dispersed across many cities), difference in speed limit between trucks and cars, and even culture. Therefore, we conduct a sensitivity analysis of the speed model we proposed in this study by repeating the computational experiments for a selected number of alternatives. We selected the following four alternatives for the maximum speed drops, which can be seen as extreme cases of the key elements of peak hour traffic congestion:

1. Only speed drops on highways. In this alternative, there are no speed drops during the peak hours on secondary roads, but only on the highways. This alternative can be seen as a case in which the majority of the road users does not consider alternative paths through secondary roads as an option to avoid peak hour traffic congestion. It may also be the case

that secondary roads are less sensitive to traffic congestion due to better traffic flows as a result of a lower maximum driving speed. Table 5.9^a presents the maximum relative speed drops for this alternative.

2. All roads the same maximum relative speed drop. This alternative is basically the opposite of Alternative 1. It represents the case in which many road users choose alternative paths during the peak hours, such that also secondary roads get congested. Therefore, the maximum relative speed drops depend no longer on the road category. Table 5.9^b presents the maximum relative speed drops for this alternative.
3. Similar speed drops during the morning and evening peak. Some cities lay between two areas that both attract much commuter traffic. Such an urban organization may imply that roads passing these ‘in-between cities’ get congested in both directions, and both during the morning and evening peak. To account for this alternative, we set the maximum relative speed drops independent of the road direction (which also implies that they are the same for the morning and evening peak), as Table 5.9^c presents.
4. Small speed drops. To investigate the sensitivity of the amount of the relative speed drops, we propose a fourth alternative in which the maximum relative speed drops are half the original maximum relative speed drops. Table 5.9^d presents the resulting maximum relative speed drops.

We run all computational experiments again for all alternatives. We compare the results of the different alternatives with the original speed pattern.

Table 5.10 presents the results for Alternative 1 in which speed drops during the peak hours only appear on highways. Strategy 2 results in a larger number of vehicle routes than Strategy 1, but this increase is smaller than with the original speed drops. This can be explained by the smaller speed drops, on average, in this alternative. The same holds for the increase in duty time with Strategy 2 with respect to Strategy 1. The reduction in the number of vehicles needed with Strategy 4 with respect to Strategy 3 is almost twice as big as this reduction with the original speed drops. This can be explained by the free-flow travel times on secondary roads, which are only exploited with Strategy 4. The higher reliability of the route plans when higher levels of congestion avoidance are adopted is similar to the results with the original speed drops. The reduction in additional duty time caused by traffic congestion is even more impressive than with the original speed drops: 95% instead of 87% with Strategy 4. This larger reduction is due to the bigger opportunities for selecting alternative paths during the peak hours.

Table 5.11 presents the results for Alternative 2 in which all roads have the same maximum relative speed drop. For this alternative, the reliability measures with Strategy 1 are worse than with the original speed drops because

(a)				
	road cat. 1	road cat. 2	road cat. 3	road cat. 4
Arcs towards urban areas	0.9	0.0	0.0	0.0
Arcs from urban areas	0.3	0.0	0.0	0.0

(b)				
	road cat. 1	road cat. 2	road cat. 3	road cat. 4
Arcs towards urban areas	0.9	0.9	0.9	0.9
Arcs from urban areas	0.3	0.3	0.3	0.3

(c)				
	road cat. 1	road cat. 2	road cat. 3	road cat. 4
Arcs towards urban areas	0.9	0.65	0.4	0.15
Arcs from urban areas	0.9	0.65	0.4	0.15

(d)				
	road cat. 1	road cat. 2	road cat. 3	road cat. 4
Arcs towards urban areas	0.45	0.325	0.2	0.075
Arcs from urban areas	0.15	0.125	0.1	0.075

Table 5.9: Maximum speed drop during the morning peak as a percentage of the free-flow speed for the four alternatives: a) only speed drops on highways, b) all roads the same maximum relative speed drop, c) similar speed drops during the morning and evening peak, d) small speed drops.

of the bigger speed drops, on average. As a consequence, the increase in number of vehicle routes with Strategy 2, 3, and 4 with respect to Strategy 1 is larger than for the original speed drops, especially with Strategy 2 (18% vs. 12%). Also the changes in duty times are more extreme than with the original speed drops: a larger increase with Strategy 2, and a larger decrease with Strategy 3 and 4. Although the maximum relative speed drops are the same for each road category, Strategy 4 still results in better vehicle route plans than Strategy 3. This can be explained by simply having more alternatives to choose from, but also by the fact that the same *relative* speed drop results in smaller *absolute* speed drops on roads with lower maximum speeds, which typically appear more often on alternative paths.

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.88	9.43%	0.59%	0.10%
Total duty time (hrs)	154.8	1.75%	-7.64%	-9.31%
Total travel distance	58.56	-0.24%	-0.26%	-1.44%
# late arrivals	17.15	-62.83%	-100.00%	-97.82%
# late return times	0.490	-67.35%	-100.00%	-100.00%
Maximum late time (hrs)	3.244	-57.08%	-100.00%	-99.22%
Total late time (hrs)	18.05	-79.74%	-100.00%	-99.92%
Additional duty time (hrs)	15.10	17.94%	-78.29%	-95.46%

^aabsolute figures

^brelative change with Strategy 1

Table 5.10: Results Alternative 1: only speed drops on highways.

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.88	18.23%	1.18%	0.65%
Total duty time	158.8	5.73%	-8.57%	-9.90%
Total travel distance	58.56	0.08%	0.60%	0.05%
# late arrivals	20.42	-83.64%	-99.54%	-99.85%
# late return times	0.823	-84.35%	-97.30%	-100.00%
Maximum late time	4.878	-82.36%	-98.60%	-99.96%
Total late time	23.48	-94.48%	-99.87%	-100.00%
Additional duty time	19.08	47.72%	-71.32%	-82.36%

^aabsolute figures

^brelative change with Strategy 1

Table 5.11: Results Alternative 2: all roads have the same maximum relative speed drop.

Table 5.12 presents the results for Alternative 3 in which speed drops during the morning and evening peak are similar. Due to the larger speed drops, on average, the reliability measures for Strategy 1 are worse than with the original speed drops. As a consequence, the increase in number of vehicle routes with Strategy 2, 3, and 4 with respect to Strategy 1 is larger than for the original speed drops. The larger speed drops in Alternative 3 offer, on the other hand, more possibilities for avoiding them, which results in larger duty time reductions with Strategy 3 and 4 with respect to Strategy 1 than with the original speed drops. The other results are similar to the results with the original speed drops.

Table 5.13 presents the results for Alternative 4 in which speed drops are half the original speed drops. Due to the smaller speed drops, there are fewer late arrivals than with the original speed drops, and the additional duty time caused by traffic congestion is also smaller. As a consequence, Strategy 3 and 4

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.88	15.31%	0.78%	0.57%
Total duty time	159.4	4.20%	-9.20%	-10.82%
Total travel distance	58.56	-0.29%	-0.94%	-1.70%
# late arrivals	20.97	-75.79%	-99.92%	-99.70%
# late return times	0.752	-63.96%	-97.05%	-100.00%
Maximum late time	3.589	-68.74%	-100.00%	-99.97%
Total late time	23.71	-89.06%	-100.00%	-100.00%
Additional duty time	19.67	33.99%	-74.52%	-87.66%

^aabsolute figures

^brelative change with Strategy 1

Table 5.12: Results Alternative 3: speed drops during the morning and evening peak are similar.

find solutions with approximately the same number of vehicle routes as Strategy 1. Note that the smaller number of vehicle routes with Strategy 3 than with Strategy 1 and 4 is due to the heuristic solution method: for the 15 customer problem instances (which are solved to optimality) the number of vehicle routes is the same for all strategies.

Even with the small speed drops in Alternative 4, the reductions of the additional duty time caused by traffic congestion with Strategy 3 and 4 with respect to Strategy 1 are still substantial (more than 60%). We noticed this also in the results on Kentucky, for which also the average congestion delays are much smaller than for the other states and alternatives. This strongly indicates that high levels of congestion avoidance leads to substantial cost savings for a broad range of different road networks. The reliability improvements with respect to Strategy 1 are similar to the improvements with the original speed drops.

5.6 Conclusions

We proposed four strategies in which traffic congestion is accounted for and avoided to an increasing extent. Next, we proposed a speed model on real road networks that reflects the key elements of peak hours traffic congestion. This speed model provided us a platform for testing the impact of the different strategies in a realistic setting.

The results indicated that the reliability of route plans strongly increase if traffic congestion is accounted for. However, if VRPs are modeled with time-independent travel times, then this reliability increase is achieved against more vehicle routes and larger duty times. By adopting higher levels of congestion avoidance such as solving VRPs with time-dependent travel times and solving

Performance measure	Strat. 1 ^a	Strat. 2 ^b	Strat. 3 ^b	Strat. 4 ^b
# vehicle routes	15.88	2.97%	-0.06%	0.02%
Total duty time (hrs)	141.2	1.72%	-0.67%	-0.68%
Total travel distance	58.56	-0.25%	0.40%	-1.26%
# late arrivals	3.394	-87.33%	-99.41%	-97.94%
# late return times	0.000 ^c	- ^c	- ^c	- ^c
Maximum late time (hrs)	0.328	-87.34%	-99.79%	-99.22%
Total late time (hrs)	0.494	-95.26%	-99.98%	-99.93%
Additional duty time (hrs)	1.528	158.94%	-62.11%	-63.07%

^aabsolute figures

^brelative change with Strategy 1

^cthere were no late return times with each strategy

Table 5.13: Results Alternative 4: the speed drops are half the original speed drops.

time-dependent shortest path problems, these cost measures can be reduced substantially. Solving a combination of these two problems is particularly effective, resulting in huge reliability improvements, substantial duty time reductions (almost 87% of the additional duty times caused by traffic congestion can be eliminated), and substantially reducing the number of vehicles needed (almost all extra vehicles needed due to congestion delays can be eliminated).

We conducted a sensitivity analysis of the speed model, which indicated that under various scenarios the improvements with the higher levels of congestion avoidance remain. Even in case of small speed drops during the peak hours congestion avoidance results in substantially more reliable route plans and reductions of duty times. In certain extreme cases, such as only speed drops on highways, congestion avoidance is even more powerful resulting in reductions of additional duty time caused by traffic congestion of almost 95%.

Chapter 6

Vehicle routing with traffic congestion and break scheduling

6.1 Introduction

As discussed in Chapter 1, for companies that practice vehicle routing, having compact driver duty times leads to substantial savings regarding, e.g., truck driver hiring costs and the time vehicles are unavailable for other services. Compact duty times are often even needed to obey the law: the EC social legislation on driving and working hours, for example, limits the daily driving and working times of truck drivers.

To obtain compact driver duty times, the departure times within vehicle routes must be optimized within the applicable regulations. As indicated in Chapter 2, time-dependent travel times and driving hours regulations make departure time optimization within vehicle routes particularly difficult. The combination of duty time minimization within the construction of vehicle routes, accounting for time-dependent travel times, and obeying driving hours regulations is a highly complex problem, which has - to the best of our knowledge - not been addressed so far. The objective of this chapter is to develop a solution method for the VRPTW with time-dependent travel times and the EC social legislation on driving and working hours (TDVRP-EC). Following the promising results for the VRPTW with the EC social legislation in Chapter 4, we propose a solution method for the TDVRP-EC based on the DP heuristic of Chapter 3.

In this chapter, we consider one-day planning. The choice for one-day planning is motivated by practice, since duty time minimization is applied to one-day schedules because the costs applied for night rests on duty differ from those for working times. Note that certain rules concerning one-week planning may also have an impact on one-day planning (e.g., extending the daily driving time to 10 hours at most twice a week). The solution method we propose in this chapter is flexible with respect to such rules, since the method is also applicable with other input parameters, such as extended daily driving times and reduced daily rest periods.

In the VRP literature, generally, the primary objective is to minimize the number of vehicles used and the secondary objective is to minimize the total distance traveled. However, within the VRPTW this secondary objective may lead to large waiting times, which are costly in practice. Moreover, traffic congestion makes the duration of travels (and thus also the costs of these travels) depend on the time of the day, while the distance remains the same. Therefore, a more relevant secondary objective is to minimize the total duty time (Savelsbergh, 1992). We numerically analyze both travel distance and duty time as the secondary objective. Moreover, we quantify the impact of extending the depot opening hours such that traveling before the morning peak and after the evening peak becomes possible.

This chapter¹ is organized as follows. Section 6.2 formally introduces the TDVRP-EC. Section 6.3 discusses some important assumptions considering waiting times at customers that have a strong impact on the complexity of departure time optimization. Section 6.4 proposes a solution approach for the TDVRP-EC, based on the DP heuristic of Chapter 3. In Section 6.5, we report on computational experiments to analyze the impact of different objective functions (minimize travel distance vs. minimize duty time) on the overall solution quality, and the impact of extending the depot opening hours. In Section 6.6, we summarize our main findings.

6.2 Problem description of the TDVRP-EC

We consider an extension of the classical VRPTW for which we first introduce some notation that we require throughout this chapter. Within the VRPTW we are given a set of vehicles $K = \{1, \dots, m\}$ and a set of nodes $V = \{0, \dots, n\}$ in which node 0 represents the depot. Nodes $i > 0$ represent customer requests with demands q_i and service time windows $[e_i, l_i]$. The problem is to find a set of routes, each starting and ending at the depot, such that the total demand along each route does not exceed the vehicle capacity Q , each service starts in the given time window, and some objective is satisfied.

¹This chapter is based on Kok et al. (2010)

We extend the VRPTW by considering time-dependent travel times and driving hours regulations. We assume that (aggregated) data is available for time-dependent travel speeds along customer-to-customer routes. In other words, we do not consider the underlying road network in which time-dependent shortest paths should be determined. The calculation of time-dependent shortest paths can be done in a pre-processing step and from these paths the required aggregated travel data for customer-to-customer routes can be obtained, as demonstrated in Chapter 5. To model the time-dependent travel times, we apply the time-dependent speed model of Ichoua et al. (2003) (see Chapter 2), which satisfies the non-passing property. There are two main reasons for this approach: 1) the non-passing property is a realistic assumption, 2) a more detailed travel time function (e.g., any differentiable travel time function) is hard to deduce from, e.g., historical travel time data.

In this chapter, we consider Regulation (EC) 561/2006 on driving hours (European Union, 2006), which is valid for all member countries of the European Union. Since we consider one-day planning, we assume that breaks and rests have to be scheduled at customer locations. Recall that, considering one-day planning, Regulation (EC) 561/2006 poses the following requirements per driver:

1. A period between two breaks of at least 45 minutes is called a *driving period*. The accumulated driving time in a driving period may not exceed 4.5 hours. The break that ends a driving period may be reduced to 30 minutes if an additional break of at least 15 minutes is taken anywhere during that driving period. The driving hours regulations do not allow service times at customers to be considered as break time.
2. The total accumulated driving time may not exceed 9 hours.
3. The total accumulated duty time may not exceed 13 hours.

The TDVRP-EC comprises three types of decisions: assigning customers to vehicles, sequencing customer visits for each vehicle, and selecting departure times for each vehicle. Departure times need not only be determined for the departure at the depot, but also at each customer to account for the driving hours regulations and the time windows. The possibility to schedule waiting times at customers makes this departure time scheduling problem particularly difficult, as we shall illustrate in Section 6.3. Therefore, we choose the *non-delay property* for scheduling departure times, meaning that we do not schedule waiting time at customers (i.e, delay the departure time) when this is not necessary. We discuss in the next section the scheduling of waiting times and our underlying assumptions in detail.

6.3 Waiting time assumptions

In order to construct feasible vehicle routes, we need a method that finds feasible departure times for these routes. Furthermore, the costs of such routes have to be determined in terms of duty times. In Chapter 2, we have proposed an ILP model to determine feasible vehicle departure times given the customer visit sequence of a vehicle route. Using the objective to minimize the difference between the start and completion time of a vehicle, we can use the ILP formulation to optimize a vehicle's departure times. We refer to this problem as the vehicle departure time optimization problem (VDO). When constructing vehicle routes in the restricted DP heuristic, however, computation times to solve this ILP are too large to apply it for each (partial) vehicle route that is considered.

A complicating factor for the determination of the minimum duty time in the DP heuristic is the use of *unforced waiting time*. We define unforced waiting time as waiting time that is not forced by either time windows or by driving hours regulations induced breaks. For example, if departing at time 0 from the depot leads to an arrival time of 2 at the first customer, but the time window opening time at this customer is 5, then a waiting time of 3 is introduced. We call this *unforced* waiting time, because it can be avoided by departing at time 3 from the depot (assuming time-independent travel times in this example). However, if departing from customer i at its latest feasible departure time (i.e., starting its service at its deadline l_i and departing directly after this service) still results in an early arrival at the next customer j , we call this *forced* waiting time. As an illustration of how to profitably introduce unforced waiting time, suppose that direct continuation from a customer results in a total driving time of slightly more than 4.5 hours, which requires an additional 45 minute break before completing the vehicle route. However, if postponing the departure time by a small amount of time (unforced waiting) reduces the total driving time below 4.5 hours (e.g., due to less traffic), then no additional break is required and we end up with an earlier completion.

The problem of exploiting unforced waiting time is that its advantage is difficult to measure, since it requires for each customer addition (or customer insertion, customer removal, etc.) a recheck at each visited customer for introducing unforced waiting time. To keep track of all possibly profitable unforced waiting times is thus computationally expensive. Moreover, the DP framework does not suit well for considering unforced waiting time, since we cannot determine locally whether this waiting time is profitable. Therefore, we consider the variant of the VDO in which introducing unforced waiting time is not considered. This implies that we choose the non-delay property for scheduling departure times. In addition, we choose not to schedule early breaks (which means that we also do not split up breaks in a 15 minute part and a 30 minute part), even not when there is sufficient forced waiting time.

To test the solution quality of the VDO algorithm, we numerically analyze in Section 6.5 the effect of choosing the non-delay property and not considering early breaks. We do this by optimizing the departure times in each constructed vehicle route by solving the ILP model for the VDO in which unforced waiting time and early breaks *are* considered. Note that we optimize each VDO in a post-processing step. Solving the ILP model for the VDO for partial solutions in the DP heuristic may lead to different results. Optimizing such partial solutions may lead to different states to be expanded in the next stage and, therefore, to different vehicle routes. However, computation times for solving the ILP model for the VDO are too large to be applied to such partial solutions. In the next section, we propose a solution method for the TDVRP-EC and for the VDO subproblem in which unforced waiting time and early breaks *are not* considered.

6.4 Solution approach

We solve the TDVRP-EC using the restricted dynamic programming heuristic of Chapter 3. As illustrated in Chapter 4, this heuristic can easily be extended to incorporate complex timing restrictions such as driving hours regulations.

In order to apply the DP heuristic to our problem, we need a method that checks for each state expansion whether there exists a feasible departure schedule for the corresponding partial vehicle route. Furthermore, the costs of such an expansion have to be determined in terms of duty times. In the remainder of this section, we propose a polynomial time algorithm for the VDO without unforced waiting time and early breaks. This VDO algorithm develops a time-dependent duty time function for the entire vehicle route under consideration. We describe how a duty time function based on time-dependent driving speeds can be represented in a duty time record with $O(p)$ elements, with p the maximum number of times the speed changes on a route. Section 6.4.1 describes how to update the duty time record each time a node is added to a partial vehicle route. We show that each such node addition introduces at most $O(p)$ new elements, resulting in $O(np)$ elements for the duty time record of the composite duty time function of an entire route. For simplicity, we first assume that service times are zero, no service time windows are given, and no driving hours regulations are present. Section 6.4.2 and 6.4.3 then, respectively, describe how service times and time windows can be incorporated whilst maintaining $O(np)$ running time complexity. Section 6.4.4 describes how breaks can be incorporated in order to respect the driving hours regulations. Section 6.4.5 derives the resulting running time complexity of the VDO algorithm.

6.4.1 Adding a node to a partial vehicle route

For simplicity reasons, we assume in this section that service times are zero, and time windows and driving hours regulations do not exist, which implies that driving times equal duty times. However, for reasons of generality, we set up an algorithm at the end of this section that remains valid when time windows are present.

Suppose that the number of speed changes on each route between two nodes is limited by p . These speed changes result in a piecewise linear duty time function. Figure 6.1^a presents such a speed step function for a route $0 \rightarrow i$ with distance 2, and Figure 6.1^b presents the resulting duty time function for that route as a function of the departure time. Each speed change causes the slope of the duty time function to change at most two times: 1) when the arrival time at node i equals the moment that the speed changes, 2) when the departure time from node 0 equals the moment that the speed changes. For example, the speed change at time 4 causes the slope of the duty time function to change at departure times 3 and 4. Therefore, the number of linear pieces of the duty time function is $O(p)$.

We define the function $T^z(d)$ as the function that gives the duty time needed to visit all nodes in route z for a given departure d from the first node in route z . Each duty time function $T^z(d)$ of a route z can be represented by a duty time record $r^z = (r_1^z, \dots, r_{U^z}^z)$ of $O(p)$ elements, in which U^z indicates the number of speed changes of the duty time function of route z . Each record entry $r_u^z = (d_u^z, t_u^z)$ contains two elements: the start time d_u^z of the u -th linear piece of the duty time function and the initial height of this piece (i.e., the duty time t_u^z required to (completely) travel route z when departing at time d_u^z from the first node in route z). We assume that for each route $i \rightarrow j$, the travel speeds are given for the entire planning horizon, i.e., for the depot opening hours $[e_0, l_0]$. Therefore, for each route $i \rightarrow j$ we have $d_0^{i \rightarrow j} = e_0$ and $d_{U^{i \rightarrow j}}^{i \rightarrow j} + t_{U^{i \rightarrow j}}^{i \rightarrow j} = l_0$. This allows us to construct the duty time records for each route $i \rightarrow j$ in a pre-processing step without knowing the actual nodes that will be visited before arriving at node i in a solution. Note that when time windows are present, departing at e_0 from node i may not make sense, even when we do not consider the nodes that may be visited before node i in a solution. Section 6.4.3 describes how to include time windows at customers in the construction of the duty time records during the pre-processing step, which may then result in $d_0^{i \rightarrow j} > e_0$ and $d_{U^{i \rightarrow j}}^{i \rightarrow j} + t_{U^{i \rightarrow j}}^{i \rightarrow j} < l_0$ for certain routes.

The duty time record for the duty time function in Figure 6.1^b is:

$$r^{0 \rightarrow i} = ((0, 1), (3, 1), (4, 2), (5, 2), (7, 1), (9, 1))$$

The minimum duty time equals the minimum of all duty time entries. The duty time for a given departure time can be calculated by interpolation.

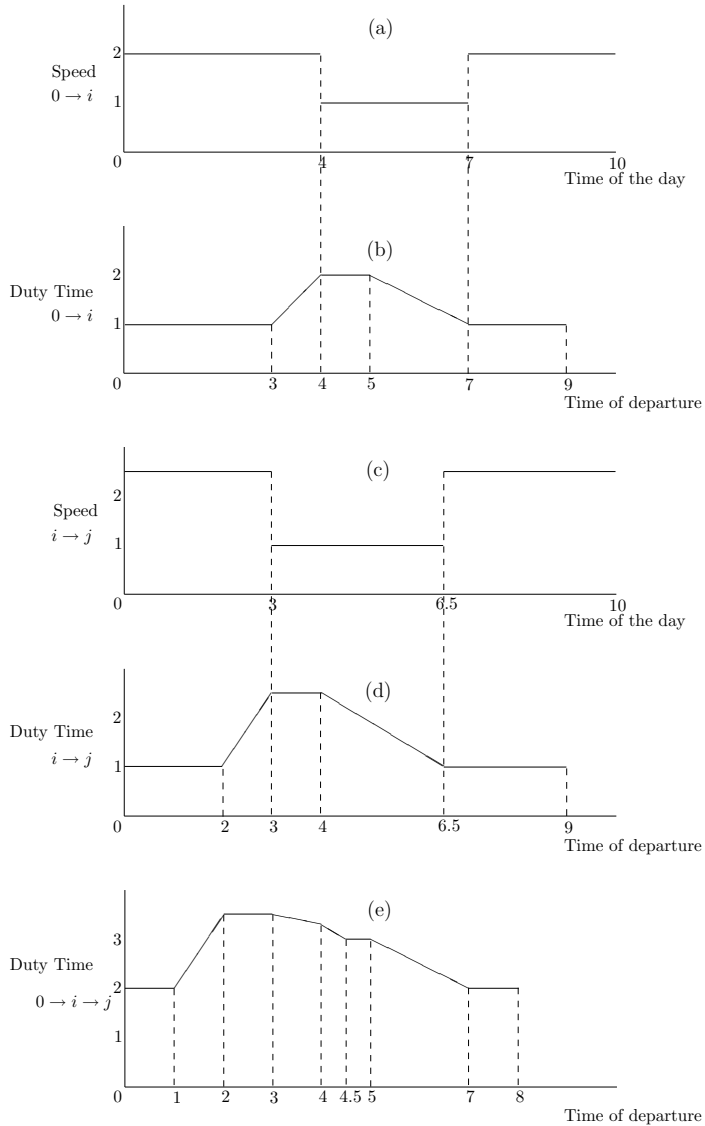


Figure 6.1: (a) speeds route $0 \rightarrow i$, (b) duty times route $0 \rightarrow i$, (c) speeds route $i \rightarrow j$, (d) duty times route $i \rightarrow j$, (e) duty times route $0 \rightarrow i \rightarrow j$.

The duty time for a given arrival time a at the last node in route z can also be calculated using the duty time record r^z . Each departure time d_u^z from the first node in route z results in an arrival time of $a_u^z = d_u^z + t_u^z$ at the last node in route z . This arrival time a_u^z corresponds to a duty time of t_u^z . We can determine the duty time for a given arrival time a at the last visited node in route z by interpolation. We define the function $F^z(a)$ as the function that gives the departure time d from the first node in route z that exactly results in an arrival time of a at the last node in route z (i.e., the difference between arrival time a at the last node in route z and the corresponding duty time). A call to this function requires a run through the duty time record. However, the calls we make in Algorithm 1 are with non-decreasing arrival times a . Therefore, we only require one run through the duty time record for all calls to $F^z(a)$ in Algorithm 1. We can do this by storing for each call the required positions in the duty time record to calculate $F^z(a)$, and to continue the search from these positions for the successive call.

We now describe how to derive a new duty time record when a node is added to the end of a partial vehicle route. Suppose that we add a node j to the end of a partial vehicle route corresponding to a state (S, i) , i.e., route $i \rightarrow j$ is added to the partial vehicle route. Then, we need to determine the duty time record r^{new} of the new partial vehicle route, which is the composite record of the duty time record r^{old} of the old partial vehicle route from node 0 to node i and the duty time record r^{add} of route $i \rightarrow j$. The duty time function of the new route is the composite function of two piecewise linear functions, which in our case is again a piecewise linear function.

Suppose that r^{old} is the duty time record of the duty time function in Figure 6.1^b (i.e., the old partial vehicle route is route $0 \rightarrow i$). Furthermore, suppose that the distance of route $i \rightarrow j$ is 2.5 with a speed step function as in Figure 6.1^c, and resulting duty times as in Figure 6.1^d. Then we get:

$$\begin{aligned} r^{old} &= ((0, 1), (3, 1), (4, 2), (5, 2), (7, 1), (9, 1)) \\ r^{add} &= ((0, 1), (2, 1), (3, 2.5), (4, 2.5), (6.5, 1), (9, 1)) \end{aligned}$$

The earliest feasible departure time from the first node in the new route $0 \rightarrow i \rightarrow j$ equals d_0^{old} (an earlier departure is not possible and departing at this time does not lead to any waiting time at node i). Therefore, $d_0^{new} := d_0^{old} = 0$. This departure time from node 0 results in an arrival time of 1 at node i . Then, departing at node i at time 1 results in an additional duty time of 1 for traveling from node i to node j (since $d_0^{add} = 0$, $d_1^{add} = 2$, and $t_0^{add} = t_1^{add} = 1$), which results in a total duty time for route $0 \rightarrow i \rightarrow j$ of $t_0^{new} := 2$. Next, we need to determine the first departure time from node 0 after d_0^{new} at which the slope of the duty time function of the new route changes. This happens at $\min\{d_1^{old}, F^{old}(d_1^{add})\}$. We have $d_1^{old} = 3$ and $F^{old}(d_1^{add}) = F^{old}(2) = 1$. Therefore, $d_1^{new} := 1$ with corresponding duty time $t_1^{new} := 2$.

We continue this process, each time determining which departure time is the first to change the slope of the duty time function and calculating the corresponding duty time. This process continues until either $d_{U^{old}}^{old}$ or $F^{old}(d_{U^{add}}^{add})$ has been added. This leads to:

$$r^{new} = ((0, 2), (1, 2), (2, 3.5), (3, 3.5), (4, 3.3), (4.5, 3), (5, 3), (7, 2), (8, 2))$$

Figure 6.1^e presents the duty time function of the new route.

Algorithm 1 describes a general procedure for determining the composite duty time record r^{new} of the duty time records of the old route r^{old} and the route to be added r^{add} . Recall that when time windows are present, d_0^{add} does not need to be equal to 0. We already account for such cases in Algorithm 1. Note that $F^{old}(a)$ is only defined for the interval $[d_0^{old} + t_0^{old}, d_{U^{old}}^{old} + t_{U^{old}}^{old}]$. We now describe the steps of the algorithm.

In the initialization, we abort if no feasible departure time from the first node in the new route exists (Line 1 to 3). Next, we check whether departing at the latest feasible departure time from the first node in the old route, i.e. $d_{U^{old}}^{old}$, still results in an early arrival at the first node of the route to be added (Line 4). If this is the case, then the only feasible departure time from the first node in the new route without *unforced* waiting time is $d_{U^{old}}^{old}$. The duty time is then the difference between the earliest completion time at the last node in the new route (which equals $d_0^{add} + t_0^{add}$) and the latest feasible departure time from the first node in the new route (Line 5 and 6). For the remainder, we know that there are multiple feasible departure times without unforced waiting time from the first node in the new route. The earliest of such departure times is either d_0^{old} or $F^{old}(d_0^{add})$ (Line 9 to 13). Note that we cannot use $F^{old}(d_0^{add})$ in the check in Line 9, since it is not defined when $d_0^{old} + t_0^{old} > d_0^{add}$. The duty time t_0^{new} is equal to the sum of the duty time needed for visiting the nodes in the old route and the duty time needed for visiting the nodes in the route to be added (Line 14). The next step is to determine the latest feasible departure time from the first node in the new route (Line 15 to 19). This departure time equals either $d_{U^{old}}^{old}$ or $F^{old}(d_{U^{add}}^{add})$. The final step in the initialization is to initialize v and u^{add} (Line 21 and 22). Index v represents the index of the current entry in r^{new} . Index u^{add} is the index of the entry in r^{add} that contains the earliest departure time from the first node in the route to be added that requires a new record entry for r^{new} (i.e., when departing later than d_v^{new} from the first node in the new route, arrival time $d_{u^{add}}^{add}$ at the first node in the route to be added is the earliest arrival time at this node that changes the slope of the duty time function of the new route).

The main procedure adds record entries to r^{new} for each change in the slope of the duty time function of the new route until an entry with departure time d_{max}^{new} is added. A later departure time than d_v^{new} may cause a change in the slope of the duty time function of the new route both because of a change

Algorithm 1 VDO algorithm

```

// Initialization
1: if  $d_0^{old} + t_0^{old} > d_{U^{add}}^{add}$  then
2:   STOP
3: end if
4: if  $d_{U^{old}}^{old} + t_{U^{old}}^{old} < d_0^{add}$  then
5:    $d_0^{new} \leftarrow d_{U^{old}}^{old}$ 
6:    $t_0^{new} \leftarrow d_0^{add} + t_0^{add} - d_{U^{old}}^{old}$ 
7:   STOP
8: end if
9: if  $d_0^{old} + t_0^{old} \geq d_0^{add}$  then
10:   $d_0^{new} \leftarrow d_0^{old}$ 
11: else
12:   $d_0^{new} \leftarrow F^{old}(d_0^{add})$ 
13: end if
14:  $t_0^{new} \leftarrow T^{old}(d_0^{new}) + T^{add}(d_0^{new} + T^{old}(d_0^{new}))$ 
15: if  $d_{U^{old}}^{old} + t_{U^{old}}^{old} \leq d_{U^{add}}^{add}$  then
16:   $d_{max}^{new} \leftarrow d_{U^{old}}^{old}$ 
17: else
18:   $d_{max}^{new} \leftarrow F^{old}(d_{U^{add}}^{add})$ 
19: end if
20:  $v \leftarrow 0$ 
21:  $u^{add} \leftarrow 0$ 

// Main procedure
22: while  $d_v^{new} < d_{max}^{new}$  do
23:   $u^{old} \leftarrow \arg \min_u \{d_u^{old} \mid d_u^{old} > d_v^{new}\}$ 
24:  while  $d_{u^{add}}^{add} \leq d_v^{new} + T^{old}(d_v^{new})$  do
25:     $u^{add} \leftarrow u^{add} + 1$ 
26:  end while
27:   $v \leftarrow v + 1$ 
28:  if  $d_{max}^{new} + T^{old}(d_{max}^{new}) \geq d_{u^{add}}^{add}$  then
29:     $d_v^{new} \leftarrow \min \{d_{u^{old}}^{old}, F^{old}(d_{u^{add}}^{add})\}$ 
30:  else
31:     $d_v^{new} \leftarrow d_{u^{old}}^{old}$ 
32:  end if
33:   $t_v^{new} \leftarrow T^{old}(d_v^{new}) + T^{add}(d_v^{new} + T^{old}(d_v^{new}))$ 
34: end while

```

in the slope of the duty time function of the ‘old’ part of the new route and because of a change in the slope of the duty time function of the ‘added’ part of the new route. Therefore, we determine the earliest departure time from the first node in the old route later than d_v^{new} that changes the slope of the duty time function of the old route (Line 23) and we determine u^{add} (Line 24 to 26). Next, we increase index v (Line 27), and we determine d_v^{new} (Line 28 to 32). Note that we have to be careful again with the usage of $F^{old}(a)$. If $d_{max}^{new} + T^{old}(d_{max}^{new}) < d_{u^{add}}^{add}$, then $F^{old}(d_{u^{add}}^{add})$ is not defined. When this situation appears, only departure times corresponding to $d_{u^{old}}^{old}$ will be added until $u^{old} = U^{old}$.

6.4.2 Incorporating service times

Service times can be incorporated by adding them to the driving times. Since service times are constant, they do not affect any of the calculations described before. What typically happens is that the duty time function for a route $i \rightarrow j$ is shifted up and to the left by the service time at node i . By doing this, the duty times include both driving times and service times.

6.4.3 Incorporating time windows

Suppose we have a route $i \rightarrow j$ with corresponding duty time function (e.g., as in Figure 6.1^d), and given time windows $[e_i, l_i]$ and $[e_j, l_j]$ for starting service at node i and node j , respectively. For ease of explanation, we again assume that service times are zero. Then, three cases may appear.

Case 1: $e_i + T^{i \rightarrow j}(e_i) > l_j$. In that case, the route $i \rightarrow j$ is infeasible, since the time window opening time at node i is already too late to arrive ultimately at l_j at node j .

Case 2: $l_i + T^{i \rightarrow j}(l_i) < e_j$. This means that, even if we start service at node i as late as possible, we arrive before the time window opening time at node j . In this case, the only way to avoid introducing *unforced* waiting time is to start serving node i as late as possible, implying one feasible departure time from node i : l_i . The corresponding duty time is equal to the travel time plus the forced waiting time: $T^{i \rightarrow j}(l_i) + (e_j - (l_i + T^{i \rightarrow j}(l_i))) = e_j - l_i$.

Case 3: the interval of possible arrival times at node j intersects with $[e_j, l_j]$. We then restrict the feasible departure times from node i to the interval in which we arrive in time at node j (i.e., before or at l_j) and we do not introduce unforced waiting time (i.e., we do not arrive before e_j). This implies that for the earliest feasible departure time from node i without unforced waiting time at node j we get $d_0^{i \rightarrow j} := e_i$ if $e_i + T^{i \rightarrow j}(e_i) \geq e_j$, and $d_0^{i \rightarrow j} := F^{i \rightarrow j}(e_j)$ otherwise. Furthermore, we get $d_{U^{i \rightarrow j}}^{i \rightarrow j} := l_i$ if $l_i + T^{i \rightarrow j}(l_i) \leq l_j$, and $d_{U^{i \rightarrow j}}^{i \rightarrow j} := F^{i \rightarrow j}(l_j)$ otherwise.

Suppose in our example node i has a time window $[2, 9]$ and node j has a time window $[6, 10]$ and the duty time record without time windows is:

$$r^{i \rightarrow j} = ((0, 1), (2, 1), (3, 2.5), (4, 2.5), (6.5, 1), (9, 1))$$

The time window at node i causes the feasible departure time interval to be restricted to $[2, 9]$, such that:

$$r^{i \rightarrow j} := ((2, 1), (3, 2.5), (4, 2.5), (6.5, 1), (9, 1))$$

Next, the time window at node j causes that departing from node i earlier than time 3.5 will result in unforced waiting time at node j , resulting in:

$$r^{i \rightarrow j} := ((3.5, 2.5), (4, 2.5), (6.5, 1), (9, 1))$$

Figure 6.2^a presents the resulting duty time function.

We construct the duty time records for each route between two nodes in this way during the pre-processing step. Then, we apply Algorithm 1 again to obtain the duty time records for the (partial) vehicle routes. Note that the time windows may substantially reduce the number of record entries. In the extreme case, only one feasible departure time remains, which implies that there is forced waiting time on the route and continuing ASAP is the best we can do in the remainder. Figure 6.2^b presents the duty time function of the new route in our example. The number of record entries reduces from 9 to 7.

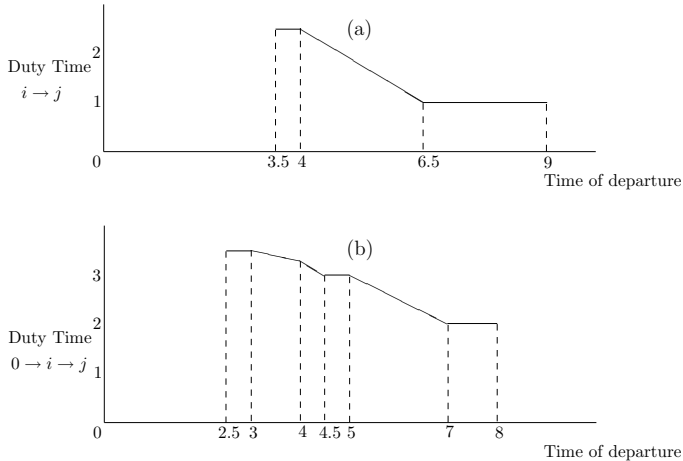


Figure 6.2: duty time records with time windows: (a) duty times route $i \rightarrow j$, (b) duty times route $0 \rightarrow i \rightarrow j$

6.4.4 Scheduling breaks

To comply with the EC social legislation, we have to schedule a 45 minute break whenever the accumulated driving time of a partial vehicle route is about to exceed 4.5 hours. To account for the accumulated driving time, we add an element ta_u^z to each duty time record entry r_u^z , indicating the total accumulated driving time in route z since the last break taken at a customer. Note that the accumulated driving time depends on the chosen departure time d_u^z from the first node in route z . Therefore, we have to account for the accumulated driving time ta_u^z for each departure time d_u^z from the first node in route z . For simplicity reasons, we again assume all service times to be zero. Since we only allow breaks at customer sites, the values of $ta_u^{i \rightarrow j}$ for the duty time records of each route $i \rightarrow j$ equal the driving times $t_u^{i \rightarrow j}$.

We assume that driving times between node pairs do not exceed 4.5 hours. In case a route $i \rightarrow j$ has a departure time that results in more than 4.5 hours of driving time, we assume this route is infeasible. Note that such a route is very unlikely to be selected in a good VRP solution, since the shortest vehicle route in such a solution would be the tour $depot \rightarrow i \rightarrow j \rightarrow depot$ and the total driving time in this tour is likely to exceed its maximum of 9 hours. Within the problem instances used for the computational experiments in Section 6.5 the driving time between each pair of nodes and for each departure time does not exceed 4.5 hours. If VRPs with a long planning horizon are considered, or VRPs with only few customers per vehicle, then it might become necessary to include also routes between two nodes exceeding 4.5 hours of driving time. This can be done by, e.g., modeling parking lots along such routes, or by assuming that breaks can be taken anywhere along the routes. These model assumptions do not affect the algorithmic framework, they only affect the calculation of the duty time records.

Now, suppose we add a node j to a partial vehicle route represented by a state (S, i) , again with duty time records r^{old} , r^{add} , and r^{new} for the duty time functions of the old route, the route to be added, and the new route, respectively. Let \tilde{r}^{new} be the duty time record of the new route in which we ignore that a break may have to be scheduled at node i . Each record entry \tilde{r}_u^{new} contains a departure time \tilde{d}_u^{new} , a corresponding duty time \tilde{t}_u^{new} , and a corresponding accumulated driving time \tilde{ta}_u^{new} without a possibly needed break at node i . We can derive \tilde{r}^{new} by applying Algorithm 1 in which we can calculate each \tilde{ta}_u^{new} in a similar way as how we calculate each \tilde{t}_u^{new} . Then, three cases may appear:

1. After adding route $i \rightarrow j$, $\tilde{ta}_u^{new} \leq 4.5$ for all $u = 0, \dots, U^{new}$.
2. After adding route $i \rightarrow j$, $\tilde{ta}_u^{new} > 4.5$ for all $u = 0, \dots, U^{new}$.
3. After adding route $i \rightarrow j$, $\tilde{ta}_u^{new} > 4.5$ for some, but not all $u = 0, \dots, U^{new}$.

In Case 1, we do not need to schedule a break for any feasible departure time and we get $r^{new} = \tilde{r}^{new}$. We describe the other two cases in detail.

In Case 2, a break is required at node i regardless of the departure time from the first node in the old route, since we assume that breaks are only taken at customers. With this break, the departure time from node i is delayed by 45 minutes. The same procedure as in Algorithm 1 can be applied to determine the duty times of the new route, but with 45 minutes added to all duty times in r^{old} . Since a break is taken at node i , such that the accumulated driving time is reset to 0 when departing from node i , all ta_u^{new} are set to $t_u^{i \rightarrow j}$.

In Case 3, we have to split the new duty time record, such that for each partial duty time record either a break is scheduled at node i for each departure time, or no break is scheduled for any departure time. Therefore, we first determine the series of departure times δ_w at which the new duty time record should be split. This is the case if departure time δ_w results in exactly 4.5 hours of accumulated driving time (when no break is scheduled at node i), while departing directly before or directly after δ_w results in more than 4.5 hours of accumulated driving time. Suppose that u_w is such that $\tilde{d}_{u_w}^{new}$ is the earliest departure time larger than δ_w (if $\delta_w = \tilde{d}_{U^{new}}^{new}$, we set $\tilde{d}_{u_w}^{new} := \delta_w$). Then, each departure time δ_w results in exactly 4.5 hours of accumulated driving time, while $\tilde{t}_{u_w-1}^{new} > 4.5$ or $\tilde{t}_{u_w}^{new} > 4.5$. This leads to a series of strictly increasing departure times $\{\delta_1, \dots, \delta_{W^{new}}\}$ at which the new duty time record should be split. Let's set $\delta_0 := \tilde{d}_0^{new}$ and $\delta_{W^{new}+1} := \tilde{d}_{U^{new}}^{new}$. Then, we split the duty time record of the new route in duty time records r^{new_w} , $w = 0, \dots, W^{new}$ with earliest and latest departure times δ_w and δ_{w+1} , respectively. Now, for each duty time record r^{new_w} either Case 1 applies, such that we follow the procedure described in Case 1 for this duty time record, or we follow the procedure described in Case 2. There is one exception: when $\tilde{t}_{u_w-1}^{new} > 4.5$ and $\tilde{t}_{u_w}^{new} > 4.5$. In that situation, we apply the procedure described in Case 2 to the departure intervals $[\delta_{w-1}, \delta_w]$ and $[\delta_w, \delta_{w+1}]$. However, we also have to consider departing exactly at δ_w without scheduling a break at node i . We resolve this by creating an additional duty time record with only one feasible departure time (δ_w) for which Case 1 applies.

For example, suppose a node k is added to the route $0 \rightarrow i \rightarrow j$ presented in Figure 6.2^b. Furthermore, suppose that all service times are 0 such that the duty times in Figure 6.2^b equal the accumulated driving times. Finally, suppose that the travel time from node j to node k is 1.5 hours, independent of the time of departure. Then, for departure times 2.5 until 4.5 from node 0, the accumulated driving times exceed 4.5 hours. This results in 2 duty time records with departure intervals $[2.5, 4.5]$ and $[4.5, 8]$, respectively. For the first interval we have to apply the procedure described in Case 2, for the second interval we have to apply the procedure described in Case 1. Figure 6.3^a and 6.3^b present the resulting duty times and accumulated driving times, respectively.

Note that, for example, departing at time 4 from node 0 leads to a later

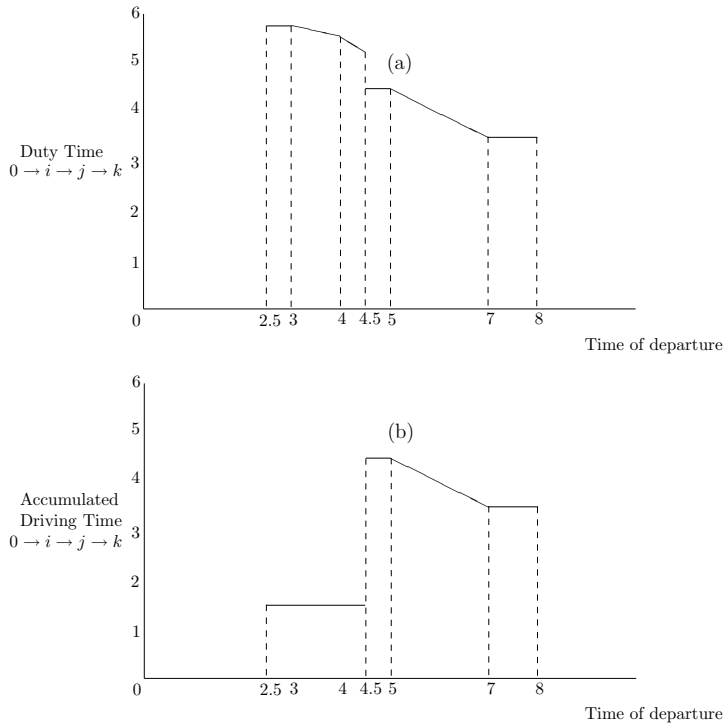


Figure 6.3: duty time records with time windows and breaks: (a) duty times route $0 \rightarrow i \rightarrow j \rightarrow k$, (b) accumulated driving times route $0 \rightarrow i \rightarrow j \rightarrow k$

arrival time at node k than departing at time 4.5. Time windows might allow departure at time 4.5, but not at time 4. Therefore, there might be gaps between succeeding feasible departure intervals.

To account for the total driving time available for each day, we add an element to each duty time record entry accounting for the total accumulated driving time over the entire route. If this element exceeds the total available driving time of 9 hours for a certain departure time, then we determine a similar series of departure times as described in Case 3 above. However, the intervals corresponding to total accumulated driving times exceeding 9 hours are left out of consideration, thereby possibly introducing gaps between departure intervals. We follow a similar strategy for the total duty times, such that non-feasible departure times are left out of consideration.

6.4.5 Running time complexity

The procedure for adding the breaks increases the running time complexity of the VDO algorithm. To derive this complexity, it is crucial to know how many breaks could maximally be scheduled in a route for a certain departure time from the first node in that route. In Appendix B, we derive that this number equals 4. We now derive how many additional duty time record entries each break might introduce.

Suppose that after adding a route $i \rightarrow j$ to a partial solution we would have $\tilde{t}a_u^{new} < 4.5$ for some entry \tilde{r}_u^{new} and $\tilde{t}a_{u+1}^{new} > 4.5$ for the next entry \tilde{r}_{u+1}^{new} . Then, the break requirement introduces two duty time record entries ($r_{U^{new_w}}^{new_w}, r_0^{new_w+1}$) for two successive duty time records r^{new_w} and r^{new_w+1} , both with the same departure time, but with different duty times and accumulated driving times. The first entry $r_{U^{new_w}}^{new_w}$ represents the case where no break is scheduled at node i , while the second entry $r_0^{new_w+1}$ represents the case where a break is scheduled at node i . Suppose next that $\tilde{t}a_{u+2}^{new} < 4.5$. Then, again the break requirement introduces two duty time record entries: $r_{U^{new_w+1}}^{new_w+1}$ and $r_0^{new_w+2}$. When another node is added to the route, a similar procedure may apply to the successive record entries ($r_{U^{new_w-1}}^{new_w}, r_{U^{new_w}}^{new_w}$) and the successive record entries ($r_0^{new_w+2}, r_1^{new_w+2}$). In the worst case, each node addition results in four new duty time record entries caused by the break requirement for the original duty time record entries \tilde{r}_u^{new} and \tilde{r}_{u+1}^{new} , because of ascending (descending) $\tilde{t}a_u^{new}$ that cross the 4.5 hours driving limit. Since there are at most $n + 1$ node additions per vehicle route, this leads to at most $2(n + 1)$ additional entries for the original entry \tilde{r}_u^{new} (and $2(n + 1)$ additional entries for the original entry \tilde{r}_{u+1}^{new}).

Since the number of existing entries without considering breaks is $O(np)$, the total number of entries with at most one break scheduled is $O(n^2p)$. The same procedure applies for each additional break, i.e., introducing at most $2n$ entries for each existing entry. Therefore, given that at most 4 breaks will be scheduled for each departure time, the running time complexity of the algorithm with scheduling breaks is $O(n^5p)$.

6.5 Computational experiments

In this section, we test the solution approach described in Section 6.4. We ran our experiments on a PC with a Core 2 Quad, 2.83 GHz CPU and 4 GB of RAM. Section 6.5.1 describes our test instances, Section 6.5.2 describes our test approach, and Section 6.5.3 presents the results.

6.5.1 Test instances

To test our heuristic, we use a modification of the set of benchmark instances for the VRPTW with time-dependent travel times proposed by Figliozzi (2009). These Figliozzi benchmark instances are themselves modifications of the well-known Solomon (1987) benchmark instances for the VRPTW. We selected these instances, because the Solomon benchmarks are standard reference in the VRP literature and they represent an extensive set of VRPTW instances with various characteristics. Moreover, Figliozzi's modification of the Solomon instances for the VRPTW with time-dependent travel times is - to the best of our knowledge - the only set of benchmark instances available in the literature for this type of problem. Below we explain both (Figliozzi's and our) modifications with respect to the Solomon instances.

Figliozzi proposed the following modification of the Solomon instances to make them applicable to the VRPTW with time-dependent travel times. The opening hours of the depot ($[e_0, l_0]$) are divided in 5 equally spread time intervals. The first and the last time interval correspond to the morning and evening peak with a reference speed of 1.00. In the remaining intervals, the speeds are higher. Figliozzi proposed the following three speed patterns, representing traffic congestion during the peak hours to an increasing extent:

$$\begin{aligned} TD1 &= [1.00, 1.60, 1.05, 1.60, 1.00] \\ TD2 &= [1.00, 2.00, 1.50, 2.00, 1.00] \\ TD3 &= [1.00, 2.50, 1.75, 2.50, 1.00] \end{aligned}$$

We add one speed pattern ($TD0$) in which speeds are constant (1.00) over the day.

Since these benchmarks do not include driving hours regulations, we modify them for the TDVRP-EC as follows. We assume that the opening hours of the depot correspond to a working day of 12 hours: from 7AM until 7PM. With Figliozzi's speed patterns, this implies that the morning and evening peak last from 7AM until 9:24AM and from 4:36PM until 7PM, respectively. To obtain these depot opening hours, we scale the time windows and travel distances in each problem instance. In summary, the resulting problem instances for the TDVRP-EC consist of the scaled modified Solomon instances with the speed patterns proposed by Figliozzi, and the EC social legislation on driving and working hours. We refer to this test set as Set 1.

The speed patterns in Set 1 do not allow driving before the morning peak or after the evening peak. Moreover, since the depot is open for 12 hours, the EC regulation on daily duty times - which restricts daily duty times to 13 hours - is always satisfied. In order to quantify the benefits of allowing travels before the morning peak and after the evening peak, we propose a second

test set in which driving before and after the morning peak is possible, and for which the EC regulation on daily duty times can be restrictive. For this purpose, we introduce Set 2 in which we extend the depot opening hours to 16 by advancing the opening time by 2 hours and by postponing the closing time by 2 hours. The speeds during these new periods represent free-flow speeds before the morning peak and after the evening peak, respectively. Therefore, we set the speed during these periods to the maximum speed for each speed pattern, i.e., we get the following speed patterns:

$$TD0' = [1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00]$$

$$TD1' = [1.60, 1.00, 1.60, 1.05, 1.60, 1.00, 1.60]$$

$$TD2' = [2.00, 1.00, 2.00, 1.50, 2.00, 1.00, 2.00]$$

$$TD3' = [2.50, 1.00, 2.50, 1.75, 2.50, 1.00, 2.50]$$

Note that the first and the last speed last for 2 hours, while the other speeds last for 2.4 hours.

In addition to these extra depot opening hours, we adjust a selection of the customer service time windows in Set 2. If the opening (closing) time of a time window is non-restrictive in the original Solomon instance, we make it also non-restrictive in the new problem instance. This implies that if the opening time in the original Solomon instance equals the opening time of the depot, then we set this opening time accordingly in Set 2. The closing times in the original Solomon instances are integer and they are constructed such that they always allow a direct return to the depot after starting service at this closing time. Therefore, we consider closing times non-restrictive if starting service at this closing time and directly returning to the depot results in an arrival time (after rounding up) equal to the closing time of the depot. In our new test set, we set such closing times equal to the closing time of the depot. We refer to this test set as Set 2. Note that Set 2 is less restrictive than Set 1, since some time windows are increased and the average travel speed is increased (every feasible solution in Set 1 is also a feasible solution in Set 2). However, the EC regulation on daily duty times can be restrictive in Set 2 as opposed to Set 1.

6.5.2 Test approach

Our test approach is as follows. We solve all problem instances twice. Both times we set the primary objective to minimize the number of vehicles used. However, the first time we set the secondary objective to minimize the total travel distance, and the second time we set it to minimize the total duty time. In the remainder we refer to the DP heuristic with minimizing travel distance as secondary objective as DP^{dist} , and we refer to the DP heuristic with minimizing duty time as secondary objective as DP^{duty} . We compare the results of these

two heuristics in terms of all relevant cost factors (number of vehicles, travel distance, duty time).

For both DP heuristics we set $H = 10,000$, which means that in each stage in the DP heuristic only the 10,000 best states are selected to be expanded in the next stage. For this selection procedure, we use the following hierarchical criteria: 1) number of vehicles used, 2) earliest completion time of vehicle route being constructed, 3) secondary objective. We added the secondary cost criterion ‘earliest completion time’, because preliminary tests showed that this criterion has a positive impact on minimizing the number of vehicles used. Within the DP heuristic, the primary criterion ‘number of vehicles used’ starts to play a role when a node representing the depot is about to be added to a state. However, when a customer with a late window opening time is selected, there is little room for adding customers to the end of this partial vehicle route, such that extra vehicles may be needed in the complete solution. Setting the secondary selection criterion to ‘earliest completion time of the partial vehicle route being constructed’ increases the room for adding customers such that less vehicles are needed in the complete solution. In our experiments we set the dominance criteria to these 3 criteria together with the remaining capacity (state A should be at least as good as state B in all 4 criteria to dominate state B).

6.5.3 Test results

Table 6.1 presents the results for the two heuristics on Set 1 in terms of number of vehicles used, total travel distance, total duty time, and the required cpu time (in seconds). DP^{dist} leads to better results than DP^{duty} in terms of travel distance (-4.1%, on average), and in terms of number of vehicles used (-5.7%, on average). The latter result can be explained as follows. If the secondary objective is set to minimize the total duty time, then routes that start late and complete early are preferable. Therefore, customers with either an early or late time window are not preferable with this objective. The first two criteria (number of vehicles used and earliest completion time of the route being constructed) for selecting the H best states in each stage tries to avoid missing such customers, but only for the route that is being constructed. These criteria do not have any effect on the routes that have already been completed in the partial solution. Therefore, for those completed routes only the tertiary criterion plays a role. Since for DP^{duty} this criterion is ‘total duty time’, it is likely that only few customers with either an early or a late time window are in the completed routes in a partial solution. Therefore, such customers have to be selected at a later stage in which they may not combine well and extra vehicles are needed.

The duty times are substantially smaller with DP^{duty} than with DP^{dist} (-3.5%, on average, but against an increase in the number of vehicle routes

of 5.7%, on average). This is of particular interest, since the total duty time defines the total amount of vehicle hours that is needed to serve all customers. Since transport costs are directly related to this amount of vehicle hours, any reduction in duty time leads to costs savings. Note that the computation times are much smaller for the TD0 speed pattern. Speeds are constant with TD0, such that the number of duty time record entries is substantially smaller with this speed pattern (this number is either 1 in case there is forced waiting time along the route, or 2: the earliest and latest feasible departure time without introducing unforced waiting time).

Speed pattern	DP^{dist}				DP^{duty}			
	# veh	dist	duty	cpu(s)	# veh	dist	duty	cpu(s)
TD0	9.18	1294	4992	148	9.34	1314	4860	148
TD1	8.23	1261	4730	397	8.82	1318	4540	397
TD2	7.75	1265	4501	407	8.18	1326	4352	408
TD3	7.48	1258	4413	415	8.18	1330	4228	415
average	8.16	1269	4659	342	8.63	1322	4495	342

Table 6.1: Results of heuristics DP^{dist} and DP^{duty} on problem Set 1

Table 6.2 presents the results for Set 2. Allowing travels before the morning peak and after the evening peak substantially reduces the number of vehicles needed (-4.4% and -3.1% for DP^{dist} and DP^{duty} , respectively). The total travel distance (2.5% and 3.5%, respectively) and total duty time (2.0% and 0.8%, respectively), however, increase.

Computation times are a bit larger for Set 2 than for Set 1. This difference can be explained by the average number of duty time record entries, which is larger for Set 2 than for Set 1. The longer planning horizon in Set 2 allows for more possible departure times for each partial vehicle route. In addition, longer routes are allowed, such that more breaks have to be scheduled.

Speed pattern	DP^{dist}				DP^{duty}			
	# veh	dist	duty	cpu(s)	# veh	dist	duty	cpu(s)
TD0	8.68	1297	5096	161	9.00	1340	4902	160
TD1	7.96	1304	4847	645	8.55	1369	4575	582
TD2	7.45	1298	4556	584	8.11	1370	4389	592
TD3	7.13	1304	4515	612	7.79	1394	4261	618
average	7.80	1301	4753	500	8.36	1368	4532	488

Table 6.2: Results of heuristics DP^{dist} and DP^{duty} on problem Set 2

We also tested the solution quality of the VDO algorithm with the non-delay property. For this purpose, we optimize the departure times of the vehicle routes in the solutions for Set 2 using the ILP model of Chapter 2, which includes unforced waiting times and early breaks. We choose the solutions for Set 2, because in the problem instances in set 2 all restrictions considering one-

day planning (e.g., maximum daily duty time) may play a role. We solved the ILP model with CPLEX 11.0 for each vehicle route obtained from the solutions for Set 2 and compared the minimum duty times with the duty times found by our VDO algorithm. Table 6.3 presents the average improvements in duty time by solving the ILP models.

Speed pattern	DP^{dist} Improvement	DP^{duty} Improvement
TD0	0.29%	0.11%
TD1	0.50%	0.43%
TD2	0.61%	0.34%
TD3	0.28%	0.19%
average	0.42%	0.27%

Table 6.3: Improvements by solving ILP model for the VDO

We observe that the improvements are very small (smaller than 0.5%, on average). The improvements are slightly larger for DP^{dist} . This can be explained by less tight routes when travel distance is the secondary objective than routes when duty time is the secondary objective. For less tight routes it is more likely that there is room for improvement by introducing unforced waiting time. Although the improvements are small on average, there are problem instances for which the average improvement over all routes is more than 3.7%. Therefore, optimizing departure times using the ILP model of Chapter 2 as a post-processing step of solving a TDVRP-EC may lead to substantial cost savings.

6.6 Conclusions

We extended the DP heuristic of Chapter 3 to a solution method for the TDVRP-EC. To the best of our knowledge, this is the first solution approach that considers both time-dependent travel times and driving hours regulations within one vehicle routing model.

We proposed a heuristic for the vehicle departure time optimization problem to estimate the minimum duty time of partial vehicle routes. This heuristic is an exact polynomial-time algorithm for the VDO with the non-delay property and without early breaks. Computational results show that this heuristic finds close to optimal solutions for the VDO.

The DP heuristic is flexible with respect to various extensions of the VRP. Therefore, the solution approach proposed in this chapter can also be applied to those extensions of the VRP. The DP heuristic is also flexible with respect to different objective functions. This is demonstrated with the computational experiments in which duty time minimization as the secondary objective, which

is often considered in practice, is compared with travel distance minimization, which is often considered in the VRP literature. Therefore, this solution approach is very promising for real life vehicle routing problems.

The computational results show that duty time minimization as the secondary objective leads to substantial reductions of duty times, but against the cost of more vehicle routes and longer travel distances. Moreover, the results show that extending the depot opening hours, such that traveling before the morning peak and after the evening peak becomes possible, may result in substantial cost savings.

Chapter 7

An alternative: distributed decision making

In the VRP literature, vehicle routing problems have been generally approached from a central planning perspective. In practice, however, a centralized planning in which all decisions are taken by the planner may not always be realistic. For example, drivers are often responsible for scheduling their own breaks, and in certain cases they also partially decide upon their routes. In such cases, drivers may have their own objective, which may deviate from the company's objective. The question that arises is whether putting a lot of effort in a very detailed centralized planning pays off if drivers change their routes according to their own objectives.

In this chapter¹, we analyze the problem of combined vehicle routing and break scheduling from a distributed decision making perspective. We approach this problem with a hierarchical planning process, in which both planners and drivers are responsible for parts of the planning process. The main question we attempt to answer is whether it pays off to make a very detailed centralized vehicle route plan if subsequently each driver optimizes his own route according to his own objectives.

7.1 Introduction

The problem of combined vehicle routing and break scheduling basically comprises three subproblems: clustering the customer requests, routing, and break scheduling. In practice, these subproblems are distributed among different decision makers: the planner and the drivers. Semi-structured interviews with 5

¹This chapter is based on Meyer et al. (2009a,b)

medium-sized logistic service providers in Germany (Onken, 2009) pointed out that in practice planners are typically responsible for the clustering of the customer requests, while the drivers schedule their breaks. Whether the routing is done by the planner or the driver depends on the application at hand.

In the literature, various strategies have been proposed to deal with the problem of break scheduling within vehicle routing. Some authors propose explicit break scheduling in which the rules of Regulation (EC) No 561/2006 are explicitly considered in the VRP (our DP heuristic of Chapter 4; Goel, 2009; Zäpfel and Bögl, 2008). Others propose implicit break scheduling, in which slack travel time is created by using a lower average speed (e.g., Bartodziej et al., 2009). This latter approach to account for breaks is often used in practice (Onken, 2009). Explicit break scheduling is more complex, but allows for finding better results and guarantees feasible break schedules for the resulting vehicle routes. This trade-off between complexity and solution quality has, to the best of our knowledge, never been analyzed and quantified so far.

To account for a situation in which different decision makers are involved, we analyze the problem of combined vehicle routing and break scheduling from a distributed decision making (DDM) perspective. For this purpose, we apply the framework for DDM proposed by Schneeweiss (2003). In this framework, two types of decision makers are considered: in our case the (central) planner and the drivers. The planner's main task is to determine the customer clusters. His decision results in an instruction to each driver to visit a certain customer set, thereby implying a hierarchical planning framework. In order to account for the fact that the drivers have to find feasible routes and break schedules, the central planner attempts to anticipate the driver's behavior to a certain extent. As we argue in Section 7.3.1, we consider all relevant degrees of anticipation for the problem of combined vehicle routing and break scheduling. The planner's decision is based on the clustering problem and his anticipation of the drivers' behavior.

We assume the planner's objectives to meet the company's objectives: the primary objective is to minimize the number of clusters and the secondary objective is to minimize the total travel distance. We select minimizing travel distance as secondary objective in this chapter, because we consider one-week planning for our analysis. As mentioned previously, minimizing duty times is typically applied to one-day schedules. Travel distance is a good measure for evaluating week schedules and using this objective also suffices for the analysis in this chapter. For similar reasons, we do not include time-dependent travel times in this chapter.

For the drivers' behavior, we consider two different cases. In the first case, the drivers' objective meets the company's secondary objective, e.g., the minimization of the total travel distance. In the second case, which in our opinion better represents practice in many cases, the drivers apply their own objective, e.g., the minimization of the return time. This objective may conflict with

the company's objective. The practical problem of conflicting objectives is another motivation to study the problem of combined vehicle routing and break scheduling from a distributed decision making perspective.

By proposing different anticipation functions of the drivers' behavior, we can compare different strategies for break scheduling, e.g., implicit and explicit break scheduling. We perform computational experiments to quantify the quality of these alternative strategies. Clearly, some of the strategies are more complex (explicit break scheduling) than others (implicit break scheduling). By quantifying the quality of these strategies, we demonstrate the trade-off between complexity and solution quality.

This chapter is organized as follows. In Section 7.2, we discuss the distribution of tasks between planners and drivers within the problem of combined vehicle routing and break scheduling. In Section 7.3, we show how the problem can be embedded into a framework for DDM. We will consider two different settings. In the first setting, the drivers act in a way that fits within the planner's objective, while in the second setting the drivers deviate from the planner's objective, following their own criterion. In Section 7.4, we analyze the impacts of various anticipation functions of the drivers' behavior using computational experiments. We consider both the setting in which drivers' and planner's objectives coincide and the setting in which these objectives conflict. Finally, in Section 7.5, we summarize our main findings.

7.2 Problem description

The problem of combined vehicle routing and break scheduling from a central planning perspective basically involves a VRPTW-EC. From a distributed decision making perspective, it comprises three interconnected partial planning problems: the clustering of customer requests, the routing of vehicles, and the planning of breaks and rest periods. These problems can be solved either simultaneously - which involves solving a VRPTW-EC - or in sequence. Since in our case the tasks are divided among different decision makers, we consider the case in which these problems are addressed sequentially by planners and drivers. First, the planner clusters the customer requests, thereby attempting to minimize the number of vehicles used to serve the customers in order to save costs for the company. His secondary objective is to minimize the total travel distance to serve all customers. The customer clusters are then passed on to the drivers. Within their assigned set of customers, each driver carries out the routing and break scheduling such that the EC social legislation is satisfied on his route. We choose this division of the three partial planning problems, because Onken (2009) indicated that this division is more often used in practice than, e.g., the case in which the planner is also responsible for the routing.

To avoid infeasible problems in the drivers' subsequent planning, the central

planner might take into account the drivers' planning approach when generating customer clusters. This means that when performing their planning task, planners anticipate the routing and break scheduling that will be performed subsequently. However, since the legislation on driving and working hours is rather complex, planners might not anticipate the exact planning process but instead use some simplified approach to anticipate the drivers' planning model. We will propose different degrees of anticipation and analyze their impact on the resulting vehicle schedules.

The drivers may follow different objectives. If they act according to the company's objectives, they will try to minimize the travel distance in order to save costs. Schneeweiss (2003) refers to this situation as the team situation between planner and drivers. However, since the drivers may freely decide on their route, they might also follow their own objectives, even if these are not in line with the company's objectives. For example, instead of minimizing the travel distance, each driver may try to minimize his return time to the depot in order to maximize his leisure time. This hidden action cannot be observed by the planner. We refer to this non-compliant setting as the non-team situation.

7.3 Framework for distributed decision making

The framework for DDM presented by Schneeweiss (2003) was first introduced as a framework for hierarchical planning by Schneeweiss (1995). This framework for hierarchical planning has been applied successfully to investigate problems in various areas, such as production planning (Gfrerer and Zäpfel, 1995), resource planning (Pesenti, 1995), supply chain management (Schneeweiss and Zimmer, 2004), managerial accounting (Eichin and Schneeweiss, 2001), contract design (Schenk-Mathes, 1995), and financial planning (Goedhart and Spronk, 1995).

The framework for DDM presented by Schneeweiss (2003) considers two decision making units (DMUs). In the case of hierarchies in distributed decision making, these DMUs are situated on different levels: the top-level and the base-level. The top-level uses a planning model $M^T(C^T, A^T)$ and takes its decision in such a way that it optimizes its criterion C^T over all possible actions a^T within its decision field A^T . The top-level's criterion consists of a private criterion C^{TT} and a top-down criterion C^{TB} (which depends on the base-level's behavior), i.e., $C^T = \{C^{TT}, C^{TB}\}$. The top-level derives an optimal instruction $IN^* = IN(a^{T*})$ and communicates it to the base-level. Subsequently, the base-level takes its decision based on the top-level's instruction using its planning model $M^B(C^B, A^B)$ such that its criterion C^B is optimized.

To improve the overall planning results, the top-level can try to anticipate the base-level's subsequent planning in order to avoid giving an infeasible instruction or to account for the base-level's influence on the top-down criterion.

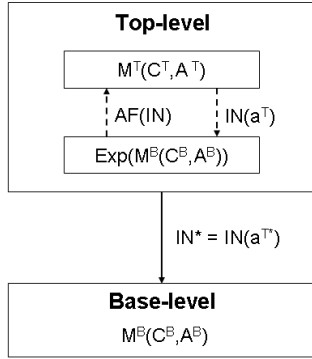


Figure 7.1: Coordination in Hierarchical Systems (Schneeweiss, 2003)

Therefore, the top-level can apply an anticipation function $AF(IN)$, which is a function of the top-level's instruction and gives possible reactions of the expected base-level's behavior. The anticipation function does not need to be a precise representation of the base-level's reaction but can also be an approximation of the expected base-level's planning model $Exp(M^B(C^B, A^B))$. Figure 7.1 depicts this hierarchical coordination structure.

Schneeweiss (2003) distinguishes four different degrees of anticipation: perfect reactive anticipation, approximately perfect reactive anticipation, implicit reactive anticipation, and non-reactive anticipation. Perfect reactive anticipation means that the base-level's planning model is exactly known and it is anticipated by the top-level without any approximations. In the case of approximately perfect reactive anticipation, the base-level's planning model is taken into account only approximately, e.g. by making simplifying assumptions. Implicit reactive anticipation means that only some features of the base model are considered and the anticipation function does no longer explicitly describe the base-level's decision model. These three degrees of anticipation incorporate the base-level's planning behavior as a reaction to the top-level's instruction. In the case of non-reactive anticipation, such an anticipation function does not exist, but only some general features of the base-level may be included in the top-level's criterion.

7.3.1 Planner's and Drivers' Models

In the problem of combined vehicle routing and break scheduling, the planner constitutes the top-level and the drivers constitute the base-level. In the pure top-down hierarchy where the planner does not account for the drivers' planning, the planner's criterion C^T is to minimize the number of vehicles required

to serve all customer requests. Since the planner does not account for the drivers' planning, the planner's criterion only comprises his private criterion, i.e., $C^T = C^{TT}$. The top-down criterion C^{TB} plays a role when the planner anticipates the driver's planning problem, which we describe in Section 7.3.2 and 7.3.3. Without any anticipation of the driver's planning (neither of his planning behavior, nor of his planning objectives, nor of his planning problem, the planner's decision field A^T comprises all possible customer clusters satisfying the capacity restrictions of the vehicles. All other restrictions (time windows, driving hours regulations) relate to the drivers' planning problem, which can only be considered by the planner if he anticipates (to some extent) the drivers' planning problem. Therefore, in the case that the planner does not anticipate the drivers' planning problem, the planner's decision problem results in solving the following assignment problem with capacity restrictions:

$$C^T : \text{Minimize } \sum_{k \in K} z_k \quad (7.1)$$

$$A^T : \sum_{k \in K} x_{ik} = 1, \quad \forall i \in V \quad (7.2)$$

$$Qz_k \geq \sum_{i \in V} q_i x_{ik}, \quad \forall k \in K \quad (7.3)$$

where

$V = \{1, \dots, n\}$: set of customers, index i

$K = \{1, \dots, m\}$: set of vehicles, index k

$x_{ik} \in \{0, 1\}$: takes value 1 iff customer i is served by vehicle k

$z_k \in \{0, 1\}$: takes value 1 iff vehicle k is used

Q : capacity of one vehicle

q_i : demand of customer i

After deriving the customer clusters $CL_k \subseteq \{1, \dots, n\}$ for each vehicle k , the planner passes them on to the drivers who constitute the base-level. Therefore, the instruction equals the customer clusters, i.e., $IN^* = \{CL_k^* | k \in K\}$, with $CL_k^* = \{i \in V | x_{ik}^* = 1\}$ and x_{ik}^* the optimal assignments in (7.1) - (7.3).

The drivers have to perform the routing and break scheduling within their customer clusters. Therefore, each driver k has to solve a traveling salesman problem with time windows and EC social legislation (TSPTW-EC) for his customer cluster CL_k^* . A mathematical description of this problem can be found in Kopfer and Meyer (2009) who propose a position based ILP-formulation for the TSPTW-EC.

When performing the routing and break scheduling, we assume that drivers try to exploit the *modified* rules of the EC social legislation in order to better

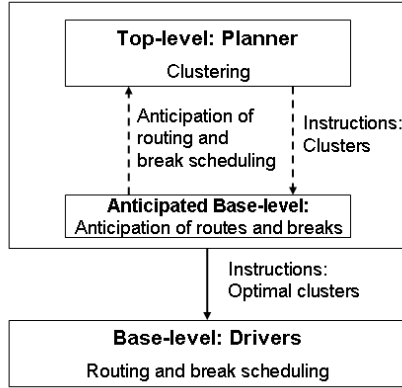


Figure 7.2: Decision structure of the DDM problem

fulfill their objectives. This results in the decision field A^B , which comprises the set of all possible routes for the customer clusters CL_k , such that the customer time windows are met and the legislation is fulfilled, including all modified legal rules.

Clearly, for a given set of customers derived by the planner, A^B could be empty (e.g., in (7.1) - (7.3) the time windows are not considered). In order to avoid infeasible customer clusters, the planner anticipates the drivers' planning behavior through an anticipation function $AF(IN)$. We propose three such anticipation functions below. We only consider reactive anticipation, since non-reactive anticipation does not make sense in this case (in case of non-reactive anticipation, the time windows and the EC social legislation are not considered, such that A^B is very likely to be empty). Figure 7.2 depicts the entire planning situation arising in combined vehicle routing and break scheduling.

7.3.2 Team Situation

If we assume a team situation between planner and drivers, then the planner rightfully expects each driver k to use the criterion of minimizing the total travel distance within their customer cluster:

$$Exp(C^B) = Min \sum_{i \in CL_k \cup \{0\}} \sum_{j \in CL_k \cup \{n+1\}} d_{ij} x_{ij}^k$$

where

$i \in \{0, n+1\}$: depot

$x_{ij}^k \in \{0, 1\}$: takes value 1 iff vehicle k serves customer j directly after customer i

d_{ij} : travel distance from customer i to customer j

Now, the planner is able to account for the drivers' criterion. The planner's private criterion C^{TT} is to minimize the number of vehicles needed. If there are different customer clusters resulting in the minimal number of vehicles, he will select the best ones using his top-down criterion C^{TB} , which is to minimize the expected total travel distance. However, to estimate the total travel distance, he needs to anticipate the routing that is performed by the drivers. We propose the following anticipation functions for the three relevant degrees of anticipation.

1. Perfect reactive anticipation. In this case, the planner considers the full planning model used by the drivers, which means that he expects each driver to solve a full TSPTW-EC, i.e., $Exp(M^B) = M^B$. By anticipating this driver model, the planner is able to instruct customer clusters that allow feasible routes and break schedules for the drivers.
2. Approximately perfect reactive anticipation. In this case, the planner simplifies the drivers' planning model. We propose to do this by leaving out the modified rules of the legislation. This means that the planner's anticipated base model is reduced to a TSPTW-EC without the modified rules. By including only the basic rules of the legislation, the planner can still guarantee that the clusters he instructs to the drivers allow for feasible routes and break schedules.
3. Implicit anticipation. In this case, the planner only considers some features of the base model. We model this planning approach by assuming that the planner does not explicitly consider the task of break scheduling. However, since the planner knows that the drivers require breaks and rest

periods, he considers a lower travel speed than the average travel speed in order to create slack travel time that can be used to schedule breaks and rest periods. Consequently, the anticipated base model is a basic TSPTW.

7.3.3 Non-Team Situation

In practice, the drivers' objective may deviate from the company's objective, which results in a principal-agent setting. In this case, the drivers follow their own objective C'^B , instead of following the company's objective, represented by C^{TB} . To model this case, we assume that instead of minimizing the travel distance, the drivers try to minimize their return time in order to finish their duty as early as possible:

$$C'^B = \text{Min } t_{return}^k,$$

where t_{return}^k is the return time to the depot of driver k . In this case, we assume that the drivers' planning behavior is not correctly observable by the planner. The drivers' objective differs from the planner's anticipated base criterion, implying $\text{Exp}(C^B) \neq C'^B$. However, since this is a situation in which the drivers have some hidden action, the planner cannot account for the drivers' behavior correctly and will still use the minimization of the travel distance as the anticipated base criterion. Therefore, in this situation the planner's model M^T and also his anticipation functions $AF(IN)$ are maintained. Moreover, the drivers' decision space A^B still comprises all vehicle routes within their assigned customer clusters such that the EC social legislation is fulfilled. Applying the new base criterion, the base model M^B changes to $M'^B = (C'^B, A^B)$.

7.4 Computational experiments

We conduct various computational experiments to quantify the impacts of the three anticipation functions. This quantification allows us to compare different strategies to schedule breaks and rest periods within vehicle routing. We test both the team- and the non-team situation. To solve the planner's and drivers' problems, we use the following approach.

In all scenarios, each driver has to solve a TSPTW-EC. Only the objective is depending on the team character of the situation considered. The planner's problem is a clustering problem, in which the decision space A^T is restricted by the anticipation function of the drivers' behavior; also the top-down criterion C^{TB} is estimated through the anticipation function. For each degree of anticipation, we describe the resulting problem that is to be solved. After describing the different problems, we describe the solution algorithm.

In the case of perfect anticipation, while minimizing the number of customer clusters, the planner expects each driver to solve a TSPTW-EC with the objective of minimizing the travel distance. The planner's problem can be addressed by solving a VRPTW-EC with minimizing the number of vehicles as the primary objective. All rules of the EC social legislation are considered. The planner's secondary objective is to minimize the total travel distance.

In the case of approximately perfect reactive anticipation the planner also considers a TSPTW-EC for each driver. However, he ignores the modified rules. Therefore, the planner's problem is a VRPTW-EC without considering the modified rules of the EC social legislation.

With implicit anticipation the planner considers a TSPTW for each driver, but with driving time estimations based on a lower travel speed than the average travel speed. Therefore, in this case the planner's problem can be solved by solving a VRPTW with minimizing the number of vehicles as the primary objective and minimizing the travel distance as the secondary objective.

We solve all problems (VRPTW-EC, VRPTW-EC without modified rules, VRPTW, TSPTW-EC) with the restricted dynamic programming (DP) heuristic of Chapter 4. We use this approach, since the DP heuristic can solve all problem types and is currently the only heuristic available to solve the VRPTW-EC with the full EC social legislation. In Chapter 4 we described how to solve the VRPTW-EC and the VRPTW-EC without modified rules with this DP heuristic. Furthermore, to solve a VRPTW we can simply relax all break scheduling constraints by setting the allowed accumulated driving and working times very high (e.g., to the time horizon of the problem instance). Finally, since the VRPTW-EC is a generalization of the TSPTW-EC, the DP heuristic can also solve the TSPTW-EC.

We test the different anticipation functions on the benchmark instances proposed by Goel (2009), who adjusted the well-known Solomon benchmark instances (Solomon, 1987) for the VRPTW to the VRPTW-EC. For each problem instance and anticipation function, we first solve the planner's problem as described above and then the resulting drivers' problem for each driver. We ran our experiments on a Pentium 4, 3.40GHz CPU and 1.00 GB of RAM and set $H = 10,000$ for all experiments.

7.4.1 Team Situation

We first describe the results of the team situation in which the drivers' objective is to minimize his travel distance. For perfect and approximately perfect anticipation the planner's solution results in feasible vehicle routes for the drivers. We assume that a driver only changes the route and break schedule found by the planner if the driver finds a better route and break schedule in terms of his objective.

Table 7.1 presents the results on perfect and approximately perfect anticipation. The first column indicates the different problem sets and, between brackets, the number of problem instances. Next, the results on the situation with perfect anticipation and the situation with approximately perfect anticipation contain three columns each: the average (over all problem instances in each problem set) number of clusters found by the planner, the average travel distance if the routes found by the planner are followed, and the average distance of the final routes found by the drivers. Note that, even in case of perfect anticipation, the drivers may find better routes than the planner, since the state space of each driver’s problem is smaller than the state space of the planner’s problem, while the state space restrictions for both problems are the same ($H = 10,000$).

Problem Set	Perfect Ant.			Approx. Perfect Ant.		
	# clust.	Pl. Dist.	Dr. Dist.	# clust.	Pl. Dist.	Dr. Dist.
c1 (9)	10.00	947	946	10.33	952	950
c2 (8)	5.50	787	785	5.63	817	811
r1 (12)	9.42	1158	1154	9.75	1159	1153
r2 (11)	7.27	1093	1091	7.73	1106	1102
rc1 (8)	10.25	1333	1332	10.13	1298	1290
rc2 (8)	7.88	1220	1219	8.50	1270	1261

Table 7.1: Results for team situation: perfect and approximately perfect anticipation

The results demonstrate that perfect anticipation clearly outperforms approximately perfect anticipation. For all but one problem set, the average number of clusters is smaller in case of perfect anticipation. On average over all problem instances, perfect anticipation results in 3.5% less clusters than approximately perfect anticipation. Also the travel distances are smaller in case of perfect anticipation than in case of approximately perfect anticipation. On average, the difference is 0.90% for the routes found by the planner, and 0.56% for the final routes found by the drivers.

The improvements found by the drivers with respect to the routes found by the planner in terms of reduced travel distance are not too big. However, for a substantial number of customer clusters, the drivers find better routes than the planner: 7.7% in case of perfect anticipation and 19% in case of approximately perfect anticipation. The larger portion in case of approximately perfect anticipation is due to the larger solution space that the drivers consider by including the modified rules, which are ignored by the planner in this case. The average reductions of the travel distances for these customer clusters are 1.4% and 2.4% in case of perfect and approximately perfect anticipation, respectively.

Table 7.2 presents the results for implicit anticipation. We tested different speeds applied by the planner to account for the breaks that the drivers have to schedule. We conducted experiments for the speeds of 2, 3, 4, and 5 distance

units per hour, where 5 is the reference speed in each problem instance. The case in which the speed is set to the reference speed is the extreme situation in which the planner neglects all breaks and rest periods that the drivers have to schedule. However, since the planner still does account for the routing including time windows, this is also a case of implicit anticipation.

Since with implicit anticipation certain customer clusters may not allow for feasible routes and break schedules, we have to consider such infeasibilities. Therefore, we report for each problem set the average number of clusters found by the planner (column 3), the average number of routes found by the planner that allow feasible break schedules (column 4), and the average number of clusters for which the drivers can find feasible routes and break schedules (column 6). Next, to make a fair comparison between the travel distances found by the planner and by the drivers, we present for each problem set the average travel distance of those customer clusters for which the routes found by the planner allow feasible break schedules. We present these travel distances both for the routes found by the planner (column 5), and for the routes found by the drivers (column 7). We do not include the travel distances of the routes found by the planner and the drivers for those customer clusters for which the routes found by the planner do not allow feasible break schedules, because it does not make sense to compare travel distances of infeasible vehicle routes.

The results indicate that higher speeds result in less clusters, but also in many infeasible clusters. For example, speeds 4 and 5 result in 13% and 30% infeasible clusters, respectively, and for the rc2 instances these values are even 22% and 46%, respectively. Therefore, these speeds are not suitable for practice. Speed 2 is also clearly not suitable, since this speed results in 84% more clusters than with perfect reactive anticipation, and 78% more clusters than with approximately perfect reactive anticipation. Speed 3 seems better than the other speeds, but still results in 16% more clusters than perfect reactive anticipation of which 4% is infeasible. Therefore, in practice breaks should explicitly be scheduled, also on a central planning level.

7.4.2 Non-Team Situation

We now consider the non-team situation in which the drivers' (hidden) objective is to optimize their return time. This case can be easily implemented within the DP heuristic, since we only need to adjust the objective function for the TSPTW-EC. This objective can be set by changing the secondary cost criterion of each state from the total distance traveled to the completion time of the last visited customer. We only run the experiments in the non-team situation for perfect and approximately perfect reactive anticipation, since the results in Section 7.4.1 indicated that implicit anticipation does not meet practical requirements.

Table 7.3 presents the results for the non-team situation with perfect and

Speed	Problem Set	# clust.	Planner		Drivers	
			Feas. ^a	Dist. ^b	Feas. ^c	Dist. ^d
2	c1	13.89	13.89	1220	13.89	1220
	c2	14.38	13.00	1285	13.50	1285
	r1	14.25	13.50	1229	14.00	1229
	r2	13.82	13.18	1212	13.82	1210
	rc1	18.75	18.63	1841	18.75	1840
	rc2	19.38	19.13	1894	19.38	1890
3	c1	10.00	9.89	891	10.00	891
	c2	8.00	6.88	764	7.25	764
	r1	10.08	7.92	897	9.42	896
	r2	9.00	6.64	805	8.91	804
	rc1	11.00	9.13	1075	10.75	1073
	rc2	10.50	8.63	1030	9.88	1028
4	c1	10.00	9.22	876	9.78	875
	c2	6.00	4.25	575	5.00	575
	r1	9.08	5.25	622	7.58	621
	r2	7.64	3.64	492	6.09	491
	rc1	10.13	6.75	845	9.63	843
	rc2	8.38	3.50	439	6.50	437
5	c1	10.00	8.22	802	9.22	799
	c2	5.25	3.75	510	4.13	510
	r1	8.58	3.67	454	5.92	452
	r2	6.55	1.82	270	3.18	270
	rc1	9.50	4.63	629	7.88	629
	rc2	7.63	2.25	273	4.13	273

^aAverage number of routes found by the planner that allow feasible break schedules

^bAverage travel distance of the feasible routes found by the planner

^cAverage number of customer clusters for which the drivers could find feasible routes and break schedules

^dAverage travel distance of the routes found by the drivers for those customer clusters for which the routes found by the planner allow feasible break schedules

Table 7.2: Results for team situation: implicit anticipation

approximately perfect anticipation. The number of clusters is the same as in the team situation, since the planner's problem does not change. Table 7.3 presents the average total travel distance and the average return time for the routes found by the planner, and for the routes found by the drivers.

The results indicate that drivers can improve the routes found by the planner with respect to the drivers' (hidden) objective. The average return time reductions over all problem instances are 1.9% and 3.9% in case of perfect and approximately perfect anticipation, respectively. However, by improving the routes according to their private criterion, the drivers deteriorate the planner's secondary objective, the total travel distance, by 7.1% and 11%, respectively. The percentage of routes that could be improved by the drivers in terms of their

Anticipation Function	Problem Set	Planner		Drivers	
		Dist.	Ret.	Dist.	Ret.
Perfect	c1	947	979	1045	957
	c2	787	684	792	681
	r1	1158	960	1257	938
	r2	1093	821	1152	806
	rc1	1333	1041	1473	1012
	rc2	1220	922	1279	914
Approximately Perfect	c1	952	1009	1056	966
	c2	817	702	851	689
	r1	1159	1014	1350	960
	r2	1106	879	1201	848
	rc1	1298	1065	1473	1021
	rc2	1270	1015	1390	985

Table 7.3: Results for non-team situation: perfect and approximately perfect anticipation

objective is 30% in case of perfect anticipation and 57% in case of approximately perfect reactive anticipation. Since the planner's top-down criterion (minimizing travel distance) conflicts with the base level's criterion (minimizing return time), there is much more room for improvement by the drivers than in the team situation in which these criteria are in line (both minimizing travel distance).

Another interesting observation for the non-team situation is that the difference between perfect and approximately perfect anticipation is bigger with respect to the company's secondary objective than in the team situation. The difference between perfect and approximately perfect anticipation in terms of travel distance for the routes found by the drivers is 4.71%. This is much larger than the 0.56% in case of the team situation, because perfect anticipation results in tighter routes found by the planner than approximately perfect anticipation. Therefore, if drivers find other routes, better with respect to their objective, it is unlikely that the total travel distance is much bigger. However, in case of less tight routes with approximately perfect anticipation there may be larger increases in the travel distance. The difference in return time between perfect and approximately perfect anticipation is also substantial: 5.2% for the routes found by the planner and 3.0% for the routes found by the drivers. These results indicate that a more precise representation of the base level's model within a non-team situation has a positive impact on the quality of the planning, both in terms of the top-down criterion and in terms of the base-level's criterion. Within a non-team situation, this impact is even larger than within a team situation.

Note that the number of clusters and the *total* return time is less with perfect anticipation than with approximately perfect anticipation. However, the *average* return time is larger with perfect anticipation than with approximately

perfect anticipation, because of the fewer customer clusters with perfect anticipation. Therefore, on average, the drivers' objective value is larger (0.63%) with perfect anticipation than with approximately perfect anticipation.

7.5 Conclusions

In practice, vehicle routing and break scheduling often involves a distributed decision making process, in which planners and drivers are both responsible for parts of the planning process. Therefore, the in the VRP literature generally used approach of centralized planning may not always be realistic in practice. Drivers may mess up the centrally planned routes by trying to optimize them according to their own objectives. Therefore, the main question in this chapter is whether it pays off to make a very detailed centralized plan, when drivers subsequently change their planned routes according to their own objectives.

To answer this question, we analyzed the problem of combined vehicle routing and break scheduling from a distributed decision making perspective. We investigated the impact of both a team and a non-team situation between planners and drivers on the resulting vehicle schedules. We proposed different degrees of anticipation for the drivers' planning behavior. The computational experiments showed that the explicit anticipation functions are superior to the implicit anticipation functions, both in terms of the planner's objectives and the drivers' objectives. Even when only a small reduction in travel speed is applied, the average number of customer clusters is still substantially larger than in case of perfect anticipation. Moreover, small speed reductions result in many infeasible customer clusters. Therefore, in practice planners should explicitly account for break scheduling already on a central planning level.

In the case that the drivers do not follow the company's objectives, but instead optimize their own hidden criterion, the planner's main objective - minimize the number of vehicles used - is not affected. However, the planner's secondary objective - minimize the total travel distance - is deteriorated substantially by the drivers' hidden actions. Here it turns out that a more precise anticipation of the drivers' planning model results in a less severe deterioration of the planner's top-down criterion. This is due to the fact that drivers do not have so many possibilities for deviating from the routes found by the planner.

Our results indicate that, even though drivers may try to improve on their routes, it pays off to make a detailed centralized planning by the planner. The benefit of having a detailed centralized planning is even stronger in case drivers' objectives deviate from the company's objective. This is explained by the fact that a detailed centralized planning results in tight vehicle routes, such that drivers have few possibilities to change their routes according to their private criterion. Therefore, detailed vehicle routing methods are highly valuable in practice.

Chapter 8

Conclusions and Recommendations

In this chapter, we discuss the main findings of this thesis in view of the research questions we pose in Chapter 1. Next, we give some recommendations for further research.

8.1 Conclusions

In this thesis, we consider vehicle routing problems with two common real life restrictions: *time-dependent travel times* and *driving hours regulations*. Time-dependent travel times result from traffic congestion, which regularly appears in urban areas. Driving hours regulations are imposed by law (partly to prevent accidents) and have to be respected by all persons involved in road transport. Since European law poses a set of driving hours regulations that is valid for all member countries in the European Union and traffic congestion forms a major problem for all these countries, solution methods that consider these timing restrictions are highly valuable in practice. Therefore, the research objective of this thesis is

To design an off-line vehicle routing approach that improves delivery reliability and reduces transport costs by avoiding traffic congestion whenever possible taking into account the EC social legislation on driving and working hours.

Three key elements indicate the value of such a solution method in practice: solution quality, computation time, and flexibility. The solution quality of

a VRP solution method is important, since it has a direct impact on transportation costs. Next, the computation time for solving problem instances of realistic sizes should be small enough for practical use. Finally, the solution method should be flexible towards the adaptation of new restrictions, since practice is a dynamic environment in which problem restrictions depend on the application at hand and may even change over time. Within such a dynamic environment, computation time becomes even more important. We discuss the research questions in view of these three key elements.

8.1.1 Research question 1

What is the state of the art in the literature on VRPs with time-dependent travel times and driving hours regulations?

Chapter 1 reveals that both the literature on the VRP with time-dependent travel times and the literature on the VRP with driving hours regulations is scarce. Moreover, there is - to the best of our knowledge - no literature that considers the VRP with both time-dependent travel times and driving hours regulations. The literature on the VRP with time-dependent travel times does not consider hard time windows. Moreover, duty time minimization as the objective function has been ignored. The literature on the VRP with driving hours regulations does not consider the full Regulation (EC) No 561/2006 on driving hours, and Directive 2002/15/16 on working hours has been completely ignored. Moreover, duty time minimization as the objective function has again been ignored. The proposed solution methods in both research areas focus on local search (improvement) methods. However, constructive methods are more suitable for including complex timing restrictions such as time-dependent travel times and driving hours regulations.

8.1.2 Research question 2

What impact do traffic congestion and driving hours regulations have on the performance of vehicle routes constructed with state of the art vehicle routing methods?

Chapter 2 shows that state of the art vehicle routing methods fail in practice when traffic congestion and driving hours regulations are present. In Chapter 2, we propose an ILP model for the departure time scheduling problem that determines for a given customer visit sequence whether there exists a feasible set of departure times respecting time-dependent travel times and driving hours regulations. Computational experiments show that for many routes developed by state of the art vehicle routing methods that ignore time-dependent travel

times or driving hours regulations, no feasible departure schedule can be found. This is not only the case for tight routes obtained by state of the art vehicle routing methods, but also for routes obtained by less sophisticated methods, and in case some slack on the driving times is introduced to gain more flexibility in scheduling the departure times. Therefore, the main conclusion of Chapter 2 is that new vehicle routing methods should be developed that account for both time-dependent travel times **and** driving hours regulations.

8.1.3 Research question 3

What type of solution framework is suitable for handling different types of vehicle routing problems and incorporating complex timing restrictions such as time-dependent travel times and driving hours regulations?

In Chapter 3, we propose a restricted DP heuristic for solving vehicle routing problems. We demonstrate that this heuristic is a flexible solution method by illustrating how various VRP extensions can be incorporated in the solution framework. Computational experiments demonstrate a trade-off between solution quality and computation time, which is controlled by 2 parameters (H and E) that restrict the state space in order to obtain practical computation times. The solution method is a constructive method as opposed to the majority of the solution approaches proposed in the VRP literature, which are focused on improvement methods. Complex timing restrictions such as time-dependent travel times and driving hours regulations are hard to incorporate in improvement methods. Therefore, the restricted DP heuristic suits better to include these timing restrictions than most solution approaches in the VRP literature.

8.1.4 Research question 4

How can driving hours regulations be incorporated in off-line vehicle routing methods?

Chapter 4 extends the restricted DP heuristic proposed in Chapter 3 with a break scheduling algorithm that checks for (partial) vehicle routes whether feasible break schedules exist. This break scheduling algorithm considers all relevant rules of the EC social legislation for one-week planning. The break scheduling algorithm decides *locally* whether breaks or rest periods should be scheduled, i.e., no time consuming checks downwards the partial vehicle routes are done to find other break schedules. This approach may miss some feasible break schedules, but has the advantage that the running time complexity of the DP heuristic does not increase. The quality of the solution method is demonstrated by substantially improving the solutions for a set of benchmark

instances obtained by the state of the art solution method for this problem, which is based on local search. Moreover, these improved results are obtained with substantially smaller computation times. In addition, Chapter 4 indicates that exploiting the modified rules in the EC social legislation (which allow for more flexibility) is beneficial, and that Directive 2002/15/EC on working hours has a substantial impact on the VRP solutions, which shows that this directive cannot be ignored.

8.1.5 Research question 5

What impact do different congestion avoidance strategies in off-line vehicle route plans have on the real-time performance of these plans?

In Chapter 5, we propose four strategies to avoid traffic congestion to an increasing extent within off-line vehicle routing. These congestion avoidance strategies are all based on not being at the wrong place at the wrong time. Within each strategy, we solve a (time-dependent) shortest path problem and a (time-dependent) VRP. To investigate the impact of these strategies in a realistic setting, we propose a time-dependent speed model for real road networks based on the three key elements of peak hour traffic congestion: heavy congestion in urban areas, heavy congestion on roads towards urban areas in the morning peak (and vice versa in the evening peak), and heavy congestion on roads with a high maximum speed (highways). We develop a set of TD-VRP instances with time windows on these road networks using the proposed speed model. Computational experiments demonstrate that avoiding traffic congestion to a large extent results in substantial cost savings in terms of, e.g., number of vehicles used and total duty time, and substantial reliability improvements in terms of, e.g., number of late arrivals. Solving a TDVRP (for which we use the DP heuristic of Chapter 3) instead of a standard VRP increases delivery reliability and reduces the number of vehicles and the total duty time substantially. If, in addition, *time-dependent* shortest paths are calculated in a pre-processing phase, then the number of vehicles and the total duty time reduce even further. Therefore, solving a TDVRP instead of a VRP and a TDSPP instead of an SPP substantially increases the quality of the off-line vehicle route plan. Since the DP heuristic always adds new customers to the end of partial vehicle routes, the computation time for solving VRPs *does not* substantially increase when time-dependent travel times are considered. The computation time for solving SPPs *does* substantially increase when time-dependent travel times are considered, since in that case we must solve an SPP for multiple departure times. However, since SPPs can be solved in a pre-processing phase, computation times play only a minor role in practice for these problems.

8.1.6 Research question 6

How can we account for both time-dependent travel times and driving hours regulations during the construction of vehicle routes with duty time minimization as the objective function?

In Chapter 6, we extend the DP heuristic of Chapter 3 with a VDO heuristic that checks for a (partial) vehicle route whether feasible departure schedules exist and what the minimal costs of these departure schedules are in terms of total duty time. The VDO heuristic is in fact an exact approach for the variant of the VDO in which unforced waiting times - waiting times that are not required by obligatory breaks and that can be reduced by postponing departure times - and early breaks are not considered. The VDO heuristic runs in polynomial time, so the running time complexity of the restricted DP heuristic remains polynomial. Computational experiments demonstrate that substantial duty time reductions can be achieved by setting the objective to minimize total duty time, but at the cost of more travel distance and vehicles. Moreover, they illustrate the benefits of extending the depot opening hours such that traveling before the morning peak and after the evening peak becomes possible. We also demonstrate that the VDO heuristic obtains good results by solving the ILP formulation of Chapter 2 (which does consider unforced waiting times) with duty time minimization as the objective function for each obtained vehicle route, which results in only minor improvements in total duty time, on average.

8.1.7 An alternative: distributed decision making

In the VRP literature, vehicle routing problems have been generally approached from a centralized planning perspective in which the planner makes all the decisions. In practice, however, combined vehicle routing and break scheduling often involves a distributed decision making process, in which both planners and drivers are responsible for parts of the planning process. The question that arises is whether it pays off to put effort in obtaining high quality and detailed solution methods for the central planning level, when drivers may subsequently change their route according to their own objectives.

In Chapter 7, we analyze combined vehicle routing and break scheduling from this alternative distributed decision making perspective. We consider the planner to be responsible for the clustering of the customer requests, and the drivers to be responsible for their routing and break scheduling. We analyze both a team situation, in which planner's and drivers' objectives coincide (with the company's objectives), and a non-team situation in which they conflict. The latter may better represent practice, for example if the planner's objective is to minimize travel distance (corresponding to the company's ob-

jective), while a truck driver tries to complete his work as early as possible. When performing his main task ‘clustering the customer visits’, the planner anticipates the drivers’ behavior. The main conclusion of Chapter 7 is that the more precise this anticipation is, the better the overall planning is, *both in terms of the planner’s and the drivers’ objectives*, and both in the team and the non-team situation. Nevertheless, it may be difficult for planners to have such precise anticipations, since they make the planning problem much more complex. Therefore, putting effort in improving VRP solution methods that include many precise restrictions, e.g., the modified rules of the EC social legislation, is highly valuable for practice.

8.2 Recommendations for further research

The restricted DP heuristic we propose in Chapter 3 is a new and flexible approach for solving vehicle routing problems. Since we propose it for including difficult timing restrictions, it is beyond the scope of this thesis to make an in-depth investigation on improvements of this method for existing vehicle routing problem variants. For example, the criteria for selecting the most promising states to take to the next stage in order to restrict the state space has only been based on some properties of the partial solutions represented by the states (e.g., total distance traveled, number of vehicles used). However, good estimations of the costs of completing these partial solutions may further improve the selection process in each stage. Such estimations can be based on lower bounds by solving a relaxation of the problem of completing the partial solution, or on upper bounds by generating a complete solution through some fast procedure (e.g., by applying a nearest neighbor heuristic in each state to the nodes that still have to be visited). Lower bounds not only help in selecting the most promising states when performing a restricted DP search, but may also speed up the exact version of the DP algorithm. Namely, instead of computing all states, the DP algorithm may also perform a shortest path search through the state network. Since there are several speed up techniques available for shortest path searches that guarantee optimality, they may also be applied within the DP algorithm. An example is the A* algorithm of Hart et al. (1968), which is applied to routing problems by Fabri (2008). Note that the quality and computation time of such bounding procedures may depend on the VRP variant at hand.

In general, when restrictions are added to the problem (e.g., time windows), also state dimensions need to be added to account for these extra restrictions. These additional state dimensions may increase the *state space*, since different partial solutions represented by the same visited node set S and end node $j \in S$ may be non-dominated by each other, resulting in multiple non-dominated states. This seems contrary to the fact that additional restrictions *reduce* the *solution space*. Therefore, an in-depth investigation of the impacts of additional

restrictions on the size of the state space and the size of the solution space is an interesting topic for further research. The question arises whether the DP formulation is still helpful when many restrictions are added to the problem, or whether it suffices to perform a search through the solution space instead of the state space.

In Chapter 5, we investigate the impact of several congestion avoidance strategies on VRP instances generated on real road networks, but with a theoretical speed model. This speed model allows an in-depth investigation of the impacts of the strategies for different parameter settings. However, a case study with real travel times would be a welcome addition to this study. Since historical travel speeds on highways are available in the Netherlands (the Department of Waterways and Public Works collects data through so-called Inductive Loop Detectors) and more information on travel speeds for secondary roads is expected in the near future (e.g., through floating vehicle data), a case study on the congestion avoidance strategies with real life travel speeds is a promising direction for further research.

Robustness of the different congestion avoidance strategies is another important topic for further research. In this thesis, we focus on constructing good vehicle route plans in a deterministic setting. However, disturbances that appear in practice (i.e., an unexpected event such as a car accident), may substantially change the performance of vehicle route plans when they are executed. Such disturbances cause delays and may propagate downstream the vehicle routes. Different congestion avoidance strategies may also have different impacts on the robustness of the resulting vehicle route plans. For example, avoiding traffic congestion by choosing alternative routes (i.e., solving a TD-SPP) aims at avoiding the location of the congestion. Therefore, a disturbance at the beginning of the day may not have too big impacts on the travel times at later times of the day. Choosing alternative visit sequences and alternative customer-vehicle assignments by solving a TDVRP, however, may result in traffic congestion avoidance by traversing routes sensitive to traffic congestion at favorable hours. Such plans may be sensible to disturbances and, therefore, may be less robust.

The research topic of robustness is of particular importance for the VDO method we propose in Chapter 6. VDO only avoids traffic congestion by better timing of the departure times. Therefore, disturbances may have large impacts on the resulting performances of the vehicle route plans while being executed. An in-depth investigation of the impact of disturbances on vehicle route plans optimized by solving a VDO is an interesting topic for further research, especially when these disturbances are based on real life data.

The solution methods we propose in this thesis consider either one-day or one-week planning. Most solution methods are flexible with respect to adaptations to other planning horizons. For example, the dynamic programming framework we propose suits well for a rolling horizon framework, since the

state dimensions can be set such that they meet the conditions at the start of the planning horizon. This is an interesting property since, e.g., truck drivers at logistic service providers are often on duty for several days, while not all customer demands for these days are known in advance. Therefore, a VRP with certain initial conditions (e.g., driver states: driver location, amount of remaining driving/working time until next daily/weekly rest period) has to be solved in such applications. The difficulty in this research may not be how to account for such initial conditions, but by obtaining good results using the best search strategy through the state space. Since the size of the state space is restricted and initial driver conditions basically changes the vehicle fleet in a heterogeneous vehicle fleet, different orderings of the vehicles may result in different solutions. Therefore, adapting the solution methods to other planning horizons is an important topic for further research.

Our final recommendation is the usage of our solution methods on a strategic level. For example, the DP heuristic of Chapter 4 can be easily adapted to modifications of the EC social legislation. The impacts of such modifications can be analyzed through some sensitivity analysis using (adaptations of) our solution methods. Another example is the idea of separate road lanes for trucks, which has been proposed as a solution for the traffic congestion problems the transport sector has to face with every day. A sensitivity analysis of such strategic decisions is now possible with the solution methods we propose in this thesis.

Bibliography

- Aarts EHL, Lenstra JK (1997) *Local Search in Combinatorial Optimization*. Wiley, New York
- Ahn BH (1991) Vehicle-routing with time windows and time-varying congestion. *The journal of the Operational Research Society* 42(5):393–400
- Alfa AS, Heragu SS, Chen M (1991) A 3-opt based simulated annealing algorithm for the vehicle routing problem. *Computers and Industrial Engineering* 21:635–639
- ANWB Reisinformatie (2010) (In Dutch) Dagelijks drukke trajecten. World Wide Web, last checked on January 14th 2010, <http://www.anwb.nl/verkeer/nederland/verkeersinformatie/verkeersverwachting/Dagelijkse-drukke-trajecten.html>
- Archetti C, Savelsbergh MWP (2009) The trip scheduling problem. *Transportation Science* 43(1):417–431
- Augerat P (1995) *Approche polyédrale du problème de tournées de véhicules*. PhD thesis, Institut National Polytechnique de Grenoble
- Bartodziej P, Derigs U, Malcherek D, Vogel U (2009) Models and algorithms for solving combined vehicle and crew scheduling problems with rest constraints: an application to road feeder service planning in air cargo transportation. *OR Spectrum* 31(2):405–429
- Beasley JE (1983) Route-first cluster-second methods for vehicle routing. *Omega* 11(4):403–408
- Bellman RE (1957) *Dynamic Programming*. Princeton Univ. Press
- Bellman RE (1962) Dynamic programming treatment of the travelling salesman problem. *Journal of the Association for Computing Machinery* 9(1):61–63
- Berger J, Barkaoui M (2004) A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* 31(12):2037–2053

- Bisiani R (1987) Beam search. In: Shapiro S (ed) *Encyclopedia of Artificial Intelligence*, Wiley & Sons, pp 56–58
- Campbell AM, Savelsbergh MWP (2004) Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* 38:369–378
- Christofides N, Eilon S (1969) An algorithm for the vehicle-dispatching problem. *Operation Research Quarterly* 20(3):309–318
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12(4):568–581
- Cordeau JF, Gendreau M, Laporte G (1997) A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks* 30(2):105–119
- Cordeau JF, Laporte G, Mercier A (2001) A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* 52:928–936
- Cordeau JF, Desaulniers G, Desrosiers J, Solomon MM, Soumis F (2002a) VRP with time windows. In: Toth P, Vigo D (eds) *The Vehicle Routing Problem*, SIAM, Philadelphia, pp 157–193
- Cordeau JF, Gendreau M, Laporte G, Potvin JY, Semet F (2002b) A guide to vehicle routing heuristics. *Journal of the Operational Research Society* 53(5):512–522
- Cordeau JF, Gendreau M, Hertz A, Laporte G, Sormany J (2005) New heuristics for the vehicle routing problem. In: Langevin A, Riopel D (eds) *Logistics Systems: Design and Optimization*, Springer-Verlag, New York, pp 279–297
- Coyle JJ, Bardi EJ, Langley Jr CJ (1996) *The Management of Business Logistics* (6th edition). West Publishing Company, USA
- Czech ZJ, Czarnas P (2002) Parallel simulated annealing for the vehicle routing problem with time windows. In: 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp 376–383
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Science* 6(1):80–91
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* pp 269–271
- Donati AV, Montemanni R, Casagrande N, Rizzoli AE, Gambardella LM (2008) Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operational Research* 185(3):1174–1191
- Dueck G (1993) New optimization heuristics. *J Comput Phys* 104(1):86–92

- Dueck G, Scheuer T (1990) Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. *Journal of Computational Physics* 90(1):161–175
- Dumas Y, Desrochiers J, Gelinas E, Solomon MM (1995) An optimal algorithm for the traveling salesman problem with time windows. *Operations Research* 43(2):367–371
- Eglese RW, Maden W, Slater A (2006) A road timetableTM to aid vehicle routing and scheduling. *Computers & Operations Research* 33(12):3508–3519
- Eichin R, Schneeweiss C (2001) Determining depreciations as a two-stage problem. *Schmalenbach Business Review* 53(2):66–81
- Eilon S, Watson-Gandy CDT, Christofides N, de Neufville R (1974) Distribution management: Mathematical modelling and practical analysis. *Systems, Man and Cybernetics, IEEE Transactions on* 4(6):589–589
- European Union (1985) Regulation (EEC) No 3820/85 of 20 December 1985 on the harmonization of certain social legislation relating to road transport. *Official Journal of the European Union L* 370, 31-12-1985
- European Union (2002) Directive 2002/15/EC of the European Parliament and of the Council of 11 March 2002 on the organisation of the working time of persons performing mobile road transport activities. *Official Journal of the European Union L* 080, 23-03-2002
- European Union (2006) Regulation (EC) No 561/2006 of the European Parliament and of the Council of 15 March 2006 on the harmonisation of certain social legislation relating to road transport and amending Council Regulations (EEC) No 3821/85 and (EC) No 2135/98 and repealing Council Regulation (EEC) No 3820/85. *Official Journal of the European Union L* 102/1, 11-04-2006
- Fabri A (2008) Behandlung des dynamischen pickup and delivery vehicle routing problems mit zeitfenstern und ladebedingungen mittels spezieller statusgraphen. Master's thesis, University of Dortmund
- Fahrion R, Wrede M (1990) On a principle of chain-exchange for vehicle-routing problems (1-*vrp*). *Journal of the Operational Research Society* 41:821–827
- Federal Motor Carrier Safety Administration (2008) Hours-Of-Service Regulations. <http://www.fmcsa.dot.gov/rules-regulations/topics/hos/index.htm>
- Figliozzi MA (2009) A route improvement algorithm for the vehicle routing problem with time dependent travel times. In: *Proceedings of the 88th Transportation Research Board Annual Meeting*, Washington DC. USA

- Fisher ML, Jaikumar R (1981) Generalized assignment heuristic for vehicle routing. *Networks* 11(2):109–124
- Fleischmann B, Gietz M, Gnutzmann S (2004) Time-varying travel times in vehicle routing. *Transportation Science* 38(2):160–173
- Ford LR, Fulkerson DR (1958) A suggested computation for maximal multi-commodity network flows. *Management Science* 5(1):97–101
- Funke B, Grünert T, Irnich S (2005) Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics* 11(4):267–306
- Garey MR, Johnson DS (1979) *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, New York
- Gendreau M, Hertz A, Laporte G (1994) A tabu search heuristic for the vehicle routing problem. *Management Science* 40(10):1276–1290
- Gfrerer H, Zäpfel G (1995) Hierarchical model for production planning in the case of uncertain demand. *European Journal of Operational Research* 86(1):142–161
- Ghaziri H (1991) Solving routing problems by a self-organizing map. In: Kohonen T, Makisara K, Simula O, Kangas J (eds) *Artificial Neural Networks*, North-Holland, Amsterdam, pp 829–834
- Gietz M (1994) *Computergestützte Tourenplanung mit zeitkritischen Restriktionen*. Physica-Verlag, Heidelberg
- Gillett BE, Miller LR (1974) A heuristic algorithm for the vehicle-dispatch problem. *Operations Research* 22(2):340–349
- Goedhart MH, Spronk J (1995) An interactive heuristic for financial planning in decentralized organizations. *European Journal of Operational Research* 86(1):162–175
- Goel A (2009) Vehicle scheduling and routing with drivers' working hours. *Transportation Science* 43(1):17–26
- Goel A, Gruhn V (2006) Solving a dynamic real-life vehicle routing problem. In: Haasis HD, Kopfer H, Schönberger J (eds) *Operations Research Proceedings 2005*, Springer, pp 367–372
- Goel A, Kok AL (2009a) Efficient scheduling of team truck drivers in the European Union. Working Paper, University of Leipzig
- Goel A, Kok AL (2009b) Efficient truck driver scheduling in the United States. Working Paper, University of Leipzig

- Gromicho J, van Hoorn J, Kok AL, Schutten MJM (2008) The flexibility of restricted dynamic programming for the VRP. Beta working paper series 266
- Haghani A, Jung S (2005) A dynamic vehicle routing problem with time-dependent travel times. *Computers & Operations Research* 32(11):2959–2986
- Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems Science and Cybernetics* 4:100–107
- Hashimoto H, Ibaraki T, Imahori S, Yagiura M (2006) The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics* 154(16):2271–2290
- Hashimoto H, Yagiura M, Ibaraki T (2008) An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization* 5:434–456
- Held M, Karp RM (1962) A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics* 10(1):196–210
- Hill AV, Benton WC (1992) Modelling intra-city time-dependent travel speeds for vehicle scheduling problems. *The Journal of the Operational Research Society* 43(4):343–351
- Hsu CI, Hung SF, Li HC (2007) Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering* 80(2):465–475
- Ichoua S, Gendreau M, Potvin JY (2003) Vehicle dispatching with time-dependent travel times. *European Journal of Operational Research* 144(2):379–396
- Johnson DS, McGeoch LA (1997) The travelling salesman problem: a case study in local optimization. In: Aarts EHL, Lenstra JK (eds) *Local Search in Combinatorial Optimization*, Wiley, New York, pp 215–310
- Jorritsma P, Berveling J, Harms L, Kolkman J, Koopmans C, Lijesen M, van der Loop H, Olde Kalter MJ, van Ooststroom H, Visser J, Warffemius P (2008) Mobiliteitsbalans. Kennisinstituut voor Mobiliteitsbeleid pp 1–90
- Kawamuro H, Yamamoto M, Mitamura T, Suzuki K, Ohuchi A (1998) Cooperative search on pheromone communication for vehicle routing problems. *IEEE TRANSACTIONS on Fundamentals* E81-A(6):1089–1096
- Kinderwater GAP, Savelsbergh MWP (1997) Vehicle routing: Handling edge exchanges. In: Aarts EHL, Lenstra JK (eds) *Local Search in Combinatorial Optimization*, Wiley, New York, pp 337–360

- Kok AL, Hans EW, Schutten JMJ (2008) Optimizing departure times in vehicle routes. Beta working paper series 236
- Kok AL, Hans EW, Schutten JMJ (2009a) Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Beta working paper series 267
- Kok AL, Meyer CM, Kopfer H, Schutten JMJ (2009b) Dynamic programming algorithm for the vehicle routing problem with time windows and EC social legislation. Beta working paper series 270
- Kok AL, Hans EW, Schutten JMJ (2010) Vehicle routing with traffic congestion and drivers' driving and working rules. Beta working paper series 305
- Kopfer H, Meyer CM (2009) A model for the traveling salesman problem including the EC regulations on driving hours. In: Tuma A, Fleischmann B, Borgwardt KH, Klein R (eds) *Operations Research Proceedings 2008*, Springer, Berlin Heidelberg New York, pp 289–294
- Kopfer H, Meyer CM, Wagenknecht A (2007) Die EU-Sozialvorschriften und ihr Einfluss auf die Tourenplanung. *Logistik Management* 9(2):32–47
- Land AH, Doig AG (1960) An automatic method of solving discrete programming problems. *Econometrica* 28(3):497–520
- Laporte G (1992) The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3):345–358
- Laporte G, Nobert Y (1987) Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics* 31:147–187
- Li H, Lim A (2001) A metaheuristic for the pickup and delivery problem with time windows. pp 160–167
- Li H, Lim A (2003) Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research* 150(1):115–127
- Lin S (1965) Computer solution of the traveling salesman problem. *Bell System Technical Journal* 44:2245–2269
- Malandraki C, Daskin MS (1992) Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. *Transportation Science* 26(3):185–200
- Malandraki C, Dial RB (1996) A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem. *European Journal of Operational Research* 90(1):45–55

- Menger K (1930) Das botenproblem. In: Menger K (ed) *Ergebnisse eines Mathematischen Kolloquiums*, Teubner, Leipzig, pp 11–12
- Meyer CM, Kok AL, Kopfer H, Schutten JMJ (2009a) Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective. *Beta working paper series 289*
- Meyer CM, Kopfer H, Kok AL, Schutten JMJ (2009b) Distributed decision making in combined vehicle routing and break scheduling. In: *Proceedings of LogDynamics International Conference 2009*
- Moscato P, Cotta C (2003) A gentle introduction to memetic algorithms. In: Glover F, Kochenberger GA (eds) *Handbook of Metaheuristics*, Kluwer, Boston, pp 105–144
- Onken M (2009) Umsetzung der lenk- und ruhezeiten-verordnung eine empirische studie. Master's thesis, University of Bremen
- Or I (1976) Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. PhD thesis, Ph.D. Thesis, Northwestern University, Evanston, IL
- Orda A, Rom R (1990) Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the Association for Computing Machinery* 37(3):607–625
- Osman IH (1993) Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* 41:421–451
- Parragh SN, Doerner KF, Hartl RF (2008a) A survey on pickup and delivery problems. part i: Transportation between customers and depot. *Journal für Betriebswirtschaft* 58(1):21–51
- Parragh SN, Doerner KF, Hartl RF (2008b) A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft* 58(2):81–117
- Pearl J (1984) *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley
- Pesenti R (1995) Hierarchical resource planning for shipping companies. *European Journal of Operational Research* 86(1):91–102
- Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. *Computers & Operations Research* 34(8):2403–2435
- Ralphs T (2003) Branch cut and price resource web. World Wide Web, <http://www.brachandcut.org>

- Rao MR, Zionts S (1968) Allocation of transportation units to alternative trips—a column generation scheme with out-of-kilter subproblems. *Operations Research* 16(1):52–63
- Reeves F (2003) Genetic algorithms. In: Glover F, Kochenberger GA (eds) *Handbook of Metaheuristics*, Kluwer, Boston, pp 55–82
- Rego C, Glover F (2002) Local search and metaheuristics. In: Gutin G, Punnen A (eds) *Traveling Salesman Problem and Its Variations*, Kluwer Academic Publishers Dordrecht, pp 309–367
- Renaud J, Boctor FF, Laporte G (1996) A fast composite heuristic for the symmetric traveling salesman problem. *INFORMS Journal on Computing* 8:134–143
- Robuste F, Daganzo CF, Souleyrette RR (1990) Implementing vehicle routing models. *Transportation Research Part B: Methodological* 24(4):263–286
- Rochat Y, Semet F (1994) A tabu search approach for delivering pet food and flour in switzerland. *Journal of the Operational Research Society* 45:1233–1246
- Rochat Y, Taillard ÉD (1995) Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics* 1(1):147–167
- Ropke S, Pisinger D (2006) An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science* 40(4):455–472
- Savelsbergh MWP (1992) The vehicle routing problem with time windows: minimizing route duration. *ORSA Journal on Computing* 4(2):146–154
- Savelsbergh MWP, Sol M (1998) Drive: Dynamic routing of independent vehicles. *Operations Research* 46:474–490
- Schenk-Mathes HY (1995) The design of supply contracts as a problem of delegation. *European Journal of Operational Research* 86(1):176–187
- Schneeweiss C (1995) Hierarchical structures in organizations: A conceptual framework. *European Journal of Operational Research* 86(1):4–31
- Schneeweiss C (2003) Distributed decision making - a unified approach. *European Journal of Operational Research* 150(2):237–252
- Schneeweiss C, Zimmer K (2004) Hierarchical coordination mechanisms within the supply chain. *European Journal of Operational Research* 153(3):687–703
- Schrank D, Lomax T (2007) The 2007 urban mobility report. The Texas A&M University System pp 1–53

- Semet F, Taillard ÉD (1993) Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations Research* 41(4):469–488
- Shaw P (1997) A new local search algorithm providing high quality solutions to vehicle routing problems. Working paper, University of Strathclyde, Glasgow, Scotland
- Skabardonis A, Varaiya P, Petty KF (2003) Measuring recurrent and nonrecurrent traffic congestion. *Transportation Research Record* 1856:118–124
- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35(2):254–265
- Stumpf P (1998) Tourenplanung im speditionellen Güterverkehr. GVB, Nürnberg
- Taillard ÉD, Badeau P, Gendreau M, Guertin F, Potvin JY (1997) A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31(2):170–186
- Thompson PM, Psaraftis HN (1993) Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations Research* 41(5):935–946
- Tiger/line (2002) United States road networks. <http://www.dis.uniroma1.it/~challenge9/data/tiger/>
- Toth P, Vigo D (2002) The vehicle routing problem. *SIAM Monographs on Discrete Mathematics and Applications*, Philadelphia
- van Breedam A (1994) An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints. PhD thesis, University of Antwerp
- van Breedam A (1995) Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research* 86(3):480–490
- Van Woensel T, Kerbache L, Peremans H, Vandaele N (2008) Vehicle routing with dynamic travel times: A queueing approach. *European Journal of Operational Research* 186(3):990–1007
- Woeginger GJ (2003) Exact algorithms for NP-hard problems: A survey. In: *Combinatorial Optimization Eureka! You shrink!*, Lecture Notes in Computer Science, Vol. 2570, Springer, Berlin, pp 185–207
- Xu H, Chen ZL, Rajagopal S, Arunapuram S (2003) Solving a practical pickup and delivery problem. *Transportation Science* 37:347–364
- Zäpfel G, Bögl M (2008) Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics* 113(2):980–996

Appendices

A Complexity theory

This appendix gives a short introduction in complexity theory. Complexity theory plays a major role in the search for efficient algorithms for problems such as the VRP. Complexity theory can be used to determine whether it is likely that an efficient exact algorithm exists for a problem. If such an algorithm does not exist, then finding a provable optimal solution for practical problem instances often becomes computationally expensive. In general, to solve practical instances of such problems requires to resort to heuristic solution methods. For a detailed introduction on complexity theory, we refer to Garey and Johnson (1979).

The VRP belongs to the class of *combinatorial optimization* (CO) problems. Each CO problem has two properties: 1) there is a finite number of possible solutions and 2) there is some function that gives a value to each solution. The objective is to find the solution for which the value is optimal (minimal or maximal, depending on whether it is a minimization or maximization problem, respectively).

Each CO problem has a corresponding *decision problem*. The solution of a decision problem is either *yes* or *no*. For the TSP, for example, the corresponding decision problem is: ‘Does there exist a solution with value at most K ’, where K can be any given value.

The complexity of a problem depends on the most efficient algorithm that exists for this problem. The efficiency of algorithms is determined by the *running time* of an algorithm. The running time is measured by the maximum number of arithmetic operations as a function of the *input size* of a problem instance. The input size of a problem instance is the number of symbols needed to represent the input. For the TSP, the input size of a problem is in general measured by n , the number of nodes in the problem. The running time of an algorithm is then represented by $O(f(n))$, which means that there exists values k and n_0 , such that the number of arithmetic operations of the algorithm is never greater than $k \cdot f(n)$ for all instances with $n \geq n_0$.

The most straightforward way to solve the TSP is complete enumeration. Since the total number of visit sequences for a TSP with n nodes and a fixed start and end node is $(n - 1)!$, the running time of complete enumeration is $O((n - 1)!)$. This algorithm is a so-called *exponential time algorithm*. If the running time is bounded by some polynomial function $f(n)$, then the algorithm is a *polynomial time algorithm*.

Within complexity theory, problems for which a polynomial time algorithm exists are considered easier than problems for which no polynomial time algorithm exists. If a polynomial time algorithm exists for a certain problem then it belongs to the complexity class \mathcal{P} . A superset of the problems in \mathcal{P} are the problems belonging to the complexity class \mathcal{NP} . The problem classes \mathcal{P} and \mathcal{NP} are defined for decision problems. A problem belongs to the class \mathcal{NP} if for every yes-instance of the problem a certificate exists that can be verified in polynomial time. A certificate for a yes-instance of the TSP is a list of the nodes in visit sequence. Verifying that this visit sequence has value at most K can be done in $O(n)$ arithmetic operations.

A special class of problems in \mathcal{NP} are the \mathcal{NP} -complete problems. A problem is \mathcal{NP} -complete if and only if it belongs to \mathcal{NP} and every problem in \mathcal{NP} is *polynomially reducible* to it. A problem A is *polynomially reducible* to problem B if and only if there exists a polynomial-time function τ that transforms A -instances into B -instances such that x is a yes-instance for A if and only if $\tau(x)$ is a yes-instance for B . \mathcal{NP} -complete problems are considered the hardest problems in \mathcal{NP} , since a polynomial time algorithm for any \mathcal{NP} -complete problem can be transformed into a polynomial time algorithm for any other problem in \mathcal{NP} . Note that if such a polynomial time algorithm exists, this would imply $\mathcal{P} = \mathcal{NP}$. Although the question whether $\mathcal{P} = \mathcal{NP}$ is still an open question, it is widely believed that $\mathcal{P} \neq \mathcal{NP}$. This belief is strengthened as over several decades, despite tremendous efforts, nobody has found a polynomial time algorithm for an \mathcal{NP} -complete problem.

A combinatorial optimization problem is \mathcal{NP} -hard if the corresponding decision problem is \mathcal{NP} -complete. The TSP is an \mathcal{NP} -hard problem (Garey and Johnson, 1979). Since the VRP is a generalization of the TSP, it is at least as hard as the TSP. Since the VRP also belongs to the class \mathcal{NP} (a certificate that can be checked in polynomial time is, e.g., a sequence of visit sequences by concatenating the visit sequences of the vehicles) the VRP is an \mathcal{NP} -hard problem.

B Maximum number of breaks per day

We show that the maximum number of breaks required for a certain departure time for one vehicle route and one-day planning equals 4. We first construct an example where exactly 4 breaks are required and next, we show that there cannot exist departure times which require more than 4 breaks.

Suppose that the first break, say at customer i , must be scheduled after a very small amount of accumulated driving time, say $\epsilon > 0$. This happens if the driving time to the next customer j equals 4.5 (see Figure 1^a). Next, assume that the driving time from i to j reduces to $3.75 + \epsilon$ if a break of 0.75 is taken at customer i (see Figure 1^b). This is possible under the non-passing property. Then, after $3.75 + 2\epsilon$ of total driving time, and $3.75 + \epsilon$ of accumulated driving time since the last break, we arrive at customer j . If the driving time to the next customer k equals 0.75, then we also have to schedule a break at customer j . Under the non-passing property, it is possible that after the break of 0.75, the driving time to customer k has reduced to ϵ (see Figure 1^c). Therefore, when arriving at customer k , $3.75 + 3\epsilon$ of total driving time has passed. Furthermore, the accumulated driving time is ϵ , which is the same as at customer i . Next, we repeat the procedure to schedule two other breaks. By making ϵ arbitrarily small, the fourth break is required after 7.5 of total driving time.

$D = \text{driving}$
 $B = \text{break}$

	i			j
(a)	D	D		
	ϵ	4.5		

	i		j		k
(b)	D	B	D		D
	ϵ	0.75	$3.75 + \epsilon$		0.75

	i		j		k
(c)	D	B	D		B
	ϵ	0.75	$3.75 + \epsilon$		0.75
				D	ϵ

Figure 1: (a) route $0 \rightarrow i \rightarrow j$ with no break at i , (b) route $0 \rightarrow i \rightarrow j \rightarrow k$ with a break at i and no break at j , (c) route $0 \rightarrow i \rightarrow j \rightarrow k$ with breaks at i and j .

A fifth break may never be required because of the following. Observe that when the second break is taken, at least 3.75 of total accumulated driving time must have passed. This is, because the accumulated driving time before the first break at customer i , added to the driving time of the next travel,

say to customer j , must exceed 4.5 (otherwise no break would be required). The non-passing property allows this total accumulated driving time to reduce by at most 0.75 during the first break. Therefore, before the second break is taken, at least 3.75 of total driving time must have passed. Next, after the second break is taken, the accumulated driving time is 0 again. With the same reasoning, we can derive that before the fourth break is taken, at least 7.5 of total driving time must have passed. Since the remaining driving time after this fourth break is 1.5, while the accumulated driving time directly after the fourth break is 0, a fifth break can never be required.

C Glossary of symbols

C.1 parameters

V	set of customers (index $i = 0, \dots, n + 1$)
K	set of vehicles (index $k = 1, \dots, m$)
Q	capacity of a vehicle
Q_k	capacity of vehicle k
e_i	time window opening time at customer i
l_i	time window closing time at customer i
s_i	service time at customer i
q_i	demand of customer i
$c_i(X_i^d)$	travel time from customer i to customer $i + 1$ at departure time X_i^d
t_{max}	maximum daily driving time (9 hours)
t_{dp}	maximum accumulated driving time in a driving period (4.5 hours)
b_{min}^1	minimum duration break of type 1 (0.5 hours)
b_{min}^2	minimum duration break of type 2 (0.25 hours)
b_{total}	minimum total duration of breaks to end a driving period (0.75 hours)
d_{max}	maximum daily duty time (13 hours)
$g_{i,r}$	indicate the times at which the slope of the travel time function from customer i to customer $i + 1$ changes ($r = 1, \dots, m_i$)
$a_{i,r}$	travel time from customer i to customer $i + 1$ at time $g_{i,r}$
$b_{i,r}$	slope of the travel time function from customer i to customer $i + 1$ between $g_{i,r}$ and $g_{i,r+1}$
c_{ij}	travel costs from customer i to customer j
d_{ij}	travel distance from customer i to customer j
p	maximum number of times the speed changes on a route
r^z	duty time record of route z
r^{zw}	partial duty time record $w = 0, \dots, W^z$ of route z
r_u^z	duty time record entry $u = 0, \dots, U^z$
\tilde{r}_u^z	duty time record entry u when no break is scheduled at the last but one visited node in route z
d_u^z	departure time from the first visited node in route z
t_u^z	duty time for visiting all nodes in route z with departure time d_u^z

continued on next page

continued from previous page

a_u^z	arrival time at the last node in route z corresponding to departure time d_u^z
$T^z(d)$	duty time needed to visit all nodes in route z when departing at time d from the first node in route z
$F^z(a)$	departure time from the first node in route z corresponding to arrival time a at the last node in route z
ta_u^z	total accumulated driving time since the last break for departure time d_u^z
CL_k	customer cluster k
S	subset of customer set $V \setminus 0$
O	set of origin nodes (index $o = o^1, \dots, o^m$)
D	set of destination nodes (index $d = d^1, \dots, d^m$)
R	set of request nodes (index $r = r^1, \dots, r^n$)
A	arc set representing feasible travels between nodes
H	maximum stage width
E	maximum number of state expansions per state

C.2 variables

t_{nbw}	accumulated non-break working time
t_{nbd}	accumulated non-break driving time
t_{nr}	accumulated non-rest time
t_{dd}	accumulated daily driving time
t_{ww}	accumulated weekly working time
t_{wd}	accumulated weekly driving time
a_i	arrival time at customer i
δ_{ij}	remaining driving time from customer i to customer j
n_{dte} :	number of driving time extensions taken by the active vehicle
n_{rr} :	number of rest reductions taken by the active vehicle
t_{dw} :	accumulated daily working time

C.3 decision variables

X_i^s	time to start service at customer i
X_i^d	departure time from customer i
W_i^s	waiting time before time to start service at customer i
W_i^d	waiting time before departure from customer i
$B_i^{p,l}$	indicates the break time of break type $l = 1, 2$ before ($p = s$) or after ($p = d$) serving customer i

continued on next page

continued from previous page

$Y_i^{p,l}$	binary variable that can only take value 1 if the waiting time W_i^p is sufficient to be considered a break of type l
V_{ij}	binary variable that takes value 1 if a driving period starts at customer i and ends at customer $j > i$
$U_{i,r}$	binary variable that takes value 1 if and only if the departure time X_i^d from customer i is between $g_{i,r}$ and $g_{i,r+1}$
$X_{i,r}^d$	takes the value of X_i^d if the corresponding $U_{i,r}$ takes value 1, and 0 otherwise
C_i	travel time from customer i to customer $i + 1$
L_i	binary variable that takes value 1 if service start late at customer i , and can take value 0 otherwise
A_i	late time at customer i
A_i^{max}	maximum late time
$B_i^{p,rest}$	indicates the rest time before ($p = s$) or after ($p = d$) serving customer i
$Y_i^{p,rest}$	binary variable that can only take value 1 if the waiting time W_i^p is sufficient to be considered a rest
V_{ij}^{daily}	binary variable that takes value 1 if a driving period starts at customer i and ends at customer $j > i$
E_i	binary variable that takes value 1 if the daily driving time starting at customer i can be extended to 10 hours
x_{ik}	binary variable that takes value 1 if and only if customer i is served by vehicle k
x_{ij}^k	binary variable that takes value 1 if and only if customer i is succeeded by customer j with vehicle k
z_k	binary variable that takes value 1 if and only if vehicle k is used
t_{return}^k	return time of vehicle k

C.4 dynamic programming

(S, j)	state: minimal cost route through customers $S \subseteq V(0)$, ending at customer $j \in S$
$C(S, j)$	costs of state (S, j)

C.5 framework for DDM

A^T	top-level's decision field with actions $a^T \in A^T$
A^B	base-level's decision field with actions $a^B \in A^B$
C^T	top-level's criterion
C^{TT}	top-level's private criterion

continued on next page

continued from previous page

C^{TB}	top-level's top-down criterion
C^B	base-level's criterion
$M^T (C^T, A^T)$	top-level's planning model
$M^B (C^B, A^B)$	base-level's planning model
$IN (a^T)$	top-level's instruction to base level depending on top level's action a^T
$IN^* (a^{T*})$	optimal top-level's instruction to base level with top level's optimal action a^{T*}
$AF (IN)$	top-level's anticipation function of top-level's instruction
$Exp(M^B)$	approximation of the expected base-level's planning model

C.6 abbreviations

VRP	vehicle routing problem
TSP	traveling salesman problem
GTR	giant-tour representation
CVRP	capacitated vehicle routing problem
VRPTW	vehicle routing problem with time windows
PDP	pickup and delivery problem
PDPTW	pickup and delivery problem with time windows
MDVRP	multi-depot vehicle routing problem
SDVRP	split-delivery vehicle routing problem
OVRP	open vehicle routing problem
PVRP	periodic vehicle routing problem
VRPTW-EC	vehicle routing problem with time windows and the EC social legislation
TSPTW-EC	traveling salesman problem with time windows and the EC social legislation
TDVRP	time-dependent vehicle routing problem
SPP	shortest path problem
TDSPP	time-dependent shortest path problem
TDVRP-EC	time-dependent vehicle routing problem with time windows and the EC social legislation

Index

- A**
anticipation function 113
 approximately perfect 113
 implicit 113
 non-reactive 113
 perfect 113
- B**
base-level 112
branch & bound 9
- C**
combinatorial optimization 143
complexity class 144
 \mathcal{NP} 144
 \mathcal{P} 144
complexity theory 9, 143
 \mathcal{NP} -hard 8, 144
congestion avoidance strategy .. 69, 71
- D**
daily driving time 57
decision making units 112
decision problems 143
degree of urbanization 76
departure time scheduling 19
 ILP formulation 22
deterministic annealing 12
Dijkstra's algorithm 73
distributed decision making . 110, 112
driving hours regulations 2
duty time record 91, 92
duty time record entry 92
dynamic programming 9
 restricted 13, 40
 TSP 37
 VRP 39
- E**
EC social legislation 2, 52
exact method 8
exponential time 144
extended daily driving time 60
- G**
giant-tour representation 37
global optimum 11
- H**
heuristic 8
 constructive 10
 improvement 10
 meta- 11
- I**
input size 143
inter-route improvement 11
intra-route improvement 11
- L**
linear programming 9
local flexibility 56
local optimum 11
local search 10
- M**
modified rules 51
- N**
nearest neighbor heuristic 10
neighborhood 11
non-break driving time 57
non-break working time 57
non-passing property 13, 24
non-rest time 57

- non-team situation 112, 117, 120
- P**
- pickup and delivery problem 8
 time windows 36
- piecewise linear 24
- planning
 deterministic 4
 off-line 4
- polynomial time 144
- polynomially reducible 144
- R**
- reduced rest period 60
- running time 143
- S**
- savings heuristic 10
- shortest path problem 70
 time-dependent travel times . . 70
- simulated annealing 11
- Solomon instances 26
- speed model 70, 73
 sensitivity analysis 80
- speed pattern 26, 103
- split break 61
- split rest period 62
- state dimension 41
- sweep algorithm 10
- T**
- tabu search 12
- team situation 112, 116, 118
- time-dependent travel times 2
- top-level 112
- traveling salesman problem 7
- two-phase method 11
- V**
- VDO 90
- vehicle routing problem 7
 capacitated 7
 EC social legislation 51
 multi-depot 8, 36
 multi-route 8
 open 8
- periodic 8
- site-dependent 8
- split-delivery 8
- time windows 7, 36, 88
- time-dependent 12
- time-dependent and EC social leg-
 islation 87
- W**
- waiting time
 forced 90
 unforced 90
- weekly driving time 57
- weekly working time 57

Samenvatting

Routeplanning is een complex probleem waar bedrijven als logistieke dienstverleners en distributiebedrijven dagelijks mee te maken hebben. Het hebben van een goede planning is noodzakelijk om de transportkosten laag te houden en om de leveringen op tijd plaats te laten vinden. Het realiseren van een dergelijke planning is echter niet eenvoudig. Planners hebben te maken met grote aantallen orders die ze moeten toewijzen aan de voertuigen. Ook moeten ze voor elk voertuig bepalen in welke volgorde de toegewezen orders uitgevoerd moeten worden. Hierbij moeten planners rekening houden met vele restricties, zoals tijdsvensters voor de leveringen, voertuigcapaciteiten en productcompatibiliteiten. Twee restricties die het planningsprobleem in het bijzonder lastig maken zijn de dagelijkse files en het rijtijdenbesluit opgesteld door de Europese Unie. In dit proefschrift onderzoeken we routeplanningsproblemen onder deze twee tijdsrestricties en ontwikkelen efficiënte oplossingsmethoden hiervoor.

In 2002 berekende Transport Logistiek Nederland dat 10% van de werktijd van chauffeurs verloren gaat als gevolg van de dagelijkse files. De totale directe transportkosten als gevolg van files bedroegen in dat jaar ongeveer 1,2 miljard euro. Sindsdien is het fileprobleem alleen maar toegenomen. Doordat files tijdelijk zijn, hangen reistijden sterk af van het gekozen vertrektijdstip. Dit veroorzaakt een nieuw probleem binnen de routeplanning: het bepalen van het optimale vertrektijdstip van elk voertuig.

In 2006 introduceerde de Europese Unie een vernieuwd rijtijdenbesluit met als doel de Europese wegen veiliger te maken. Elk jaar vinden er namelijk ongeveer 1,5 miljoen ongelukken plaats op de Europese wegen, waarvan meer dan 40 duizend met dodelijke afloop. Vermoedheid bij chauffeurs wordt gezien als een belangrijke oorzaak van deze ongelukken. Daarom stelt het rijtijdenbesluit restricties aan de totale rij- en werktijd, waarna een pauze moet worden genomen. Deze pauzes vinden vaak bij klanten plaats, waardoor niet alleen het vertrektijdstip van elk voertuig aan het begin van de dag, maar ook bij elke klant bepaald moet worden.

Bestaande oplossingsmethoden voor routeplanningsproblemen houden geen rekening met files of met het rijtijdenbesluit. Er wordt vanuit gegaan dat er

na het bepalen van de routes genoeg speling is om de verplichte pauzes in te plannen en vertragingen als gevolg van files op te vangen. Hoofdstuk 2 van dit proefschrift laat echter zien dat dit in het algemeen niet zo is, waardoor er allerlei kostbare acties ondernomen moeten worden (zoals het inhuren van een extra voertuig) om toch alle leveringen op tijd plaats te laten vinden.

Het merendeel van de oplossingsmethoden voor routeplanningsproblemen zijn zogeheten verbetermethoden. Deze methoden proberen een bestaande planning telkens te verbeteren door middel van kleine aanpassingen, zoals het verplaatsen van een order van de ene route naar de andere. Het voordeel van zulke kleine aanpassingen is dat ze in het algemeen erg snel doorgerekend kunnen worden, zodat in korte tijd veel oplossingen geëvalueerd kunnen worden. Deze methoden zijn echter minder geschikt voor lastige tijdsrestricties als files en het rijtijdenbesluit. Het invoegen van een order in een route bijvoorbeeld verandert de vertrektijdstoppen bij alle klanten die na deze order bezocht moeten worden (en daarmee mogelijk ook de reistijden). Daarom is het evalueren van een dergelijke aanpassing onder deze tijdsrestricties erg rekenintensief.

In dit proefschrift ontwikkelen wij een oplossingsmethode voor routeplanningsproblemen gebaseerd op dynamisch programmeren. Deze methode breidt op een efficiënte manier bestaande deelroutes uit door telkens een order aan het eind van één van deze deelroutes toe te voegen. Het voordeel van deze constructieve manier van het creëren van de routes is dat lastige tijdsrestricties gemakkelijk meegenomen kunnen worden. Daarnaast is de methode ook zeer flexibel met betrekking tot vele andere restricties, zoals volgordebetrekkingen, verschillende trailercompartimenten, meerdere depots en heterogene voertuigen.

De kwaliteit van onze oplossingsmethode demonstreren we in hoofdstuk 4, waar we het routeplanningsprobleem onder het rijtijdenbesluit (maar zonder rekening te houden met files) onderzoeken. Onze methode vindt veel betere oplossingen (gemiddeld 18% minder voertuigen en 5% minder reisafstand) voor een uitgebreide verzameling standaardproblemen dan de state-of-the-art oplossingsmethode voor dit probleem.

Door tijdens de routeplanning al rekening te houden met files, kun je ze ook vermijden, bijvoorbeeld door dagelijks drukke trajecten te mijden tijdens de spitsuren. Dit kan op verschillende manieren: kies een andere route tussen twee locaties, verander de volgorde waarin een voertuig orders uitvoert, of wijs een order toe aan een ander voertuig. In hoofdstuk 5 analyseren we de impact van 4 verschillende strategieën om files te vermijden. Voor deze analyses maken we gebruik van echte wegennetwerken en ontwikkelen we een snelheidsmodel dat gebaseerd is op observaties van de ANWB over de dagelijks drukke trajecten. De resultaten laten zien dat met de beste filevermijdingsstrategieën niet alleen de betrouwbaarheid van de planning substantieel toeneemt, maar dat ook bijna alle extra kosten (in de vorm van extra reistijd of extra voertuigen) als gevolg van files vermeden kunnen worden.

Een speciale manier van filevermijding is vertrektijdoptimalisatie. Door later te vertrekken bijvoorbeeld kun je een dagelijks druk traject na de spits passeren. Doordat er binnen de routeplanning rekening moet worden gehouden met het rijtijdenbesluit en andere tijdsrestricties zoals klantspecifieke tijdsvensters, is vertrektijdoptimalisatie een complex probleem. Het bepalen van de optimale vertrektijdstoppen voor een voertuig kan daarom soms erg lang duren. In hoofdstuk 6 ontwikkelen wij een snelle heuristiek voor het vertrektijdoptimalisatieprobleem dat oplossingen vindt die gemiddeld minder dan 0.5% van de optimale oplossing verwijderd zijn. Ook laten we zien dat het rijden voor de ochtend- en na de avondspits fors op de kosten kan besparen.

De toepassingen van de oplossingsmethoden ontwikkeld in dit proefschrift beperken zich niet tot de problemen waarvoor ze ontwikkeld zijn. De oplossingsmethoden zijn erg flexibel met betrekking tot nieuwe restricties. Daarom zijn ze uitermate geschikt voor de praktijk: een dynamische omgeving waarin restricties snel kunnen veranderen. Daarnaast laat deze flexibiliteit het gebruik van de oplossingsmethoden binnen andere studies toe. Hoofdstuk 7 geeft hier een voorbeeld van, waarin routeplanning geanalyseerd wordt vanuit een gedistribueerd beslissingsproces. In een dergelijk beslissingsproces zijn zowel de planner als de chauffeurs verantwoordelijk voor een deel van het planningsproces. In de praktijk gebeurt het bijvoorbeeld vaak dat de planner verantwoordelijk is voor de toewijzing van de orders aan de voertuigen, terwijl elke chauffeur zelf de volgorde bepaalt waarin hij zijn toegewezen orders uitvoert en zijn eigen pauzes inplant. Om ervoor te zorgen dat de chauffeurs overal op tijd kunnen zijn, moet de planner wel rekening houden met de beslissingen die de chauffeurs nog moeten maken. Hoofdstuk 7 laat zien dat een gedetailleerde planning, waarin zoveel mogelijk restricties meegenomen worden door de planner, tot de beste resultaten leidt.

Dankzij de flexibiliteit van de oplossingsmethoden uit dit proefschrift kunnen deze ook gebruikt worden bij belangrijke beleidsmatige beslissingen, zoals veranderingen in het rijtijdenbesluit of het aanleggen van een aparte rijstrook voor vrachtwagens om files te vermijden. Met deze methoden is het voor het eerst mogelijk om op brede schaal de impact van zulke beslissingen te evalueren.

About the author

Leendert Kok was born on August 22, 1982 in Stolwijk, the Netherlands. He followed his pre-university education at the Emelwerda College in Emmeloord, which he completed in 2000. In the same year, he started his studies in Applied Mathematics at the University of Twente. He followed a master program at the Department of Discrete Mathematics and Mathematical Programming. In 2006 he obtained his MSc (ir.) under the supervision of prof. dr. J.L. Hurink and dr. ir. J.J. Paulus. His master's thesis was entitled 'Spatial Resource Constrained Project Scheduling; the Allocation and Sequencing of Activity Groups to Spatial Resources'.

In 2006 he started as a Ph.D. student at the University of Twente, School of Management and Governance, Department of Operational Methods for Production and Logistics, under the supervision of prof. dr. W.H.M. Zijm, dr. ir. J.M.J. Schutten, and dr. ir. E.W. Hans. He started a research project entitled 'Vehicle Routing under Time-Dependent Travel Times', which resulted in his dissertation entitled 'Congestion Avoidance and Break Scheduling within Vehicle Routing'. He did part of his research abroad during a two-month visit at the University of Bremen, Germany. Here he collaborated with prof. dr.-ing H. Kopfer and C.M. Meyer, MSc.