Sustainable city logistics : fleet planning, routing and scheduling problems

SUSTAINABLE CITY LOGISTICS

Fleet planning, routing and scheduling problems

Anna Franceschetti

Beta
Research School for Operations Management and Logistics

Sustainable City Logistics

Fleet planning, routing and scheduling problems

Printed by Proefschriftmaken.nl || Uitgeverij BOXPress
Cover design by Alessandro de Gaspari

This thesis is number D194 of the thesis series of the Beta Research School for Operations Management and Logistics. The Beta Research School is a joint effort of the School of Industrial Engineering and the Department of Mathematics and Computer Science at Eindhoven University of Technology, and the Center for Production, Logistics and Operations Management at the University of Twente.

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-3921-5

This research has been partially funded by the Dutch Institute for Advanced Logistics (DINALOG), within the context of the 4C4D: City Distribution project.

Sustainable City Logistics
Fleet planning, routing and scheduling problems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 22 september 2015 om 16:00 uur

door

Anna Franceschetti geboren te Legnago, Italië

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. I.E.J. Heynderickx
1e promotor:	prof.dr. T. Van Woensel
2e promotor:	prof.dr.ir. J.C. Fransoo
leden:	prof.dr. G. Woeginger
	dr. D. Honhon (University of Texas at Dallas)
	prof.dr. G. Laporte (HEC Montréal)
	prof.dr. F. Semet (Ecole Centrale de Lille)

più si perdeva in quartieri sconosciuti di città lontane, più capiva le altre città che aveva attraversato per giungere fin là ⋆

Italo Calvino, *Le città invisibili*

⋆ *the more one was lost in unfamiliar quarters of distant cities, the more one understood the other cities he had crossed to arrive there*

If I have seen further it is by standing
on the shoulders of Giants.
Isaac Newton, *Letter To Robert Hooke,*
February 5, 1675

Acknowledgments

I would like to express my greatest gratitude to all my *giants* for their help, support and encouragement over the past, nearly four, years.

Firstly, I would like to thank my promotor Tom Van Woensel, who has been a great support for me right from the beginning of my PhD. His critical view on my work has substantially improved the quality of this thesis. Thank you for constantly believing in my work, for your encouragement and your patience.

I was very lucky to have Dorothée Honhon as a daily supervisor. Dorothée is a mentor and a friend, from whom I have learned a countless amount of things. Thank you for putting all of your passion and enthusiasm into our work. I cannot think of a better supervisor to have.

During my PhD I had the great privilege to work with Gilbert Laporte, one of the most brilliant minds I've ever encountered. Thank you for your guidance, friendship, and willingness to share your knowledge with me. I take great pride and consider myself to be very fortunate to have the opportunity to work with you.

I express my gratitude to Tolga Bektaş, with whom I collaborated during the first year of my PhD. Thank you for the great time and for your valuable feedbacks. It was a great pleasure to work with you.

A special mention for my co-promotor Jan Fransoo, for his guidance and his valuable feedbacks. Thank you for the support and the insightful conversations.

I thank Gerhard Woeginger and Frederic Semet for being part of my thesis committee. Thank you for your valuable comments, which have considerably improved the quality of this thesis.

I am grateful to all my current and former colleagues at the OPAC department. Especially, I would like to thank Emrah Demir and Mark Stobbe, with whom I had the pleasure to collaborate during the last year. A special thank goes to Kasper van der Vliet, José Vidal Vieira, Shaunak Dabadghao, Stefano Fazi, Maximiliano Udenio, Maryam Steadie Seifi, Yousef Ghiami, Vaeceslav Ghilas, Taimaz Soltani, Loe Schlicher, Baoxiang Li, Zümbül Atan, Tarkan Tan, Derya Sever, Duygu Taş and Hande de Korte-Cetinay. Lastly I would like to thank Claudine Hulsman-Paul, José van Dijk-Kok, Jolanda Verkuijlen-Nelissen, Christel van Berlo-Verlijsdonk and Geertje Kramer for their invaluable help.

As a PhD student I had the great pleasure to travel, visit new places and meet inspiring people. I take this chance to thank all those special people that I encountered during my travels. A special thank goes to Eva Barrena and Ola Jabali.

I would like to thank Alessandro de Gaspari for designing the perfect cover for my thesis. You caught what I had in mind and made it a thousand times better.

I am grateful to all my precious friends scattered around the world. Special thanks goes to Jenny Murari and Nadia Mantovani for their support and their help throughout difficult times.

Thanks to Maria Eriksson and Nemanja Cvijanovic, my "dutch" family, for keeping our house cozy and fun.

I am very grateful to my brother and my father for their belief in me and for their support.

I dedicate this thesis to the memory of my mother, my greatest *giant*, my constant source of courage and inspiration.

Eindhoven, 31^{th} July 2015

Contents

Parfois, la réalité est trop complexe. Une bonne histoire lui donne meilleure forme.★
Jean-Luc Godard

1 Introduction

Imagine you are in your favourite capital city. Most likely, you are now thinking about monuments, historical buildings, cozy cafés, nice restaurants, shops, cinemas, theatres and other attractions that remind you of that city. Now imagine you are in your favourite capital city during the peak hour. Most likely, the lovely picture you had in mind a few seconds ago has just vanished. In its place there is now the smell of gasoline and smog, the noise of vehicles and the screams of infuriated drivers stuck in a traffic jam.

Sadly, the coexistence of these two different faces is a common feature of many big cities. This phenomenon partially derives from the current patterns of urban development, which are not putting any of our cities on a sustainable pathway (Sorensen et al., 2004). The growing amount of these side effects, resulting from a policy of growth purely focused on economical aspects, have drawn attention towards the concept of sustainable development. Such a concept has been defined by Wheeler (1998) as "the development that improves the long-term social and ecological health of cities and towns".

Cities are constantly involved in complex and multiple processes of change. An essential contribution to the grow of a city comes from freight distribution, which ensures the attractiveness and the economic power of the city. As urban stores want to keep their inventory levels as low as possible, freight transport activities within urban areas mainly consist in frequent small volume deliveries, often referred as "last mile" activities. The great majority of these transport activities are performed by diesel trucks, as they are perceived to be the most suitable means of transport to perform deliveries within urban areas. As a consequence, a large number of diesel vehicles travels daily on the city roads, increasing traffic congestion and producing a substantial amount of greenhouses gas emissions.

★Sometime reality is too complex. A good story gives it form.

In this context, addressing the environmental externalities caused by transport activities, such as traffic congestion, noise pollution and car accidents, has become one of the major challenges of urban transport systems. This constant attempt to be innovative and competitive while limiting the negative environmental impacts, has been defined by Ehmke (2012b) as a fundamental dilemma of urban freight transportation.

1.1 City logistics

The very first studies on city logistics date back to the early 70's (see Taniguchi and Thompson (2014) and Crainic et al. (2009) for an overview of historical facts), since then this topic has been widely studied, (see e.g. Taniguchi et al., 2001; Crainic et al., 2009; Anand et al., 2012a; Ehmke, 2012b; Gonzalez-Feliu et al., 2014; Taniguchi and Thompson, 2014). A frequently quoted definition of city logistics is the following one from Taniguchi et al. (2014):

> *"the process for totally optimizing the logistics and transport activities by private companies with support of advanced information systems in urban areas considering the traffic environment, the traffic congestion, the traffic safety and the energy savings within the framework of a market economy".*

This definition highlights two important features of city logistics. The first one is the inherent sustainability of city logistics systems, that is, the integration of economical and environmental targets. The second one is the total optimization of logistics activities of private companies rather than local optimization (Taniguchi, 2014). A total optimization requires consideration of the needs of all stakeholders involved. According to Taniguchi (2014), the major stakeholders involved in the city logistics domain are shippers, city logistics service providers, city administrations and residents. These four stakeholders act autonomously without any centralized control in order to fulfill their own interests (Anand et al., 2012b). In particular, shippers want their products to be delivered within a set time-frame at the lowest price. Freight carriers try to meet the requirements of the shippers while optimizing the the usage of their own resources and minimizing the total transportation cost. City administrations want to improve the attractiveness of the city, by promoting economic health, green image and sustainability. Finally, the residents want to have safer and liveable cities. The heterogeneity of the stakeholders involved in this system and their different objectives increase the complexity of city logistics initiatives and undermine their efficiency. This in turn leads to the visible problems in urban freight transport, e.g., poor economic and environmental sustainability.

In the domain of city logistics during the last years several studies have been conducted on defining and improving the sustainability of a city logistic systems, see e.g., Taniguchi et al. (2013); Anand et al. (2012b); Browne et al. (2012); Gonzalez-Feliu et al. (2014). Browne et al. (2012) investigate the relationship between features and negative impacts of urban freight transport (see Fig 1.1).

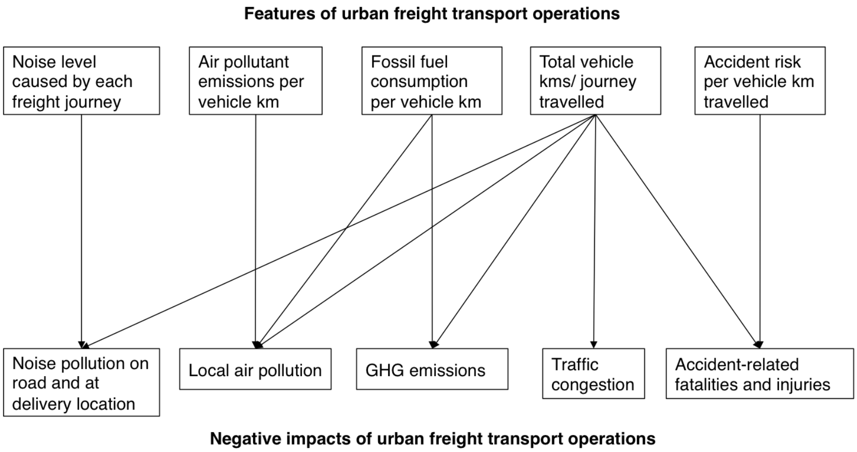


Figure 1.1 Relationship between features and negative impacts of urban freight transport (adapted from Browne et al., 2012).

This study shows that the feature with the largest negative impact is the total distance travelled by vehicles. Taniguchi et al. (2014) present an overview on modelling approaches developed in the last years for forecasting the entity of the negative impacts of urban freight transport and for investigating the benefits of targeted policies. The authors distinguish the following categories of modelling approaches: (i) fleet management, (ii) routing models, (iii) network modelling, (iv) life cycle analysis and (v) other models, including vehicle scheduling. These categories correspond to different levels of planning activities, i.e. strategic, tactical, operational and real-time (see Roy, 2001). In the remainder of this thesis we will mainly focus on the first two categories: (i) fleet management and (ii) routing and scheduling. In particular, one can think of this thesis as composed of two main parts. The first part, i.e. Chapter 2, focuses on including sustainability issues in the decision process of managing a fleet of vehicles at a strategic level. The second part, i.e. Chapters 3, 4 and 5, focuses on incorporating sustainability issues at the operational decision level. These two parts are discussed more in detail in the following sections.

1.2 Decision problems and research objectives

In the previous sections we sketched the environment for which the models presented in this thesis apply. In this section we briefly present an overview of the decision problems studied and we list the main research objectives of this thesis. At this stage we do not position the contribution of each chapter with respect to the literature. This is done individually in each chapter.

1.2.1 Fleet management

Planning the composition and the activities of a vehicle fleet in order to satisfy transportation service demands is a core strategic decision for most freight carriers. Its complexity, however, is such that fleet managers need the help of a decision support system in order to perform it adequately (Couillard and Martel, 1990).

In 2012, Transport for London (TfL), a local government body responsible for most aspects of the transport system in Greater London in England, published a guide (for London , TfL) to assist with the process of implementing sustainable fleet management. According to this guide, among other benefits, having a sustainable fleet contributes to (i) minimize the fuel costs and optimize carbon dioxide (CO2) based tax liabilities, (ii) minimize exposure to congestion and make more efficient use of company transport, (iii) support corporate sustainability goals, and (iv) provide a competitive edge in a market where environment credentials are becoming increasingly important to clients.
Driven by those reasons fleet managers have begun to reshape their priorities. Nowadays having a more green and sustainable fleet has become a common target. In a recent interview (Gray, 2013), Tim Anderson from the Energy Savings Trust reports:

> *"Saving money is a primary concern for all fleet managers. A greener fleet essentially means a more efficient fleet, which saves you money in the long run. Assembling a green fleet used to be very expensive but that's no longer the case and the safety benefits of having a newer fleet that's well maintained are marked."*.

This awareness towards sustainability has also affected city administrators, who have started to implement urban regulations that prioritize the usage of "green vehicles". As discussed in Quak and de Koster (2006) and Quak and de Koster (2009), a growing number of city administrators, with the aim of improving air quality and reducing the noise level in city centres, are prioritizing the access to central areas to low-emission vehicles by means of the implementation of Low Emission Zones (LEZ) or Zero Emission Zones (see ENCLOSE project report, 2014). As a consequence, freight companies operating in these cities are compelled to account for such constraints when planning the composition of their fleet of vehicles, as otherwise the company's business might be seriously affected.

In Chapter 2 we investigate how such regulations, imposed by municipalities or other institutions in order to improve the sustainability of the urban areas, affect the fleet composition of a logistic company working in these areas. Specifically, we study the strategic problem of managing a heterogeneous fleet of vehicles operating in a urban areas where access restrictions are applied to certain categories of vehicles. We consider different categories of vehicles, e.g. electric and diesel, which differ in terms of fixed cost, e.g., leasing cost, operative costs, e.g., fuel costs, electricity costs and capacities. We model the problem as an area partitioning problem where a rectangular service region has to be divided into sectors, each served by a single vehicle. We use a continuous approximation model (Daganzo, 1984a,b, 1987a,b) to calculate the distances traveled to serve each sector. The objective is to determine the best fleet composition and to assign each vehicle to a service sector so as to minimize the sum of ownership or leasing, transportation and labor costs, while satisfying the vehicle capacity constraints and the access restriction limits. To the best of our knowledge, this is the first study where operational restrictions such as city access regulations are incorporated in a fleet management problem. We develop an efficient dynamic programming (DP)-based algorithm to calculate the optimal solution for the case of two types of vehicles (e.g., electric and diesel), and we use a mixed integer linear programming (MILP) formulation for more general settings. We also derive some interesting insight on how the optimal fleet composition changes depending on the vehicle parameters. Finally we discuss the impact of city access restrictions on fleet composition.

The research objectives addressed in the first part of this thesis, dedicated to the strategical problem of optimizing the vehicle fleet composition, are the following:

Research objective 1 Develop a fleet management model to manage a (possibly heterogenous) fleet of vehicles to serve a city in the presence of access restrictions.

Research objective 2 Investigate the impact of traffic restrictions on urban fleet planning.

1.2.2 Routing and scheduling

Vehicle routing and scheduling problems are operational decision problems which consist of determining the optimal routes and the optimal schedules for a fleet of vehicles which has to visit a set of customer sites. The objective is to serve all customers at a minimum cost, under consideration of all restrictions, e.g. vehicle capacity and customer time window limits. These types of problem are solved on a regular basis by most of the logistics companies in order to efficiently plan the transport activities and minimize the operating costs.

The vehicle routing problem was introduced for the first time in 1959 by Dantzig and Ramser (1959). Since then it has been widely studied and several mathematical formulations have been proposed. As reported in Toth and Vigo (2014), there are more than 50 several types of vehicle routing problems that differ from each other in a number of ways. Some examples are the the pick-up and delivery problems, the multi-depot delivery problems, the delivery problems with stochastic travel time, the heterogeneous fleet delivery problems, and many others. For a comprehensive overview on vehicle routing problems we refer to Toth and Vigo (2014).

In the last years researchers have started to explore a new growing line of research, known as "green logistics", which aims to minimize the harmful effects of transportation activities. The characteristic of this research is the incorporation of environmental aspects in the routing and scheduling models, in addition to the traditional economical issues (see Toth and Vigo, 2014). According to Demir et al. (2014a) in August 2013 a total of 58 publications were associated with the reduction of fuel consumption in vehicle routing and scheduling. The authors show that between 2009 and 2013 the increase in the number of publications addressing sustainability issues in vehicle routing and scheduling problems is about 430%. In particular, most of these studies focus on optimizing the transportation system with respect to minimize the amount of CO_2e emissions produced by the vehicles. This is mainly due to the fact that there is a rich body of literature on analytical expressions for calculating the amount of emissions produced by the vehicles, while this is not the case for the other externalities, e.g. noise or accidents (Toth and Vigo, 2014). As shown in Figure 1.2, this amount depends on several factors correlated to the vehicles type, to the road features, etc. However, most of the studies on green logistics focus mainly on optimizing the vehicle load and the travel speed. In particular, the travel speed has been shown to be crucial in determining the amount of fuel consumed by vehicles, and consequently the amount of vehicles emissions (Demir et al., 2014a). To give the reader an idea on how these two factors are correlated we present Figure 1.3, which depicts the amount of fuel consumed by a vehicle as a function of the travel speed, calculated using the *Comprehensive Modal Emissions Modeling* (CMEM) by Barth et al. (2005) and Scora and Barth (2006). As is it known that the amount of emissions produced by a vehicle is directly proportional to the amount of fuel consumed (see e.g., Barth et al., 2005; Scora and Barth, 2006), Figure 1.3 shows the convexity of the fuel consumption upon the travel speed. This behaviour of the fuel function implies that traveling at very low or very high speed levels leads to higher fuel consumption, and therefore increases the amount of emissions produced by vehicles. For this reason when vehicles are stuck in traffic congestion they produce a larger amount of CO_2e emissions.

In this context, one novel scheduling problem has been identified, which consists of optimizing the travel speed of a vehicle visiting a given sequence of customer

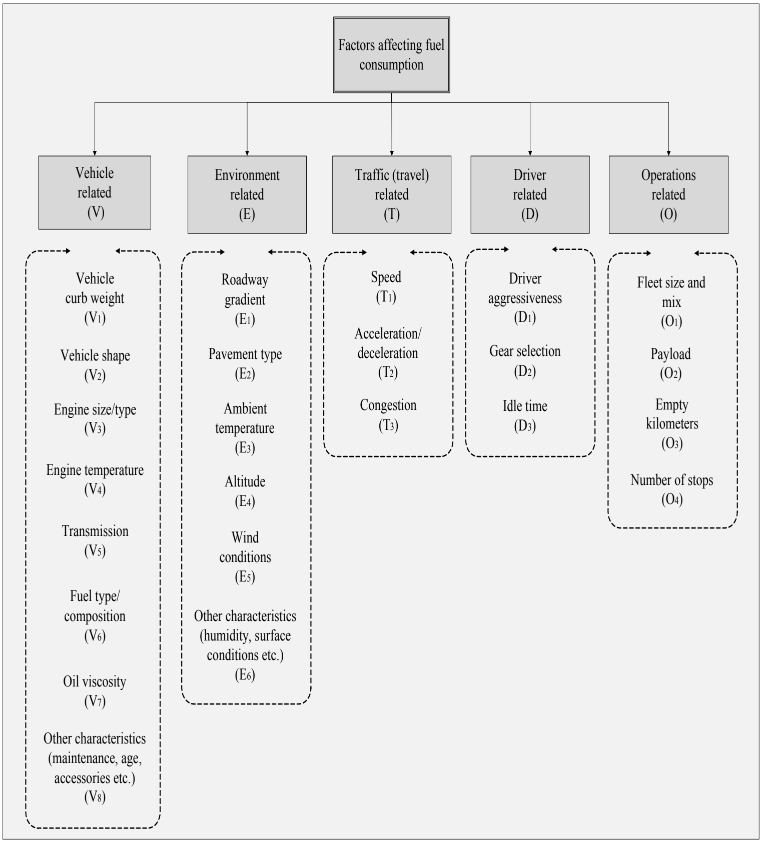


Figure 1.2 Factors affecting the amount of emissions produced by vehicles (adapted from Demir et al. (2014a))

locations (Toth and Vigo, 2014). This problem was first introduced by Hvattum et al. (2010, 2013) to optimize the sailing speed of a vessel with the aim to minimize the amount of fuel consumed. Recently, driven by the growing pressure on achieving sustainability targets, this speed optimization problem has found applications also in road transportation.

In the second part of this thesis, i.e., Chapters 3, 4 and 5, we focus on such operational decision problems, where the objective is to improve the efficiency and in particular the sustainability of the transport activities. In Chapters 3 and 4 we study the operational problem of routing a homogeneous fleet of vehicle in a presence of traffic congestion, which at peak periods, limits the travel speed of the vehicles and increases the amount of emissions produced. The objective is

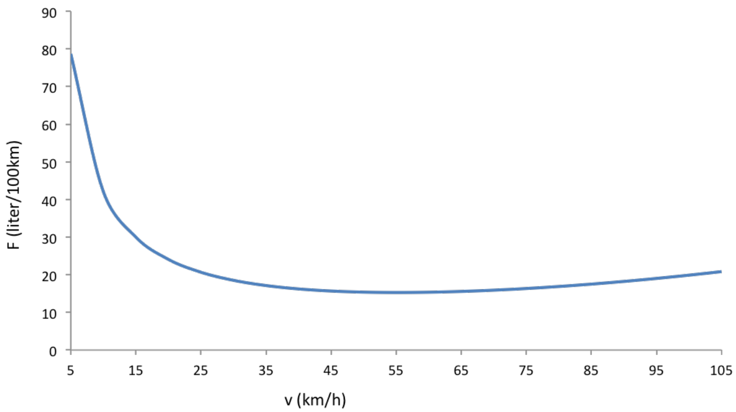


Figure 1.3 Fuel use rate F as a function of speed v

to determine the optimal set of routes and the optimal travel speed on each leg of a route so as to minimize a total transportation cost made of labour cost and emissions cost. To the best of our knowledge, this is the first study where both vehicle emissions and peak hours traffic congestion are accounted in a routing. We show that allowing the vehicle to wait at a customer site after the service has been completed, can be used as an effective strategy to avoid traveling in congestion and therefore to reduce the transportation costs. Next, we investigate the trade-off between emissions cost and driver wage and we propose a metaheuristic algorithm for solving the problem based on an Adaptive Large Neighborhood Search (ALNS) algorithm. The ALNS is a general framework introduced by Pisinger and Ropke (2007) and Ropke and Pisinger (2006a). The basic idea is to search for better solutions by partially destroying the current solution and by reconstructing it according some predefined criteria. To asses the quality of the algorithm we solve some benchmark instances and we compare our results with others from the literature. We perform some sensitivity analysis to better understand the efficiency of the destroy and repaid methods and we present the results of extensive computational experimentation. In Chapter 5 we study the problem of optimizing the travel speed of a vehicle visiting a given sequence of customer locations. Also in this case the objective is the minimization of the labour and emissions costs. First we formulate the problem as a dynamic program and we study the properties of the value function. Next we show how to recast the problem as a shortest path problem, exploiting some of the theoretical findings. This way, we provide a method to solve the problem to optimality in a quadratic time. Finally, we provide a heuristic algorithm for solving he scheduling problem in a presence of peak hours traffic congestion. To the best of our knowledge, this is the first study where both vehicle emissions and peak hours traffic congestion are accounted in a scheduling problem.

The research objectives addressed in the second part of this thesis, dedicated to operational problems such as vehicle routing and scheduling, are the following:

Research objective 3 Study the problem of routing and scheduling a homogeneous fleet of vehicles in a presence of traffic congestion which, at peak periods, limits the vehicles travel speed and increases the amount of emissions produced. The objective is the minimization of a total cost function including labour and emissions cost. Formulate the problem as a mathematical model and develop heuristic algorithm to solve to solve medium and large size instances in a reasonable amount of time.

Research objective 4 Study how idle waiting either at the depot or at a customer node affects the emissions and the labour costs, in a presence of traffic congestion.

Research objective 5 Study the scheduling problem of a vehicle visiting a given sequence of locations. The objective is to determine the optimal departure times and the travel speed on each leg of the route so as to minimize the sum of labour and emissions costs. Formulate the problem in mathematical terms and develop an exact algorithm for solving the problem. Extend the study to the case where traffic congestion limits the vehicle speed during peak periods. Develop a heuristic algorithm to solve this latter problem.

1.3 Outline of the thesis

Table 1.1 displays an online of this thesis based on the research objective listed in the previous section and on the research methodology used in each chapter, i.e. mixed integer programming formulation (MIP), dynamic programming (DP) and heuristic algorithm (HA).

Chapter 2 focuses on a fleet management problem. After a brief problem introduction, a MIP formulation is presented, followed by a DP-based algorithm. Chapters 3, 4 and 5 focus on vehicle routing and scheduling problems. Chapter 3 presents a novel MIIP formulation for a vehicle routing problem which account for both vehicle emissions and peak hour traffic congestion. Chapter 4 presents a metaheuristic algorithm for solving the problem introduced in Chapter 3. Finally, Chapter 5 focus on vehicle scheduling problems. The first part of the chapter presents an exact method for optimizing the travel speed and the departure times of a vehicle visiting a given sequence of nodes. The second part presents a heuristic algorithm for solving the same scheduling problem in a presence of peak hour traffic traffic congestion.

Table 1.1 Navigating the thesis by research objective and methodology

	Research objective					Research methodology		
Chapter	1	2	3	4	5	MIP	DP	HA
2	✔	✔				✔	✔	
3			✔	✔		✔		
4			✔	✔				✔
5					✔		✔	✔

Although this may seem a paradox, all exact science
is dominated by the idea of approximation.
Bertrand Russell, *The Scientific Outlook*

2 Strategic Fleet Planning for City Logistics

In this chapter we study the strategic problem of a logistics service provider managing a possibly heterogeneous fleet of vehicles to serve a city in the presence of access restrictions. We model the problem as an area partitioning problem where a rectangular service region has to be divided into sectors, each served by a single vehicle. The length of the routes, which depends on the dimension of the sectors and on customer density in the area, is calculated using a continuous approximation. The aim is to partition the area and to determine the type of vehicles to use in order to minimize the sum of ownership or leasing, transportation and labor costs. We formulate the problem as a mixed integer problem and as a dynamic program. We develop efficient algorithms to obtain an optimal solution and present some structural properties regarding the optimal partition of the service region and the set of vehicle types used. We also derive some interesting insights, namely we show that in some cases, traffic restrictions may actually increase the number of vehicles on the streets. Finally we study the benefits of operating a heterogeneous fleet of vehicles.

2.1 Introduction

Cities increasingly depend on efficient and sustainable freight transportation systems to ensure their attractiveness, economic power, and quality of life. The high concentration of small commercial activities which characterizes urban areas generally results in a very high number of vehicles movements, often uncoordinated and performed with less-than-truckload shipments. These have a substantial economic, environmental and social impact as cities are confronted with more traffic, congestion, noise and air pollution. The need for efficient and environmentally acceptable urban transportation schemes has given rise to the concept of city logistics (Ehmke, 2012a).
As discussed in the previous chapter, these growing environmental and economic concerns led to strategies aimed at improving the efficiency of transportation systems focused on the reduction of energy consumption and of vehicles emissions. One example is the introduction of electric vehicles into logistics fleets (Roumboutsos et al., 2014). Because of their high densities and relatively short distances, cities are particularly suited to the early adoption of alternative types of mobility (European Commission, 2013). In order to encourage the use of electric vehicles instead of diesel vehicles and to provide a clear incentive for investment in new low energy consumption vehicles, several cities have passed regulations limiting urban freight transport. Urban access regulations are often introduced to prioritize access for certain types of vehicles. There are currently no standard guidelines for such regulations: they may apply permanently or only at certain times of the day; similarly, they may be based on specific vehicle characteristics such as dimension, type of energy consumed, engine type, etc. (Muñuzuri et al., 2005). One example is the Dutch city of 's-Hertogenbosch, where a specific regulation was implemented to limit access of commercial traffic to the inner city. Green and silent trucks are allowed to enter the city center at any time, whereas other commercial freight vehicles are admitted only between 7:00 and 12:00, and between 18:00 and 20:00. Similar restrictions have been implemented in Utrecht, where a sustainable inner city delivery service was introduced. This service, called Cargohopper, performs last mile deliveries from a distribution center to the city center using a multi-trailer road train powered by a solar and battery-electric motor. Similarly, several Italian cities such as Rome, Milan, Bologna and Florence, now restrict the access of diesel vehicles to the city center at certain times of the day (e.g. from 7:00 to 20:00 in Bologna). These restrictions are known as ZTL (Limited Traffic Zone).

Faced with increasingly restrictive access regulations and with the need to reduce costs, energy use and greenhouse gas emissions, logistics service providers are looking for ways to better manage their vehicles fleet in order to increase their profitability and sustainability. Stewart (2012) conducts a series of interviews with fleet managers to gain some understanding of their purchasing policies. He reports that around half of the organizations surveyed would be willing to pay a

10% premium ownership costs for an electric vehicle, due to fuel savings, benefits of CO_2 reduction, as well as "green branding" (Stewart).

While there exists a rich body of literature on the fleet composition problem at the operational level, e.g. Golden et al. (1984) and Koç et al. (2014), relatively little has been done at the strategic level. One of the first publications of the fleet composition problem is due to Kirby (1959) who considers a homogeneous fleet. Loxton and Lin (2011) study a multi-period heterogeneous fleet dimensioning problem where the cost function is the sum of fixed, variable and hiring costs. They assume that the number of vehicles of a given type required in a certain period is known. Loxton et al. (2012) investigate a stochastic version of the problem, in which the future vehicle requirements follow a given probability distribution. Both studies present a solution method based on dynamic programming and golden section method. Finally, Jabali et al. (2012a) develop a continuous approximation model for the heterogeneous fleet composition problem. These authors present a mixed integer non-linear formulation along with upper and lower bounding procedures. Their study is the first in which operational aspects, such as vehicles routes, are incorporated within a strategic decision model.

In this chapter, we consider the strategic problem of determining an optimal fleet composition for a logistics service provider serving an urban area in the presence of access restrictions for certain types of vehicles. The problem is to determine the number and the types of vehicles to use, such as electric and diesel, in order to minimize the sum of ownership or leasing, transportation and labor costs. Specifically, we consider a rectangular urban area, called the *service region*, with a depot located on the edge of the area (this is motivated by the widespread policy of operating an urban consolidation center at the entrance of a city (Quak and de Koster, 2006)). We use a continuous approximation model (Daganzo, 1984a,b, 1987a,b) to calculate the distances traveled and we assume that customer demand is uniformly distributed over the service region. We partition this service region into contiguous rectangular blocks called *service sectors*, each served by a single vehicle. This partitioning policy is described in detail in §2.2. As observed in Huang et al. (2013) this way of distributing the workload among vehicles is useful in practical settings since it allows the drivers to be responsible for a particular area.

The contribution of this chapter is multifold. First, to the best of our knowledge, this is the first study where operational restrictions such as city access regulations are incorporated within a fleet management problem. Second, we propose an efficient dynamic programming (DP)-based algorithm to calculate the optimal solution for the case of two types of vehicles (e.g., electric and diesel), and we use a mixed integer linear programming (MILP) formulation for more general settings. We also establish structural results for the optimal partition of the service area served by a heterogeneous fleet of vehicles. Finally, we show how the optimal fleet composition changes depending on the vehicle parameters, and we discuss the impact of city access restrictions on fleet composition.

The remainder of this chapter is organized as follows. In §2.2, we describe the problem and we provide a MILP formulation. In §2.3, we present our DP formulation and derive analytical results: we first consider the single-strip-single-type case, then the single-strip-multiple-types case, and finally the multiple-strip-multiple-types case. In §2.4, we report some numerical results on the impact of city access restrictions and on the benefits of using a heterogeneous fleet of vehicles. We then compare the performance of our MILP and DP formulations.

2.2 Model

In this section we describe the problem setting. Subsequently we present the routing strategy and the partitioning policy. Finally, we introduce the MILP formulation.

2.2.1 Problem setting

We consider a rectangular service region of length L and width W, with a depot located in the south at a distance φ from the midpoint of the bottom edge of the rectangle (Figure 2.1). We refer to the closest edge of the rectangle as the 'bottom' edge and the furthest edge as the 'top' edge. We also use the term 'width' to refer to the size of horizontal edges and 'length' to the size of vertical edges, even if the vertical distance is smaller than the horizontal distance. We use the L_1 (Manhattan) norm to calculate distances. A number e of customers are located in this region, and are distributed according to a density function $\delta(x)$, where x is a point within the region. As in Daganzo (2005) (2005) and Huang et al. (2013), we assume that the density function $\delta(x)$ does not vary significantly within the region and therefore, without any significant loss of accuracy, it is approximated by a continuous function $\delta \approx e/WL$. This approximation is reasonable in megacities and metropolitan areas where a large number of retailers are distributed evenly.

Different vehicle types can be used to perform the deliveries within the service region, for example, electric and diesel vehicles. The vehicle types differ in their capacity, as well as in their usage cost, which is made up of two components: a fixed cost (if the vehicles are purchased, this cost is the depreciation on the purchase amount; if they are leased, this cost is the rental price paid per vehicle per shift), and a variable cost, which is proportional to the distance traveled. We also consider a limit on the duration of the vehicle routes as in Jabali et al. (2012a) and Langevin and Soumis (1989); Langevin et al. (1996); Langevin and Soumis (1989). However, in our setting the time limit is allowed to differ across vehicle types. This time limit is motivated by the city center traffic restrictions discussed

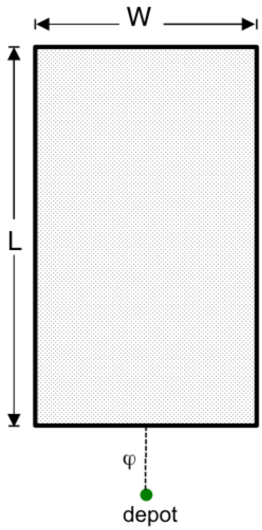


Figure 2.1 Urban service area.

in the introduction of this chapter. We assume that the vehicle types with the larger transportation capacity have larger variable costs and stricter time access restrictions, since in practice, cities tend to impose further access restrictions on larger delivery vehicles which are also more expensive to operate. We do not make any assumption on how the fixed costs compare across vehicle types.

Our problem consist of partitioning the service region into contiguous rectangles corresponding to the service sectors, each served by a single vehicle, so that all customers are served by delivery vehicles which do not exceed the capacity and route duration constraints. The objective is the minimization of the total travel cost which is the sum of the fixed vehicle cost, the variable vehicle cost and the driver wages. We describe how these costs are calculated in the following sections.

2.2.2 Routing strategy

We use a continuous approximation model of the type first proposed by Daganzo (1987a), and known as the *dual strip* strategy or *half-width routing* strategy, to calculate the total distance traveled by a vehicle in a service sector. Let w be the width of the service sector and y be its length, so that the number of customers to visit in this sector is δwy. Also, let μ be the distance between the depot and the midpoint of the bottom edge of the sector. According to the half-width routing strategy, the sector is divided into two halves along its width. The vehicle enters from the middle point of the bottom edge then makes a single round trip within the sector visiting all customers without backtracking, and finally exits the sector from the point of entry (see Figure 2.2). According to the half-width routing

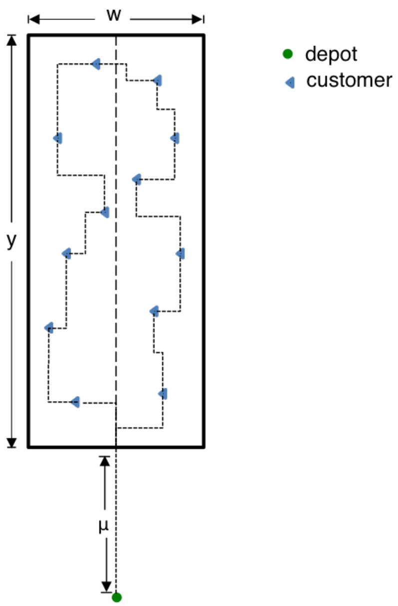


Figure 2.2 Delivery tour of a vehicle in a rectangular service sector.

strategy, the total distance γ made up of the *transit distance* and of the distance traveled within the sector, can be approximated by

$$\gamma = 2\mu + 2y + \frac{yw^2\delta}{6}. \tag{2.1}$$

In this expression, 2μ is the transit distance, $2y$ is the approximate vertical distance traveled by the vehicle within the sector, and $yw^2\delta/6$ is the approximate horizontal distance within the sector (seeDaganzo (1987a) for more details).

2.2.3 Partitioning policy

We assume that the rectangular service area is partitioned into s strips having the same width W/s. Each strip is then divided into a number of sectors. As depicted Figure 2.3, we assume that the strips are numbered from left to right. Similarly, the sectors in each strip are numbered from bottom to top. Let y_{ij}^s denote the length of the j^{th} sector in strip i when the service area is partitioned into s strips. Let φ_i^s denote the distance from the depot to the middle point of the bottom edge

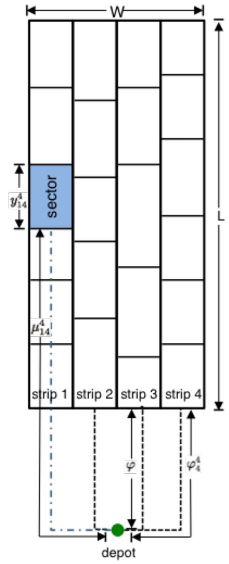


Figure 2.3 Example of area partitioning.

of strip i. This value is $\varphi_i^s = \varphi + \frac{|s+1-2i|}{s}\frac{W}{2}$, where φ is the vertical distance between the depot and the middle point of the bottom edge of the service area, and the second set of terms is the horizontal distance from this point to the middle point of the bottom edge of strip i. The vehicle that services the j^{th} sector in strip i must first drive through sectors 1 to $j-1$ in order to reach the assigned sector. Therefore, the total transit distance from the depot to the bottom edge of the j^{th} sector is $\mu_{ij}^s = \varphi_i^s + \sum_{l=1}^{j-1} y_{il}^s$, where the first term is the distance between the depot and the bottom edge of strip i, and the second one is the distance between the bottom edge of the strip and that of the sector.

From (2.1), the total distance γ_{ij}^s traveled by a vehicle to serve customers in the j^{th} sector in strip i when there are s strips can be approximated by

$$\gamma_{ij}^s = 2\mu_{ij}^s + 2y_{ij}^s + \frac{y_{ij}^s \delta W^2}{6s^2} = 2\left(\varphi + \frac{|s+1-2i|}{s}\frac{W}{2} + \sum_{l=1}^{j} y_{il}^s\right) + \frac{y_{ij}^s \delta W^2}{6s^2}. \quad (2.2)$$

2.2.4 MILP formulation

We now show how to formulate the area partitioning problem as a MILP. Let K be the number of possible vehicle types. For every type $k \in \{1, \ldots, K\}$, let Q_k denote the vehicle capacity, let f_k be the vehicle fixed cost, and let o_k be its variable cost. Let T_k be the maximum route duration for a vehicle of type k. We label the vehicle types so that $Q_1 \leq \ldots \leq Q_K$, $o_1 \leq \ldots \leq o_K$ and $T_1 \geq \ldots \geq T_K$.

Let v be the travel speed, which is assumed to be constant and identical for all vehicle types (this is a realistic assumption in the context of a congested city center), let h be the service time at the customer locations (i.e., the time to unload the goods), and let d denote the driver wage (in £ per time unit).

Let $\overline{s}$ be the maximum number of strips in which the region can be partitioned, and let $\overline{m}_i^s$ be the maximum number of sectors in which the i^{th} strip can be partitioned when there are s strips in total. We show how to calculate these values in §2.3 (Lemmas 2.2) and in Appendix 2.A.

The decision variables are as follows:

- x^s: binary variable equal to 1 if the area is partitioned into s strips, 0 otherwise;
- z_{ij}^{ks}: binary variable equal to 1 if the area is partitioned into s strips, and sector $j \in \{1, \ldots, \overline{m}_i^s\}$ in strip $i \in \{1, \ldots, \overline{s}\}$ is served by vehicle $k \in \{1, \ldots, K\}$, 0 otherwise;
- y_{ij}^{ks}: length of sector j in strip i when the area is partitioned into s strips;
- γ_{ij}^{ks}: total distance traveled by vehicle k to serve sector j in strip i when the area is partitioned into s strips;
- μ_{ij}^{ks}: distance between the depot and the beginning of sector j in strip i when the area is partitioned into s strips.

The value of the last three variables is positive if the area is partitioned into s strips and sector j in strip i is served by vehicle k, otherwise it is 0.

The formulation is

$$\text{Minimize} \sum_{s=1}^{\overline{s}} \sum_{i=1}^{s} \sum_{j=1}^{\overline{m}_i^s} \sum_{k=1}^{K} f_k z_{ij}^{ks} + \sum_{s=1}^{\overline{s}} \sum_{i=1}^{s} \sum_{j=1}^{\overline{m}_i^s} \sum_{k=1}^{K} o_k \gamma_{ij}^{ks} + \sum_{s=1}^{\overline{s}} \sum_{i=1}^{s} \sum_{j=1}^{\overline{m}_i^s} \sum_{k=1}^{K} d \frac{\gamma_{ij}^{ks}}{v} + dh\delta LW \tag{2.3}$$

subject to

$$\sum_{s=1}^{\overline{s}} x^s = 1 \tag{2.4}$$

$$\sum_{k=1}^{K} z_{ij}^{ks} \leq x^s \qquad s = 1, \ldots, \overline{s}, i = 1, \ldots, s, j = 1, \ldots, \overline{m}_i^s \tag{2.5}$$

$$\delta y_{ij}^{ks} \frac{W}{s} \leq Q_k z_{ij}^{ks} \qquad s = 1, \ldots, \overline{s}, i = 1, \ldots, s, j = 1, \ldots, \overline{m}_i^s, k = 1, \ldots, K \tag{2.6}$$

$$\sum_{j=1}^{\overline{m}_i^s}\sum_{k=1}^{K} y_{ij}^{ks} = Lx^s \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s \tag{2.7}$$

$$\mu_{ij}^{ks} \geq \varphi_i^s + \sum_{l=1}^{j-1}\sum_{k=1}^{K} y_{il}^{ks} - M(1 - z_{ij}^{ks}) \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.8}$$

$$\gamma_{ij}^{ks} = 2(\mu_{ij}^{ks} + y_{ij}^{ks}) + y_{ij}^{ks}\delta W^2/(6s^2) \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.9}$$

$$\gamma_{ij}^{ks}/v + y_{ij}^{ks}\delta Wh \leq T_k \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.10}$$

$$x^s \in \{0,1\} \qquad s = 1,\ldots,\overline{s} \tag{2.11}$$

$$z_{ij}^{ks} \in \{0,1\} \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.12}$$

$$y_{ij}^{ks} \geq 0 \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.13}$$

$$\gamma_{ij}^{ks} \geq 0 \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K \tag{2.14}$$

$$\mu_{ij}^{ks} \geq 0 \qquad s = 1,\ldots,\overline{s}, i = 1,\ldots,s, j = 1,\ldots,\overline{m}_i^s, k = 1,\ldots,K. \tag{2.15}$$

The objective function is the sum of four terms: the vehicle fixed cost, the vehicle variable cost, the driver wage for the time spent traveling, and the driver wage for the time spent serving the customer, the last term being a constant. Constraint (2.4) guarantees that the area is partitioned into a positive number of strips. Constraints (2.5) ensure that each sector is served by at most one vehicle type. Constraints (2.6) mean that the capacity of the vehicle is not exceeded. Constraints (2.7) guarantee that the sum of the lengths of the sectors in every strip is equal to L. Constraints (2.8) compute the distance between the depot and the beginning of sector j in strip i (to speed up the calculations, M can be replaced by $\varphi + L$). Constraints (2.9) calculate the total distance traveled by a vehicle to service sector j in strip i. Constraints (2.10) ensure that the total time required to service sector j in strip i does not exceed the maximum tour length. Finally the domains of the variables are defined in the last five constraints. As shown in §2.4 this MILP may be slow to generate a solution. In the remainder of this chapter we study some analytical properties of the problem, which will be used as a basis for developing a fast solution procedure.

2.3 Analytical results

The notation used in the chapter is presented in Table 2.1. Without loss of generality in the rest of the chapter we assume $d = 0$.

Table 2.1 Summary of the notation.

Symbol	Definition
L	length of service region
W	width of service region
φ	distance from depot to bottom edge of the service area
φ_i^s	distance from depot to middle point of bottom edge of strip i when there are s strips
δ	customer density
d	driver wage
v	vehicle speed
h	service time
K	number of vehicle types
f_k	fixed cost for vehicle of type k
o_k	variable cost for vehicle of type k
T_k	maximum tour duration for a vehicle of type k
Q_k	capacity of vehicle of type k
$\overline{s}$	maximum number of strips
s	number of strips
m	number of sectors
w	width of each strip
γ	total distance traveled by vehicle
t_j	vehicle type used in sector j when there is only 1 strip
t_{ij}^s	vehicle type used in sector j of strip i when there are s strips
y_j	length of sector j when there is only 1 strip
y_{ij}^s	length of sector j in strip i when there are s strips
μ_{ij}^2	transit distance from depot to sector j in strip i when there are s strips
$\overline{m}_i^s$	maximum number of sectors in which the i^{th} strip can be partitioned when there are s strips

2.3.1 Single strip, one vehicle type

Here we consider a special case of our problem where there is only one strip, i.e., $s = 1$ and only one vehicle ($K = 1$) with fixed cost f, variable cost o, capacity Q, and maximum tour length T. Let $w = W$ denote the width of the strip. The problem is to determine the number and therefore the length of the sectors in the strip: let y_j be the length of the j^{th} sector and m be the chosen number of sectors. Let C denote the total cost. The problem can be written as

$$\begin{aligned}\min_{m,(y_1,\ldots,y_m)} C(y_1,\ldots,y_m) &= mf + o\sum_{j=1}^{m}\left(2\left(\varphi+\sum_{i=1}^{j}y_i\right)+\frac{y_j w^2\delta}{6}\right) \\ &= m\left(f+2o\varphi\right)+2o\left(my_1+(m-1)y_2+....+y_m\right) \\ &\quad +o\frac{\delta w^2}{6}L \end{aligned} \tag{2.16}$$

subject to

$$\sum_{j=1}^{m} y_j = L \tag{2.17}$$

$$\delta y_j w \leq Q \qquad j = 1,...,m \tag{2.18}$$

$$\frac{1}{v}\left[2\left(\varphi+\sum_{i=1}^{j} y_i\right)+\frac{y_j w^2\delta}{6}\right]+h\delta y_j w \le T \qquad j=1,...,m \tag{2.19}$$

$$y_j \ge 0 \qquad j=1,...,m \tag{2.20}$$

$$m \in \mathcal{N}_+. \tag{2.21}$$

The first term of the objective function is the total fixed cost and the second term is the total variable cost. Constraint (2.17) guarantees that the entire strip is covered by sectors, constraints (2.18) and (2.19) ensure than the vehicle capacity and maximum tour length are not exceeded. Constraints (2.20) and (2.21) define the domains of the decision variables. Note that for the problem to be feasible we need $T > 2(\varphi + L)/v$, otherwise reaching the top of the strip would take more than T units of time, leaving no time to serve the customers.

We formulate this problem as a DP. Let $g(y;l)$ be the cost of serving a sector of length y with a top edge at a distance of l from the bottom of the strip (and therefore a bottom edge at a distance of $l-y$ from the bottom of the strip), where $l \in [0, L]$. From (2.1), we have $g(y;l) = f + o\gamma = f + o\big(2(\varphi + l) + yw^2\delta/6\big)$. Let $V(l)$ be the value function, which is the total cost of serving the customers located at a vertical distance less than $l + \varphi$ from the depot, or equivalently, at a vertical distance of l from the bottom of the strip. Our goal is to calculate $V(L)$. The DP recursion is

$$V(l) \quad = \quad \begin{cases} \min_{0\le y\le \overline{y}(l)} g(y;l) + V(l-y) & \text{if } 0 < l \le L \\ 0 & \text{if } l \le 0, \end{cases} \tag{2.22}$$

where $\overline{y}(l)$ is the maximum length for a sector with a top edge at a distance of l from the bottom of the strip:

$$\overline{y}(l) \quad = \quad \min\left\{l, \frac{Q}{\delta w}, \frac{6(Tv - 2(l+\varphi))}{w^2\delta + 6whv\delta}\right\}. \tag{2.23}$$

In this expression, the first term comes from the fact that the length of the sector cannot exceed the remaining uncovered portion of the strip, the second term comes from rewriting (2.18) as an equation and solving it for y_j, and the third term is obtained by rewriting (2.19) as an equation with $l = \sum_{i=1}^{j} y_i$ and solving it for y_j. Note that for $l \in [0, L]$, $\overline{y}(l) \ge 0$ by the feasibility condition.

Our first result states that it is always optimal to set the length of a sector equal to its maximum value. Let $l_j = \sum_{i=1}^{j-1} y_i$ denote the distance from the top of sector j to the bottom of the strip.

Proposition 2.1 *It is optimal to set the length of each sector equal to its maximum, i.e.,* $y_i = \overline{y}(l_i)$ *for* $i = 1, ..., m$.

Proof: The proof is by contradiction. Let $\hat{s}$ be the lowest index such that $y_{\hat{s}} \neq \overline{y}(l_{\hat{s}})$ in the optimal solution. Since $\overline{y}(l)$ is the maximum value satisfying constraints (2.18) and (2.19), we must have $y_{\hat{s}} < \overline{y}(l_{\hat{s}})$. Also $\hat{s} > 1$ since by (2.17), $y_1 = l_1 = \overline{y}(l_1)$. Consider an alternate solution with the same number of sectors and same sector length for all sectors, except sectors $\hat{s}$ and $\hat{s}-1$, such that the length of sector $\hat{s}$ is increased by ϵ and the length of sector $\hat{s}-1$ is decreased by ϵ, where ϵ is a small positive value. The difference in total cost between the optimal and the alternate solutions is

$$\begin{aligned}
& g(y_{\hat{s}}, l_{\hat{s}}) + g(y_{\hat{s}-1}, l_{\hat{s}} - y_{\hat{s}}) - g(y_{\hat{s}} + \epsilon, l_{\hat{s}}) - g(y_{\hat{s}-1} - \epsilon, l_{\hat{s}} - y_{\hat{s}} - \epsilon) \\
= \; & o\left(2l_{\hat{s}} + y_{\hat{s}}\frac{w^2\delta}{6}\right) + o\left(2(l_{\hat{s}} - y_{\hat{s}}) + y_{\hat{s}+1}\frac{w^2\delta}{6}\right) - o\left(2l_{\hat{s}} + (y_{\hat{s}} + \epsilon)\frac{w^2\delta}{6}\right) \\
& -o\left(2(l_{\hat{s}} - y_{\hat{s}} - \epsilon) + (y_{\hat{s}+1} - \epsilon)\frac{w^2\delta}{6}\right) \\
= \; & 2o\epsilon < 0,
\end{aligned}$$

which is a contradiction. □

Based on Proposition 2.1, we propose Algorithm 1, which is a recursive method to calculate the optimal partition of the strip into sectors. The intuition behind

Algorithm 1: Optimal partition of strip into sectors with one vehicle type.

Step 0: Set $l = L$ and $j = 1$.
Step 1: $y_j = \overline{y}(l)$.
if $\overline{y}(l) > l$ **then**
  set $l = L - \overline{y}(l)$ and $j = j + 1$ then repeat Step 1
else
  Stop
Step 2: $m = j$. Renumber the sectors: $y_j := y_{m-j+1}$ for $j = 1, ..., m$.

Proposition 2.1 and Algorithm 1 is that we need to make the sectors as long as possible, that is, as long as permitted by the capacity of the vehicle and the maximum route duration. The only sector for which these constraints might be unbinding is the one closest to the depot: the vehicle assigned to that sector just covers the leftover part of the strip. It is optimal to make the shortest sector the one closest to the depot because vehicles need to drive through previous sectors on their way to their service sector, and therefore it is optimal to keep the distance to the start of each sector as low as possible. We see from Equation (2.16) that the length y_1 of the first sector has the largest multiplier. Hence, it should be minimized.

In the special case where $T \geq 12(L + \varphi) + Q(w + 6h)/(6v)$, we can provide a closed-form expression for the optimal solution: $m = \lceil L\delta w/Q \rceil$, $y_j = Q/\delta w$ for $j = 2, ..., m$ and $y_1 = L - (m-1)Q/\delta w$; in this case, the constraint on the maximum tour length is so loose that the length of sectors 2 to m is determined by the maximum vehicle capacity Q.

Example 2.1 Let $L = 55$, $w = 10$, $\delta = 0.5$, $\varphi = 0$ and $v = 30$. There is one type of vehicle with $f = 70$, $o = 5$, $Q = 50$ and $T = 6$.

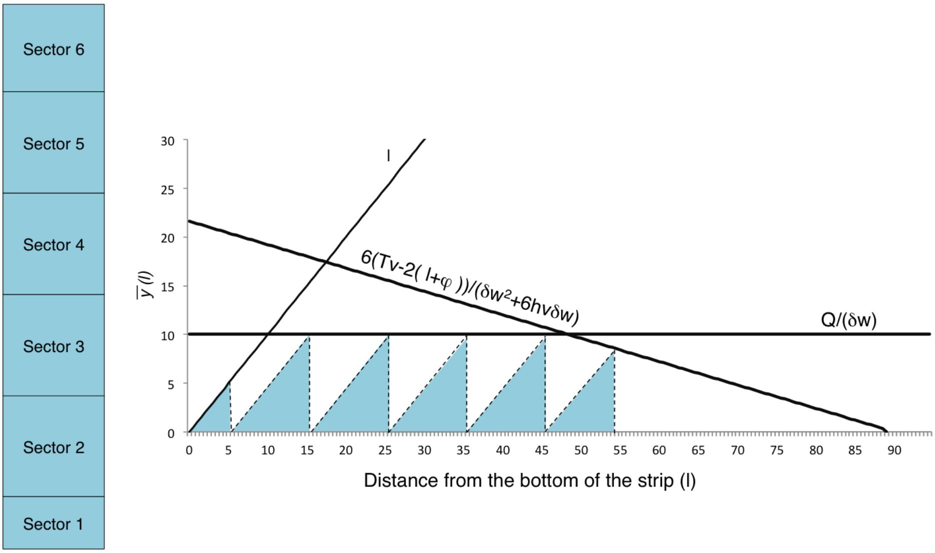


Figure 2.4 Optimal solution.

We provide a two-part graphical representation of the optimal partition of the strip in Figure 2.4, which is helpful in understanding how it is obtained. The right part of the figure is a graph where the X-axis represents the distance from the bottom of the strip (which is the distance from the depot, minus φ) and the Y-axis represents the three components of the $\overline{y}(l)$ function represented by the solid black lines. The optimal solution can be obtained graphically as follows: (i) start at a value equal to L on the X-axis and measure the height of the $\overline{y}(l)$ function at this point; this value is the optimal length of the last sector, (ii) from this point on the $\overline{y}(l)$ curve, draw a line parallel to the 45-degree line until reaching the X-axis again; this value is the distance from the bottom of the strip to the top of the second last sector, (iii) measure the height of the $\overline{y}(l)$ function at this point; this value is the optimal length of the second last sector. Repeat these steps until reaching the origin.

In this example, the optimal solution contains six sectors. Sector 6, which is the furthest away from the depot is constrained by the maximum route duration and has length 8.4. Sectors 2 to 5 are constrained by the vehicle capacity and have length 10. Finally the remainder of the strip length is allocated to the first sector, which has length 6.6.

Next we derive some monotonicity properties for the optimal number of vehicles.

Lemma 2.1 *The optimal number of sectors m^* is independent of f and o, is non-decreasing in L, φ, δ and w, and non-increasing in T and Q.*

Proof: First we show that m^* is non-decreasing in L. Consider two strips with respective lengths L' and L'' such that $L' < L''$. For both strips, we use Algorithm 1 to obtain the optimal number of sectors. Let l' and l'' be the variable used in this algorithm when the length of the strip is L' and L'' respectively. In the first iteration, we have $l' = L' < l'' = L''$. There are three cases: (i) if $\overline{y}(l'') = l''$, then $\overline{y}(l') = l'$ and the number of sector is the same for both strips; (ii) if $\overline{y}(l'')<l''$ and $\overline{y}(l') = l'$, then the method stops for l' but not for l'', which means that there is at least one more sector with l''; (iii) if $\overline{y}(l'') < l''$ and $\overline{y}(l') < l'$, then the algorithm continues for both strips. Also in this case, we must have $\overline{y}(l') \geq \overline{y}(l'')$ because $\overline{y}(l)$ is non-increasing in l for values of l such that $\overline{y}(l) < l$ (see Figure 2.4). As a result the next iteration starts with $l'' := l'' - \overline{y}(l'')$ and $l' := l' - \overline{y}(l')$, and $l' < l''$, which is a similar starting point. We can therefore repeat the same argument. Since there is no case in which the strip with the greater length stops the recursive method before the strip with the shorter length does, the result must be true.

The fact that m^* is non-decreasing in φ, δ and w and non-increasing in T and Q follows directly from the fact that $\overline{y}(l)$ is non-increasing in φ, δ and w and non-decreasing in T and Q, as can be seen from (2.23). Given that $\overline{y}(l)$ does not depend on f and o, it follows that m^* is independent of these two cost parameters. □

The optimal solution always minimizes the total number of sectors and hence, the total number of vehicles used. For this reason, when there is a single vehicle type, the fixed and variable vehicle cost parameters f and o are not relevant, that is, the solution obtained from Algorithm 1 remains optimal for any value of o and f, including $f = 0$. The other relationships are as follows: the total number of vehicles is non-decreasing in the length of the strip, the distance from the depot and the number of customers, and non-increasing in the vehicle capacity and the maximum route length. Note that some of these intuitive relationships no longer hold when there are several types of vehicle, as shown in §2.3.2.

2.3.2 Single strip, multiple vehicle types

Here, we keep the assumption that there is a single strip of width w, but we allow the firm to choose between vehicles of K different types. As in §2.3.1, let y_j denote the length of the j^{th} sector and $l_j = \sum_{i=1}^{j-1} y_i$ denote the distance from the top of sector j to the bottom of the strip. Let $t_j \in \{1, ..., K\}$ denote the type of the vehicle serving the j^{th} sector, so that f_{t_j}, o_{t_j}, Q_{t_j} and T_{t_j} respectively denote the fixed cost, variable cost, the capacity and the maximum tour duration of the vehicle used to serve the j^{th} sector. This problem can be written as

$$\min_{m,(y_1,\ldots,y_m),(t_1,\ldots,t_m)} C(y_1, \ldots, y_m, t_1, ..., t_m) =$$
$$= \sum_{j=1}^{m} f_{t_j} + \sum_{j=1}^{m} o_{t_j} \left(2\left(\varphi + \sum_{k=1}^{j} y_k\right) + \frac{y_j \delta w^2}{6} \right)$$

subject to

$$\begin{aligned}
\sum_{j=1}^{m} y_j &= L && \\
\delta w y_j &\le Q_{t_j} && j = 1, ..., m \\
\frac{1}{v}\left[2\left(\varphi + \sum_{i=1}^{j} y_i\right) + \frac{y_j w^2 \delta}{6}\right] + h\delta y_j &\le T_{t_j} && j = 1, ..., m \\
y_j &\ge 0 && j = 1, ..., m \\
t_j &\in \{1, 2\} && j = 1, ..., m \\
m &\in \mathcal{N}_+. &&
\end{aligned}$$

We need $\max_{k=1,\ldots,K} T_k > 2(\varphi + L)/v$, for the problem to be feasible. As in the previous section, we formulate the problem as a DP. Let $g_k(y; l)$ denote the cost of serving a sector of length y, which has a top edge at a distance of l from the bottom of the strip, with a vehicle of type k. We have $g_k(y; l) = f_k + o_k \left(2(\varphi + l) + yw^2\delta/6\right)$. The DP recursion in this case is

$$V(l) \quad = \quad \begin{cases} \min_{k=1,\ldots,K} \left\{ \min_{0 \le y \le \overline{y}_k(l)} \{g_k(y; l) + V(l - y)\} \right\} & \text{if } 0 \le l \le L \\ 0 & \text{if } l \le 0 \end{cases} \tag{2.24}$$

where, for every vehicle type $k \in \{1, ..., K\}$, the maximum length for a sector ending at a distance of l from the bottom of the strip is $\overline{y}_k(l) = \min\{l, Q_k/(\delta w), 6(T_k v - 2(l + \varphi))/(\delta w^2 + 6hv\delta w)\}$ (if $\overline{y}_k$ is negative, vehicles of type k cannot be used to feasibly serve the area). Note that $\overline{y}_k(l)$ depends on the vehicle type k, through the capacity Q_k and maximum tour duration T_k. As before, our goal is to calculate $V(L)$. We first analyze the structure of the optimal solution.

Proposition 2.2 *An optimal solution contains at most one sector with a length shorter than its maximum value, i.e., we have* $y_i = \overline{y}_{t_i}(l_i)$, *for all sectors* $i = 1, ..., m$, *except possibly for one of them. If such a sector exists, then assume it is sector* j, *i.e.,* $y_j < \overline{y}_{t_j}(l_j)$. *In this case, the following properties must hold: (i)* $j > 1$, *(ii)* $o_{t_j} > o_{t_i}\left(12 + \delta w^2\right)/\delta w^2$ *for* $i = 1, \dots j-1$, *(iii)* $y_i = \overline{y}_{t_i}(l_i) = \frac{Q_{t_i}}{w\delta}$ *for* $i = 1, ..., j-1$, *and (iv)* $\frac{Q_{t_{j-1}}}{w\delta} \leq y_j$.

Proof: Property (i) holds because the length of the first sector is always equal to its maximum since $y_1 = l_1 = \overline{y}_{t_1}(l_1)$. Property (ii) can be proven by contradiction. Suppose there exists a sector $k < j$ such that $o_{t_j} \leq o_{t_k}(12+\delta w^2)/(\delta w^2)$. Consider an alternate solution with the same number of sectors m and the same vehicle types used in each sector, but with sector lengths $y'_1, \dots, y'_m$ such that $y'_i = y_i$ for $i \neq k, j$, $y'_j = y_j + \epsilon$ and $y'_k = y_k - \epsilon$, with $\epsilon \in (0, \overline{y}_{t_j}(l_j) - y_j)$. In other words only sectors j and k are different in the two solutions. This alternate solution is feasible since $\overline{y}_{t_i}(l_i)$ is non-increasing in l_i for $i > 1$. The difference in total cost between the optimal and the alternate solution is given by

$$
\begin{aligned}
& \sum_{i=k}^{j} g_{t_i}\left(y_i, L - \sum_{l=i+1}^{m} y_l\right) - \sum_{i=k}^{j} g_{t_i}\left(y'_i, L - \sum_{l=i+1}^{m} y'_l\right) \\
= \; & f_{t_j} + o_{t_j}\left[2(l_j+\varphi) + \frac{y_j w^2\delta}{6}\right] + f_{t_k} + o_{t_k}\left[2(l_k+\varphi) + \frac{y_k w^2\delta}{6}\right] + \sum_{i=k+1}^{j-1} f_{t_i} \\
& + \sum_{i=k+1}^{j-1} o_{t_i}\left[2(l_i+\varphi) + \frac{y_i w^2\delta}{6}\right] - f_{t_j} - o_{t_j}\left[2(l_j+\varphi) + \frac{y'_j w^2\delta}{6}\right] - f_{t_k} \\
& - o_{t_k}\left[2(l_k+\varphi-\epsilon) + \frac{y'_k w^2\delta}{6}\right] - \sum_{i=k+1}^{j-1} f_{t_i} - \sum_{i=k+1}^{j-1} o_{t_i}\left[2(l_i+\varphi-\epsilon) - \frac{y'_i w^2\delta}{6}\right] \\
= \; & o_{t_j}\left[2(l_j+\varphi) + \frac{y_j w^2\delta}{6}\right] + o_{t_k}\left[2(l_k+\varphi) + \frac{y_k w^2\delta}{6}\right] + \sum_{i=k+1}^{j-1} o_{t_i}\left[2(l_i+\varphi) + \frac{y_i w^2\delta}{6}\right] \\
& - o_{t_j}\left[2(l_j+\varphi) + \frac{(y_j+\epsilon) w^2\delta}{6}\right] - o_{t_k}\left[2\left(l_k+\varphi-\epsilon\right) + \frac{(y_k-\epsilon) w^2\delta}{6}\right] \\
= \; & -\sum_{i=k+1}^{j-1} o_{t_i}\left[2\left(l_i+\varphi-\epsilon\right) - \frac{y_i w^2\delta}{6}\right] - o_{t_j}\frac{\epsilon w^2\delta}{6} + o_{t_k}\left(2\epsilon + \frac{\epsilon w^2\delta}{6}\right) + 2\epsilon\sum_{i=k+1}^{j-1} o_{t_i} \\
> \; & \left(o_{t_k}\frac{12+w^2\delta}{6} - o_{t_j}\frac{w^2\delta}{6}\right)\epsilon.
\end{aligned}
$$

The last term is positive since $o_{t_j} \leq o_{t_k}\left(\frac{12+\delta w^2}{\delta w^2}\right)$ by the contradiction assumption; hence, we have a contradiction. The proof of property (iii) consists of two parts:

(a) we show that it is never optimal to have a sector $k < j$ such that $\overline{y}_{t_k}(l_k) = \frac{6(T_{t_k}v-2(l_k)-\varphi)}{w^2\delta+6whv\delta}$; (b) we show that it is never optimal to have a sector $k < j$ such that $\overline{y}_{t_k}(l_k) < \frac{Q_{t_k}}{w\delta}$.

Part (a) The proof is by contradiction. Suppose there exists a sector $k < j$ such that $y_k \leq \overline{y}_{t_k}(l_k) = \frac{6(T_{t_k}v-2l_k+\varphi)}{w^2\delta+6whv\delta}$. Let $\alpha = \overline{y}_{t_k}(l_k) - y_k \geq 0$. Also let $\epsilon = \overline{y}_{t_k}(l_j) - y_j$, which is strictly positive by definition of sector j. By property (ii), $o_{t_k} < o_{t_j}$, therefore, given the vehicle numbering, we also have $T_{t_k} \geq T_{t_j}$ and $Q_{t_k} \leq Q_{t_j}$, which implies that $\overline{y}_{t_k}(l_j) = \frac{6(T_{t_k}v-2(l_j)-\varphi)}{w^2\delta+6whv\delta} > \overline{y}_{t_j}(l_j) = \frac{6(T_{t_j}v-2(l_j)-\varphi)}{w^2\delta+6whv\delta}$. Since $y_j < \overline{y}_{t_j}(l_j)$, we have $y_j < \overline{y}_{t_k}(l_j)$. There can be two cases: (1) $\alpha \leq \epsilon$, (2) $\alpha > \epsilon$. In case (1), consider an alternate solution S' with the same number of sectors, but with lengths $y'_1, \ldots, y'_m$ such that $y'_i = y_i$ for $i = 1, \ldots, m$ with $i \neq k, j$, $y'_j = y_j + (\epsilon - \alpha)$ and $y'_k = y_k - (\epsilon - \alpha)$, and vehicle types $t'_1, \ldots, t'_m$ such that $t'_i = t_i$ for $i = 1, ..., m$ and $i \neq k, j$, $t'_j = t_k$ and $t'_k = t_j$. In other words sector j gets longer and k gets shorter and they switch their vehicle types. This solution is feasible since $y'_j = \overline{y}_{t_k}(l_j) - \alpha < \overline{y}_{t_k}(l_j)$ and $y'_k = \overline{y}_{t_k}(l_k) - \epsilon = \overline{y}_{t_k}(l_k) - \left(\overline{y}_{t_k}(l_j) - \overline{y}_{t_j}(l_j) + \overline{y}_{t_j}(l_j) - y_j\right) = \overline{y}_{t_k}(l_k) - \left(\overline{y}_{t_k}(l_k) - \overline{y}_{t_j}(l_k) + \overline{y}_{t_j}(l_j) - y_j\right) = \overline{y}_{t_j}(l_k) - \left(\overline{y}_{t_j}(l_j) - y_j\right) < \overline{y}_{t_j}(l_k) \leq \overline{y}_{t_j}(l_j - y'_j) = \overline{y}_{t_k}(l_k - \epsilon + \alpha)$.

The cost difference between the original and the alternate solution is

$$\begin{aligned}
& g_{t_j}(y_j, l_j) + g_{t_k}(y_k, l_k) + \sum_{i=k+1}^{j-1} g_{t_i}(y_i, l_i) - g_{t_k}(y_j + (\epsilon - \alpha), l_j) \\
& -g_{t_j}(y_k - (\epsilon - \alpha), l_k - (\epsilon - \alpha)) - \sum_{i=k+1}^{j-1} g_{t_i}(y_i, l_i - (\epsilon - \alpha)) \\
= \quad & o_{t_j}\left[2(l_j + \varphi) + \frac{y_j w^2\delta}{6}\right] + o_{t_k}\left[2(l_k + \varphi) + \frac{y_k w^2\delta}{6}\right] + \sum_{i=k+1}^{j-1} o_{t_i}\left[2(l_i + \varphi) + \frac{y_i w^2\delta}{6}\right] \\
& -o_{t_k}\left[2(l_j + \varphi) + \frac{(y_j + (\epsilon - \alpha))w^2\delta}{6}\right] - o_{t_j}\left[2\left(l_k + \varphi - (\epsilon - \alpha)\right) + \frac{(y_k - (\epsilon - \alpha))w^2\delta}{6}\right] \\
& - \sum_{i=k+1}^{j-1} o_{t_i}\left[2(l_i + \varphi - (\epsilon - \alpha) + \frac{y_i w^2\delta}{6}\right] \\
= \quad & 2o_{t_j}(\epsilon - \alpha) + (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}(y_j - y_k + \epsilon - \alpha)\right) + \sum_{i=k+1}^{j-1} o_{t_i}(\epsilon - \alpha) \\
= \quad & 2o_{t_j}(\epsilon - \alpha) + (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}(\overline{y}_{t_k}(l_j) - \overline{y}_{t_k}(l_k))\right) + \sum_{i=k+1}^{j-1} o_{t_i}(\epsilon - \alpha)
\end{aligned}$$

$$
\begin{aligned}
&= 2o_{t_j}(\epsilon-\alpha) + (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}\frac{12(l_k - l_j)}{w^2\delta + 6whv\delta}\right) + \sum_{i=k+1}^{j-1} o_{t_i}(\epsilon - \alpha)\\
&= 2o_{t_j}(\epsilon-\alpha) + (o_{t_j} - o_{t_k})\left(2(l_j - l_k)\left(1 - \frac{w^2\delta}{w^2\delta + 6whv\delta}\right)\right) + \sum_{i=k+1}^{j-1} o_{t_i}(\epsilon - \alpha)
\end{aligned}
$$

which is positive because $\epsilon > \alpha$, $o_{t_j} > o_{t_k}$ and $l_j > l_k$.
In case (2), consider an alternative solution S' with the same number of sectors and the same sector lengths, but with vehicle types $t'_1, ..., t'_m$ such that $t'_i = t_i$ for $i = 1, ..., m$ and $i \neq k, j$, $t'_j = t_k$ and $t'_k = t_j$. In other words, sectors k and j exchange their vehicle types. This solution is feasible for $\alpha > \epsilon$ since $y_j < \overline{y}_{t_j}(l_j) \leq \overline{y}_{t_k}(l_j)$ and $y_k \leq \overline{y}_{t_k}(l_k) - \alpha \leq \overline{y}_{t_k}(l_k) - \epsilon \leq \overline{y}_{t_k}(l_k) - \left(\overline{y}_{t_k}(l_j) - \overline{y}_{t_j}(l_j)\right) = \overline{y}_{t_k}(l_k) - \left(\overline{y}_{t_k}(l_k) - \overline{y}_{t_j}(l_k)\right) = \overline{y}_{t_j}(l_k)$. The cost difference between the original and the alternate solution is

$$
\begin{aligned}
& g_{t_j}(y_j, l_j) + g_{t_k}(y_k, l_k) + \sum_{i=k+1}^{j-1} g_{t_i}(y_i, l_i) - g_{t_k}(y_j, l_j) - g_{t_j}(y_k, l_k) - \sum_{i=k+1}^{j-1} g_{t_i}(y_i, l_i)\\
&= o_{t_j}\left[2(l_j + \varphi) + \frac{y_j w^2\delta}{6}\right] + o_{t_k}\left[2(l_k + \varphi) + \frac{y_k w^2\delta}{6}\right] - o_{t_k}\left[2(l_j + \varphi) + \frac{y_j w^2\delta}{6}\right]\\
&\quad - o_{t_j}\left[2(l_k + \varphi) + \frac{y_k w^2\delta}{6}\right]\\
&= (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}(y_j - y_k)\right)\\
&= (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}(\overline{y}_{t_k}(l_j) - \overline{y}_{t_k}(l_k)) + (\alpha - \epsilon)\right)\\
&\geq (o_{t_j} - o_{t_k})\left(2(l_j - l_k) + \frac{w^2\delta}{6}\frac{12(l_k - l_j)}{w^2\delta + 6whv\delta}\right)\\
&= (o_{t_j} - o_{t_k})\left(2(l_j - l_k)\left(1 - \frac{w^2\delta}{w^2\delta + 6whv\delta}\right)\right),
\end{aligned}
$$

which is positive since $o_{t_j} > o_{t_k}$ and $l_j > l_k$.

2.3.2.0.1 Part (b) The proof is by contradiction. Suppose there exists a sector $k < j$ such that $y_k < Q_{t_k}/(w\delta)$. Consider an alternate solution S' with the same number of sectors and vehicle types serving each sector, but with lengths $y'_1, \ldots, y'_m$ such that $y'_i = y_i$ for $i = 1, \ldots, m$ with $i \neq k, j$, $y'_j = y_j + \epsilon$ and $y'_k = y_k - \epsilon$, where ϵ is a (positive or negative) value such that $y'_k \geq 0$, $y'_j > 0$ and $y'_j < \overline{y}_{t_j}(l_j)$. In other words, we shift some of the lengths of sector j to sector k or the other way around and we keep the length of all other sectors unchanged. This alternate solution is feasible since $\overline{y}_{t_i}(l_i) = \min\{l_i, Q_{t_i}/(w\delta)\}$

for $i = 2, \ldots, j-1$ by Part (a). The difference in cost between solution S and S' is $\epsilon \left(2 \sum_{k=k}^{j-1} o_{t_k} - (w^2\delta)/6(o_{t_j} - o_{t_k}) \right)$. Depending on whether ϵ is positive or negative, this value can be made positive, hence we have a contradiction. Note that property (iii) implies that there is at most one sector with length shorter than its maximum possible value.

The proof of property (iv) is by contradiction. Suppose that $y_j < Q_{t_{j-1}}/w\delta$. By Property (iii) we know that $y_{j-1} = Q_{t_{j-1}}/w\delta$ and therefore this would imply that $y_j < y_{j-1}$. Consider an alternate solution S' with the same number of sectors with lengths $y'_1, ..., y'_m$ such that $y'_{j-1} = y_j$, $y'_j = y_{j-1}$, $t'_{j-1} = t_j$, $t'_j = t_{j-1}$, $y'_k = y_k$ and $t'_k = t_k$ for $k = 1, \ldots, j-2, j+1, \ldots, m$. In other words, the lengths and types of sectors $j-1$ and j are switched. This solution is feasible since $\overline{y}_{t_j}(l_j)$ is non-increasing in l_j for $j > 1$. The cost difference between two solutions is $2(o_{t_j} y_{j-1} - o_{t_{j-1}} y_j)$, which is positive since $o_{t_j} > o_{t_{j-1}}$ by property (ii) and $y_{j-1} > y_j$. Hence, we have a contradiction. □

From Proposition 2.2(ii), it follows that, when the variable cost of the vehicles are not too different, in the sense that they satisfy $o_K < o_1 \left(12 + \delta w^2\right)/\delta w^2$, it is optimal to set the length of each sector equal to its maximum, given the type of vehicle by which it is served. However, if this condition is not satisfied, then it is possible that one sector has length shorter than its maximum. This sector must be preceded only by sectors served by vehicles with a lower variable cost (property (ii)) which operate at full truckload (property (iii)).

The following lemma provides an upper bound on the optimal number of sectors. We use this value to calculate the $\overline{m}_i^s$ values in our MILP formulation from §2.2.4.

Lemma 2.2 *The optimal number m of sectors in which the strip is partitioned is bounded by $\overline{m}$, which is the optimal number of sectors in which the strip would be partitioned if it was served solely by vehicles of type 1 with a maximum route duration that depends on the distance from the bottom of the strip l and is equal to*

$$T(l) = \begin{cases} T_K & l \in (0, (T_K v)/2 - \varphi] \\ T_{K-1} & l \in ((T_K v)/2 - \varphi, (T_{K-1} v)/2 - \varphi] \\ \cdots & \\ T_1 & l \in ((T_2 v)/2 - \varphi, (T_1 v)/2 - \varphi] \,. \end{cases}$$

Proof: Let $\overline{y}_{min}(l) = \min \left\{ l, Q_1/(\delta w), \frac{6(T(l)v - 2(l+\varphi))}{w^2\delta + 6whv\delta} \right\}$. For all $l \in [0, L]$, we have $\overline{y}_{min}(l) \leq \overline{y}_k(l)$ for $k = 1, ..., K$. The proof is by contradiction. Suppose there exists an optimal solution S^* where $m^* > \overline{m}$. This is only possible if there is a sector j such that $y_j < \overline{y}_{min}(l_j)$. By Proposition 2.2 properties (i) and (iv), it must be $j > 1$ and $y_j > Q_{t_{j-1}}/(\delta w)$. But this would imply that $\overline{y}_{min}(l_j) > Q_{t_{j-1}}/(\delta w) \geq \overline{y}_{t_{j-1}}(l_j)$, which is a contradiction. □

Note that $\overline{m}$ can be easily calculated by Algorithm 1 except that $T(l)$ is used instead of T in the expression for $\overline{y}(l)$.

In what follows, we focus on the special case of $K = 2$. As before we assume $o_1 \leq o_2$, $Q_1 \leq Q_2$ and $T_1 \geq T_2$ so that, for example, one can think of the vehicles of type 1 as electric vehicles and vehicles of type 2 as diesel vehicles.

From Proposition 2.2 it follows that, when $o_1 \leq o_2 \leq o_1 \left(12 + \delta w^2\right)/\delta w^2$, property (ii) cannot be satisfied so that each sector has a length equal to its maximum and the value function reduces to

$$V(l) = \begin{cases} \min\left\{g_1(\overline{y}_1(l); l) + V(l - \overline{y}_1(l)), g_2(\overline{y}_2(l); l) + V(l - \overline{y}_2(l))\right\} & \text{if } 0 \leq l \leq L \\ 0 & \text{if } l \leq 0, \end{cases} \tag{2.25}$$

i.e., it is sufficient to compare only two values at each step of the DP. However, this method may not work when $o_2 > o_1 \left(12 + \delta w^2\right)/\delta w^2$ because there can be (at most) one sector served by a vehicle of type 2 with length lower than its maximum, as shown in Figure 2.5 (in this example the third sector is shorter than its maximum). Nevertheless, in this case we can exploit the solution properties given in Proposition 2.2 to simplify the DP resolution, so that at most three values need to be compared at each step:

$$V(l) = \begin{cases} 0 & \text{if } l \leq 0 \\ \min\left\{g_1(\overline{y}_1(l); l) + V(l - \overline{y}_1(l)),\right. & \\ \left. g_2(\overline{y}_2(l); l) + V(l - \overline{y}_2(l)), TC(l)\right\} & \text{if } 0 < l \leq \min\left\{\frac{6(T_2 v - 2\varphi) - Q_1(w + 6hv)}{12}, L\right\} \\ \min\left\{g_1(\overline{y}_1(l); l) + V(l - \overline{y}_1(l)),\right. & \\ \left. g_2(\overline{y}_2(l); l) + V(l - \overline{y}_2(l))\right\} & \text{otherwise,} \end{cases} \tag{2.26}$$

where $TC(l)$ is the cost of serving the sector ending at distance l from the bottom of the strip by a vehicle of type 2 operating at less than full capacity, and all the previous sectors use vehicles of type 1 at full capacity. A formula used to calculate this value is derived in Appendix §2.B.

We now analyze the properties of the optimal solution. To this end, we first provide an example to illustrate the tradeoffs between the two types of vehicle.

Example 2.2 Consider two alternative solutions for the partitioning of a strip of length L, as depicted on Figure 2.6. In solution A (Figure 2.6a), there exists a sector of length x, served by a vehicle of type 2 at the top of the strip and a sector of length $L - x$, served by a vehicle of type 1 at bottom of the strip, such that $x \geq L/2$. In solution B (Figure 2.6b), the order of the two sectors is reversed: the sector of length $L - x$ served by a vehicle of type 1 is at the top of the strip, and the sector of length x served by a vehicle of type 2 is at the bottom of the strip.

We compare the two solutions in terms of the total distance traveled by the two vehicles. From (2.1), we can see that the total vertical and traversal distances within the strip are equal in both

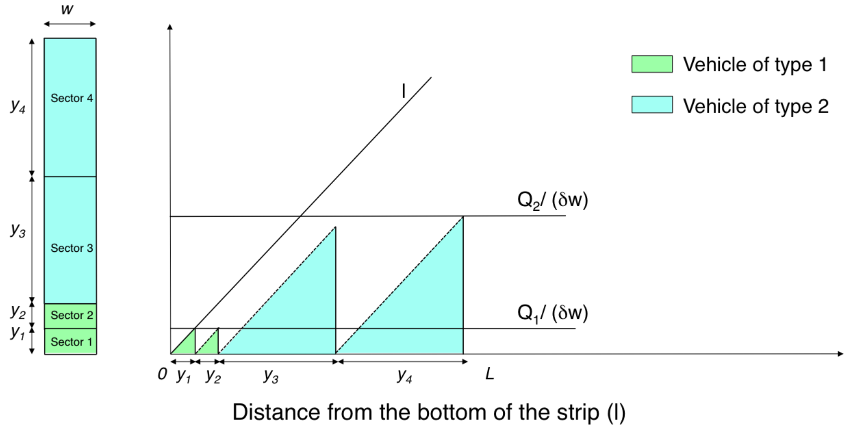


Figure 2.5 Optimal solution for $L = 46$, $w = 8$, $\delta = 0.2$, $\varphi = 0$, $f_1 = 223$, $f_2 = 75$, $o_1 = 8$, $o_2 = 25.5$, $Q_1 = 6$, $Q_2 = 32$, $T_1 = T_2 = \infty$, $v = 30$.

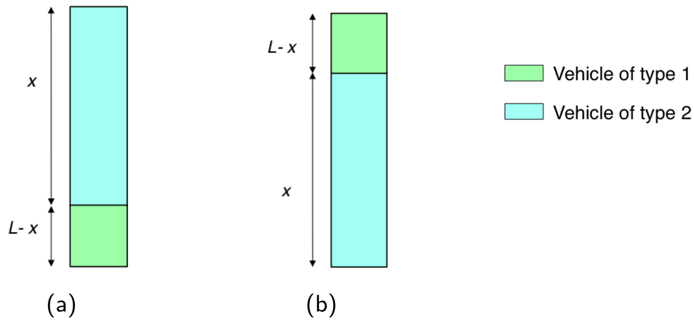


Figure 2.6 Example of two solutions with two vehicles.

solutions. The difference lies in the total transit distance. In solution A, the vehicle of type 1 starts touring the bottom part of the strip immediately, while the vehicle of type 2 needs to drive $L - x$ units before reaching its service sector. In contrast, in solution B, the vehicle of type 1 needs to drive x units before reaching its service sector, and the vehicle of type 2 starts service immediately. Since the total fixed cost is identical in both solutions and $L - x < x$, the tradeoff between total transit distance and variable costs determines which solution is optimal: if $(L - x)o_2 < xo_1$, or equivalently $(L - x)/x < o_1/o_2$, then solution A is preferred, otherwise solution B is.

In the absence of route duration constraints, we are able to prove some structure of the optimal solution.

Lemma 2.3 *If there are no route duration constraints, i.e., $T_1 = T_2 = \infty$, then there exists an optimal solution which has one of the following three structures: (i) All sectors are served by the same vehicle type (see Figures 2.7a and 2.7b); (ii) The first sectors next to the depot are* *served by one vehicle type and the remaining ones are served by the other type (see Figures 2.7c and 2.7d); (iii) The first and last sectors are served by the same vehicle type, while the sectors in the middle are served by the other one (see Figures 2.7e and 2.7f).*

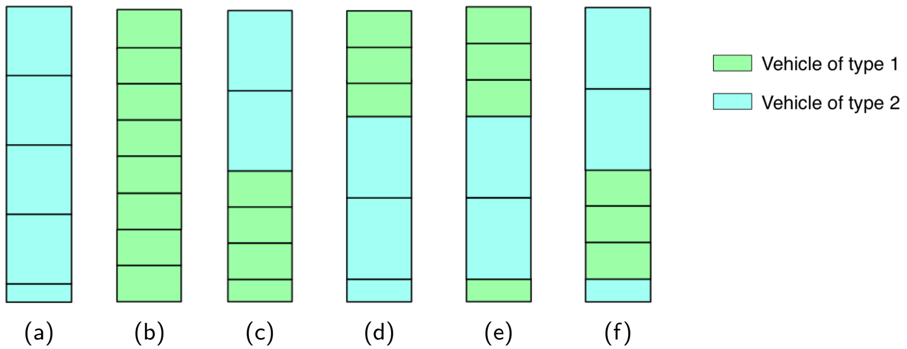


Figure 2.7 Six examples of optimal solutions structures when $T_1 = T_2 = \infty$.

Proof: Consider a solution S with two non-consecutive sectors k and i, such that $k+1 < i$, which are served by the same type of vehicles, i.e., $t_k = t_i$, such that all the sectors in between are served by the other vehicle type, i.e. $t_j = 3 - t_k$, for $j = k+1, ..., i-1$. We show that for S to be optimal it must be $k = 1$. This will prove that there cannot be other structures than (A), (B) or (C). In particular any structure with the vehicle types switching more than twice from the depot to the end of the region will not be possible, since it would violate this property.

The proof is by contradiction, i.e., suppose $k > 1$. There are two cases (i) $t_k = t_i = 1$ and (ii) $t_k = t_i = 2$. In Case (i), consider an alternate solution S' obtained by swapping sectors k and $k+1$, that is, setting $t'_k = 2$, $t'_{k+1} = 1$, $y'_{k+1} = y_k$ and $y'_k = y_{k+1}$ (and leaving all other sectors unchanged). This solution is feasible since, for $T_1 = T_2 = \infty$, $y_k = \overline{y}_1(l_k) = \overline{y}_1(l_{k+1})$. Also, $\overline{y}_2(l_{k+1} - y_k) = \min\{Q_2/(w\delta), l_{k+1} - y_k\}$. We know that (i) $y_{k+1} \leq \overline{y}_2(l_{k+1}) \leq Q_2/(w\delta)$. Moreover, since $k > 1$ it must be $y_k + y_{k+1} < l_{k+1}$ and therefore (ii) $l_{k+1} - y_k > y_{k+1}$ From (i) and (ii) it follows $y_{k+1} \leq \overline{y}_2(l_{k+1} - y_k)$. The difference in costs between solution S and S' is $\Delta_{S,S'} = 2\,(o_2 y_k - o_1 y_{k+1})$, which must be negative if S is optimal. Next, let S'' be an alternate solution obtained by swapping sectors $i-1$ and i, that is, setting $t''_i = 2$, $t''_{i-1} = 1$, $y''_i = y_{i-1}$ and $y''_{i-1} = y_i$ (and leaving all other sectors unchanged). This solution is feasible since $y_{i-1} \leq \overline{y}_2\,(l_{i-1}) \leq \overline{y}_2(l_i)$ and $y_i = \overline{y}_1(l_i) = \overline{y}_1(l_i - y_{i-1})$. The difference in costs between solution S and S'' is $\Delta_{S,S''} = 2\,(o_1 y_{i-1} - o_2 y_i)$.

From Proposition 2.2, if there is a sector with length less than its maximum, then it must be the first sector served by a vehicle of type 2 starting from the bottom of the strip, i.e., sector $k+1$. Hence, we have $y_k = \overline{y}_1(l_k)$, $y_{k+1} \leq \overline{y}_2(l_{k+1})$, $y_{i-1} = \overline{y}_2(l_{i-1})$ and $y_i = \overline{y}_1(l_i)$. Moreover since $k > 1$ (and $T_1 = T_2 = \infty$), we must have $\overline{y}_1(l_k) = \overline{y}_1(l_i) = Q_1/(w\delta)$ and $\overline{y}_2(l_{k+1}) = \max\{l_{k+1}, Q_2/(w\delta)\} \leq \overline{y}_2(l_{i-1}) = \max\{l_{i-1}, Q_2/(w\delta)\}$ by (2.23). This implies that $y_k = y_i$ and $y_{k+1} \leq y_{i-1}$. Therefore, $\Delta_{S,S''} = 2\,(o_1 y_{i-1} - o_2 y_k) \geq 2\,(o_1 y_{k+1} - o_2 y_k) = -\Delta_{S,S'} > 0$, which is a contradiction.

In case (ii), note that sector k cannot be preceded by a vehicle of type 1, since otherwise we could be in case (i) as well. Therefore by Proposition 2, all sectors lengths have to be equal to their maximum value so that it is easy to show that $\Delta_{S,S'} = -\Delta_{S,S''}$, which is a contradiction to S being optimal. $\square$

In some rare cases there could be multiple optimal solutions, some of which may not have one of the three structures stated in this lemma (this may happen in particular if $o_1/o_2 = Q_1/Q_2$). However, in these cases, we can guarantee that there is always at least one optimal solution which does. Lemma 2.3 implies that in the absence of route duration constraints, there can be at most two switches in the vehicle types which are used to serve the sectors when going from the bottom to the top of the strip. Note that this may not be the case if there are route duration restrictions. In fact, the (unique) optimal solution may be such that vehicle types used to serve the sectors change several times from top to bottom, as shown in Figure 2.8.

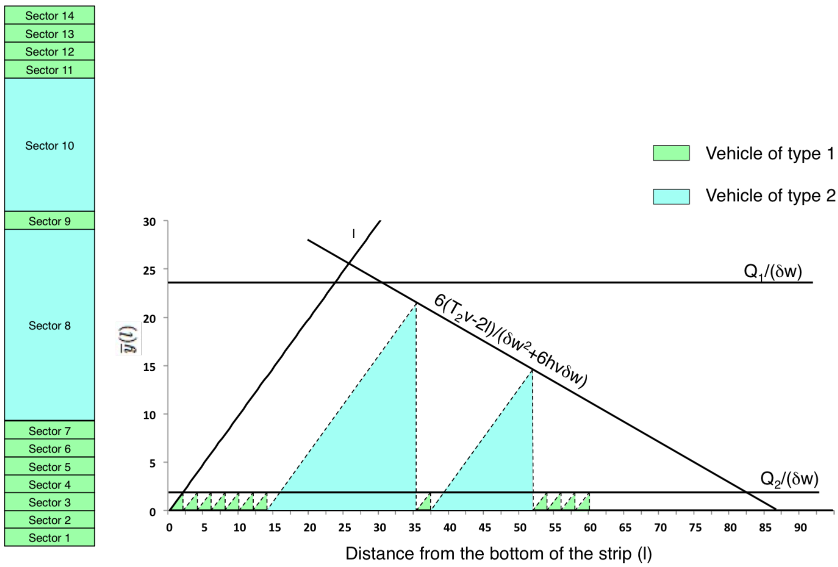


Figure 2.8 Optimal solution for $w = 10$, $L = 60$, $\varphi = 1$, $\delta = 0.3$, $v = 30$, $f_1 = 150$, $f_2 = 90$, $o_1 = 5$, $o_2 = 25$, $Q_1 = 6$, $Q_2 = 70$, $T_1 \geq 6$, $T_2 = 6$.

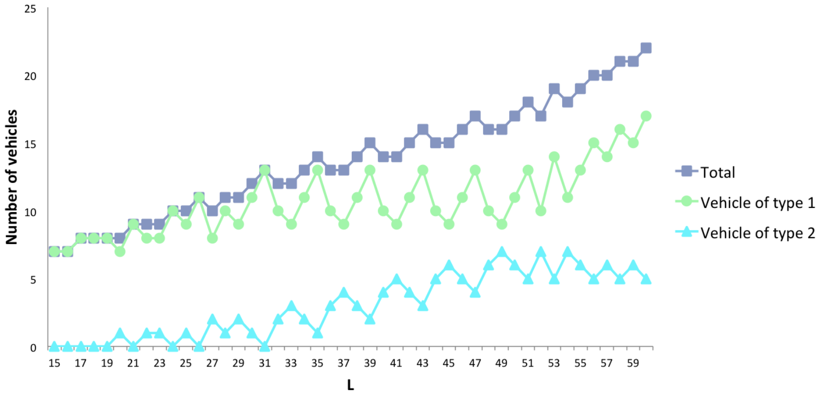
Finally, contrary to what happens in the single vehicle type case, if there are several types of vehicle, the optimal number of sectors (and hence of vehicles) is not necessarily monotone in L (Figure 2.9a) and in φ (Figure 2.9b).

The following lemma provides sufficient conditions under which the optimal solution uses only one type of vehicle.

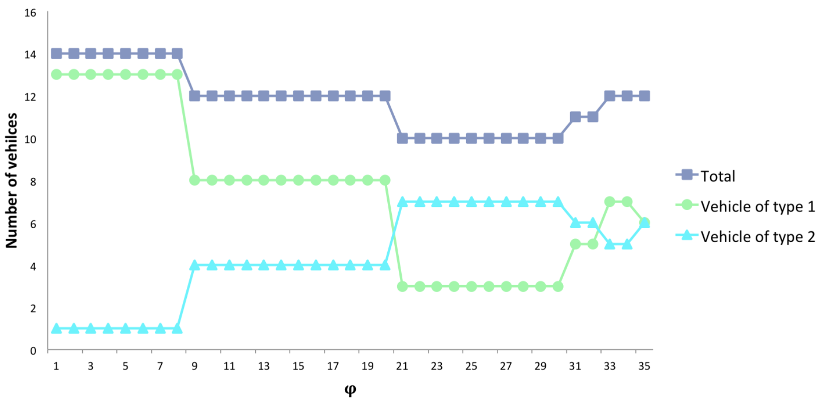
Lemma 2.4 *(a) If $f_1 \leq f_2$ and $Q_1 > \frac{6(T_2 v - 2\varphi)\delta w}{\delta w^2 + 6hv\delta w + 12}$, then it is optimal to use only vehicles of type 1 (b) If both vehicle types have the same route duration limit, i.e. ($T_1 = T_2$) and $f_1 \geq f_2 + (o_2 - o_1)\left(2(L+\varphi) + \frac{wQ_2}{6}\right)$, then it is optimal to use only vehicles of type 2.*

Proof: We first prove (a). If $Q_1 > \frac{6(T_2 v - 2\varphi)\delta w}{\delta w^2 + 6hv\delta w + 12}$ then for all $l \geq 0$, then we have

$$\bar{y}_2(l) = \begin{cases} l & \text{for } l \leq \frac{6(T_2 v - 2\varphi)}{\delta w^2 + 6hv\delta w + 12} \\ \frac{6(T_2 v - 2(l+\varphi))}{w^2\delta + 6whv\delta} & \text{for } \frac{6(T_2 v - 2\varphi)}{\delta w^2 + 6hv\delta w + 12} < l \leq \frac{T_2 v}{2} - \varphi \end{cases}$$



(a) $W = 10$, $\varphi = 7$, $\delta = 0.5$, $v = 30$, $f_1 = f_2 = 0$, $o_1 = 7$, $o_2 = 10$, $Q_1 = 12$, $Q_2 = 20$, $T_1 = 5 = T_2$.



(b) $W = 10$, $L = 35$, $\delta = 0.5$, $v = 30$, $f_1 = f_2 = 0$, $o_1 = 7$, $o_2 = 10$, $Q_1 = 12$, $Q_2 = 20$, $T_1 = 5 = T_2$.

Figure 2.9 Optimal number of vehicles as a function of L (2.9a) and of φ (2.9b).

and

$$\overline{y}_1(l) = \begin{cases} l & \text{for } l \leq \min\left\{\frac{Q_1}{\delta w}, \frac{6(T_1 v - 2\varphi)}{\delta w^2 + 6hv\delta w + 12}\right\} \\ \frac{Q_1}{\delta w} & \text{for } \min\left\{\frac{Q_1}{\delta w}, \frac{6(T_1 v - 2\varphi)}{\delta w^2 + 6hv\delta w + 12}\right\} \leq l \leq \frac{6(T_1 v - 2\varphi) - Q_1(w + 6hv)}{12} \\ \frac{6(T_1 v - 2\varphi)}{\delta w^2 + 6hv\delta w + 12.} & \text{for } \frac{6(T_1 v - 2\varphi) - Q_1(w + 6hv)}{12} < l \leq \frac{T_1 v}{2} - \varphi. \end{cases}$$

The comparison of these two equations implies that $\overline{y}_1(l) \geq \overline{y}_2(l)$ for all $l < \frac{T_2 v}{2} - \varphi$. Hence, for $l < \frac{T_2 v}{2} - \varphi$, we can write the DP recursion as $V(l) = \min\{\min_{0 \leq y \leq \overline{y}_2(l)}\{g_1(y,l), g_2(y,l)\} + V(l-y)\}, \min_{\overline{y}_2(l) < y \leq \overline{y}_1(l)}\{g_1(y,l) + V(l -$

$y)\}\}$. When $f_1 \leq f_2$, $g_1(y,l) \leq g_2(y,l)$ for all $l > 0$ and $y \leq \overline{y}_2(l)$, which means that it is optimal to select vehicle 1. For $l \in [T_2 v/2 - \varphi, T_1 v/2 - \varphi)$, only vehicles of type 1 are feasible. Therefore it is optimal to use vehicles of type 1 for all values of l such that the problem is feasible.

We next prove (b). Consider the first iteration of the DP recursion

$$V(L) = \min\left\{ \min_{0\leq y\leq \overline{y}_1(L)} \{g_1(y;L) + V(L-y)\}, \min_{0\leq y\leq \overline{y}_2(L)} \{g_2(y;L) + V(L-y)\} \right\} \tag{2.27}$$

Let $\hat{l}(y) = \frac{6(f_1-f_2)-(o_2-o_1)(w^2\delta y+12\varphi)}{12(o_2-o_1)}$ be the maximum value of l such that, for a given y, and all $l \leq \hat{l}(y)$, $g_2(y;l) = f_2 + o_2(2(l+\varphi) + w^2\delta y/6) \leq g_1(y;l) = f_1 + o_1(2(l+\varphi) + w^2\delta y/6)$. If $\hat{l}(\frac{Q_2}{\delta w}) \geq L$, which is equivalent to $f_1 \geq f_2 + (o_2 - o_1)\left(2(L+\varphi) + \frac{wQ_2}{6}\right)$, then $g_2(\frac{Q_2}{\delta w};L) \leq g_1(\frac{Q_2}{\delta w};L)$. Since $o_2 > o_1$, $\hat{l}(y)$ is decreasing in y, and therefore $g_2(y;L) \leq g_1(y;L)$ for all $y \in \left[0, \frac{Q_2}{\delta w}\right]$. As a result, $\min_{0\leq y\leq \overline{y}_2(L)} g_2(y;L) < \min_{0\leq y\leq \overline{y}_1(L)} g_1(y;L)$ since for $T_1 = T_2$ it is $\overline{y}_1(L) \leq \overline{y}_2(L) \leq \frac{Q_2}{\delta w}$. It follows that the sector which is the furthest away from the depot should be served by vehicle 2 since the second term in (2.27) achieves the minimum. Let y_m be its length. The next sector will end at a distance of $L - y_m$ from the bottom of the strip. Since $L - y_m < L < \hat{l}(\frac{Q_2}{\delta w})$ and $\overline{y}_1(l) \leq \overline{y}_2(l) \leq \frac{Q_2}{\delta w}$ for all l we have $\min_{0\leq y\leq \overline{y}_2(L)} g_2(y;L-y_m) < \min_{0\leq y\leq \overline{y}_1(L)} g_1(y;L-y_m)$. Therefore, by a similar argument we conclude that the next sector should also be served by vehicle 2. We can repeat the same argument up to the bottom of the strip. □

In Lemma 2.4(a), the route duration constraint for vehicles of type 2 is so strict that these vehicles are not able to carry more than the vehicles of type 1, hence, they lose their unique advantage over these and are never used. The condition in Lemma 2.4(b) implies that the increase in fixed cost when switching from a type 2 to a type 1 vehicle more than offsets the maximum possible decrease in variable cost which comes with this switch, and therefore type 1 vehicles are never used.

In practice, a mixed fleet may bring extra logistical and maintenance complexity, so that some logistics service providers may prefer to have only one type of vehicle. The following lemma establishes when it is optimal to use only vehicles of type 1 versus only vehicles of type 2.

Lemma 2.5 *Assume that there is no route duration limitation, i.e. $T_1 = T_2 = \infty$ and that only one type of vehicle can be used. Then it is cheaper to use only vehicles of type 1 if and only if $o_1/o_2 \leq \alpha$, where*

$$\alpha = \frac{o_2\left(\lceil\frac{Lw\delta}{Q_2}\rceil\left(2(L+\varphi) - (\lceil\frac{Lw\delta}{Q_2}\rceil - 1)\frac{Q_2}{w\delta}\right) + \frac{w^2\delta}{6}L\right) + f_2\lceil\frac{Lw\delta}{Q_2}\rceil - f_1\lceil\frac{Lw\delta}{Q_1}\rceil}{o_2\left(\lceil\frac{Lw\delta}{Q_1}\rceil\left(2(L+\varphi) - (\lceil\frac{Lw\delta}{Q_1}\rceil - 1)\frac{Q_1}{w\delta}\right) + \frac{w^2\delta}{6}L\right)}. \tag{2.28}$$

Proof: If the strip is served only by vehicles of type k and $T_k \geq \frac{12(L+\varphi)+Q_k(w+6h)}{6v}$ we can provide a closed-form expression for the optimal solution. Let $m = \left\lceil \frac{L\delta w}{Q_k} \right\rceil$, $y_j = Q_k/(\delta w)$ for $j = 2, ..., m$ and $y_1 = L - (m - 1)Q_k(\delta w)$.

$$
\begin{aligned}
TC_k &= g_k\left(L-(m-1)\frac{Q_k}{w\delta}, L-(m-1)\frac{Q_k}{w\delta}\right) + \sum_{i=2}^{m} g_k\left(\frac{Q_k}{w\delta}, L-(m-i)\frac{Q_k}{w\delta}\right) \\
&= mf_k + o_k\left(2(L-(m-1)\frac{Q_k}{w\delta}+\varphi) + \left(L-(m-1)\frac{Q_k}{w\delta}\right)\frac{w^2\delta}{6}\right. \\
&\quad \left. + \sum_{i=2}^{m}\left(2(L-(m-i)\frac{Q_k}{w\delta}+\varphi) + \frac{Q_k}{w\delta}\frac{w^2\delta}{6}\right)\right) + \\
&= mf_k + o_k\left[2m(L+\varphi) - (m-1)m\frac{Q_k}{w\delta} + \frac{w^2\delta}{6}L\right] \\
&= mf_k + mo_k\left[2(L+\varphi) - (m-1)\frac{Q_k}{w\delta}\right] + o_k\frac{w^2\delta}{6}L \\
&= \left\lceil \frac{L\delta w}{Q_k} \right\rceil f_k + \left\lceil \frac{L\delta w}{Q_k} \right\rceil o_k\left[2(L+\varphi) - \left(\left\lceil \frac{L\delta w}{Q_k} \right\rceil - 1\right)\frac{Q_k}{w\delta}\right] + o_k\frac{w^2\delta}{6}L.
\end{aligned}
$$

Equation 2.28 is obtained after simplification of $TC_2 \leq TC_1$. □

In Figure 2.10, we plot the value of α from (2.28) as a function of the ratio Q_1/Q_2 (we have kept Q_2 fixed and we have let Q_1 vary in the interval $[0, Q_2]$ so that the ratio varies between 0 and 1). We see that the curve is concave and located above the 45-degree line. The shape and location of this curve imply that if the variable cost of type 1 vehicles is half that of type 2 vehicles, then type 2 vehicles need to have more than twice the capacity of type 1 vehicles in order to be used optimally (when the fixed costs of the two vehicle types are equal).

2.3.3 Multiple strips, multiple vehicle types

In this section we consider the case of multiple strips, that is, the rectangular region is partitioned into s strips of equal width, and $w = W/s$. Let $V_{w,\varphi}(L)$ denote the optimal cost of serving customers in a strip of width w and length L, so that the middle of the bottom edge is located at a distance of φ from the depot. This value can be calculated using recursion (2.25) or (2.26), depending on whether o_2 is lower or greater than $o_1\left(12 + \delta w^2\right)/\delta w^2$, as explained in §2.3.2. Let $C(s)$ denote the optimal cost when the region is partitioned into s strips; we have $C(s) = \sum_{i=1}^{s} V_{W/s,\varphi_i^s}(L)$, where φ_i^s is given by (2.2). The following lemma establishes the symmetry of the optimal solution around the middle strip(s).

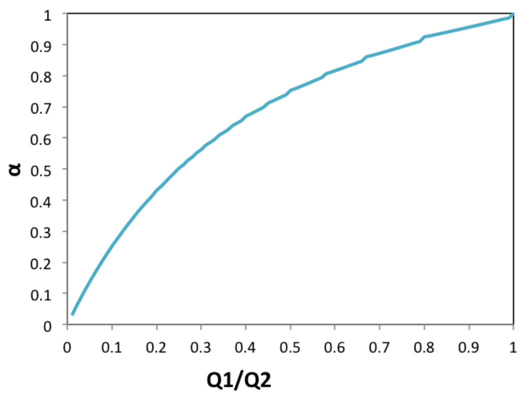


Figure 2.10 Value of α as function of Q_1/Q_2 for $W = 10$, $L = 40$, $\varphi = 3$, $\delta = 0.5$, $Q_2 = 50$, $o_2 = 20$, $f_1 = f_2 = 0$, $W = 10$.

Lemma 2.6 *The optimal solution for a given number of strips is always symmetric around the middle strip(s), i.e.,* $y_{i,j}^{s^*} = y_{s^*-i+1,j}^{s^*}$ *and* $t_{i,j}^{s^*} = t_{s^*-i+1,j}^{s^*}$ *for* $i = 1, \dots, \lceil s^*/2 \rceil$*, where* s^* *is the optimal number of strips.*

Proof: The result follows directly for the realization that for all s, $\varphi_i^s = \varphi_{s-i+1}^s$ for $i = 1, ..., \lceil s/2 \rceil$, therefore $V_{w,\varphi_i^s}(L) = V_{w,\varphi_{s-i+1}^s}(L)$ for all w, L and s. □

From Lemma 2.6 we can write $C(s) = 2\sum_{i=1}^{\lfloor s/2 \rfloor} V_{W/s,\varphi_i^s}(L)$ for s even and $C(s) = 2\sum_{i=1}^{\lfloor s/2 \rfloor} V_{W/s,\varphi_i^s}(L) + V_{W/\lceil s/2 \rceil,\varphi_{\lceil s/2 \rceil}^s}(L)$ for s odd. The minimum cost for the area partitioning problem is $C^* = \min_{s=1,\dots,\overline{s}} C(s)$. Because $C(s)$ is not necessarily monotone in s, as shown in Figure 2.11, the optimal value of s can only be obtained by comparison of $C(1), \dots, C(\overline{s})$, where $\overline{s}$ is an upper bound on the number of strips, which we show how to calculate in Appendix 2.A.

2.4 Numerical analysis

The purpose of our numerical study is threefold. First, we study the impact of city access restrictions on overall traffic congestion. Second, we study the optimal fleet composition; in particular, we quantify the benefits of having a heterogeneous versus a homogeneous fleet of vehicles. Third, we compare our MILP formulation from §2.2.4 and DP formulation from §2.3.2 in terms of computational time.

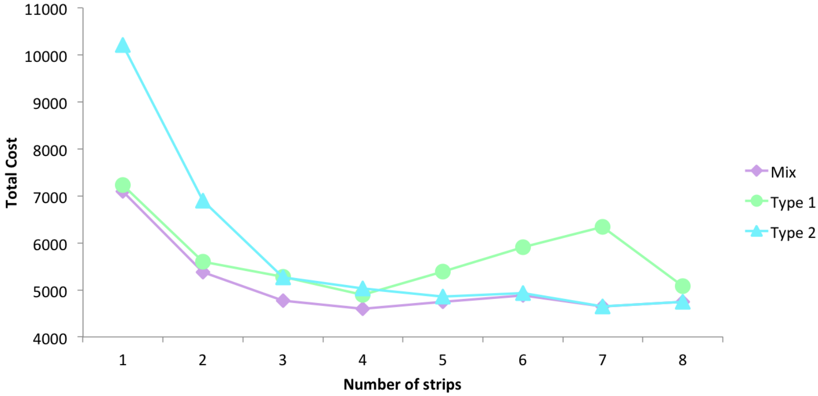


Figure 2.11 Total transportation cost as a function of the number of strips for $W = 20$, $L = 40$, $\varphi = 1$, $\delta = 0.4$, $v = 30$, $f_1 = 200$, $f_2 = 50$, $o_1 = 4$, $o_2 = 5$, $Q_1 = 50$, $T_1 = 12$, $T_2 = 4$.

2.4.1 Impact of city access restrictions

We first investigate how the time access restrictions affect the optimal fleet composition for a problem with two vehicle types, i.e., $K = 2$. We refer to type 1 as the electric vehicles and to type 2 as the diesel vehicles. In Figure 2.12, we see that the total number of vehicles tends to decrease as the time restrictions for vehicles of type 2 become stricter, but locally, it can actually increase, as is the case where T_2 increases from 3.4 to 3.6. In this example, when the maximum route duration constraints for vehicles of type 2 are not binding, i.e., $T_2 > 4.2$, it is optimal to use only vehicles of type 2. Note that for T_2 between 3.8 and 4.4, the number of vehicles of type 2 decreases with T_2 but for values between 3.2 and 3.8, it increases. These results show that limiting access or banning the use of large vehicles may in some cases actually accentuate traffic congestion and increase the number of diesel vehicles on the streets.

2.4.2 Optimal fleet composition

We next investigate how the optimal fleet composition, i.e., the number of vehicles of each type, changes depending on the area and vehicle parameters. As in §2.2.1 we assume there are two vehicle types with $Q_1 \leq Q_2$ and $o_1 \leq o_2$.

Figure 2.13 represents how the percentage of electric vehicles in the optimal solution to the area partitioning problem varies with the ratios o_1/o_2 and Q_1/Q_2.

The pictures are based on the calculation of the optimal solution for $10,000$ discrete

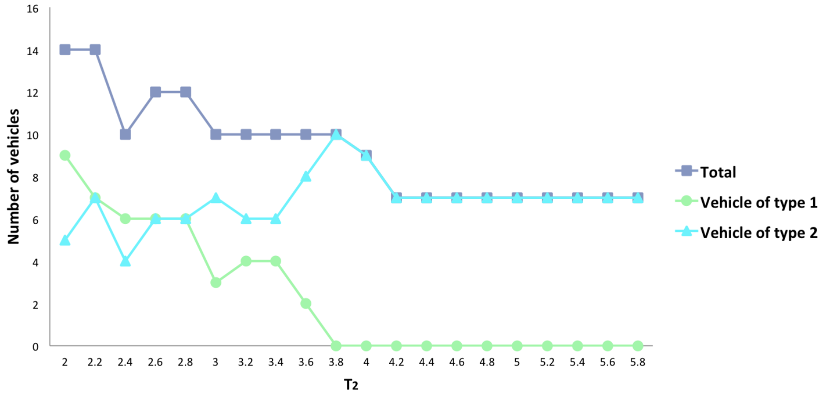


Figure 2.12 Optimal number of vehicles as a function of T_2 given $W = 20$, $L = 40$, $\varphi = 1$, $\delta = 0.4$, $v = 30, f_1 = 200$, $f_2 = 50$, $o_1 = 4$, $o_2 = 5$, $Q_1 = 25$, $Q_2 = 50$, $T_1 = 12$.

points, when both ratios vary between 0 and 1 in increments of 0.01. A value of 100% corresponds to an all-electric vehicle fleet, while a value of 0% corresponds to an all-diesel vehicle fleet. Any intermediate value indicates that it is optimal to use a heterogeneous fleet of vehicles. Figure 2.13a is the base scenario where the parameters are set as follows: $W = 10$, $L = 30$, $\varphi = 3$, $\delta = 0.5$, $v = 30$ $h = 0$, $f_1 = f_2 = 0, o_2 = 20, Q_2 = 50, T_1 = T_2 = \infty$. We have generated several problem instances and obtained very similar pictures for other sets of parameters. Hence we are confident that this base case is a good representation of the tradeoffs involved.

The four graphs of Figure 2.13a all exhibit the same basic pattern: if the electric vehicle has a small capacity and a high variable cost, i.e. $Q_1/Q_2 \to 0$ and $o_1/o_2 \to 1$, then it is preferable to use a homogeneous fleet of diesel vehicles (i.e, the proportion of electric vehicles used is 0%). On the other hand, if the electric vehicle has a large capacity and a low variable cost, i.e. $Q_1/Q_2 \to 1$ and $o_1/o_2 \to 0$, then it is preferable to use a homogeneous fleet of electric vehicles (i.e., the proportion of electric vehicles used is 100%). The cases where a heterogeneous fleet of vehicles is optimal all lie on the frontier between these two extreme regions. This suggests that the benefits of having a heterogeneous fleet are the greatest when there is not too much difference between the cost of operating an all-electric vehicle fleet and the cost of operating an all-diesel vehicle fleet. Further, this region appears to be delimited by a concave line located mostly above the 45-degree line, which is reminiscent of the shape of the α function discussed in §2.3.2. In the base case of Figure 2.13a, 60% of the instances have an all-electric optimal fleet, 37% have an all-diesel optimal fleet and the remaining 3% have a heterogeneous fleet.

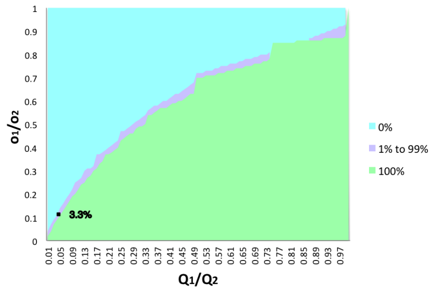
In each of the four graphs of Figure 2.13, we also identify the instance yielding the maximum cost savings resulting from using a heterogeneous fleet of vehicles versus a homogeneous one. So for example, the maximum cost savings is 3.3% in Figure 2.13a.

Figure 2.13b, 2.13c, 2.13d illustrate how the three regions change when one parameter varies while the others are kept constant. In Figure 2.13b, the fixed cost of the electric vehicle is increased to 130 and this leads to a decrease in the number of electric vehicles used, moving the heterogeneous fleet region down, closer to the 45-degree line. Compared to the previous case, the proportion of instances where it is optimal to have an all-electric fleet decreases to 50%, while the proportion of instances where it is optimal to have an all-diesel fleet increases to 49%. Also, the proportion of instances where it is optimal to have a heterogeneous fleet reduces to 1%.

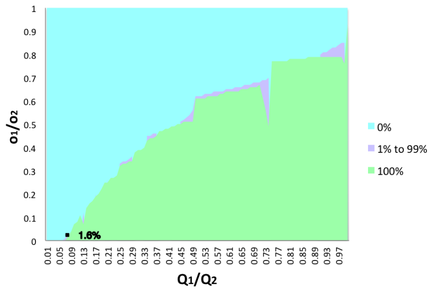
In Figure 2.13c, the maximum route duration of the diesel vehicles is set equal to 3, but there is still no time restriction on the usage of the electric vehicles. This leads to a decrease in the proportion of diesel vehicles used (to 25%) and an increase in the proportion of electric vehicles used (to 69%). Also note that the proportion of solutions with a heterogeneous fleet increases to 6% compared for the base case. Note that if Q_1/Q_2 is sufficiently large ($Q_1/Q_2 \geq 0.776$), it becomes optimal to use only electric vehicles, even if $o_1 = o_2$. This is because the maximum length of the sector that can be served by an electric vehicle is longer than the maximum sector length that can be served by a diesel vehicle due to the time restrictions of the latter. Therefore, in this situation, the diesel vehicle provides no advantage compared with the electric vehicle since it has to make shorter trips and is more expensive.

Finally Figure 2.13d depicts the results obtained when the same number of customers are distributed over a longer area, i.e., the length of the area is increased from $L = 30$ to $L = 50$ and the density is reduced from $\delta = 0.5$ to $\delta = 0.3$, so that the total number of customers remains constant at 150. This change leads to a decrease in the proportion of solutions for which only diesel vehicles are used (to 31.7%) and to an increase in the proportion of solutions for which a heterogeneous fleet is optimal (7.68%). This is because that the area is longer and the travel distances are increased so that it is optimal to use more electric vehicles which are cheaper to operate per unit of distance.

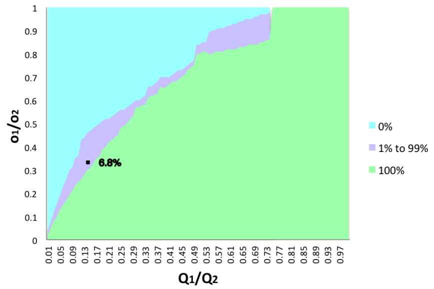
Although we have found some examples where the cost saving yielded from using an heterogeneous fleet is very large (more than 50%), in our numerical study, the maximum cost saving resulting from using a heterogeneous fleet is about 10% and the average is only about 0.1%. These cost figures suggest that the benefits are generally small, the vast majority of cases being such that a homogeneous fleet of vehicles is optimal. In practice, using different vehicle types generates extra logistical complexity, which may not be outweighed by the average cost savings



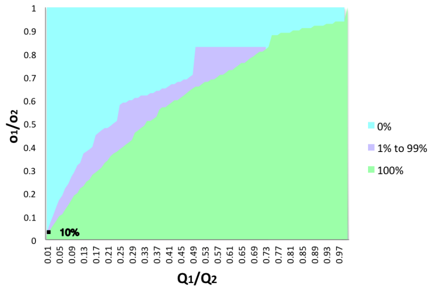
(a) Case 1: $W = 10$, $L = 30$, $f_1 = f_2 = 0$, $T_1 = T_2 = \infty$, $\delta = 0.5$.



(b) Case 2: $W = 10$, $L = 30$, $f_1 = 130, f_2 = 0$, $T_1 = T_2 = \infty$, $\delta = 0.5$.



(c) Case 3: $W = 10$, $L = 30$, $f_1 = f_2 = 0$, $T_1 = \infty, T_2 = 3$, $\delta = 0.5$.



(d) Case 4: $W = 10$, $L = 50$, $f_1 = f_2 = 0$, $T_1 = T_2 = \infty$, $\delta = 0.3$.

Figure 2.13 Percentage usage of electric vehicles.

we have observed.

2.4.3 MILP versus DP

In the experiments of this section we solve 100 problem instances to compare the computational performance of our DP algorithm and solving our MILP formulation. Both methods were coded in a C++ environment and we used the CPLEX V12.5.1 concert libraries to solve the MILP. The computational tests were performed on an 2.70GHz Intel Xeon CPU E5-2680 processor, with 32 GB RAM, operating on Windows 7. Table 2.2 reports the average computational time to solve a set of 100 instances. The value *Opt.* indicates the number of instances solved to optimality within a time limit of one hour. The instances where generated using the same input parameters used in Figure 2.13b. The full results for individual instances are presented in Table 2.3 in Appendix §2.C.

While the MILP could solve only 98 instances to optimality within an average time

Table 2.2 Comparison between MILP and DP.

Number of instances	MILP Opt.	MILP Average* time (sec.)	DP Opt.	DP Average* time (sec.)
100	98	116.73	100	0.059

* Calculated on instances solved to optimality by both MILP and DP.

of 116.7 seconds, the DP algorithm could solve all instances within less than one second. This confirms the efficiency of our method for problems with two vehicle types.

2.5 Conclusions

We have studied the strategic problem of a logistics service provider managing a fleet of vehicles operating in a city under access restrictions. We have represented the city as a rectangular service region divided into sectors, each served by a single vehicle. The length of the routes was calculated by means of a continuous approximation formula. The objective was the minimization of the total cost of fleet ownership or leasing, the fuel cost and labor cost. We have formulated the problem as a mixed integer partitioning problem, and as a dynamic program. We have provided an efficient method to compute an optimal solution by exploiting some key structural properties of the problem; this method was shown to be much faster than solving the MILP on a large number of problem instances. We have also shown that the optimal solution may have a complex structure, with vehicles of different types serving adjacent sectors, and some vehicle running at less than capacity or returning to the depot before the route duration limit. Numerically, we found that city access restrictions may, in some cases, be counter-productive as they can induce more traffic congestion. We also found that on average, operating a heterogeneous fleet only leads to a small decrease in cost, which may not be outweighed by the increase in logistical complexity resulting from operating several vehicle types.

2.A Solution properties and formula for calculating the value of $\overline{s}$

In this section of the appendix, we prove the existence of an upper bound on the number of strips and we provide a formulation for calculating it. We first provide a Lemma which is useful in obtaining this result.

Lemma 2.7 *The total distance traveled by all the vehicles serving a region, where each strip is served by a single vehicle, is increasing in s for $s \geq \left\lceil\sqrt{W^2\delta/12}\right\rceil$.*

Proof: Proof Let $s \geq \left\lceil\sqrt{W^2\delta/12}\right\rceil$, the difference in total distance traveled between solutions with $s+1$ and s strips, denoted $\Delta_{s,s+1}$, is equal to

$$\begin{aligned}\Delta_{s,s+1} &= \sum_{i=1}^{s+1}\left(2(\varphi_i^{s+1}+L)+\frac{L\delta W^2}{6s^2}\right)-\sum_{i=1}^{s+1}\left(2(\varphi_i^{s+1}+L)+\frac{L\delta W^2}{6(s+1)^2}\right)\\ &= 2\left(\sum_{i=1}^{s+1}\varphi_i^{s+1}-\sum_{i=1}^{s}\varphi_i^{s}\right)+2L-\frac{L\delta W^2}{6s(s+1)}.\end{aligned}$$

Given the definition of φ_i^s, it is easy to check that $\sum_{i=1}^{s+1}\varphi_i^{s+1}-\sum_{i=1}^{s}\varphi_i^{s}$ is equal to $\frac{W}{4}\frac{s}{s+1}+\varphi$ if s is even and to $\frac{W}{4}\frac{s+1}{s}+\varphi$ if s is odd, which is positive in both cases. Therefore $\Delta_{s,s+1} \geq 2L-\frac{L\delta W^2}{6s(s+1)} \geq 2L-\frac{L\delta W^2}{6\overline{s}^2}$, which is positive because $s+1 > s \geq \lceil\sqrt{W^2\delta/12}\rceil$. □

The next lemma provides a formulation for the upper bound computation.

Lemma 2.8 *The optimal solution has a number of strips s^* which is less or equal to $\overline{s} = \max\left\{\max_{k=1,\dots,K} s_k, \left\lceil\sqrt{W^2\delta/12}\right\rceil\right\}$, where for $k = 1, \dots K$,*

$$s_k = \max\left\{\left\lceil\frac{WL\delta}{Q_k}\right\rceil,\right. \tag{2.29}$$

$$\left.\left\lceil\frac{6W_k(1-\delta hLv)+\sqrt{(6W_k\delta hLv-6W_k)^2-4\delta LW_k^2(6(2L+2\varphi-T_kv)+6W_k)}}{12(W_k+2(L+\varphi)-T_kv)}\right\rceil\right\}$$

and $W_k = \min\{W, T_kv-2(L+\varphi)-\epsilon\}$ where ϵ is a small positive number.

Proof: For a given vehicle type k, $W_k/2$ measures the maximum horizontal distance from the middle point of the top edge of the service area that can be reached by a vehicle of type k. Vehicles such that $W_k = W$ can reach the upper left corner of the rectangle, but vehicles such that $W_k < W$ cannot be used to serve the top left and right corners of the service area because they are unable to reach these customers within their route duration limit T_k.

The variable s_k measures the minimum number of strips needed so that vehicles of type k can feasibly use one vehicle per strip to serve the rectangle of height L and width W_k (with $W_k/2$ on each side of the depot). It is the smallest value satisfying $\frac{WL\delta}{s_k} \leq Q_k$ and $2(\varphi + \frac{s_k-1}{s_k}\frac{W_k}{2} + L) + \frac{W^2L\delta}{6s_k^2} + \frac{WL\delta hv}{s_k} \leq T_k v$. Solving the first inequality as an equation gives $s_k = \frac{WL\delta}{Q_k}$, which is the first term (2.29) for the second we obtain a quadratic equation in s_k:

$$6s_k^2[T_k v - 2(L+\varphi) - W] + 6Ws_k(1 - Lhv\delta) - W^2L\delta = 0.$$

Since $T_k v - 2(L+\varphi) - W_k = \epsilon > 0$, this equation has only one positive root, which is the second term in (2.29). As a result, for all values of $s > s_k$, it is always feasible to serve all strips within a distance $W_k/2$ of the depot with only one vehicle of type k.

The rest of the proof is by contradiction. Suppose there exists an optimal solution S' for which the area is divided into s' strips such that $s' > \overline{s}$. In this solution, there must be only one vehicle per strip since all vehicles types used in a strip can feasibly serve it entirely, therefore it is optimal to use the vehicle that can serve the entire strip at the lowest cost. Let n'_i denote the number of strips served by vehicles of type i in solution S'. Consider an alternative solution S'' with $s'' = \overline{s}$, where each strip is served by a single vehicle. Let n''_i denote the number of strips served by vehicles of type i in solution S''. We set $n''_1 = \min\{n'_1, \overline{s}\}, n''_2 = \min\{n'_2, \overline{s} - n''_1\}, \ldots, n''_K = \min\{n'_K, \overline{s} - \sum_{i=1}^{K-1} n''_i\}$. Let $k \in \{1, \ldots, K\}$ be the largest value such that $n''_k > 0$. We have $n''_i = n'_i$ for $i = 1, ..., k-1$, $n''_k \leq n'_k$ and $n''_i = 0 < n'_i$ for $i = k+1, ..., K$. Let D'_i and D''_i be the total distances traveled by all the vehicles of type $i = \{1, \ldots, K\}$ in solutions S' and S'', respectively. We know that $\sum_{i=1}^K D'_i \geq \sum_{i=1}^K D''_i$ since, by Lemma 2.7, the total distance travelled increases in the number of strips as $\overline{s} \geq \left\lceil \sqrt{W^2\delta/12} \right\rceil$. Also, $D''_i \geq D'_i$ for $i = 1, ..., k-1$ since the number of strips served by vehicles i in S'' is the same as for S, but these strips are wider.

We have:

$$\begin{aligned}
TC(S') &= \sum_{i=1}^{K} (f_i n'_i + o_i D'_i) \\
&= \sum_{i=1}^{K} f_i n'_i + \sum_{i=1}^{k-1} o_i D'_i + \sum_{i=k}^{K} o_i D'_i \\
&\geq \sum_{i=1}^{K} f_i n'_i + \sum_{i=1}^{k-1} o_i D'_i + o_k \left(\sum_{i=1}^{K} D'_i - \sum_{i=1}^{k-1} D'_i \right)
\end{aligned}$$

$$
\begin{aligned}
&> \sum_{i=1}^{K} f_i n_i'' - \sum_{i=1}^{k-1} (o_k - o_i) D_i'' + o_k \sum_{i=1}^{K} D_i'' \\
&= \sum_{i=1}^{K} f_i n_i'' + \sum_{i=1}^{k-1} o_i D_i'' + o_k D_k'' = TC(S''),
\end{aligned}
$$

where the first inequality holds because $o_k \leq o_i$ for $i = k+1, ..., K$ and the second one holds because $n_i' \geq n_i''$ for all $i = 1, ..., K$ and $\sum_{i=1}^{K} D_i' \geq \sum_{i=1}^{K} D_i''$ and $D_i' < D_i''$ for $i = 1, ..., k-1$. Since $TC(S'') < TC(S')$ we have a contradiction. □
We observed that we can obtain a tighter upper bound on the value of s^* using a more complex algorithm. We used this alternate bound in our numerical study since it contributes to reducing the computational time. However, as this does not affect the main results of this chapter, we decided to present a simpler formulation which calculates a looser upper bound.

2.B $TC(l)$ Formulation

In this section of the appendix, we present the formulation for calculating $TC(l)$ from §2.3.2. Formally, we have

$$
\begin{aligned}
TC(l) &= \min_{k=\left\lceil \frac{lw\delta - \overline{y}_2(l)}{Q1} \right\rceil, ..., \left\lfloor \frac{lw\delta}{Q_1} \right\rfloor - 1} \left\{ \sum_{i=1}^{k} g_1\left(\frac{Q_1}{w\delta}, i\frac{Q_1}{w\delta}\right) + g_2\left(l - i\frac{Q_1}{w\delta}, l\right) \right\} \\
&= \min_{k=\left\lceil \frac{lw\delta - \overline{y}_2(l)}{Q1} \right\rceil, ..., \left\lfloor \frac{lw\delta}{Q_1} \right\rfloor - 1} \left\{ kf_1 + o_1 \left(2\sum_{i=0}^{k-1} \frac{Q_1}{w\delta}(k-i) + 2k\varphi + \frac{Q_1 w}{6}k \right) + f_2 \right. \\
&\quad \left. + o_2\left(2(l+\varphi) + \frac{l - kQ_1 w}{6}\right) \right\},
\end{aligned}
$$

where $TC(l)$ is set equal to ∞ if $\lceil lw\delta - \overline{y}_2(l)/Q_1 \rceil > \lfloor lw\delta/Q_1 \rfloor - 1$. The minimum and maximum values for k come from the observation that, by Proposition 2.2(iv), the sector must have a length y such that $Q_1/(\delta w) = \overline{y}_1(l) < y < \overline{y}_2(l) \leq Q_2/(w\delta)$, hence k must be such that $Q_1/(\delta w) < l - kQ_1/(\delta w) < \overline{y}_2(l)$.

2.C Comparison of the MILP and DP formulations

In this section of the appendix, we provide the extended results table for §2.4.3.

Table 2.3 Computational comparison of MILP and DP.

Instance	L	W	φ	δ	v	h	f_1	f_2	o_1	o_2	Q_1	Q_2	T_1	T_2	MILP Solution (£)	MILP CPU Time (sec.)	DP Solution (£)	DP CPU Time (sec.)
1	30	10	3	0.5	30	0	0	0	0.2	20	0.5	50	3	15000	-	no sol.	2216	0.529
2	30	10	3	0.5	30	0	0	0	0.2	20	5.45	50	3	15000	254.792	21.33	254.792	0.03
3	30	10	3	0.5	30	0	0	0	0.2	20	10.4	50	3	15000	153.12	4.034	153.12	0.005
4	30	10	3	0.5	30	0	0	0	0.2	20	15.35	50	3	15000	117.88	3.09	117.88	0.002
5	30	10	3	0.5	30	0	0	0	0.2	20	20.3	50	3	15000	99.6187	2.095	99.6187	0.002
6	30	10	3	0.5	30	0	0	0	0.2	20	25.25	50	3	15000	83.02	1.104	83.02	0
7	30	10	3	0.5	30	0	0	0	0.2	20	30.2	50	3	15000	79.456	0.656	79.456	0
8	30	10	3	0.5	30	0	0	0	0.2	20	35.15	50	3	15000	77.788	0.659	77.788	0.001
9	30	10	3	0.5	30	0	0	0	0.2	20	40.1	50	3	15000	69.3	0.808	69.3	0
10	30	10	3	0.5	30	0	0	0	0.2	20	45.05	50	3	15000	69.3	0.64	69.3	0.001
11	30	10	3	0.5	30	0	0	0	2.18	20	0.5	50	3	15000	-	no sol.	5893.33	0.532
12	30	10	3	0.5	30	0	0	0	2.18	20	5.45	50	3	15000	2777.23	77.131	2777.23	0.027
13	30	10	3	0.5	30	0	0	0	2.18	20	10.4	50	3	15000	1669.01	9.643	1669.01	0.007
14	30	10	3	0.5	30	0	0	0	2.18	20	15.35	50	3	15000	1284.89	4.985	1284.89	0.002
15	30	10	3	0.5	30	0	0	0	2.18	20	20.3	50	3	15000	1085.84	2.604	1085.84	0.002
16	30	10	3	0.5	30	0	0	0	2.18	20	25.25	50	3	15000	904.918	1.436	904.918	0.001
17	30	10	3	0.5	30	0	0	0	2.18	20	30.2	50	3	15000	866.07	0.975	866.07	0.001
18	30	10	3	0.5	30	0	0	0	2.18	20	35.15	50	3	15000	847.889	0.951	847.889	0.001
19	30	10	3	0.5	30	0	0	0	2.18	20	40.1	50	3	15000	755.37	0.983	755.37	0.001
20	30	10	3	0.5	30	0	0	0	2.18	20	45.05	50	3	15000	755.37	0.931	755.37	0
21	30	10	3	0.5	30	0	0	0	4.16	20	0.5	50	3	15000	5893.33	2120.46	5893.33	0.53
22	30	10	3	0.5	30	0	0	0	4.16	20	5.45	50	3	15000	5299.67	239.574	5299.67	0.015
23	30	10	3	0.5	30	0	0	0	4.16	20	10.4	50	3	15000	3184.9	14.461	3184.9	0
24	30	10	3	0.5	30	0	0	0	4.16	20	15.35	50	3	15000	2451.9	5.71	2451.9	0
25	30	10	3	0.5	30	0	0	0	4.16	20	20.3	50	3	15000	2072.07	2.996	2072.07	0.001
26	30	10	3	0.5	30	0	0	0	4.16	20	25.25	50	3	15000	1726.82	1.709	1726.82	0
27	30	10	3	0.5	30	0	0	0	4.16	20	30.2	50	3	15000	1652.68	1.061	1652.68	0
28	30	10	3	0.5	30	0	0	0	4.16	20	35.15	50	3	15000	1617.99	1.482	1617.99	0
29	30	10	3	0.5	30	0	0	0	4.16	20	40.1	50	3	15000	1441.44	1.138	1441.44	0
30	30	10	3	0.5	30	0	0	0	4.16	20	45.05	50	3	15000	1441.44	1.186	1441.44	0
31	30	10	3	0.5	30	0	0	0	6.14	20	0.5	50	3	15000	5893.33	1495.9	5893.33	0.499

32	30	10	3	0.5	30	0	0	0	6.14	20	5.45	50	3	15000	5893.33	52.886	5893.33	0.015
33	30	10	3	0.5	30	0	0	0	6.14	20	10.4	50	3	15000	4700.78	21.918	4700.78	0
34	30	10	3	0.5	30	0	0	0	6.14	20	15.35	50	3	15000	3618.92	8.268	3618.92	0.016
35	30	10	3	0.5	30	0	0	0	6.14	20	20.3	50	3	15000	3058.29	3.791	3058.29	0
36	30	10	3	0.5	30	0	0	0	6.14	20	25.25	50	3	15000	2548.71	1.966	2548.71	0
37	30	10	3	0.5	30	0	0	0	6.14	20	30.2	50	3	15000	2439.3	1.154	2439.3	0
38	30	10	3	0.5	30	0	0	0	6.14	20	35.15	50	3	15000	2388.09	1.108	2388.09	0
39	30	10	3	0.5	30	0	0	0	6.14	20	40.1	50	3	15000	2127.51	1.67	2127.51	0
40	30	10	3	0.5	30	0	0	0	6.14	20	45.05	50	3	15000	2127.51	1.077	2127.51	0
41	30	10	3	0.5	30	0	0	0	8.12	20	0.5	50	3	15000	5893.33	1543.89	5893.33	0.499
42	30	10	3	0.5	30	0	0	0	8.12	20	5.45	50	3	15000	5893.33	24.368	5893.33	0.015
43	30	10	3	0.5	30	0	0	0	8.12	20	10.4	50	3	15000	5893.33	25.085	5893.33	0
44	30	10	3	0.5	30	0	0	0	8.12	20	15.35	50	3	15000	4785.93	9.734	4785.93	0
45	30	10	3	0.5	30	0	0	0	8.12	20	20.3	50	3	15000	4044.52	5.226	4044.52	0
46	30	10	3	0.5	30	0	0	0	8.12	20	25.25	50	3	15000	3370.61	2.168	3370.61	0
47	30	10	3	0.5	30	0	0	0	8.12	20	30.2	50	3	15000	3225.91	1.747	3225.91	0
48	30	10	3	0.5	30	0	0	0	8.12	20	35.15	50	3	15000	3158.19	1.217	3158.19	0
49	30	10	3	0.5	30	0	0	0	8.12	20	40.1	50	3	15000	2813.58	1.623	2813.58	0
50	30	10	3	0.5	30	0	0	0	8.12	20	45.05	50	3	15000	2813.58	1.124	2813.58	0
51	30	10	3	0.5	30	0	0	0	10.1	20	0.5	50	3	15000	5893.33	1409.22	5893.33	0.515
52	30	10	3	0.5	30	0	0	0	10.1	20	5.45	50	3	15000	5893.33	12.449	5893.33	0.109
53	30	10	3	0.5	30	0	0	0	10.1	20	10.4	50	3	15000	5893.33	11.981	5893.33	0.015
54	30	10	3	0.5	30	0	0	0	10.1	20	15.35	50	3	15000	5848.48	11.856	5848.48	0.016
55	30	10	3	0.5	30	0	0	0	10.1	20	20.3	50	3	15000	5030.74	6.334	5030.74	0.016
56	30	10	3	0.5	30	0	0	0	10.1	20	25.25	50	3	15000	4192.51	2.231	4192.51	0
57	30	10	3	0.5	30	0	0	0	10.1	20	30.2	50	3	15000	4012.53	1.669	4012.53	0
58	30	10	3	0.5	30	0	0	0	10.1	20	35.15	50	3	15000	3928.29	2.231	3928.29	0
59	30	10	3	0.5	30	0	0	0	10.1	20	40.1	50	3	15000	3499.65	2.044	3499.65	0
60	30	10	3	0.5	30	0	0	0	10.1	20	45.05	50	3	15000	3499.65	1.373	3499.65	0
61	30	10	3	0.5	30	0	0	0	12.08	20	0.5	50	3	15000	5893.33	856.265	5893.33	0.499
62	30	10	3	0.5	30	0	0	0	12.08	20	5.45	50	3	15000	5893.33	10.561	5893.33	0.109
63	30	10	3	0.5	30	0	0	0	12.08	20	10.4	50	3	15000	5893.33	6.1	5893.33	0
64	30	10	3	0.5	30	0	0	0	12.08	20	15.35	50	3	15000	5893.33	6.896	5893.33	0
65	30	10	3	0.5	30	0	0	0	12.08	20	20.3	50	3	15000	5862.37	7.644	5862.37	0
66	30	10	3	0.5	30	0	0	0	12.08	20	25.25	50	3	15000	5014.41	3.276	5014.41	0
67	30	10	3	0.5	30	0	0	0	12.08	20	30.2	50	3	15000	4799.14	2.48	4799.14	0
68	30	10	3	0.5	30	0	0	0	12.08	20	35.15	50	3	15000	4698.4	2.2	4698.4	0

69	30	10	3	0.5	30	0	0	0	12.08	20	40.1	50	3	15000	4185.72	1.794	4185.72	0
70	30	10	3	0.5	30	0	0	0	12.08	20	45.05	50	3	15000	4185.72	2.106	4185.72	0
71	30	10	3	0.5	30	0	0	0	14.06	20	0.5	50	3	15000	5893.33	1362.38	5893.33	0.499
72	30	10	3	0.5	30	0	0	0	14.06	20	5.45	50	3	15000	5893.33	6.724	5893.33	0.187
73	30	10	3	0.5	30	0	0	0	14.06	20	10.4	50	3	15000	5893.33	4.976	5893.33	0.031
74	30	10	3	0.5	30	0	0	0	14.06	20	15.35	50	3	15000	5893.33	5.039	5893.33	0.015
75	30	10	3	0.5	30	0	0	0	14.06	20	20.3	50	3	15000	5893.33	5.148	5893.33	0
76	30	10	3	0.5	30	0	0	0	14.06	20	25.25	50	3	15000	5836.31	3.151	5836.31	0
77	30	10	3	0.5	30	0	0	0	14.06	20	30.2	50	3	15000	5585.76	3.011	5585.76	0
78	30	10	3	0.5	30	0	0	0	14.06	20	35.15	50	3	15000	5468.5	1.778	5468.5	0.016
79	30	10	3	0.5	30	0	0	0	14.06	20	40.1	50	3	15000	4871.79	2.215	4871.79	0
80	30	10	3	0.5	30	0	0	0	14.06	20	45.05	50	3	15000	4871.79	2.263	4871.79	0
81	30	10	3	0.5	30	0	0	0	16.04	20	0.5	50	3	15000	5893.33	1045.95	5893.33	0.514
82	30	10	3	0.5	30	0	0	0	16.04	20	5.45	50	3	15000	5893.33	4.087	5893.33	0.265
83	30	10	3	0.5	30	0	0	0	16.04	20	10.4	50	3	15000	5893.33	1.591	5893.33	0.094
84	30	10	3	0.5	30	0	0	0	16.04	20	15.35	50	3	15000	5893.33	5.102	5893.33	0.047
85	30	10	3	0.5	30	0	0	0	16.04	20	20.3	50	3	15000	5893.33	3.51	5893.33	0.032
86	30	10	3	0.5	30	0	0	0	16.04	20	25.25	50	3	15000	5893.33	3.213	5893.33	0.032
87	30	10	3	0.5	30	0	0	0	16.04	20	30.2	50	3	15000	5893.33	2.075	5893.33	0.016
88	30	10	3	0.5	30	0	0	0	16.04	20	35.15	50	3	15000	5893.33	2.964	5893.33	0.015
89	30	10	3	0.5	30	0	0	0	16.04	20	40.1	50	3	15000	5557.86	2.2	5557.86	0.015
90	30	10	3	0.5	30	0	0	0	16.04	20	45.05	50	3	15000	5557.86	2.184	5557.86	0
91	30	10	3	0.5	30	0	0	0	18.02	20	0.5	50	3	15000	5893.33	859.432	5893.33	0.499
92	30	10	3	0.5	30	0	0	0	18.02	20	5.45	50	3	15000	5893.33	2.87	5893.33	0.219
93	30	10	3	0.5	30	0	0	0	18.02	20	10.4	50	3	15000	5893.33	1.076	5893.33	0.11
94	30	10	3	0.5	30	0	0	0	18.02	20	15.35	50	3	15000	5893.33	2.761	5893.33	0.047
95	30	10	3	0.5	30	0	0	0	18.02	20	20.3	50	3	15000	5893.33	2.574	5893.33	0.047
96	30	10	3	0.5	30	0	0	0	18.02	20	25.25	50	3	15000	5893.33	2.262	5893.33	0.031
97	30	10	3	0.5	30	0	0	0	18.02	20	30.2	50	3	15000	5893.33	2.247	5893.33	0.016
98	30	10	3	0.5	30	0	0	0	18.02	20	35.15	50	3	15000	5893.33	1.981	5893.33	0.016
99	30	10	3	0.5	30	0	0	0	18.02	20	40.1	50	3	15000	5893.33	2.34	5893.33	0.031
100	30	10	3	0.5	30	0	0	0	18.02	20	45.05	50	3	15000	5893.33	2.168	5893.33	0.016

Vladimir : That passed the time.
Estragon : It would have passed in any case.
Samuel Beckett, *Waiting for Godot*

3 The Time-Dependent Pollution-Routing Problem

In this chapter we study the Time-Dependent Pollution-Routing Problem (TDPRP), namely the problem of routing a homogeneous fleet of vehicles in order to serve a set of customers and determining the speeds on each leg of the routes. The cost function includes emissions and driver costs, taking into account traffic congestion which, at peak periods, significantly restricts vehicle speeds and increases emissions. Next to describing the TDPRP, we present an integer linear programming formulation of the problem and provide illustrative examples to motivate the problem and give insights about the tradeoffs it involves. We also provide an analytical characterization of the optimal solutions for a single-arc version of the problem, identifying conditions under which it is optimal to wait idly at certain locations in order to avoid congestion and to reduce the cost of emissions. Finally, using benchmark instances, we present results on the computational performance of the proposed formulation.

3.1 Introduction

Traffic congestion occurs when the capacity of a particular transportation link is insufficient to accommodate an incoming flow at a particular point in time. Congestion has a number of adverse consequences, including longer travel times and variations in trip duration which result in decreased transport reliability, increased fuel consumption and more carbon dioxide equivalent (CO_{2e}) emissions. The latter measures, for a given mixture and amount of greenhouse gas, the amount of CO_2 that would have the same global warming potential (GWP) (Wikipedia, 2013). It is known that CO_{2e} emissions are proportional to fuel consumption and depend on vehicle speed. Heavy congestion results in low speeds with fluctuations, often accompanied by frequent acceleration and deceleration, and greatly contributes to CO_{2e} emissions (Barth and Boriboonsomsin, 2008). According to the International Road Transport Union (IRU), around 100 billion liters of wasted fuel, or 250 billion tonnes of CO_{2e}, were attributed to traffic congestion in the United States in 2004 (IRU, 2012). Noise is another externality resulting from congestion. In particular, noise from a vehicle's power unit comprising the engine, air intake and exhaust becomes dominant at low speeds of 15–20 mph and at high acceleration rates of 2 m/s^2, as reported by the World Business Council for Sustainable Development (2004). Congestion is at its highest during rush hour, which typically lasts from 6am or 7am to 9am or 10am in the morning, although this varies from one city to another, e.g., 6am–9am in Sydney, Brisbane and Melbourne, and 4am–9am in New York City (Wikipedia, 2012).

Our aim is to study the effect of congestion and CO_{2e} emissions within the context of the Vehicle Routing Problem (VRP), defined as the problem of routing a fleet of vehicles to serve a set of customers subject to various constraints, such as vehicle capacities (see e.g., Cordeau et al., 2007b). Previous VRP research assumes constant vehicle speed, which is not realistic for most practical applications. Van Woensel et al. (2001) show that solving the VRP under this assumption can lead to deviations of up to 20% in CO_{2e} emissions for gasoline vehicles on an average day and up to 40% in congested traffic. Indeed, vehicle speed varies throughout the day (Van Woensel et al., 2008), which affects CO_{2e} emissions. Maden et al. (2009) present an approach for the time-dependent vehicle routing problem which allows for the planning of more reliable routes and schedules. It is based on a tabu search algorithm, which minimizes the total travel time and reduces emissions by avoiding congestion. The authors have applied this algorithm to a real-life case study and have obtained reductions of about 7% in CO_{2e} emissions.

Accounting for emissions in the context of the VRP is relatively new. For a general introduction to the topic we refer the reader to Sbihi and Eglese (2007b). Figliozzi (2010) presents the emissions minimizing VRP (EVRP), a variant of the time-dependent VRP (TDVRP) with time windows, which takes into account congestion so as to minimize speed-dependent CO_{2e} emissions, using a function described by Hickman et al. (1999). The EVRP is modeled on a partition of the working time, and a set of speeds on each arc (i, j) of the network is defined as a function of the departure time from node i. A model for the EVRP described by Figliozzi (2010) uses route and departure times as decision variables, but the model also optimizes speeds as a consequence of the objective function. Conrad and Figliozzi (2010) and Figliozzi (2011) present results related to a variant of the EVRP on a case study in Portland, Oregon, where scenarios with and without congestion are considered. These papers focus on finding approximate, rather than optimal, solutions to the problems, and hence heuristic algorithms are used to generate solutions. Jabali et al. (2012b) take a similar approach by using the same emissions function in a formulation of the time-dependent VRP (without time windows), with speed as an additional decision variable. Travel times are modeled by partitioning the planning horizon into two parts, where one part corresponds to a peak period in which there is congestion and the vehicle speed is fixed, whereas the other part assumes free-flow speeds which can be optimized. Jabali et al. (2012b) describe a tabu search heuristic for this problem.

Another contribution along these lines is due to Bektaş and Laporte (2011) who present the Pollution-Routing Problem (PRP) as an extension of the classical Vehicle Routing Problem with Time Windows (VRPTW). The PRP consists of routing a number of vehicles to serve a set of customers within preset time windows, and determining their speed on each route segment, so as to minimize a function comprising emissions and driver costs. The emissions function used within the PRP is based on a comprehensive emissions model for heavy-duty vehicles described by Barth et al. (2005), and differs from previous work in that it allows to optimize *both* load and speed. The PRP formulation described by Bektaş and Laporte (2011) considers only free-flow speeds of 40 km/h or higher. Demir et al. (2012) extend the PRP formulation to take into account lower speeds, but without looking at congestion *per se*, and describe a heuristic that can solve large-size instances.
A common assumption in the VRPTW is to allow arrival at a customer location before the opening of the time window, but service can only start within the time window. None of the work mentioned above has allowed for idle waiting after service completion as a strategy to avoid congestion. In this chapter we incorporate, for the first time, congestion into the PRP framework so as to adequately account for the adverse effects of low speeds caused by congestion, and we make use of the "idle waiting" strategy.

In this chapter we introduce the Time-Dependent Pollution-Routing Problem (TDPRP), which extends the PRP by explicitly taking into account traffic congestion, and we describe an integer linear programming formulation of the TDPRP where the vehicles speeds are optimized among a set of discrete values. We also provide an analytical characterization of the optimal solutions for a single-arc version of the problem. Finally we report computational experiments with the integer programming formulation on benchmark instances.

The contribution of this chapter is multi-fold and can be stated as follows: (i) we break away from the literature on congestion-aware VRP by using a comprehensive emissions function which includes factors such as load and speed, (ii) we demonstrate how the total travel cost can be significantly reduced by allowing the vehicle to wait at the depot or at a customer node, after the service has been completed, (iii) we propose an integer linear programming formulation of the TDPRP which computationally improves upon the PRP formulation, (iv) we generate new insights on the trade-off between emissions cost and driver wage.

It should be noted at this point that all results derived in this chapter also hold for the special case of zero pollution costs. In other words, our results apply to the problem of optimizing vehicle speeds and departure times in contexts characterized by driver costs, time windows and traffic congestion only.

The remainder of the chapter is structured as follows. The next section presents a formal description of the TDPRP and our general modeling framework. §3.3 provides illustrative examples to motivate the problem. §3.4 describes an integer linear programming formulation of the TDPRP. A complete analytical characterization of the optimal solutions for a single-arc version of the problem is provided in §3.5. Computational results obtained on benchmark instances with the proposed TDPRP formulation are presented in §3.6. Conclusions follow in §3.7.

3.2 Problem description

The TDPRP is defined on a complete graph $G = \{N, A\}$ where N is the set of nodes, 0 is the depot, $N_0 = N \setminus \{0\}$ is the set of customers, and A is the set of arcs between every pair of nodes. The distance between two nodes $i \neq j \in N$ is denoted by d_{ij}. A homogeneous fleet of K vehicles, each with a capacity limit of Q units, is available to serve all customers, where each customer $i \in N_0$ has a non-negative demand q_i. To each customer $i \in N_0$, corresponds a service time h_i and a hard time window $[l_i, u_i]$ in which service must start. In particular, if a vehicle arrives at node i before l_i, it waits until time l_i to start service. Without loss of generality we assume that the vehicle can depart from the depot at time zero (we relax this assumption in §3.5), i.e the service time at the depot is 0.

The following sections present the way in which time dependency and congestion are modeled in the TDPRP, and how CO_{2e} emissions are calculated.

3.2.1 Time-dependency

In the PRP (Bektaş and Laporte, 2011), the travel time of a vehicle depends only on distance and speed, and the latter can be chosen freely. In the TDPRP, the speed also depends on the departure time of the vehicle because it is constrained during periods of traffic congestion. Here, we make use of time-dependent travel times and model traffic congestion using a two-level speed function as in Jabali et al. (2012b). We assume there is an initial period of congestion, lasting a units of time, followed by a period of free-flow. This modeling framework is suitable for routing problems which must be executed in the first half of a given day, e.g., starting from a peak-morning period where traffic congestion is expected, and following which it will dissipate. In the peak-period, the vehicle travels at a *congestion speed* v_c whereas in the period that follows, it is only limited by the speed limits v^{min} and v^{max}, meaning that the vehicle drives at *free-flow speed* $v_f \in [v^{min}, v^{max}]$. These bounds may be explicitly imposed by the driving code or may implicitly result from traffic regulations such as a ban on overtaking for heavy vehicles.

For practical reasons we assume that the speed v_c and the time a are constants which can be extracted from archived travel data (e.g., Hansen et al., 2005) and that the same values hold between every pair of locations.

To model time-dependency, consider two locations spaced out by a distance of d. Let $T(w, v_f)$ denote the travel time of a vehicle between the two locations, that is the time spent by the vehicle on the road depending on its departure time w from the first location, and the chosen free-flow speed v_f. It can be calculated using the following formulation proposed by Jabali et al. (2012b):

$$T(w, v_f) = \begin{cases} \frac{d}{v_c} & \text{if } w \leq \left(a - \frac{d}{v_c}\right)^+ \\ \frac{v_f - v_c}{v_f}(a - w) + \frac{d}{v_f} & \text{if } \left(a - \frac{d}{v_c}\right)^+ < w < a \\ \frac{d}{v_f} & \text{if } w \geq a. \end{cases} \tag{3.1}$$

The calculation of $T(w, v_f)$ suggests that the planing horizon can be divided into three consecutive time regions in terms of the departure time w, as follows:

- The first one $w \in \left[0, \left(a - \frac{d}{v_c}\right)^+\right]$ is called the *all congestion* region: the vehicle leaving the first location within this region makes the entire trip during the congestion period and arrives at the second location after d/v_c units of time.
- The second one $w \in \left[\left(a - \frac{d}{v_c}\right)^+, a\right]$, is called the *transient* region: the vehicle leaving within this region traverses a distance of length $(a - w)v_c$ at

speed v_c and the remaining distance of length $d - (a - w)v_c$ at the chosen free-flow speed v_f.

- The last one $w \in [a, \infty)$, is called the *all free-flow* region, in which the vehicle makes the entire trip at the free-flow speed v_f and completes the journey in d/v_f units of time.

Figure 3.1(left) shows the speed of a vehicle as a function of time for $v_f > v_c$. Figure 3.1(right) shows how T varies with the departure time w given free-flow speed v_f.

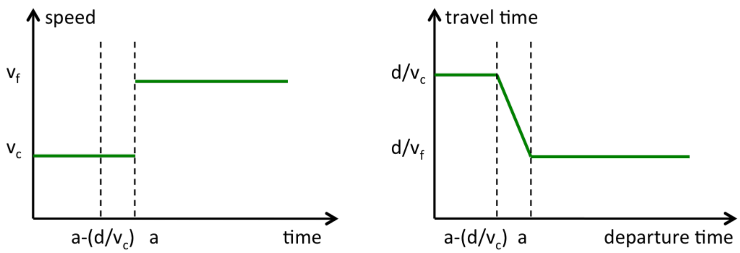


Figure 3.1 Time-dependent speed and travel time profiles.

3.2.2 Modeling emissions

Our modeling of emissions follows the same approach as in Bektaş and Laporte (2011). Here we provide a brief exposition for the sake of completeness. Since CO_{2e} emissions are directly proportional to the amount of fuel consumed, we use the fuel use rate as a proxy to estimate the total amount of CO_{2e} emissions. To calculate fuel consumption, we use the *comprehensive emissions model* of Barth et al. (2005) and Barth and Boriboonsomsin (2008), according to which the instantaneous fuel use rate, denoted FR (liter/s), when traveling at a constant speed v (m/s) with load f (kg) is estimated as

$$FR(v, f) = \frac{\xi}{\kappa\psi}\left(kN_eV + \frac{0.5C_d\rho Av^3 + (\mu + f)v\,(g\sin\phi + gC_r\cos\phi)}{1000\varepsilon\varpi}\right), \quad (3.2)$$

where ξ is fuel-to-air mass ratio, κ is the heating value of a typical diesel fuel (kJ/g), ψ is a conversion factor from grams to liters from (g/s) to (liter/s), k is the engine friction factor (kJ/rev/liter), N_e is the engine speed (rev/s), V is the engine displacement (liter), ρ is the air density (kg/m^3), A is the frontal surface area (m^2), μ is the vehicle curb weight (kg), g is the gravitational constant (equal

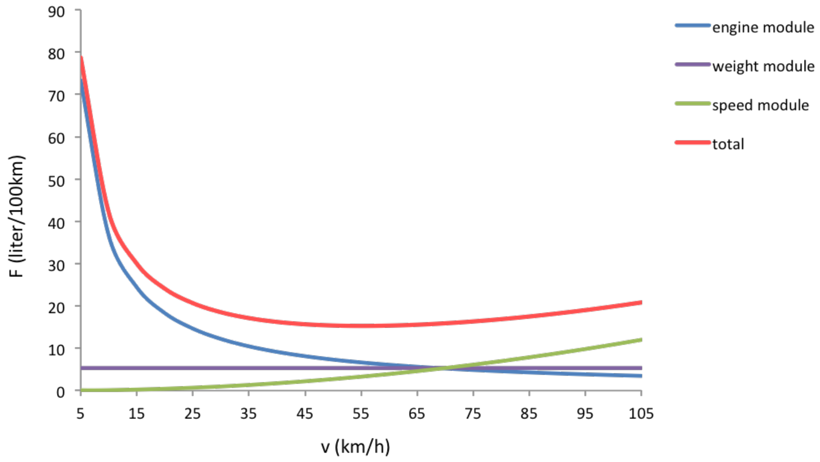


Figure 3.2 Fuel use rate F as a function of speed v

to $9.81 m/s^2$), ϕ is the road angle, C_d and C_r are the coefficient of aerodynamic drag and rolling resistance, ε is vehicle drive train efficiency and ϖ is an efficiency parameter for diesel engines. Using $\alpha = g \sin\phi + gC_r \cos\phi$, $\beta = 0.5C_d A\rho$, $\gamma = 1/(1000\varepsilon\varpi)$ and $\lambda = \xi/\kappa\psi$, (3.2) can be simplified as

$$FR(v, f) = \lambda \left(kN_e V + \gamma \left(\beta v^3 + \alpha(\mu + f)v\right)\right). \tag{3.3}$$

The total amount of fuel used, denoted F (liters), for traversing a distance d (m) at *constant* speed v (m/s) with load f (kg) is equal to the fuel rate multiplied by the travel time d/v:

$$F(v, f) = \lambda \left(kN_e V \frac{d}{v} + \gamma\beta d v^2 + \gamma\alpha(\mu + f)d\right). \tag{3.4}$$

Expression (3.4) contains three terms in the parentheses. We refer to the first term, namely $kN_e V d/v$, as the *engine module* which is linear in the travel time. The second term, $\gamma\beta d v^2$, is called the *speed module*, which is quadratic in the speed. The last term, $\gamma\alpha(\mu + f)d$, is the *weight module*, which is independent of travel time and speed. Figure 3.2 shows the relationship between F and v for a vehicle traveling a distance of 100 km. Other parameters used in generating the figure are given in Table 3.1.

Figure 3.2 shows a U-shape curve between fuel consumption and speed, which is consistent with the behavior of functions suggested by other authors (e.g., Demir et al., 2011), confirming that low speeds (as in the case of traffic congestion) lead to very high fuel use rate. The figure also shows the engine module as the main

Table 3.1 Setting of vehicle and emissions parameters

Notation	Description	Value
ξ	fuel-to-air mass ratio	1
κ	heating value of a typical diesel fuel (kJ/g)	44
ψ	conversion factor (g/liter)	737
k	engine friction factor (kJ/rev/liter)	0.2
N_e	engine speed (rev/s)	33
V	engine displacement (liter)	5
ρ	air density (kg/m^3)	1.2041
A	frontal surface area (m^2)	3.912
μ	curb-weight (kg)	6350
g	gravitational constant (m/s^2)	9.81
ϕ	road angle	0
C_d	coefficient of aerodynamic drag	0.7
C_r	coefficient of rolling resistance	0.01
ε	vehicle drive train efficiency	0.4
ϖ	efficiency parameter for diesel engines	0.9
f_c	fuel price per liter (£)	1.4
d_c	driver wage (£/s)	0.0022

driver of this trend, which contributes considerably to the increase in the amount of emissions at low speeds.

To model the emissions in a time-dependent setting, we rewrite the fuel consumption function F as a function of the departure time w and the free-flow speed v_f on a given arc of length d. If a vehicle traverses the arc in the *all congestion* region, then

$$F(w, v_f) = \lambda \left[kN_eV\ T(w, v_f) + \gamma\beta\ T(w, v_f)v_c^3 + \gamma\alpha(\mu + f)d\right].$$

Similarly, in the *all free-flow* region,

$$F(w, v_f) = \lambda \left[kN_eV\ T(w, v_f) + \gamma\beta\ T(w, v_f)v_f^3 + \gamma\alpha(\mu + f)d\right].$$

When a vehicle traverses the arc in the *transient* region, the speed module needs to be split into two terms since the speed changes before and after the end of the congestion period. In this case

$$F(w, v_f) = \lambda \left[kN_eV\ T(w, v_f) + \gamma\beta \left[(a - w)v_c^3 + (w + T(w, v_f) - a)v_f^3\right] + \gamma\alpha(\mu + f)d\right],$$

where $a - w$ is the time spent in congestion and $w + T(w, v_f) - a$ is the time spent driving at free-flow speed.

In general, let $T^c(w) = \min\{(a - w)^+, d/v_c\}$ be the time spent by the vehicle in congestion and $T^f(w, v_f) = [d - (a - w)^+v_c]^+/v_f$ be the time spent driving at the free-flow speed. We have $T(w, v_f) = T^c(w) + T^f(w, v_f)$ and we can write

$$F(w, v_f) = \lambda kN_eVT(w, v_f) + \lambda\gamma\beta \left[T^c(w)v_c^3 + T^f(w, v_f)v_f^3\right] + \lambda\gamma\alpha(\mu + f)d.$$

3.2.3 Aim of the TDPRP

In the TDPRP, the total travel cost function is composed of the cost of the vehicle emissions and the driver cost for each arc in the network. Let f_c denote the fuel price per liter and let d_c denote the wage rate for the drivers of the vehicles. In this chapter we assume that the CO_{2e} emissions cost is equal to the fuel cost. In practice, we could modify f_c to include the cost of emissions. However, there is considerable debate on the price of CO_{2e} and the method used to estimate it is rather subjective (see the survey paper by Tol (2005) gathering 103 estimates of the marginal damage costs of CO_{2e} emissions), so we have decided not to include it in our numerical calculations.

We consider two ways of calculating the total time for which the driver is paid, which we call *driver wage policies*: (a) the driver of each vehicle is paid starting from the beginning of the planning horizon until returning back to the depot, (b) the driver is paid starting from the start of the service at the origin location until the completion of service at location n. Under driver wage policy (a), the driver reports to the origin location at the start of the planning horizon, which coincides with the start of a typical work day, but may have to wait before starting his or her service and driving duties (during this time he or she may be asked to perform some administrative duties). Under driver wage policy (b), the driver arrives at the origin location right on time to start service.

The aim of the TDPRP is to determine a set of routes, starting and ending at the depot, the speeds on each leg of the routes and departure times from each node so as to minimize the total travel cost. We provide an expression for the cost function in §3.4 and one for the special case of a network with only one arc in §3.5.

In the next section we present a number of numerical examples which illustrate the trade-offs involved in this model. In particular, we outline an important feature of the TDPRP, i.e., that it may be optimal to wait at a node, even after the service is completed, in order to reduce the time spent driving in congestion. Similarly, it may also be optimal for the vehicles not to leave the depot at the start of the planning horizon. Hence, the driver's time at a customer can be spent (i) waiting for the start of service in the case of an early arrival—we call this the *pre-service* wait, (ii) serving the customer, or (iii) waiting after service is completed and before departing to the next customer or back to the depot—we call this the *post-service* wait.

3.3 Examples

The purpose of this section is twofold. We first investigate the impact of considering traffic congestion on the routing and scheduling planning activities.

We then compare the two driver wage policies, namely paying the drivers from the beginning of the planning horizon or from their departure time from the depot. In both cases, we analyze a four-node network where node 0 is the depot at which a single vehicle is based, and $\{1, 2, 3\}$ is the set of customers. The network is depicted in Figure 3.3. Every arc has the same two-level speed profile consisting of an initial congestion period which lasts a seconds, followed by a free-flow period. In the examples below, the congestion speed v_c is set to 10 km/h, the minimum speed limit v^{min} to 50 km/h and the maximum speed limit v^{max} to 110 km/h. The examples differ with respect to the driver wage policy and the time windows at the customer nodes, which are given above each table. We assume that demand and service time at each customer node are zero. The assumption on the demand values entails no loss of generality given that the weight module does not depend on the vehicle speed, as shown in §3.2.2. The parameters used to calculate the total cost function, which are reported in Table 3.1, are taken from Demir et al. (2012).

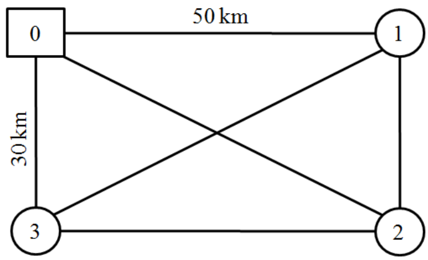


Figure 3.3 Sample four-node instance

3.3.1 Impact of traffic congestion

We consider four examples. In each one, we minimize the total travel cost using two different approaches. In the *time-independent* approach, we ignore traffic congestion when planning the vehicle route and schedule, that is, we assume that the vehicle can always drive at the chosen free-flow speed on each arc of the network. Let S_N denote the solution of the time-independent approach. In the *time-dependent* approach, we account for traffic congestion by solving the TDPRP, the solution of which we denote by S_D. However, the costs for both solutions (denoted by $TC(S_N)$ and $TC(S_D)$) are evaluated under traffic congestion. Since S_D is optimal under traffic congestion, it follows that $TC(S_D) \leq TC(S_N)$, and the difference in cost between the two solutions represents the value of incorporating traffic congestion information in the decision making process. In the example below, the length of the congestion period is equal to 14400 seconds (4 hours).

Example 3.1 (Post-service wait at depot) This example shows that ignoring traffic congestion when planning the route and schedule of the vehicle can lead to a substantial increase in costs. It also shows that adding waiting time at the depot can be used as an effective strategy to mitigate the effect of congestion and reduce the total travel cost. We assume no service time windows at customer nodes: $l_1 = l_2 = l_3 = 0$, and $u_1 = u_2 = u_3 = \infty$. The driver is paid from the beginning of the planning horizon.

The solutions to the time-independent and time-dependent approaches are displayed in Table 3.2. For each solution, the table reports (i) the set of traversed arcs in chronological order from top to bottom under column Arc, (ii) the speed(s) at which each arc is traversed (for an arc traversed during the *transient* region, both the congestion speed and free-flow speed are reported), (iii) the departure time from the origin node, (iv) the post-service waiting time at the origin node, i.e. the additional time that the driver intentionally waits once the service is completed before leaving a node (at the depot the waiting time is equal to the departure time), (v) the emissions cost F, (vi) the driver cost W and (vii) the total cost TC.

Table 3.2 Comparison of S_N and S_D in Example 1

	S_N							S_D					
Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £	Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £
(0, 1)	10, 75.34†	0	0	25.86	32.73	58.59	(0, 1)	75.34	14400	14400	11.47	36.94	48.40
(1, 2)	75.34	14877.8	0	6.88	3.15	10.03	(1, 2)	75.34	16789.2	0	6.88	3.15	10.03
(2, 3)	75.34	16311.3	0	11.47	5.26	16.72	(2, 3)	75.34	18222.7	0	11.47	5.26	16.72
(3, 0)	75.34	18700.5	0	6.88	3.15	10.03	(3, 0)	75.34	20611.8	0	6.88	3.15	10.03
Total				51.09	44.29	95.38					36.70	48.50	85.20

†*transient* region

From Table 3.2, we see that the two solutions yield the same optimal tour $(0, 1, 2, 3, 0)$ and the same set of optimal free-flow speed levels (75.34 km/h on each arc). The difference between the two solutions lies in the fact that the vehicle leaves the depot at time zero in S_N but waits until the end of the congestion period in S_D. Thus, S_D yields a higher driver cost but this increase is more than compensated by an emissions cost saving, yielding a 10.67% total cost saving over S_N (85.20 instead of 95.38).

Example 3.2 (Post-service wait at a customer node) This example shows that ignoring traffic congestion can lead to a significant cost increase when the schedule fails to include post-service wait times which help to mitigate the negative impacts of traffic congestion on emissions costs. It also highlights the difference between pre-service and post-service waits. We assume the following service time windows (in seconds) at customer nodes: $l_1 = 15000, l_2 = 0, l_3 = 11000$, $u_1 = u_2 = \infty$, $u_3 = 12000$. The driver is paid from the beginning of the planning horizon. The solutions to the time-independent and time-dependent approaches are displayed in Table 3.3.

Table 3.3 Comparison of S_N and S_D in Example 2

	S_N							S_D					
Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £	Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £
(0, 3)	10	0	0	17.67	24.20	41.87	(0, 3)	10	0	0	17.67	31.68	49.35
(3, 2)	10, 72†	11000	0	14.69	11.94	26.63	(3, 2)	75.34	14400	3400	11.47	5.26	16.72
(2, 1)	72	16427.8	0	6.75	3.30	10.05	(2, 1)	75.34	16789.2	0	6.88	3.15	10.03
(1, 0)	75.34	17927.8	0	11.47	5.26	16.72	(1, 0)	75.34	18222.7	0	11.47	5.26	16.72
Total				50.58	44.70	95.28					47.49	45.35	92.84

†*transient* region

In this example, S_N and S_D yield the same optimal route but different schedules. In both solutions, the time at which the driver arrives at node 3 is 3200 seconds before the lower limit of the time window, hence there is a positive pre-service wait time at that node. In the S_N solution, the vehicle leaves immediately after serving customer 3, while in the S_D solution it waits until the end of the traffic congestion. Hence, the pre-service and post-service waiting times at node 3 are both positive in S_D. This change in the schedule leads to cost savings of 2.56% over the time-independent solution. From this example, it can be seen that, while pre-service wait times can occur in S_N and S_D, post-service wait times are strategic decisions motivated by the impact of congestion and in this example only occur in S_D, when the driver is paid from the beginning of the planning horizon.

Example 3.3 (Late deliveries due to congestion) This example shows that ignoring traffic congestion can prevent the driver from delivering within the set time windows because he or she chose a suboptimal route and suboptimal free-flow speeds. This can have significant negative consequences in terms of future business profitability. We assume the following service time windows (in seconds) at customer nodes: $l_1 = l_2 = l_3 = 0$, $u_2 = 15500$ and $u_1 = u_3 = \infty$. The driver is paid from the beginning of the planning horizon. The solutions to the time-independent and time-dependent approaches are displayed in Table 3.4.

Table 3.4 Comparison of S_N and S_D in Example 3

	S_N							S_D					
Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £	Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £
(0, 1)	10, 75.34†	0	0	25.86	32.73	58.59	(0, 2)	10, 106.02†	5070.96	5070.96	24.81	34.10	58.91
(1, 2)	75.34	14877.8	inf.	inf.	inf.	inf.	(2, 1)	75.34	15500	0	6.88	3.15	10.03
(2, 3)	75.34	16311.3	inf.	inf.	inf.	inf.	(1, 3)	75.34	16933.5	0	13.37	6.13	19.50
(3, 0)	75.34	18700.5	inf.	inf.	inf.	inf.	(3, 0)	75.34	19719.7	0	6.88	3.15	10.03
Total											51.95	46.54	98.48

†*transient* region

We see from Table 3.4 that the optimal tour for S_N is $(0, 1, 2, 3, 0)$ and the optimal free-flow speed, without congestion, is 75.34 km/h for every arc. Under congestion, however, the vehicle is only able to reach customer 2 after $14877.8 + (30/75.34)3600 = 16311.3$ seconds, that is, with a 13.5 minute delay with respect to the upper time window limit. Because of this delay, S_N is infeasible in the presence of traffic congestion. The optimal route $(0, 2, 1, 3, 0)$ under S_D is different and so are the free-flow speeds (106.02 km/h on the first arc and 75.34 km/h afterwards). By accounting for traffic congestion, the planner realizes that the driver must go to customer 2 first. It does so after an initial waiting time of 5070.96 seconds at the depot, and then proceeds at a speed of 106 km/h to reach customer 2, exactly at the upper limit of its time window, at 15500 seconds.

Example 3.4 (Reduction of driver and emissions costs) This example shows that S_N and S_D solutions can both have strategic wait times but for reasons which are different from those mentioned above. We assume the following service time windows (in seconds) at customer nodes: $l_1 = 19000, l_2 = 0, l_3 = 11000$, $u_1 = u_2 = u3 = \infty$. Contrary to the previous three examples, the driver is now paid from the departure time. The solutions to the time-independent and time-dependent approaches are displayed in Table 3.5. Table 3.5 shows that when there

Table 3.5 Comparison of S_N and S_D in Example 4

	S_N							S_D					
Arc	Speed km/h	Departure time s	Waiting time s	F £	D £	TC £	Arc	Speed km/h	Departure time s	Waiting time s	F £	D £	TC £
(0, 3)	10, 75.34†	13743.8	13743.8	7.54	4.41	11.94	(0, 3)	75.34	14400	14400	6.88	3.15	10.03
(3, 2)	75.34	15746.4	0	11.47	5.26	16.73	(3, 2)	75.34	15833.5	0	11.47	5.26	16.72
(2, 1)	75.34	18135.6	0	6.88	3.15	10.04	(2, 1)	75.34	18222.7	0	6.88	3.15	10.03
(1, 0)	75.34	19569.1	0	11.47	5.26	16.73	(1, 0)	75.34	19656.2	0	11.47	5.26	16.72
Total				37.36	18.07	55.43					36.70	16.82	53.52

†*transient* region

are lower time window restrictions at the customers and the driver is paid from its departure time, there can be strategic post-service waiting time at the depot in both solutions S_N and S_D. In the S_N solution, the reason for delaying the vehicle's departure is to reduce the driver cost by avoiding pre-service wait at the customer node. In contrast, in S_D solution, there is another reason for delaying the vehicle's departure, which is the desire to avoid traveling in congestion, thereby reducing emissions cost.

From the four examples just presented, we conclude that ignoring traffic congestion can have detrimental consequences on the timing of deliveries. Congestion is likely to increase costs or even lead to an infeasible solution (which can be seen as a solution with infinite costs) when customer nodes have delivery time windows. This is because the planner does not incorporate strategic post-service wait times motivated by traffic congestion in the vehicle schedules. We show that these strategic wait times can occur either at the depot or at the customer nodes.

3.3.2 Impact of the driver wage policy

In this section we investigate the impact of the driver wage policy on the optimal TDPRP solution, namely whether the driver is paid from the beginning of the planning horizon or from the departure time. In the example below, the length of the congestion period is equal to 7200 seconds.

Example 3.5 (Impact of driver wage policy on wait time and routing) In this example we assume the following service time windows (in seconds) at customer nodes: $l_1 = l_2 = 9000, l_3 = 10000$, $u_1 = 19000, u_2 = 15000, u_3 = 12000$. The optimal solutions for the two driver wage policies are compared in Table 3.6.

Table 3.6 Comparison of the driver wage policies in Example 5

	S_D The driver is paid from the beginning of the planning horizon							S_D The driver is paid from departure					
Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £	Arc	Speed km/h	Departure time s	Waiting time s	F £	W £	TC £
(0, 1)	97.5	7200	7200	13.64	19.89	33.53	(0, 3)	75.34	8566.5	8566.5	6.88	3.15	10.03
(1, 2)	97.5	9046.15	0.00	8.17	2.44	10.61	(3, 2)	75.34	10000	0	11.47	5.26	16.73
(2, 3)	97.5	10153.8	0.00	13.59	4.07	17.66	(2, 1)	75.34	12389.2	0	6.88	3.15	10.03
(3, 0)	75.34	12000.00	0.00	6.88	3.15	10.03	(1, 0)	75.34	13822.7	0	11.47	5.26	16.73
Total				42.28	29.55	71.83					36.70	16.82	53.52

Table 3.6 shows that the driver wage policy may affect the resulting route. When the driver is paid from the beginning of the planning horizon, the optimal route is $(0, 1, 2, 3, 0)$ and it is optimal to wait until the end of the congestion period. When the driver is paid from the departure time, it is optimal to postpone the departure until after the end of the congestion period but this requires a change of route to $(0, 3, 2, 1, 0)$ in order to meet the delivery time windows.

In summary, we see that it is important to take the driver wage policy into account when optimizing the route and schedule of the vehicles. When the driver is paid from the departure time, he or she generally leaves the depot later than if he or she was paid from the beginning of the planning horizon, but this delay has to be compensated by either a change of route or a speed increase.

3.4 MILP formulation

This section presents a mathematical formulation for the TDPRP. The objective is to determine a set of routes for the K vehicles, all starting and ending at the depot, along with their speeds on each arc, and then departure times from each node so as to minimize a total cost function encompassing driver and emissions costs. The objective function is not linear in the speed values. To linearize it, we discretize the free-flow speed following an approach used by Bektaş and Laporte (2011). Let

$R = \{1, \ldots, k\}$ be the index set of different speed levels and $v^1, \ldots, v^k$ denote the corresponding free-flow speeds where $v_c \leq v^{min} = v^1 < \ldots < v^k = v^{max}$. Figure 3.4 illustrates the different speed values and corresponding travel time functions. Let $b^0 = 0$, $b^1 = (a - d/v_c)^+$, $b^2 = a$ and $b^3 = \infty$ and let $[b^{m-1}, b^m)$ denote the

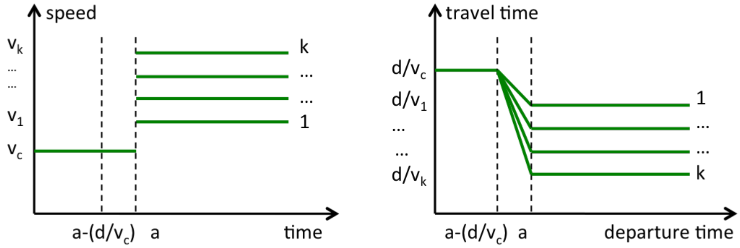


Figure 3.4 Time-dependent speed and travel time profiles

m-th time interval, where $m \in \{1, 2, 3\}$. Specifically, $m = 1$ is the *all congestion* region, $m = 2$ is the *transient* region and $m = 3$ is the *all free-flow* region. We also define ν^{mr} as the vehicle speed in time region m given free-flow speed v^r with $r \in R$, that is, $\nu^{1r} = v_c$, $\nu^{2r} = v_c$ and $\nu^{3r} = v^r$. These definitions allow us to rewrite (3.1) for arc (i, j) as $T(w, v_r) = \theta^{mr} w + \eta_{ij}^{mr}$, if $b^{m-1} \leq w < b^m$ and $r \in R$, where,

$$\theta^{mr} = \begin{cases} 0 & m = 1, 3 \\ \frac{\nu^{2r} - \nu^{3r}}{\nu^{3r}} & m = 2, \end{cases}$$

$$\eta_{ij}^{mr} = \begin{cases} \frac{d_{ij}}{\nu^{1r}} & m = 1 \\ \frac{d_{ij}}{\nu^{3r}} + \left(\frac{\nu^{3r} - \nu^{2r}}{\nu^{3r}}\right) a & m = 2 \\ \frac{d_{ij}}{\nu^{3r}} & m = 3. \end{cases}$$

The model uses the following decision variables:

x_{ij}	binary variable equal to 1 if a vehicle traverses arc $(i, j) \in A$, 0 otherwise,
z_{ij}^{mr}	binary variable equal to 1 if a vehicle traverses arc $(i, j) \in A$, leaving node i within time interval $m \in \{1, 2, 3\}$ with the free-flow speed v_r with $r \in R$, 0 otherwise,
f_{ij}	load carried on arc (i, j),
w_{ij}^{mr}	variable equal to the time instant at which a vehicle leaves node $i \in N$ to traverse arc (i, j) if within time interval $m \in \{1, 2, 3\}$ with the free-flow speed v_r with $r \in R$,

s_i	total time spent on a route that has node $i \in N_0$ as last visited before returning to the depot,
φ_i	time at which service at node $i \in N_0$ starts.

Given these variables, $\theta_{ij}^{mr} w_{ij}^{mr} + \eta_{ij}^{mr} z_{ij}^{mr}$ is equal to the travel time of a vehicle on arc $(i, j) \in A$ if the vehicle leaves node i within time interval $m \in \{1, 2, 3\}$ and uses free-flow speed v^r with $r \in R$.

We now present a mixed integer linear programming formulation for the TDPRP:

$$\text{Minimize} \quad \sum_{(i,j)\in A} \sum_{r\in R} \sum_{m=1}^{3} f_c \lambda k N_e V (\theta_{ij}^{mr} w_{ij}^{mr} + \eta_{ij}^{mr} z_{ij}^{mr}) \tag{3.5}$$

$$+ \sum_{(i,j)\in A} \sum_{r\in R} \sum_{m=1,3} f_c \lambda \gamma \beta (\nu^{mr})^3 (\theta_{ij}^{mr} w_{ij}^{mr} + \eta_{ij}^{mr} z_{ij}^{mr}) \tag{3.6}$$

$$+ \sum_{(ij)\in A} \sum_{r\in R} f_c \lambda \gamma \beta (\nu^{2r})^3 (a z_{ij}^{2r} - w_{ij}^{2r}) \tag{3.7}$$

$$+ \sum_{(i,j)\in A} \sum_{r\in R} f_c \lambda \gamma \beta (\nu^{3r})^3 (w_{ij}^{2r} + \theta_{ij}^{2r} w_{ij}^{2r} + \eta_{ij}^{2r} z_{ij}^{2r} - a z_{ij}^{2r}) \tag{3.8}$$

$$+ \sum_{(i,j)\in A} f_c \lambda \gamma \alpha_{ij} d_{ij} (\mu x_{ij} + f_{ij}) \tag{3.9}$$

$$+ \sum_{i\in N_0} d_c s_i \tag{3.10}$$

subject to

$$\sum_{j\in N} x_{0j} = K \tag{3.11}$$

$$\sum_{i\in N} x_{ij} = 1 \qquad \forall j \in N_0 \tag{3.12}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in N_0 \tag{3.13}$$

$$\sum_{j\in N} f_{ji} - \sum_{j\in N} f_{ij} = q_i \qquad \forall i \in N_0 \tag{3.14}$$

$$q_j x_{ij} \le f_{ij} \le x_{ij}(Q - q_i) \qquad \forall (i,j) \in A \tag{3.15}$$

$$z_{ij}^{mr} b_{ij}^{m-1} \le w_{ij}^{mr} \le z_{ij}^{mr} b_{ij}^{m} \qquad \forall (i,j) \in A, m \in \{1,2,3\}, r \in R \tag{3.16}$$

$$\sum_{i\in N} \sum_{m=1}^{3} \sum_{r\in R} \left(w_{ij}^{mr} + \theta_{ij}^{mr} w_{ij}^{mr} + \eta_{ij}^{mr} z_{ij}^{mr} \right) \le \varphi_j \qquad \forall j \in N_0 \tag{3.17}$$

$$\sum_{j\in N} \sum_{r\in R} \sum_{m=1}^{3} w_{ij}^{mr} \ge \varphi_i + h_i \qquad \forall i \in N_0 \tag{3.18}$$

$$l_i \le \varphi_i \le u_i \qquad \forall i \in N_0 \tag{3.19}$$

$$s_i \ge \sum_{r\in R} \sum_{m=1}^{3} (w_{i0}^{mr} + \theta_{i0}^{mr} w_{i0}^{mr} + \eta_{i0}^{mr} z_{i0}^{mr}) \qquad \forall i \in N_0 \tag{3.20}$$

$$\sum_{m=1}^{3}\sum_{r\in R} z_{ij}^{mr} = x_{ij} \qquad \forall (i,j)\in A \tag{3.21}$$

$$z_{ij}^{mr} \in \{0,1\} \qquad \forall (i,j)\in A, m\in\{1,2,3\}, r\in R \tag{3.22}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j)\in A \tag{3.23}$$

$$f_{ij} \geq 0 \qquad \forall (i,j)\in A. \tag{3.24}$$

The first five parts of the objective function represent the cost of emissions. In particular, (3.5) computes the cost induced by the *engine module*, the terms (3.6)–(3.8) measure the cost induced by the *speed module*, and (3.9) is the cost induced by the *weight module*. More precisely, (3.6) calculates the emissions cost generated by the *speed module* in the *all congestion* and in the *all free-flow* regions, while (3.7) and (3.8) represent the emissions cost generated by the speed module in the *transient* region. Finally, the last term (3.10) measures the total driver wage when the driver is paid from the beginning of the planning horizon. In contrast, if the driver was paid from its departure time, the total driver wage would be $\sum_{i\in N_0} d_c s_i - \sum_{j\in N_0}\sum_{r\in R}\sum_{m=1}^{3} d_c w_{0j}^{mr}$.

Constraint (3.11) indicates that exactly K vehicles depart from the depot. Constraints (3.12) and (3.13) guarantee that each customer is visited exactly once. Constraints (3.14) and (3.15) model the flow on each arc, and ensure that vehicle capacities are respected. The boundary conditions on the departure time are imposed by constraint (3.16). Constraints (3.17) and (3.18) are used to express the temporal relationship between arrival time and service time, and between service time and departure time, respectively. The time windows restrictions at customer nodes are imposed by constraint (3.19). Constraint (3.20) computes the time at which the vehicle returns to the depot. The relationship between speed and arc-traversal variables is expressed by constraint (3.21). Finally, constraints (3.22)–(3.24) enforce the integrality and nonnegativity restrictions on the variables.

We provide a numerical analysis of the performance of this formulation in §3.6.

3.5 Analytical results based on a single-arc network

We now consider a special case of the TDPRP on a network with two nodes spaced by a distance d, i.e., the depot and one customer node. The aim is to gain insights by analyzing this special case of the problem; as will be shown in Chapter 5 and §3.6, the results obtained in this section are useful for optimizing the TDPRP on a fixed route and for improving the computational performance of the integer linear programming formulation.

We minimize the cost of going from the depot to the customer node (hence, ignoring the return trip to the depot). The customer node has a time window $[l, u]$, service time at the customer node is equal to h. We assume, without loss of generality, that the demand at the customer is equal to zero and that there is a two-level speed profile with an initial congestion period a, as described in §3.2.1.

In this special case there are only two decision variables: the departure time w from the depot and the free-flow speed v_f for the vehicle serving the customer. We must have $v_f \in [v^{min}, v^{max}]$ and $w \geq \epsilon$, where $\epsilon \geq 0$ is the earliest time at which the vehicle can leave the depot. For example ϵ can represent loading time at the depot. We refer to ϵ as the beginning of the planning horizon; $w - \epsilon$ is the (strategic) waiting time at the depot. Without loss of generality we assume that $a \geq \epsilon$ and $\epsilon \leq l \leq u \leq \infty$ (for example, if $a < \epsilon$, then the problem can be solved by setting $a = \epsilon$).

Our objective is to minimize the total cost function $TC(w, v_f)$ so that the arrival time at the customer node does not exceed u. In other words, the optimization problem is

$$\begin{aligned} \underset{\substack{w \geq \epsilon \\ v^{min} \leq v_f \leq v^{max}}}{\text{minimize}} \quad TC(w, v_f) \quad &= \quad f_c F(w, v_f) + d_c W(w, v_f) \\ \text{subject to} \quad & \quad T(w, v_f) + w \leq u, \end{aligned}$$

where F and T are as defined in §3.2 and $W(w, v_f)$ denotes the time for which the driver is paid. If the driver is paid from the beginning of the planning horizon (i.e., ϵ), then $W(w, v_f) = \max\left\{w - \epsilon + T(w, v_f), l - \epsilon\right\} + h$. If the driver is paid from the departure time (i.e., w), then $W(w, v_f) = \max\{T(w, v_f), l - w\} + h$.

For the single-arc problem to be feasible, the vehicle must be able to reach the customer node by time u if it does not wait at the depot, i.e. if $w = \epsilon$. By leaving immediately, the vehicle is either (i) in the *all congestion* region, i.e., when $\epsilon \leq a - d/v_c$, in which case $u \geq \epsilon + d/v_c$, or (ii) in the *transient* region, i.e., when $\epsilon \geq a - d/v_c$, in which case $u \geq a + (d - (a - \epsilon)v_c)/v^{max}$. We can summarize these two conditions as follows: $u \geq \min\{a, \epsilon + d/v_c\} + (d - (a - \epsilon)v_c)^+/v^{max}$. In what follows we assume that this condition is satisfied.

Let v_w^u be the free-flow speed required for the driver to arrive at the customer exactly at time u when leaving the depot at time w. Then

$$v_w^u = \begin{cases} \frac{d - (a-w)^+ v_c}{u - \max\{a, w\}}, & \text{if } w \in [\max\{\epsilon, a - d/v_c\}, u] \text{ and } u > a \\ \infty, & \text{otherwise.} \end{cases}$$

Similarly, let v_w^l be the free-flow speed required for the driver to arrive at the customer exactly at time l when leaving the depot at w. Then

$$v_w^l = \begin{cases} \frac{d - (a-w)^+ v_c}{l - \max\{a, w\}}, & \text{if } w \in [\max\{\epsilon, a - d/v_c\}, l] \text{ and } l > a \\ \infty, & \text{otherwise.} \end{cases}$$

The departure time w from the depot must be such that $v_w^u \leq v^{max}$ otherwise it is not possible to arrive by time u. Let w_{max}^u denote the time at which the vehicle needs to depart from the depot to reach the customer at exactly time u, driving at free-flow speed v^{max}.

$$w^u_{max} = \begin{cases} u - \frac{d}{v^{max}}, & \text{if } v^{max} \geq v^u_a \text{ and } u > a \\ a - \frac{d-(u-a)v^{max}}{v_c}, & \text{if } v^{max} < v^u_a \text{ and } u > a \\ u - \frac{d}{v_c} & \text{if } \epsilon \leq u \leq a. \end{cases}$$

In other words, w^u_{max} is an upper bound on the departure time, i.e., for a value of the departure time w to be feasible we need $w \in [\epsilon, w^u_{max})$. Similarly let w^l_{max} be the maximum departure time such that the driver arrives exactly at time l driving at free-flow speed v^{max}:

$$w^l_{max} = \begin{cases} l - \frac{d}{v^{max}}, & \text{if } v^{max} \geq v^l_a \text{ and } l > a \\ a - \frac{d-(l-a)v^{max}}{v_c}, & \text{if } v^{max} < v^l_a \text{ and } l > a \\ l - \frac{d}{v_c} & \text{if } \epsilon \leq l \leq a. \end{cases}$$

We first determine the optimal free-flow speed v_f for a given departure time $w \in [\max\{\epsilon, a - d/v_c\}, w^u_{max}]$. As shown in Lemma 3.1, this can be done by comparing the speed levels v^l_w and v^u_w to two key speed levels: $\overline{v} = ((f_c\lambda k N_e V + d_c)/2f_c\lambda\beta\gamma)^{1/3}$ and $\underline{v} = (kN_eV/2\beta\gamma)^{1/3}$. The speed level $\overline{v}$ minimizes emissions and driver costs, i.e., TC, in the absence of any time window, whereas the speed $\underline{v}$ minimizes emissions consumption only, i.e., F, in the absence of any time windows. Both values are independent of the departure time w. These speed values have previously been identified by Demir et al. (2012).

Lemma 3.1 *Consider a single-arc TDPRP instance and a departure time w such that $w \in [\max\{\epsilon, a - d/v_c\}, w^u_{max}]$. The optimal free-flow speed is $\min\{v^{max}, v^*\}$, where v^* is given as follows: (i) if $v^l_w \leq \underline{v}$ then v^* is $\max\{v^{min}, \underline{v}\}$, (ii) if $\underline{v} \leq v^l_w \leq \overline{v}$ then v^* is $\max\{v^{min}, v^l_w\}$, (iii) if $v^u_w \leq \overline{v} \leq v^l_w$ then v^* is $\max\{v^{min}, \overline{v}\}$, (iv) if $\overline{v} \leq v^u_w$ then v^* is $\max\{v^{min}, v^u_w\}$.*

Proof: Let $A = f_c\lambda\gamma\alpha(\mu + f)$, $B = f_c\lambda k N_e V$ and $C = f_c\lambda\beta\gamma$, $D = d_c$. Note that $A, B, C, D \geq 0$. First note that since $w \leq w^u_{max}$, we have $v^{max} \geq v^u_w$. For a fixed w, we need to minimize TC with respect to v_f in $[\max\{v^u_w, v^{min}\}, v^{max}]$.

When the driver is paid from the beginning of the planning horizon, the total cost function TC for a fixed w as a function of the free-flow speed can be written as

$$TC(w, v_f) = \begin{cases} Ad + (B + D + Cv_c^3)T_c(w) \\ +(B + D + Cv_f^3)T_f(w, v_f) \\ +D(w - \epsilon) & \text{if } \max\{v^u_w, v^{min}\} \leq v_f \leq \max\{v^l_w, v^{min}\} \\ Ad + (B + Cv_c^3)T_c(w) \\ +(B + Cv_f^3)T_f(w, v_f) \\ +D(l - \epsilon) & \text{if } v_f \geq \max\{v^l_w, v^{min}\}. \end{cases}$$

For a fixed w, the function TC is continuous in v_f and is made of two pieces which are both convex in v_f. More precisely, the first piece is minimized at $v_f = \overline{v}$, while the second one at $v_f = \underline{v}$. Note that $\underline{v} < \overline{v}$.

In case (i) the first part is non-increasing and the second one is minimized at $\underline{v}$. If $\underline{v} > v^{max}$, the global minimum is achieved at v^{max}, otherwise it is achieved at $\max\{v^{min}, \underline{v}\}$. In case (ii) the first part is non-increasing and the second one is non-decreasing. If $v_w^l > v^{max}$, the global minimum is achieved at v^{max}, otherwise it is achieved at $\max\{v^{min}, v_w^l\}$. In case (iii) the first part is minimized at $\overline{v}$, while the second one is increasing. If $\overline{v} > v^{max}$, the global minimum is achieved at v^{max}, otherwise it is achieved at $\max\{v^{min}, \overline{v}\}$. Finally, in case (iv) both parts are non-decreasing so the global minimum is achieved at $\max\{v^{min}, v_w^u\}$.

When the driver is paid from the departure time, the total cost function has an extra $-D(w - \epsilon)$ term, which does not depend on v_f. Hence, the solution is the same. □

Note that the optimal speed for a given departure time does not depend on the driver wage policy. Using Lemma 3.1, we can reduce the problem of minimizing TC to a unidimensional optimization problem, that is, we set w as the unique decision variable.

Observe that the minimum speed limitation only affects the optimal solution if $v^{min} > \underline{v}$.

We now provide the full characterization of the optimal solution for the special case where $v^{min} \leq \underline{v}$ (for the sake of conciseness) [1]. Let $S = (w^*, v_f^*)$ denote a solution, where w^* is the optimal departure time and v_f^* is the optimal free-flow speed, Proposition 3.1 provides the solution when the driver is paid from the beginning of the planning horizon, i.e, from time ϵ, and Proposition 3.2 provides the solution when the driver is paid from the departure time i.e., from time w. Observe that whenever the vehicle traverses the entire arc during the congestion period, the free-flow speed is never used but we may still write $S = (w^*, v_f^*)$, with v_f^* being equal to any positive value.

Proposition 3.1 *Consider a single-arc TDPRP instance. If the driver is paid from the beginning of the planning horizon, the optimal solution depends mainly on the relative values of the nine speed levels:* v^{max}, $\underline{v}$, $\overline{v}$, $\hat{v} = ((kN_eV + \beta\gamma v_c^3)/3\beta\gamma v_c)^{1/2}$, $\check{v} = ((f_c\lambda kN_eV + d_c + f_c\lambda\beta\gamma(v^{max})^3)/3f_c\lambda\beta\gamma v^{max})^{1/2}$, v_ϵ^l, v_a^l, v_ϵ^u *and* v_a^u *and is given in Table 3.22 in §3.B.*

Proof: Let $A = f_c\lambda\gamma\alpha(\mu + f)$, $B = f_c\lambda kN_eV$ and $C = f_c\lambda\beta\gamma$, $D = d_c$. Note that $A, B, C, D \geq 0$. In the following tables, we use circled numbers such as ①

[1] For most practical purposes, it is reasonable to assume that the minimum speed limit is lower than the speed which minimizes emissions costs.

and ②, to refer to the pieces of the TC function. For each piece we use symbols such as $\rightarrow$, $\nearrow$, $\searrow$ and $\smile$, to indicate whether the TC function is respectively constant, non-decreasing, non-increasing or convex, with respect to w. Let $T(w) = \min_{v_f \in [v^{min}, v^{max}]} TC(w, v_f)$ such that $w + T(w, v_f) \leq u$. We consider three cases: (1) $l \leq u \leq a$, (2) $l < a < u$ and (3) $a \leq l < u$.

In case (1), we have:
footnotesize

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - \epsilon) & \text{if } \epsilon \leq w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} + Dw & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \leq w \leq u - \frac{d}{v_c}. \end{cases}$$

The first piece is constant in w and the second is increasing in w. So any departure time in $\left[\epsilon, \max\left\{\epsilon, l - \frac{d}{v_c}\right\}\right]$ is optimal. We summarize this information in Table 3.7

Table 3.7 Case 1

Case	①	②	Solution
1	$\rightarrow$	$\nearrow$	(w, v_f) with $w \in \left[\epsilon, \max\left\{\epsilon, l - \frac{d}{v_c}\right\}\right]$

where ① and ② are the time regions delimited by the breakpoints: $\max\left\{\epsilon, l - \frac{d}{v_c}\right\}$ and $u - \frac{d}{v_c}$.

In case (2), we distinguish two subcases: (2.1) $v^{max} < \overline{v}$, (2.2) $v^{max} \geq \overline{v}$.

In case (2.1):

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - \epsilon) & \text{if } \epsilon \leq w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} + Dw & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \leq w \\ & \text{and } w < \max\left\{\epsilon, a - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)(a - w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d - (a-w)^+ v_c}{v^{max}} + Dw & \text{if } \max\left\{\epsilon, a - \frac{d}{v_c}\right\} \leq w \leq w_{max}^u. \end{cases}$$

Table 3.8 gives the solution depending on which piece contains the value a.

Table 3.8 Case 2.1

Case	$a \in$	Condition 1	Condition 2	①	②	③	④	Solution
2.1.1.1	$\left[\max\left\{\epsilon, a - \frac{d}{v_c}\right\}, w_{max}^u\right)$	$v_a^u \leq v^{max}$	$\hat{v} \geq \check{v}$	$\rightarrow$	$\nearrow$	$\searrow$	$\nearrow$	(a, v^{max}) or (ϵ, v^{max})
2.1.1.2	$\left[\max\left\{\epsilon, a - \frac{d}{v_c}\right\}, w_{max}^u\right)$	$v_a^u \leq v^{max}$	$\hat{v} \leq \check{v}$	$\rightarrow$	$\nearrow$	$\nearrow$	$\nearrow$	(ϵ, v^{max})
2.1.2.1	$[w_{max}^u, \infty)$	$v_a^u \geq v^{max}$	$\hat{v} \geq \check{v}$	$\rightarrow$	$\nearrow$	$\searrow$		(w_{max}^u, v^{max}) or (ϵ, v^{max})
2.1.2.2	$[w_{max}^u, \infty)$	$v_a^u \geq v^{max}$	$\hat{v} \leq \check{v}$	$\rightarrow$	$\nearrow$	$\nearrow$		(ϵ, v^{max})

In some cases, there are two possible solutions. Then, the optimal solution can be obtained by calculating the cost associated with each one of them to find out which is the least (note that this needs to be done only if $\epsilon < a - \frac{d}{v_c}$, otherwise the solution with $w > \epsilon$ is the optimal one).

In case (2.2)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l-\epsilon) & \text{if } \epsilon \le w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} + Dw & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \le w < \max\left\{\epsilon, a - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C\overline{v}^3)\frac{d-(a-w)^+ v_c}{\overline{v}} + Dw & \text{if } \max\left\{\epsilon, a - \frac{d}{v_c}\right\} \le w < \max\left\{\epsilon, \overline{w}^u\right\} \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C(v_w^u)^3)\frac{d-(a-w)^+ v_c}{v_w^u} + Dw & \text{if } \max\left\{\epsilon, \overline{w}^u\right\} \le w \le w_{max}^u. \end{cases}$$

where

$$\overline{w}^u = \begin{cases} a - \frac{d-(u-a)\overline{v}}{v_c} & \text{if } v_a^u \ge \overline{v} \\ u - \frac{d}{\overline{v}} & \text{otherwise.} \end{cases}$$

Table 3.9 gives the solution in all possible subcases.

Table 3.9 Case 2.2

Case	$a \in$	Condition 1	Condition 2	Condition 3	①	②	③	④	⑤	Solution
2.2.1.1	$\left[\max\left\{\epsilon, a - \frac{d}{v_c}\right\}, \overline{w}^u\right)$	$v_a^u \le \overline{v}$	$\hat{v} \ge \overline{v}$		$\rightarrow$	$\nearrow$	$\searrow$	$\nearrow$	$\nearrow$	$(a, \overline{v})$ or $(\epsilon, \overline{v})$
2.2.1.2	$\left[\max\left\{\epsilon, a - \frac{d}{v_c}\right\}, \overline{w}^u\right)$	$v_a^u \le \overline{v}$	$\hat{v} \le \overline{v}$		$\rightarrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$(\epsilon, \overline{v})$
2.2.2.1.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \le \overline{v}$	$\rightarrow$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$	$(\epsilon, \overline{v})$
2.2.2.1.2	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \ge \overline{v}$	$\nearrow$	$\nearrow$				(ϵ, v_ϵ^u)
2.2.2.2.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\overline{v} \le \hat{v} \le v_a^u$	$v_\epsilon^u \le \hat{v}$	$\rightarrow$	$\nearrow$	$\searrow$	$\smile$	$\nearrow$	$(\hat{w}^u, \hat{v})$ or $(\epsilon, \overline{v})$
2.2.2.2.2	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\overline{v} \le \hat{v} \le v_a^u$	$v_\epsilon^u \ge \hat{v}$	$\nearrow$	$\nearrow$				(ϵ, v_ϵ^u)
2.2.2.3	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \ge v_a^u$		$\rightarrow$	$\nearrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v_u^a) or $(\epsilon, \overline{v})$
2.2.3.1.1	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \le \overline{v}$	$\rightarrow$	$\nearrow$	$\nearrow$	$\nearrow$		$(\epsilon, \overline{v})$
2.2.3.1.2	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \ge \overline{v}$	$\nearrow$	$\nearrow$				(ϵ, v_ϵ^u)
2.2.3.2.1	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\overline{v} \le \hat{v} \le v^{max}$	$v_\epsilon^u \le \hat{v}$	$\rightarrow$	$\nearrow$	$\searrow$	$\smile$		$(\hat{w}^u, \hat{v})$ or $(\epsilon, \overline{v})$
2.2.3.2.2	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\overline{v} \le \hat{v} \le v^{max}$	$v_\epsilon^u \ge \hat{v}$	$\nearrow$					(ϵ, v_ϵ^u)
2.2.3.3	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \ge v^{max}$		$\rightarrow$	$\nearrow$	$\searrow$	$\searrow$		(w_{max}^u, v^{max}) or $(\epsilon, \overline{v})$

where $\hat{w}^u = a - (d - (u-a)\hat{v})/v_c$.

In case (3), we distinguish three subcases: (3.1) $v^{max} < \underline{v}$, (3.2) $\underline{v} \le v^{max} < \overline{v}$, (3.3) $v^{max} \ge \overline{v}$.

In case (3.1)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l-\epsilon) & \text{if } \epsilon \le w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \le w \\ +(B + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} + D(l-\epsilon) & \text{and } w < \max\left\{\epsilon, w_{max}^l\right\} \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} + Dw & \text{if } \max\left\{\epsilon, w_{max}^l\right\} \le w \le w_{max}^u. \end{cases}$$

Table 3.10 gives the solution in all possible subcases.

Table 3.10 Case 3.1

Case	$a \in$	Condition 1	Condition 2	①	②	③	④	Solution
3.1.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \max\left\{\epsilon, w^l_{max}\right\}\right)$	$v^l_a \leq v^{max}$		$\rightarrow$	$\searrow$	$\rightarrow$	$\nearrow$	(w, v^{max}) where $w \in [a, w^l_{max}]$
3.1.2.1	$[w^l_{max}, w^u_{max}]$	$v^u_a \leq v^{max} \leq v^l_a$	$\check{v} \leq \hat{v}$	$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v^{max})
3.1.2.1	$[w^l_{max}, w^u_{max}]$	$v^u_a \leq v^{max} \leq v^l_a$	$\check{v} \geq \hat{v}$	$\rightarrow$	$\searrow$	$\nearrow$	$\nearrow$	$(\max\left\{\epsilon, w^l_{max}\right\}, v^{max})$
3.1.3.1	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\check{v} \leq \hat{v}$	$\rightarrow$	$\searrow$	$\searrow$		(w^u_{max}, v^{max})
3.1.3.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\check{v} \geq \hat{v}$	$\rightarrow$	$\searrow$	$\nearrow$		$(\max\left\{\epsilon, w^l_{max}\right\}, v^{max})$

In case (3.2)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l-\epsilon) & \text{if } \epsilon \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w \\ +(B + C\underline{v}^3)\frac{d-(a-w)^+ v_c}{\underline{v}} + D(l-\epsilon) & \text{and } w < \max\left\{\epsilon, \underline{w}^l\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \\ +(B + C(v^l_w)^3)\frac{d-(a-w)^+ v_c}{v^l_w} + D(l-\epsilon) & \text{if } \max\left\{\epsilon, \underline{w}^l\right\} \leq w < \max\left\{\epsilon, w^l_{max}\right\} \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} + Dw & \text{if} \max\left\{\epsilon, (w^l_{max})\right\} \leq w \leq w^u_{max}, \end{cases}$$

where

$$\underline{w}^l = \begin{cases} a - \frac{d-(l-a)\underline{v}}{v_c} & \text{if } v^l_a \geq \underline{v} \\ l - \frac{d}{\underline{v}} & \text{otherwise.} \end{cases}$$

Table 3.11 gives the solution in all possible subcases.

Table 3.11 Case 3.2

Case	$a \in$	Condition 1	Condition 2	Condition 3	①	②	③	④	⑤	Solution
3.2.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \underline{w}^l\right)$	$v^l_a \leq \underline{v}$			$\rightarrow$	$\searrow$	$\rightarrow$	$\nearrow$	$\nearrow$	$(w, \underline{v})$ where $w \in [a, \underline{w}^l]$
3.2.2.1.1	$[\underline{w}^l, w^l_{max})$	$\underline{v} \leq v^l_a \leq v^{max}$	$v^l_a \geq \hat{v}$	$v^l_\epsilon \leq \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$	$(\hat{w}^l, \hat{v})$
3.2.2.1.2	$[\underline{w}^l, w^l_{max})$	$\underline{v} \leq v^l_a \leq v^{max}$	$v^l_a \geq \hat{v}$	$v^l_\epsilon \geq \hat{v}$	$\nearrow$	$\nearrow$	$\nearrow$			(ϵ, v^l_ϵ)
3.2.2.2	$[\underline{w}^l, w^l_{max})$	$\underline{v} \leq v^l_a \leq v^{max}$	$v^l_a \leq \hat{v}$		$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	(a, v^l_a)
3.2.3.1.1	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$\hat{v} \leq v^{max}$	$v^l_\epsilon \leq \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$	$(\hat{w}^l, \hat{v})$
3.2.3.1.2	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$\hat{v} \leq v^{max}$	$\hat{v} \leq v^l_\epsilon \leq v^{max}$	$\nearrow$	$\nearrow$	$\nearrow$			(ϵ, v^l_ϵ)
3.2.3.1.3	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$\hat{v} \leq v^{max}$	$v^l_\epsilon \geq v^{max}$	$\nearrow$	$\nearrow$				(ϵ, v^{max})
3.2.3.2.1	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$v^{max} \leq \hat{v} \leq \check{v}$	$v^l_\epsilon \leq v^{max}$	$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	(w^l_{max}, v^{max})
3.2.3.2.2	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$v^{max} \leq \hat{v} \leq \check{v}$	$v^l_\epsilon \geq v^{max}$	$\nearrow$	$\nearrow$				(ϵ, v^{max})
3.2.3.3	$[w^l_{max}, w^u_{max})$	$v^u_a \leq v^{max} \leq v^l_a$	$\hat{v} \geq \check{v}$		$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v^{max})
3.2.4.1.1	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\hat{v} \leq v^{max}$	$v^l_\epsilon \leq \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$		$(\hat{w}^l, \hat{v})$
3.2.4.1.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\hat{v} \leq v^{max}$	$\hat{v} \leq v^l_\epsilon \leq v^{max}$	$\nearrow$	$\nearrow$				(ϵ, v^l_ϵ)
3.2.4.1.3	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\hat{v} \leq v^{max}$	$v^l_\epsilon \geq v^{max}$	$\nearrow$					(ϵ, v^{max})
3.2.4.2.1	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$v^{max} \leq \hat{v} \leq \check{v}$	$v^l_\epsilon \leq v^{max}$	$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$		(w^l_{max}, v^{max})
3.2.4.2.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$v^{max} \leq \hat{v} \leq \check{v}$	$v^l_\epsilon \geq v^{max}$	$\nearrow$					(ϵ, v^{max})
3.2.4.3	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\hat{v} \geq \check{v}$		$\rightarrow$	$\searrow$	$\searrow$	$\searrow$		(w^u_{max}, v^{max})

where $\hat{w}^l = a - (d - (l-a)\hat{v})/v_c$.

In case (3.3):

$$
TC(w) = \begin{cases}
Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l-\epsilon) & \text{if } \epsilon \le w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\
Ad + (B + Cv_c^3)(a-w)^+ & \\
+(B + C\underline{v}^3)\frac{d-(a-w)^+ v_c}{\underline{v}} + D(l-\epsilon) & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \le w < \max\left\{\epsilon, \underline{w}^l\right\} \\
Ad + (B + Cv_c^3)(a-w)^+ & \\
+(B + C(v_w^l)^3)\frac{d-(a-w)^+ v_c}{v_w^l} + D(l-\epsilon) & \text{if } \max\left\{\epsilon, \underline{w}^l\right\} \le w < \max\left\{\epsilon, \overline{w}^l\right\} \\
Ad + (B + D + Cv_c^3)(a-w)^+ & \\
+(B + D + C\overline{v}^3)\frac{d-(a-w)^+ v_c}{\overline{v}} + Dw & \text{if } \max\left\{\epsilon, \overline{w}^l\right\} \le w < \max\left\{\epsilon, \overline{w}^u\right\} \\
Ad + (B + D + Cv_c^3)(a-w)^+ & \\
+(B + D + C(v_w^u)^3)\frac{d-(a-w)^+ v_c}{v_w^u} + Dw & \text{if } \max\left\{\epsilon, \overline{w}^u\right\} \le w \le w_{max}^u
\end{cases}
$$

where

$$
\overline{w}^l = \begin{cases}
a - \frac{d-(l-a)\overline{v}}{v_c} & \text{if } v_a^l \ge \overline{v} \\
l - \frac{d}{\overline{v}} & \text{otherwise.}
\end{cases}
$$

Table 3.12 gives the solution for all subcases. □

Table 3.12 Case 3.3

Case	$a \in$	Condition 1	Condition 2	Condition 3	①	②	③	④	⑤	⑥	Solution
3.3.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \underline{v}^l\right)$	$v_a^l \le \underline{v}$			$\rightarrow$	$\searrow$	$\rightarrow$	$\nearrow$	$\nearrow$	$\nearrow$	$(w, \underline{v})$ where $w \in [a, \underline{w}^l]$
3.3.2.1.1	$[\underline{w}^l, \overline{w}^l)$	$\underline{v} \le v_a^l \le \overline{v}$	$v_a^l \ge \hat{v}$	$v_\epsilon^l \le \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$	$\nearrow$	$(\hat{w}^l, \hat{v})$
3.3.2.1.2	$[\underline{w}^l, \overline{w}^l)$	$\underline{v} \le v_a^l \le \overline{v}$	$v_a^l \ge \hat{v}$	$v_\epsilon^l \ge \hat{v}$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$			(ϵ, v_ϵ^l)
3.3.2.2	$[\underline{w}^l, \overline{w}^l)$	$\underline{v} \le v_a^l \le \overline{v}$	$v_a^l \le \hat{v}$		$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	$\nearrow$	(a, v_a^l)
3.3.3.1.1	$[\overline{w}^l, \overline{w}^u)$	$v_a^u \le \overline{v} \le v_a^l$	$\hat{v} \le \overline{v}$	$v_\epsilon^l \le \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$	$\nearrow$	$(\hat{w}^l, \hat{v})$
3.3.3.1.2	$[\overline{w}^l, \overline{w}^u)$	$v_a^u \le \overline{v} \le v_a^l$	$\hat{v} \le \overline{v}$	$\hat{v} \le v_\epsilon^l \le \overline{v}$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$			(ϵ, v_ϵ^l)
3.3.3.1.3	$[\overline{w}^l, \overline{w}^u)$	$v_a^u \le \overline{v} \le v_a^l$	$\hat{v} \le \overline{v}$	$v_\epsilon^l \ge \overline{v}$	$\nearrow$	$\nearrow$	$\nearrow$				$(\epsilon, \overline{v})$
3.3.3.2	$[\overline{w}^l, \overline{w}^u)$	$v_a^u \le \overline{v} \le v_a^l$	$\hat{v} \ge \overline{v}$		$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	$\nearrow$	$(a, \overline{v})$
3.3.4.1.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^l \le \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$	$\nearrow$	$(\hat{w}^l, \hat{v})$
3.3.4.1.2	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$\hat{v} \le v_\epsilon^l \le \overline{v}$	$\nearrow$	$\nearrow$	$\nearrow$	$\nearrow$			(ϵ, v_ϵ^l)
3.3.4.1.3	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \le \overline{v} \le v_\epsilon^l$	$\nearrow$	$\nearrow$	$\nearrow$				$(\epsilon, \overline{v})$
3.3.4.1.4	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \le \overline{v}$	$\overline{v} \le v_\epsilon^u$	$\nearrow$	$\nearrow$					(ϵ, v_ϵ^u)
3.3.4.2.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\overline{v} \le \hat{v} \le v_a^u$	$v_\epsilon^u \le \hat{v}$	$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\smile$	$\nearrow$	$(\hat{w}^u, \hat{v})$
3.3.4.2.2	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\overline{v} \le \hat{v} \le v_a^u$	$v_\epsilon^u \ge \hat{v}$	$\nearrow$	$\nearrow$					(ϵ, v_ϵ^u)
3.3.4.3	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \le v_a^u \le v^{max}$	$\hat{v} \ge v_a^u$		$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v_a^u)
3.3.5.1.1	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^l \le \hat{v}$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$\nearrow$		$(\hat{w}^l, \hat{v})$
3.3.5.1.2	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$\hat{v} \le v_\epsilon^l \le \overline{v}$	$\nearrow$	$\nearrow$	$\nearrow$				(ϵ, v_ϵ^l)
3.3.5.1.3	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$v_\epsilon^u \le \overline{v} \le v_\epsilon^l$	$\nearrow$	$\nearrow$					$(\epsilon, \overline{v})$
3.3.5.1.4	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \le \overline{v}$	$\overline{v} \le v_\epsilon^u$	$\nearrow$						(ϵ, v_ϵ^u)
3.3.5.2.1	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\overline{v} \le \hat{v} \le v^{max}$	$v_\epsilon^u \le \hat{v}$	$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\smile$		$(\hat{w}^u, \hat{v})$
3.3.5.2.2	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\overline{v} \le \hat{v} \le v^{max}$	$v_\epsilon^u \ge \hat{v}$	$\nearrow$						(ϵ, v_ϵ^u)
3.3.5.3	$[w_{max}^u, \infty)$	$v_a^u \ge v^{max}$	$\hat{v} \ge v^{max}$		$\rightarrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$		(w_{max}^u, v^{max})

where $\hat{w}^u = a - (d - (u-a)\hat{v})/v_c$ and $\hat{w}^l = a - (d - (l-a)\hat{v})/v_c$.

Proposition 3.1 suggests that, when the driver is paid from the beginning of the planning horizon, there are four important free-flow speed values: $\overline{v}$, $\underline{v}$, $\hat{v}$ and $\check{v}$, which only depend on the values from Table 3.1. In particular, the first two values are defined as in Lemma 3.1, and the latter two are comparison parameters. The intuition is as follows. Delaying the departure of the driver has two effects: on the one hand, it may increase the driver cost as the driver is paid for a longer period

of time; on the other hand, it may reduce the time spent driving in congestion, allowing the driver to reach a higher average driving speed and spend less time on the road. The engine module component of the emissions cost is decreasing in the departure time, whereas the driver cost and speed module are increasing in it. As a result, the overall impact on the total cost depends on the trade-off between these costs. More specifically, when $v^{max} \leq \overline{v}$ ($v^{max} > \overline{v}$), the total cost function is initially decreasing in the *transient* region (where both effects are active) only if $\hat{v} \geq \check{v}$ ($\hat{v} \geq \overline{v}$). In this case, it may be beneficial to postpone the departure time past time ϵ because the drop in the engine module part of the emissions cost more than offsets the increase in driver cost and speed module.
Beside the speeds just described, the optimal solution also depends on other four free-flow speed values: v^l_ϵ, v^u_ϵ, v^l_a, and v^u_a, which only depend on the instance parameters, that is, l, u, d and a.

Proposition 3.2 *Consider a single-arc TDPRP instance. If the driver is paid from the departure time, the optimal solution depends mainly on the relative values of the eight speed levels:* v^{max}, $\underline{v}$, $\overline{v}$, $\tilde{v} = \left((f_c\lambda k N_e V + d_c + f_c\lambda\beta\gamma v_c^3)/3f_c\lambda\beta\gamma v_c\right)^{1/2}$, v^l_ϵ, v^l_a, v^u_ϵ *and* v^u_a *and is given in Table 3.23 in §3.B.*

PROOF: Let $A = f_c\lambda\gamma\alpha(\mu + f)$, $B = f_c\lambda k N_e V$ and $C = f_c\lambda\beta\gamma$, $D = d_c$. Note that $A, B, C, D \geq 0$. Further, let $T(w) = \min_{v_f\in[v^{min},v^{max}]} TC(w, v_f)$ such that $w + T(w, v_f) \leq u$. We consider three cases: (1) $l \leq u \leq a$, (2) $l < a < u$ and (3) $a \leq l < u$.

In case (1), we have

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - w) & \text{if } \epsilon \leq w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \leq w \leq u - \frac{d}{v_c}. \end{cases}$$

The first piece is decreasing in w and the second is constant in w. So any departure time in $\left[\max\left\{\epsilon, l - \frac{d}{v_c}\right\}, u\right]$ is optimal. We summarize this information in Table 3.13.

Table 3.13 Case 1

Case	①	②	Solution
1	$\searrow$	$\rightarrow$	(w, v_f) with $w \in \left[\max\left\{\epsilon, l - \frac{d}{v_c}\right\}, u - \frac{d}{v_c}\right]$

In case 2 we distinguish two subcases: (2.1) $v^{max} < \overline{v}$, (2.2) $v^{max} \geq \overline{v}$.

In case (2.1)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - w) & \text{if } \epsilon \leq w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + D + Cv_c^3)(a - w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d-(a-w)^+v_c}{v^{max}} & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w \leq w^u_{max}. \end{cases}$$

Table 3.14 gives the solution in all possible subcases.

Table 3.14 Case 2.1

Case	$a \in$	Condition 1	①	②	③	④	Solution
2.1.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, w^u_{max}\right)$	$v^u_a \leq v^{max}$	$\searrow$	$\rightarrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in [a, w^u_{max}]$
2.1.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\searrow$	$\rightarrow$	$\searrow$		(w^u_{max}, v^{max})

In case (2.2)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - w) & \text{if } \epsilon \leq w < \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \\ Ad + (B + D + Cv_c^3)\frac{d}{v_c} & \text{if } \max\left\{\epsilon, l - \frac{d}{v_c}\right\} \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + D + Cv_c^3)(a - w)^+ & \\ +(B + D + C\overline{v}^3)\frac{d-(a-w)^+ v_c}{\overline{v}} & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w < \max\{\epsilon, \overline{w}^u\} \\ Ad + (B + D + Cv_c^3)(a - w)^+ & \\ +(B + D + C(v^u_w)^3)\frac{d-(a-w)^+ v_c}{v^u_w} & \text{if } \max\{\epsilon, \overline{w}^u\} \leq w \leq w^u_{max}. \end{cases}$$

Table 3.15 gives the solution in all possible subcases.

Table 3.15 Case 2.2

Case	$a \in$	Condition 1	Condition 2	Condition 3	①	②	③	④	⑤	Solution
2.2.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \overline{w}^u\right)$	$v^u_a \leq \overline{v}$			$\searrow$	$\rightarrow$	$\searrow$	$\rightarrow$	$\nearrow$	$(w, \overline{v})$ where $w \in [a, \overline{w}^u)$
2.2.2.1.1	$[\overline{w}^u, w^u_{max})$	$\overline{v} \leq v^u_a \leq v^{max}$	$\tilde{v} \leq v^u_a$	$v^u_\epsilon \leq \tilde{v}$	$\searrow$	$\rightarrow$	$\searrow$	$\smile$	$\nearrow$	$(\tilde{w}^u, \tilde{v})$
2.2.2.1.2	$[\overline{w}^u, w^u_{max})$	$\overline{v} \leq v^u_a \leq v^{max}$	$\tilde{v} \leq v^u_a$	$v^u_\epsilon \geq \tilde{v}$	$\nearrow$	$\nearrow$				(ϵ, v^u_ϵ)
2.2.2.2	$[\overline{w}^u, w^u_{max})$	$\overline{v} \leq v^u_a \leq v^{max}$	$\tilde{v} \geq v^u_a$		$\searrow$	$\rightarrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v^u_a)
2.2.3.1.1	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v^u_\epsilon \leq \tilde{v}$	$\searrow$	$\rightarrow$	$\searrow$	$\smile$		$(\tilde{w}^u, \tilde{v})$
2.2.3.1.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v^u_\epsilon \geq \tilde{v}$	$\nearrow$					(ϵ, v^u_ϵ)
2.2.3.2	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\tilde{v} \geq v^{max}$		$\searrow$	$\rightarrow$	$\searrow$	$\searrow$		(w^u_{max}, v^{max})

where $\tilde{w}^u = a - (d - (u - a)\tilde{v})/v_c$.

In case 3 we distinguish three subcases: (3.1) $v^{max} < \underline{v}$, (3.2) $\underline{v} \leq v^{max} < \overline{v}$, (3.3) $v^{max} \geq \overline{v}$.

In case (3.1)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l - w) & \text{if } \epsilon \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + Cv_c^3)(a - w)^+ & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w \\ +(B + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} + D(l - w) & \text{and } w < \max\left\{\epsilon, w^l_{max}\right\} \\ Ad + (B + D + Cv_c^3)(a - w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} & \text{if } \max\left\{\epsilon, w^l_{max}\right\} \leq w \leq w^u_{max}. \end{cases}$$

Table 3.16 gives the solution in all possible subcases.

Table 3.16 Case 3.1

Case	$a \in$	Condition 1	①	②	③	④	Solution
3.1.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, w^l_{max}\right)$	$v^l_a \leq v^{max}$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in [w^l_{max}, w^u_{max}]$
3.1.2	$\left[w^l_{max}, w^u_{max}\right)$	$v^u_a \leq v^{max} \leq v^l_a$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in [a, w^u_{max}]$
3.1.3	$\left[w^l_{max}, w^u_{max}\right)$	$v^u_a \geq v^{max}$	$\searrow$	$\searrow$	$\searrow$		(w^u_{max}, v^{max}).

In case (3.2)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + D(l-w) & \text{if } \epsilon \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \\ +(B + C\underline{v}^3)\frac{d-(a-w)^+ v_c}{\underline{v}} + D(l-w) & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w < \max\left\{\epsilon, \underline{w}^l\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \\ +(B + C(v^l_w)^3)\frac{d-(a-w)^+ v_c}{v^l_w} + D(l-w) & \text{if } \max\left\{\epsilon, \underline{w}^l\right\} \leq w < \max\left\{\epsilon, w^l_{max}\right\} \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C(v^{max})^3)\frac{d-(a-w)^+ v_c}{v^{max}} & \text{if } \max\left\{\epsilon, w^l_{max}\right\} \leq w \leq w^u_{max}. \end{cases}$$

Table 3.17 gives the solution in all possible subcases.

Table 3.17 Case 3.2

Case	$a \in$	Condition 1	①	②	③	④	⑤	Solution
3.2.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \underline{w}^l\right)$	$v^l_a \leq \underline{v}$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in \left[w^l_{max}, w^u_{max}\right]$
3.2.2	$\left[\underline{w}^l, w^l_{max}\right)$	$\underline{v} \leq v^l_a \leq v^{max}$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in \left[w^l_{max}, w^u_{max}\right]$
3.2.3	$\left[w^l_{max}, w^u_{max}\right)$	$v^u_a \leq v^{max} \leq v^l_a$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	(w, v^{max}) where $w \in [a, w^u_{max}]$
3.2.4	$[w^u_{max}, \infty)$	$v^u_a \geq v^{max}$	$\searrow$	$\searrow$	$\searrow$	$\searrow$		(w^u_{max}, v^{max})

In case (3.3)

$$TC(w) = \begin{cases} Ad + (B + Cv_c^3)\frac{d}{v_c} + Dl & \text{if } \epsilon \leq w < \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \\ Ad + (B + Cv_c^3)(a-w)^+ & \\ +(B + C\underline{v}^3)\frac{d-(a-w)^+ v_c}{\underline{v}} + D(l-w) & \text{if } \max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\} \leq w < \left(\underline{w}^l\right)^+ \\ Ad + (B + Cv_c^3)(a-w)^+ & \\ +(B + C(v^l_w)^3)\frac{d-(a-w)^+ v_c}{v^l_w} + D(l-w) & \text{if } \left(\underline{w}^l\right)^+ \leq w < \left(\overline{w}^l\right)^+ \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C\overline{v}^3)\frac{d-(a-w)^+ v_c}{\overline{v}} & \text{if } \left(\overline{w}^l\right)^+ \leq w < (\overline{w}^u)^+ \\ Ad + (B + D + Cv_c^3)(a-w)^+ & \\ +(B + D + C(v^u_w)^3)\frac{d-(a-w)^+ v_c}{v^u_w} & \text{if } (\overline{w}^u)^+ \leq w \leq w^u_{max}. \end{cases}$$

Table 3.18 gives the solution in all possible subcases. □

When the driver is paid from the departure time, delaying departure does not lead to an increase in the driver cost. In fact it may lead to a decrease since waiting may mean less driving in congestion and therefore spending less time on the road.

Table 3.18 Case 3.3

Case	$a \in$	Condition 1	Condition 2	Condition 3	①	②	③	④	⑤	⑥	Solution
3.3.1	$\left[\max\left\{\epsilon, \left(a - \frac{d}{v_c}\right)\right\}, \underline{w}^l\right)$	$v_a^l \leq \underline{v}$			$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	$\nearrow$	$(w, \overline{v})$ where $w \in [\overline{w}^l, \overline{w}^u]$
3.3.2	$[\underline{w}^l, \overline{w}^l)$	$\underline{v} \leq v_a^l \leq \overline{v}$			$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	$\nearrow$	$(w, \overline{v})$ where $w \in [\overline{w}^l, \overline{w}^u]$
3.3.3	$[\overline{w}^l, \overline{w}^u)$	$v_a^u \leq \overline{v} \leq v_a^l$			$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\rightarrow$	$\nearrow$	$(w, \overline{v})$ where $w \in [a, \overline{w}^u]$
3.3.4.1.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \leq v_a^u \leq v^{max}$	$\tilde{v} \leq v_a^u$	$v_\epsilon^u \leq \tilde{v}$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\smile$	$\nearrow$	$(\tilde{w}^u, \tilde{v})$
3.3.4.1.2	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \leq v_a^u \leq v^{max}$	$\tilde{v} \leq v_a^u$	$v_\epsilon^u \geq \tilde{v}$	$\nearrow$	$\nearrow$					(ϵ, v_ϵ^u)
3.3.4.1	$[\overline{w}^u, w_{max}^u)$	$\overline{v} \leq v_a^u \leq v^{max}$	$\tilde{v} \geq v_a^u$		$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\nearrow$	(a, v_a^u)
3.3.5.1.1	$[w_{max}^u, \infty)$	$v_a^u \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v_\epsilon^u \leq \tilde{v}$	$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\smile$		$(\tilde{w}^u, \tilde{v})$
3.3.5.1.2	$[w_{max}^u, \infty)$	$v_a^u \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v_\epsilon^u \geq \tilde{v}$	$\nearrow$						(ϵ, v_ϵ^u)
3.3.5.1	$[w_{max}^u, \infty)$	$v_a^u \geq v^{max}$	$\tilde{v} \geq v^{max}$		$\searrow$	$\searrow$	$\searrow$	$\searrow$	$\searrow$		(w_{max}^u, v^{max})

where $\tilde{w}^u = a - (d - (u - a)\tilde{v})/v_c$.

In this case the trade-off is between the speed module of the emissions cost, which is increasing in the departure time, and the driver cost and engine module which are decreasing.

We make the following remarks about the optimal solutions under both driver wage policies.

Remark 3.1 Consider a single-arc TDPRP instance.

- If there is no time window, i.e. $l = 0$ and $u = \infty$, and the driver is paid from the beginning of the planning horizon, then one of the following two solutions is optimal: either leave the depot immediately ($w^* = \epsilon$), or wait until the end of the congestion period ($w^* = a$). In both cases the optimal speed is $\overline{v}$. Alternatively, when the driver is paid from the departure time, leaving the depot at the end of the congestion period ($w^* = a$) and driving at free-flow speed $\overline{v}$ is optimal.

- When the driver is paid from the beginning of the planning horizon, there always exists an optimal solution where the driver leaves at or before the end of the congestion period, i.e., at time $w^* \leq a$. However, when the driver is paid from the departure time, it may be optimal to leave the depot after the end of the congestion period, i.e., at time $w^* > a$.

- The optimal departure time when the driver is paid from the beginning of the planning horizon is at most equal to the optimal departure time when the driver is paid from the departure time. This is due to the fact that there is an extra incentive to delay departure when the driver is paid from the departure time, which is to reduce the driver cost.

- If there is no congestion period, the TDPRP reduces to the PRP. In this case, our results show that, when the driver is paid from the beginning of the planning horizon, there always exists an optimal solution where the driver leaves the depot immediately, i.e., $w^* = \epsilon$. However, this result is not true when the driver is paid from the departure time. In this case, even in the absence of congestion, it may be optimal to delay the departure of the vehicle in order to save on the driver cost, when leaving at time ϵ would lead to a pre-service waiting time at the customer node.

- The results of this section also apply to the case where emissions costs are ignored (i.e., if f_c is set to 0) so that the objective function reduces to minimizing only the driver cost, that is, Propositions 3.1 and 3.2 can be used to obtain an optimal solution (note that $\overline{v} = \check{v} = \tilde{v} = \infty$ in this case). When the driver is paid from the beginning of the planning horizon, it is always optimal for him to leave immediately and drive at speed v^{max}. However, when the driver is paid from the departure time, it may be optimal to wait at the depot.

The following proposition establishes the relationship between the optimal departure time and the time window $[l, u]$.

Proposition 3.3 *The (earliest) optimal departure time from the depot w^* is non-decreasing in l and u. The optimal free-flow speed v^* (whenever it is used) is non-increasing in l and u.*

PROOF: The result follows from a careful comparison of the cases listed in Table 3.22 in Proposition 3.1 and in Table 3.23 in Proposition 3.2. □

The following example illustrates how the optimal solution to the TDPRP varies with l and u.

Example 3.6 The parameters in Table 3.1 imply that $\underline{v} = 55.19$ km/h and $\overline{v} = 75.34$ km/h. Let $\epsilon = 0$, $d = 100$ km , $v_c = 19$ km/h, $v^{min} = 50$km/h, $v^{max} = 110$ km/h and $a = 10000$ seconds. This implies that $\hat{v} = 77.58$ km/h and $\tilde{v} = 122.99$ km/h. Table 3.19 shows the optimal solution as a function of the lower (l) and upper (u) time windows, given in seconds.

Table 3.19 Optimal solution $S = (w^*, v_f^*)$ as a function of lower and upper time window

		Driver paid from the beginning of the planning horizon			Driver paid from departure time		
l	u	w^*	v_f^*	Arrival Time	w^*	v_f^*	Arrival Time
7500	11545	0	110 (v_m)	11545 (u)	0	110 (v_m)	11545 (u)
7500	12000	0	85 (v_ϵ^u)	12000 (u)	2631.58 ($< a$)	110 (v_m)	12000 (u)
7500	13000	3301.98 ($< a$)	77.58 ($\hat{v}$)	13000 (u)	8421.05 ($< a$)	110 (v^{max})	13000 (u)
7500	14700	10000 (a)	76.60 (v_a^u)	14700 (u)	10000 (a)	76.60 (v_u^a)	14700 (u)
7500	70000	10000 (a)	75.34 ($\overline{v}$)	14778.20 ($\in (l, u)$)	10000 (a)	75.34 ($\overline{v}$)	14778.20 ($\in (l, u)$)
15000	70000	10000 (a)	72 (v_a^l)	15000 (l)	10221.79 ($> a$)	75.34 ($\overline{v}$)	15000 (l)
25000	70000	10000 (a)	55.19 ($\underline{v}$)	16523 ($< l$)	20221.79 ($> a$)	75.34 ($\overline{v}$)	25000 (l)

We see that for low values of l and u, it is optimal for the driver to leave the depot immediately and arrive at the customer node exactly at time u. As u increases, it becomes optimal to wait at the depot and eventually arrive between l and u. Then as l is increased, the optimal arrival time becomes exactly l and then possibly (when the driver is paid from the beginning of the planning horizon) a value less than l, meaning that there is a pre-service waiting time.

Based on the properties of single-arc TDPRP instance we derive the following results which also apply to the general case.

Lemma 3.2 *Given a TDPRP instance,*
(i) it is never optimal for drivers to drive at a free-flow speed lower than $\underline{v}$*;*
(ii) if drivers are paid from their departure time, it is never optimal for them to drive on the first arc of a route at a free-flow speed lower than $\min\{\overline{v}, v^{max}\}$*.*

Proof: Proof of part (i)
The proof is by contradiction.
Suppose that there exists an optimal solution (denoted by S^*) where the speed on one arc is lower than $\underline{v}$. Without loss of generality, suppose that this arc belongs to the route $(0, \dots, n+1)$, where $n+1$ is a copy of the depot. Let w_i^* denote the optimal departure time from node i and let v_i^* denote the optimal speed on arc $(i, i+1)$. So there exists $k \in \{0, \dots, n\}$ such that $v_k^* < \underline{v}$.

The total cost associated with this route is

$$\sum_{i=0}^{n} f_c F_i(w_i^*, v_i^*) + d_c W(w_0^*, \dots, w_n^*, v_0^*, \dots, v_n^*),$$

where F_i denotes the emissions cost on arc $(i, i+1)$ and W is the total time the driver is paid for.

We construct an alternative solution (denoted by S') as follows: let $w_i' = w_i^*$ for $i = 0, \dots, n$, $v_i' = v_i^*$ for $i = 0, \dots, k-1, k+1, \dots, n$ and $v_k' = \underline{v}$. In other words, we increase the speed on arc $(k, k+1)$ to $\underline{v}$ and we keep the same departure time from node $k+1$ (unless $k = n$) by adding some extra waiting time. The resulting solution is feasible since the arrival time at each node is at most equal to that in the optimal solution. Compared to S^*, in S' the total time the driver is paid for (W) can only decrease (it decreases if $k = n$, otherwise it remains the same). Whereas the emissions cost (F_i) is the same on every arc except on arc $(k, k+1)$, where it decreases since $\underline{v}$ is the speed that minimizes the emissions cost for a given departure time in a one-arc problem as shown in §3.5. Therefore, the alternative solution S' yields a total cost lower that the optimal solution S^* and this leads to a contradiction.

Proof of part (ii) The proof is by contradiction.
Suppose that there exists an optimal solution (denoted by S^*) where the speed on the first arc of a route is lower than $\min\{\overline{v}, v_{max}\}$. Without loss of generality, suppose that this arc belongs to the route $(0,\ldots,n+1)$, where $n+1$ is a copy of the depot. Let w_i^* denote the optimal departure time from node i and let v_i^* denote the optimal speed on arc $(i, i+1)$. So we have $v_0^* \leq \min\{\overline{v}, v_{max}\}$.

The total cost associated with this route is

$$\sum_{i=0}^{n} f_c F_i(w_i^*, v_i^*) + d_c W(w_0^*, \ldots, w_n^*, v_0^*, \ldots, v_n^*),$$

where F_i denotes the emissions cost on arc $(i, i+1)$ and W is the total time the driver is paid for. This cost function can be rewritten as

$$\sum_{i=1}^{n} f_c F_i(w_i^*, v_i^*) + d_c W_{1,\ldots,n}(w_1^*, \ldots, w_n^*, v_1^*, \ldots, v_n^*) + f_c F_0(w_0^*, v_0^*) + \\ + d_c W_0(w_0^*, v_0^*) \tag{3.25}$$

where $W_{1,\ldots,n}$ is the time spent from the arrival at node 1 until the return to the depot and W_0 is the time spent from the departure from the depot to the arrival at node 1. Note that the last two terms in (3.25) correspond to the total cost function of a one-arc TDPRP when the driver is paid from his departure time.
We construct an alternative solution (denoted by S') as follows: let $w_i' = w_i^*$ for $i = 1,\ldots,n$, $v_i' = v_i^*$ for $i = 1,\ldots,n$, $v_0' = \min\{\overline{v}, v_{max}\}$ and $w_0' > w_0^*$ such that the arrival time at node 1 is the same in S' as in S^*. The departure times and free-flow speeds on arcs $(i, i+1)$ where $i = 1,\ldots,n$ remain unchanged and therefore the resulting solution is feasible. For the same reasons, in both solutions S^* and S' the first two terms of the 3.25 remain the same. Whereas, as results from the proof of Theorem 3.2, the last two terms 3.25 are lower in S' compared to S^*. Hence, we have a contradiction. □

These results will be useful to improve the efficiency of the MIP formulation, as discussed in the following section.

3.6 Computational results

This section presents the results of computational experiments using the integer linear programming formulation of the TDPRP presented in Secion 3.4. All tests were carried out using three sets of instances from the PRPLIB (`http://www.apollo.management.soton.ac.uk/prplib.htm`), with respectively 10, 15 and 20 nodes as described by Demir et al. (2012)). All experiments were conducted by using CPLEX 12.1 on a server with 2.93 GHz and 1.1 Gb RAM. The nodes in these instances represent randomly selected cities from the United Kingdom, with real distances. The time windows and service times, however, are randomly generated.

We set CPLEX to run sequentially in deterministic mode in a single thread. A common time-limit of three hours was imposed on all instances. To improve the efficiency of the formulation, we have used preprocessing to reduce the input data space by using the results of Lemma 3.2. More specifically, we downsize the set of free-flow speed levels R by setting $v^1 = \max\{v^{min}, \underline{v}\}$. We also include the values of the three speed levels $\overline{v}$, $\hat{v}$ and $\tilde{v}$ in the set of free-flow speed levels R, whenever these do not exceed the upper speed limit v^{max}. Finally, we supplement the formulation with two-node subtour breaking constraints $x_{ij} + x_{ji} \leq 1, \forall i, j \in N_0, i \neq j$, as was also done by Bektaş and Laporte (2011).

3.6.1 Performance on PRP instances

This section compares the performance of the proposed formulation for the TDPRP with that of Bektaş and Laporte (2011) reported in §3.A, for cases where there is no congestion. Table 3.24 in §3.C presents the results of this experiment using 10-node instances. The first two columns of the table are self-explanatory, whereas the columns PRP and TDPRP present the total cost produced by the respective formulations and t(PRP) and t(TDPRP) present the associated computational times (in seconds) required to solve each instance to optimality. Compared with the mathematical formulation proposed by Bektaş and Laporte (2011), the TDPRP formulation is superior in terms of the computational time required to reach optimality. The average solution time with the new formulation is indeed significantly reduced from 508.47 to 5.52 seconds. The proposed model also can solve some larger PRP instances to optimality, in particular the 15- and 20-node instances, as shown in §3.6.2. The Bektaş and Laporte (2011) formulation could not handle such sizes because of the computational time requirements. One possible explanation for our formulation to be faster, despite being more general, is that it does not include any big-M parameters. Bektaş and Laporte (2011) use such a parameter both in the time window constraints and in the calculation of the total travel time.

3.6.2 Importance of modeling traffic congestion and impact of driver wage policy

In this section, we compare the results of cases with and without congestion, as we did in §3.3, using 10-, 15- and 20-node PRP instances. More specifically, by using the integer linear programming formulation described in §3.4, we compute a *time-dependent* optimal solution S_D. Using the same formulation and fixing the congestion period to zero, we compute a *time-independent* optimal solution

S_N. We note that solving the problem by means of a *time-independent* approach may generate multiple optimal solutions which yield different total costs under a congestion scenario, in which case we select the solution with the minimum waiting time at the depot. For every instance, we assume the same two-level speed profile as defined in §3.2.1, and we consider both driver wage policies. The congestion speed v_c is set to 10 km/h and we consider two values for the length of the congestion period: 3600 and 7200 seconds. A summary of the results is provided in Tables 3.20 and 3.21 (the full results over 60 instances are reported in Tables 3.25–3.30 in §3.C). These tables report, for each set of instances the percentage of infeasible solutions S_D and S_N, the average computational time (denoted by $t(S_N)$ and $t(S_D)$) and the average saving of using a *time-dependent* formulation. The latter is calculated as Saving % $= 100(TC(S_N) - TC(S_D))/TC(S_N)$, representing the percentage decrease in costs which results from incorporating traffic congestion into planning vehicles routes and schedules.

Table 3.20 Summarized results for three sets of instances with an initial congestion period of 3600 seconds

Instances	Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
	Infeasible S_N %	Infeasible S_D %	$t(S_N)$ s	$t(S_D)$ s	Saving %	Infeasible S_N %	Infeasible S_D %	$t(S_N)$ s	$t(S_D)$ s	Saving %
UK_10	30	0	3.663	4.981	3.206	30	0	3.136	4.561	6.330
UK_15	55	5	976.610	467.797	3.478	45	5	1148.129	668.824	5.705
UK_20†	19	0	1527.273	1119.881	2.937	24	0	2179.146	1003.909	5.736

†Results calculated only on the instances solved to optimality.

Table 3.21 Summarized results for three sets of instances with an initial congestion period of 7200 seconds

Instances	Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
	Infeasible S_N %	Infeasible S_D %	$t(S_N)$ s	$t(S_D)$ s	Saving %	Infeasible S_N %	Infeasible S_D %	$t(S_N)$ s	$t(S_D)$ s	Saving %
UK_10	50	0	3.663	10.870	4.942	50	0	3.136	8.514	15.276
UK_15	80	10	976.610	463.724	5.055	85	10	1148.129	714.044	14.986
UK_20†	80	0	1527.273	3388.063	5.310	88	0	2179.146	3628.597	14.910

†Results calculated only on the instances solved to optimality.

Tables 3.20 and 3.21 show that in the presence of traffic congestion, using a *time-dependent* formulation significantly decreases the percentage of infeasible solutions. Furthermore the results also suggest that if both solutions are feasible, the *time-dependent* one can yield considerable cost savings over the *time-independent* one. The potential cost reduction increases proportionally to the length of the congestion period and can more than double when the driver is paid from the departure time. These implications support the assertions made in §3.3 by means of simple examples.

3.7 Conclusions

We have introduced and analyzed the time-dependent vehicle routing problem with time windows, which considers vehicles traveling under two subsequent periods of congestion and free-flow, respectively, and explicitly accounts for vehicle emissions which increases sharply when vehicles travel at slow speed. The modeling approach adopted in this chapter yields solution with reduced greenhouse gas emissions. We emphasize that our results also hold for the time-dependent VRP even if emissions are not considered in the objective function.

We have provided an integer linear programming formulation, which is also valid for the special case of the problem where there is no congestion (e.g., as in the PRP introduced by Bektaş and Laporte (2011). We have presented several examples that motivate idle waiting time, either pre- or post-service, at customer nodes or at the depot, in order to minimize a total cost function incorporating emissions and driver wages. We have derived a complete characterization of the optimal solution for a single-arc version of the TDPRP, identifying conditions under which it is optimal to wait before departure, and the associated amount of time. The characterization prescribes optimal speed levels under various conditions associated with time windows, the length of the congestion period and the speed limits. The analytical results derived in this chapter were used to strengthen the computational performance of the mathematical formulation. Computational results have confirmed that the proposed formulation computationally outperforms the formulation recently proposed for the PRP. Moreover, the analytical expressions for optimal speeds can easily be used as a "rule-of-thumb" for the design of vehicle routes under congestion.

3.A PRP formulation

In this section we present the PRP formulation by Bektaş and Laporte (2011). The model uses the following decision variables:

- x_{ij} binary variable equal to 1 if a vehicle traverses arc $(i,j) \in A$, 0 otherwise,
- z_{ij}^r binary variable equal to 1 if a vehicle traverses arc $(i,j) \in A$ with the free-flow speed v_r with $r \in R$, 0 otherwise,
- f_{ij} load carried on arc (i,j),
- s_i total time spent on a route that has node $i \in N_0$ as last visited before returning to the depot,
- φ_i time at which service at node $i \in N_0$ starts.

$$\text{Minimize} \quad \sum_{(i,j)\in A} \sum_{r\in R} \sum_{m=1}^{3} f_c \lambda k N_e V d_{ij} z_{ij}^r / v^r \tag{3.26}$$

$$+ \sum_{(i,j)\in A} \sum_{r\in R} \sum_{m=1,3} f_c \lambda \gamma \beta d_{ij} (v^r)^2 z_{ij}^r \tag{3.27}$$

$$+ \sum_{(i,j)\in A} f_c \lambda \gamma \alpha_{ij} d_{ij} (\mu x_{ij} + f_{ij}) \tag{3.28}$$

$$+ \sum_{i\in N_0} d_c s_i \tag{3.29}$$

subject to

$$\sum_{j\in N} x_{0j} = K \tag{3.30}$$

$$\sum_{i\in N} x_{ij} = 1 \qquad \forall j \in N_0 \tag{3.31}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad \forall i \in N_0 \tag{3.32}$$

$$\sum_{j\in N} f_{ji} - \sum_{j\in N} f_{ij} = q_i \qquad \forall i \in N_0 \tag{3.33}$$

$$q_j x_{ij} \le f_{ij} \le x_{ij}(Q - q_i) \qquad \forall (i,j) \in A \tag{3.34}$$

$$\varphi_i - \varphi_j + h_i + \sum_{r\in R} (d_{ij}/v^r) z_{ij}^r \le M_{ij}(1 - x_{ij}) \qquad \forall j \in N, j \in N_0, i \ne j \tag{3.35}$$

$$l_i \le \varphi_i \le u_i \qquad \forall i \in N_0 \tag{3.36}$$

$$\varphi_j + h_j - s_j + \sum_{r\in R} (d_{j0}/v^r) z_{ij}^r \le L(1 - x_{ij}) \qquad \forall j \in N_0 \tag{3.37}$$

$$\sum_{m=1}^{3}\sum_{r\in R} z_{ij}^{mr} = x_{ij} \qquad \forall (i,j) \in A \tag{3.38}$$

$$z_{ij}^{r} \in \{0,1\} \qquad \forall (i,j) \in A, r \in R \tag{3.39}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{3.40}$$

$$f_{ij} \geq 0 \qquad \forall (i,j) \in A. \tag{3.41}$$

The first three parts of the objective function represent the cost of emissions: (3.26) computes the cost induced by the *engine module*, the term (3.27) computes the cost induced by the *speed module*, and (3.28) computes the cost induced by the *weight module*. Finally, the last term (3.29) measures the total driver wage when the driver is paid from the beginning of the planning horizon. Constraint (3.30) indicates that exactly K vehicles depart from the depot. Constraints (3.31) and (3.32) guarantee that each customer is visited exactly once. Constraints (3.33) and (3.34) model the flow on each arc and ensure that vehicle capacities are respected. The time windows restrictions at customer nodes are imposed by constraints (3.35) and (3.36). In particular, as in Cordeau et al. (2007a), constraints (3.35) are obtained through a linearization of a set of non-linear inequalities and $M_{ij} = \max\{0, u_i + s_i + d_{ij}/v^{min} - l_j\}$. Constraint (3.37) computes the time at which the vehicle returns to the depot. The relationship between speed and arc-traversal variables is expressed by constraint (3.21). Finally, constraints (3.39)–(3.41) enforce the integrality and nonnegativity restrictions on the variables.

3.B Optimal solution tables

Table 3.22 Optimal solution when driver is paid from the beginning of the planning horizon.

Condition 1	Condition 2	Condition 3	Condition 4	Condition 5	Solution
$l \leq u \leq a$					(w, v_f) where $w \in \left[\epsilon, \max\left\{\epsilon, \left(l - \frac{d}{v_c}\right)\right\}\right]$
$l < a < u$	$v^{max} \leq \overline{v}$	$v_a^u \leq v^{max}$	$\hat{v} \geq \check{v}$		(a, v^{max}) or (ϵ, v^{max})
			$\hat{v} \leq \check{v}$		(ϵ, v^{max})
		$v_a^u \geq v^{max}$	$\hat{v} \geq \check{v}$		(w^u_{max}, v^{max}) or (ϵ, v^{max})
			$\hat{v} \leq \check{v}$		(ϵ, v^{max})
	$v^{max} \geq \overline{v}$	$v_a^u \leq \overline{v}$	$\hat{v} \geq \overline{v}$		$(a, \overline{v})$ or $(\epsilon, \overline{v})$
			$\hat{v} \leq \overline{v}$		$(\epsilon, \overline{v})$
		$\overline{v} \leq v_a^u \leq v^{max}$	$\hat{v} \leq \overline{v}$	$v_\epsilon^u \leq \overline{v}$	$(\epsilon, \overline{v})$
				$v_\epsilon^u \geq \overline{v}$	(ϵ, v_ϵ^u)
			$\overline{v} \leq \hat{v} \leq v_a^u$	$v_\epsilon^u \leq \hat{v}$	$(\hat{w}^u, \hat{v})$ or $(\epsilon, \overline{v})$
				$v_\epsilon^u \geq \hat{v}$	(ϵ, v_ϵ^u)
			$\hat{v} \geq v_a^u$		(a, v_u^a) or $(\epsilon, \overline{v})$
		$v_a^u \geq v^{max}$	$\hat{v} \leq \overline{v}$	$v_\epsilon^u \leq \overline{v}$	$(\epsilon, \overline{v})$
				$v_\epsilon^u \geq \overline{v}$	(ϵ, v_ϵ^u)
			$\overline{v} \leq \hat{v} \leq v^{max}$	$v_\epsilon^u \leq \hat{v}$	$(\hat{w}^u, \hat{v})$ or $(\epsilon, \overline{v})$
				$v_\epsilon^u \geq \hat{v}$	(ϵ, v_ϵ^u)
			$\hat{v} \geq v^{max}$		(w^u_{max}, v^{max}) or $(\epsilon, \overline{v})$
$a < l < u$	$v^{max} \leq \underline{v}$	$v_a^l \leq v^{max}$			(w, v^{max}) where $w \in [a, w^l_{max}]$
		$v_a^u \leq v^{max} \leq v_a^l$	$\hat{v} \geq \check{v}$		(a, v^{max})
			$\hat{v} \leq \check{v}$		(w^l_{max}, v^{max})
		$v_a^u \geq v^{max}$	$\hat{v} \geq \check{v}$		(w^u_{max}, v^{max})
			$\hat{v} \leq \check{v}$		(w^l_{max}, v^{max})
	$\underline{v} \leq v^{max} \leq \overline{v}$	$v_a^l \leq \underline{v}$			$(w, \underline{v})$ where $w \in \left[a, \underline{w}^l\right]$
		$\underline{v} \leq v_a^l \leq v^{max}$	$v_a^l \geq \hat{v}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$v_\epsilon^l \geq \hat{v}$	(ϵ, v_ϵ^l)
			$v_a^l \leq \hat{v}$		(a, v_a^l)
		$v_a^u \leq v^{max} \leq v_a^l$	$\hat{v} \leq v^{max}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$\hat{v} \leq v_\epsilon^l \leq v^{max}$	(ϵ, v_ϵ^l)
				$v_\epsilon^l \geq v^{max}$	(ϵ, v^{max})
			$v^{max} \leq \hat{v} \leq \check{v}$	$v_\epsilon^l \leq v^{max}$	(w^l_{max}, v^{max})
				$v_\epsilon^l \geq v^{max}$	(ϵ, v^{max})
			$\hat{v} \geq \check{v}$		(a, v^{max})
		$v_a^u \geq v^{max}$	$\hat{v} \leq v^{max}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$\hat{v} \leq v_\epsilon^l \leq v^{max}$	(ϵ, v_ϵ^l)
				$v_\epsilon^l \geq v^{max}$	(ϵ, v^{max})
			$v^{max} \leq \hat{v} \leq \check{v}$	$v_\epsilon^l \leq v^{max}$	(w^l_{max}, v^{max})
				$v_\epsilon^l \geq v^{max}$	(ϵ, v^{max})
			$\hat{v} \geq \check{v}$		(w^u_{max}, v^{max})
	$v^{max} \geq \overline{v}$	$v_a^l \leq \underline{v}$			$(w, \underline{v})$ where $w \in \left[a, \underline{w}^l\right]$
		$\underline{v} \leq v_a^l \leq \overline{v}$	$v_a^l \geq \hat{v}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$v_\epsilon^l \geq \hat{v}$	(ϵ, v_ϵ^l)
			$v_a^l \leq \hat{v}$		(a, v_a^l)
		$v_a^u \leq \overline{v} \leq v_a^l$	$\hat{v} \leq \overline{v}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$\hat{v} \leq v_\epsilon^l \leq \overline{v}$	(ϵ, v_ϵ^l)
				$v_\epsilon^l \geq \overline{v}$	$(\epsilon, \overline{v})$
			$\hat{v} \geq \overline{v}$		$(a, \overline{v})$
		$\overline{v} \leq v_a^u \leq v^{max}$	$\hat{v} \leq \overline{v}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$\hat{v} \leq v_\epsilon^l \leq \overline{v}$	(ϵ, v_ϵ^l)
				$v_\epsilon^u \leq \overline{v} \leq v_\epsilon^l$	$(\epsilon, \overline{v})$
				$\overline{v} \leq v_\epsilon^u$	(ϵ, v_ϵ^u)
			$\overline{v} \leq \hat{v} \leq v_a^u$	$v_\epsilon^u \leq \hat{v}$	$(\hat{w}^u, \hat{v})$
				$v_\epsilon^u \geq \hat{v}$	(ϵ, v_ϵ^u)
			$\hat{v} \geq v_a^u$		(a, v_a^u)
		$v_a^u \geq v^{max}$	$\hat{v} \leq \overline{v}$	$v_\epsilon^l \leq \hat{v}$	$(\hat{w}^l, \hat{v})$
				$\hat{v} \leq v_\epsilon^l \leq \overline{v}$	(ϵ, v_ϵ^l)
				$v_\epsilon^u \leq \overline{v} \leq v_\epsilon^l$	$(\epsilon, \overline{v})$
				$\overline{v} \leq v_\epsilon^u$	(ϵ, v_ϵ^u)
			$\overline{v} \leq \hat{v} \leq v^{max}$	$v_\epsilon^u \leq \hat{v}$	$(\hat{w}^u, \hat{v})$
				$v_\epsilon^u \geq \hat{v}$	(ϵ, v_ϵ^u)
			$\hat{v} \geq v^{max}$		(w^u_{max}, v^{max})

where $\hat{w}^l = a - (d - (l-a)\hat{v})/v_c$, $\hat{w}^u = a - (d - (u-a)\hat{v})/v_c$ and $\underline{w}^l = l - d/\underline{v}$.

Table 3.23 Optimal solution when driver is paid from departure time

Condition 1	Condition 2	Condition 3	Condition 4	Condition 5	Solution
$l \leq u \leq a$					(w, v_f) where $w \in \left[\max\left\{\epsilon, \left(l - \frac{d}{v_c}\right)\right\}, u - \frac{d}{v_c}\right]$
$l < a < u$	$v^{max} \leq \overline{v}$	$v_a^u \leq v^{max}$			(w, v^{max}) where $w \in [a, w_{max}^u]$
		$v_a^u \geq v^{max}$			(w_{max}^u, v^{max})
	$v^{max} \geq \overline{v}$	$v_a^u \leq \overline{v}$			$(w, \overline{v})$ where $w \in [a, \overline{w}^u]$
		$\overline{v} \leq v_a^u \leq v^{max}$	$\tilde{v} \leq v_a^u$	$v_\epsilon^u \leq \tilde{v}$	$(\tilde{w}^u, \tilde{v})$
				$v_\epsilon^u \geq \tilde{v}$	(ϵ, v_ϵ^u)
			$\tilde{v} \geq v_a^u$		(a, v_a^u)
		$v_a^u \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v_\epsilon^u \leq \tilde{v}$	$(\tilde{w}^u, \tilde{v})$
				$v_\epsilon^u \geq \tilde{v}$	(ϵ, v_ϵ^u)
			$\tilde{v} \geq v^{max}$		(w_{max}^u, v^{max})
$a < l < u$	$v^{max} \leq \underline{v}$	$v_a^l \leq v^{max}$			(w, v^{max}) where $w \in [w_{max}^l, w_{max}^u]$
		$v_a^u \leq v^{max} \leq v_a^l$			(w, v^{max}) where $w \in [a, w_{max}^u]$
		$v_a^u \geq v^{max}$			(w_{max}^u, v^{max})
	$\underline{v} \leq v^{max} \leq \overline{v}$	$v_a^l \leq \underline{v}$			(w, v^{max}) where $w \in \left[w_{max}^l, w_{max}^u\right]$
		$\underline{v} \leq v_a^l \leq v^{max}$			(w, v^{max}) where $w \in \left[w_{max}^l, w_{max}^u\right]$
		$v_a^u \leq v^{max} \leq v_a^l$			(w, v^{max}) where $w \in [a, w_{max}^u]$
		$v_a^u \geq v^{max}$			(w_{max}^u, v^{max})
	$v^{max} \geq \overline{v}$	$v_a^l \leq \underline{v}$			$(w, \overline{v})$ where $w \in [\overline{w}^l, \overline{w}^u]$
		$\underline{v} \leq v_a^l \leq \overline{v}$			$(w, \overline{v})$ where $w \in [\overline{w}^l, \overline{w}^u]$
		$v_a^u \leq \overline{v} \leq v_a^l$			$(w, \overline{v})$ where $w \in [a, \overline{w}^u]$
		$\overline{v} \leq v_a^u \leq v^{max}$	$\tilde{v} \leq v_a^u$	$v_\epsilon^u \leq \tilde{v}$	$(\tilde{w}^u, \tilde{v})$
				$v_\epsilon^u \geq \tilde{v}$	(ϵ, v_ϵ^u)
			$\tilde{v} \geq v_a^u$		(a, v_a^u)
		$v_a^u \geq v^{max}$	$\tilde{v} \leq v^{max}$	$v_\epsilon^u \leq \tilde{v}$	$(\tilde{w}^u, \tilde{v})$
				$v_\epsilon^u \geq \tilde{v}$	(ϵ, v_ϵ^u)
			$\tilde{v} \geq v^{max}$		(w_{max}^u, v^{max})

where $\tilde{w}^u = a - (d - (u - a)\tilde{v})/v_c$ and $\overline{w}^u = u - d/\overline{v}$.

3.C Computational results

3.C.1 Results on PRP instances

The PRP results in columns 2 and 3 are taken from Demir et al. (2012). The reason behind the slight discrepancy between the values in columns 2 and 4 is due to numerical approximation.

3.C.2 Results on TDPRP instances

Each table reports the two cases: (i) driver paid from the beginning of the planning horizon, (ii) driver paid from the departure time. In both cases the tables display, for each instance, the cost values of the S_D and S_N solutions (denoted by $TC(S_N)$ and $TC(S_D)$) and the CPU times (in seconds) required to construct these solutions (denoted by $t(S_N)$ and $t(S_D)$). Under the last column are reposted the cost savings of incorporating traffic congestion when planning the vehicles' routes and schedules.

Table 3.24 Comparison of PRP versus TDPRP formulations with respect to computational time

Instance	PRP	t(PRP)	TDPRP	t(TDPRP)
	£	s	£	s
UK10_01	170.66	163.40	170.66	10.71
UK10_02	204.87	113.90	204.88	3.73
UK10_03	200.33	926.00	200.34	3.36
UK10_04	189.94	396.50	189.95	5.00
UK10_05	175.61	1253.70	175.62	4.93
UK10_06	214.56	347.50	214.53	3.43
UK10_07	190.14	191.00	190.15	5.06
UK10_08	222.16	139.80	222.17	2.23
UK10_09	174.53	54.00	174.54	4.64
UK10_10	189.83	76.00	189.84	2.83
UK10_11	262.07	50.50	262.08	4.40
UK10_12	183.18	1978.70	183.19	14.71
UK10_13	195.97	1235.10	195.97	2.94
UK10_14	163.17	84.10	163.18	2.77
UK10_15	127.15	433.30	127.16	6.25
UK10_16	186.63	680.80	186.63	7.03
UK10_17	159.07	27.00	159.08	3.22
UK10_18	162.09	522.10	162.09	4.19
UK10_19	169.46	130.50	169.46	1.52
UK10_20	168.8	1365.50	168.81	17.44
Average		508.47		5.52

Table 3.25 Computational results for 10-node instances with initial congestion period of 3600 seconds

Instance	# of vehicles	Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
		$TC(S_N)$	$t(S_N)$	$TC(S_D)$	$t(S_D)$	Saving	$TC(S_N)$	$t(S_N)$	$TC(S_D)$	$t(S_D)$	Saving
		£	s	£	s	%	£	s	£	s	%
UK10_01	2	inf.	4.62	183.98	6.01	-	177.97	3.99	168.14	6.05	5.52
UK10_02	2	225.10	3.09	218.90	3.63	2.75	220.26	1.81	203.06	6.79	7.81
UK10_03	2	219.33	12.88	213.34	8.76	2.73	210.54	8.33	197.50	2.94	6.19
UK10_04	2	209.97	2.83	202.17	2.20	3.71	187.18	1.36	185.88	2.65	0.69
UK10_05	2	195.80	3.99	188.07	3.95	3.95	185.77	1.24	172.23	3.27	7.29
UK10_06	2	inf.	2.55	229.13	3.55	-	inf.	2.21	213.29	5.86	-
UK10_07	2	210.37	1.54	205.18	3.31	2.47	203.98	1.81	189.34	3.35	7.18
UK10_08	2	242.26	1.83	237.17	2.46	2.1	242.26	1.09	221.33	2.11	8.64
UK10_09	2	194.82	2.59	189.73	2.97	2.61	194.82	2.86	173.89	3.23	10.74
UK10_10	2	210.03	1.91	204.89	2.56	2.44	209.59	2.26	189.05	2.75	9.8
UK10_11	2	inf.	2.71	277.12	2.57	-	inf.	1.92	261.28	2.71	-
UK10_12	2	198.41	5.32	193.65	4.20	2.4	181.64	2.52	177.81	3.88	2.11
UK10_13	2	216.19	1.79	208.37	2.08	3.61	205.72	1.18	192.53	2.04	6.41
UK10_14	2	inf.	1.54	179.84	17.40	-	inf.	1.20	164.72	6.40	-
UK10_15	2	141.13	3.06	135.46	4.01	4.02	123.22	2.73	119.62	4.39	2.92
UK10_16	2	206.25	4.97	198.86	4.20	3.58	194.80	5.03	183.02	5.60	6.05
UK10_17	2	inf.	2.17	171.60	2.51	-	inf.	1.34	155.76	2.81	-
UK10_18	2	182.37	3.78	173.96	6.04	4.61	inf.	2.90	158.00	4.42	-
UK10_19	2	inf.	1.74	181.28	5.38	-	inf.	2.29	165.44	5.61	-
UK10_20	2	189.06	8.37	181.68	11.84	3.9	178.83	14.64	165.84	14.38	7.27

Table 3.26 Computational results for 10-node instances with initial congestion period of 7200 seconds

Instance	# of vehicles	Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
		$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %
UK10_01	2	inf.	4.62	201.76	22.61	-	inf.	3.99	170.08	20.35	-
UK10_02	2	inf.	3.09	241.31	12.88	-	inf.	1.81	210.63	23.02	-
UK10_03	2	240.03	12.88	229.69	30.30	4.31	231.47	8.33	198.01	20.99	14.46
UK10_04	2	230.84	2.83	217.56	4.89	5.75	206.4	1.36	185.88	3.54	9.94
UK10_05	2	216.68	3.99	203.91	4.32	5.89	206.71	1.24	172.23	4.36	16.68
UK10_06	2	inf.	2.55	249.98	6.61	-	inf.	2.21	218.30	12.38	-
UK10_07	2	231.31	1.54	221.31	5.90	4.32	inf.	1.81	189.63	3.74	-
UK10_08	2	263.19	1.83	253.01	2.43	3.87	263.19	1.09	221.33	1.96	15.91
UK10_09	2	215.75	2.59	205.57	4.71	4.72	215.75	2.86	173.89	5.06	19.4
UK10_10	2	230.94	1.91	220.74	3.60	4.42	230.53	2.26	189.05	3.82	17.99
UK10_11	2	inf.	2.71	296.27	3.92	-	inf.	1.92	264.59	2.92	-
UK10_12	2	219.28	5.32	208.75	21.78	4.80	202.64	2.52	177.81	4.44	12.25
UK10_13	2	inf.	1.79	224.21	2.94	-	226.65	1.18	192.54	2.63	15.05
UK10_14	2	inf.	1.54	199.36	5.32	-	inf.	1.20	167.68	5.44	-
UK10_15	2	inf.	3.06	152.87	9.70	-	inf.	2.73	121.19	8.28	-
UK10_16	2	226.95	4.97	214.70	6.53	5.40	215.73	5.03	183.02	6.01	15.16
UK10_17	2	inf.	2.17	207.46	32.35	-	inf.	1.34	175.83	16.01	-
UK10_18	2	inf.	3.78	189.68	11.33	-	inf.	2.90	158.00	7.03	-
UK10_19	2	inf.	1.74	199.15	6.48	-	inf.	2.29	167.47	5.82	-
UK10_20	2	209.99	8.37	197.52	18.80	5.94	197.25	14.64	165.84	12.47	15.92

Table 3.27 Computational results for 15-node instances with initial congestion period of 3600 seconds

Instance	# of vehicles	Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
		$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %
UK15_01	2	inf.	234.88	299.06	556.78	-	inf.	667.67	283.22	618.29	-
UK15_02	2	226	25.92	219.36	30.37	2.94	213.31	28.08	203.52	35.62	4.59
UK15_03	2	inf.	4746.72	316.59	3186.76	-	inf.	7422.00	300.75	6316.59	-
UK15_04	3	inf.	71.64	318.50	53.86	-	inf.	24.98	294.74	33.26	-
UK15_05	2	inf.	14.17	299.90	40.14	-	inf.	41.47	284.06	27.85	-
UK15_06	2	inf.	8862.00	244.05	1050.61	-	240.6	2221.46	228.21	1932.42	5.15
UK15_07	3	281.15	26.71	269.44	6.44	4.16	261.56	8.19	245.68	9.84	6.07
UK15_08	2	185.47	162.84	178.97	33.59	3.51	171.94	75.11	163.13	52.41	5.12
UK15_09	3	293.51	1138.32	281.89	70.98	3.96	278.86	126.32	258.11	105.38	7.44
UK15_10	2	234.14	40.70	227.71	42.99	2.74	225.05	30.76	211.87	53.35	5.85
UK15_11	2	inf.	20.69	275.26	232.20	-	inf.	26.45	259.42	123.38	-
UK15_12	3	340.57	24.63	330.51	19.71	2.95	331.72	38.74	306.75	36.77	7.53
UK15_13	2	inf.	909.86	265.09	1028.96	-	inf.	1939.60	249.25	1379.09	-
UK15_14	3	inf.	3083.37	359.58	2871.24	-	inf.	10130.00	335.82	2408.28	-
UK15_15	2	239.81	48.45	232.81	96.55	2.92	219	134.98	216.97	155.69	0.93
UK15_16	2	224.67	27.34	214.37	7.88	4.58	208.32	7.30	198.53	44.56	4.7
UK15_17	3	inf.	9.82	302.04	5.00	-	300.07	5.18	278.28	6.25	7.26
UK15_18	3	inf.	58.39	332.40	10.29	-	inf.	27.24	308.65	21.24	-
UK15_19	2	184.85	9.46	178.31	4.50	3.54	176.81	4.24	162.47	6.81	8.11
UK15_20	3	inf.	16.28	220.57	7.10	-	inf.	2.84	196.81	9.42	-

Table 3.28 Computational results for 15-node instances with initial congestion period of 7200 seconds

		Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
Instance	# of vehicles	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %
UK15_01	2	inf.	234.68	337.71	2489.17	-	inf.	667.67	306.42	2972.56	-
UK15_02	2	inf.	25.86	235.40	63.91	-	231.96	28.08	203.72	42.19	12.17
UK15_03	2	inf.	4858.37	inf.	476.42	-	inf.	7422.00	inf.	748.03	-
UK15_04	3	inf.	71.58	343.16	67.64	-	inf.	24.98	295.64	194.17	-
UK15_05	2	inf.	14.14	349.49	272.13	-	inf.	41.47	331.04	520.05	-
UK15_06	2	inf.	8856.99	263.74	1853.29	-	inf.	2221.46	232.06	2105.63	-
UK15_07	3	inf.	26.68	304.60	168.01	-	inf.	8.19	257.08	114.88	-
UK15_08	2	206.04	162.79	194.81	70.76	5.45	192.87	75.11	163.13	60.50	15.42
UK15_09	3	inf.	1137.68	306.18	234.59	-	inf.	126.32	258.66	290.29	-
UK15_10	2	inf.	40.65	245.23	74.02	-	inf.	30.76	213.55	44.76	-
UK15_11	2	inf.	20.68	337.12	824.14	-	inf.	26.45	308.15	790.05	-
UK15_12	3	inf.	24.61	354.41	31.77	-	inf.	38.74	69.55	72.18	-
UK15_13	2	inf.	913.76	282.77	2093.01	-	inf.	1939.60	262.93	5685.92	-
UK15_14	2	inf.	3079.17	inf.	6.48	-	inf.	10130.00	inf.	6.63	-
UK15_15	2	260.51	48.30	248.65	94.52	4.55	inf.	134.98	216.97	106.01	-
UK15_16	2	245.24	27.30	230.21	11.01	6.13	inf.	7.30	198.53	12.89	13.40
UK15_17	3	inf.	9.80	325.80	14.14	-	inf.	5.18	278.28	14.34	16.10
UK15_18	3	inf.	58.32	363.74	173.09	-	inf.	27.24	316.22	417.77	-
UK15_19	2	202.42	9.47	194.15	13.23	4.09	197.74	4.24	162.47	10.80	17.84
UK15_20	3	inf.	16.23	244.55	243.15	-	inf.	2.84	200.68	71.21	-

Table 3.29 Computational results for 20-node instances with initial congestion period of 3600 seconds

		Drivers paid from the beginning of the planning horizon					Drivers paid from departure				
Instance	# of vehicles	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %
UK20_01	3	347.16	416.29	337.9	265.72	2.68	328.90	212.49	314.10	169.66	4.50
UK20_02	3	365.84	295.04	352.9	225.98	3.54	inf.	321.04	329.12	161.04	-
UK20_03	3	233.27	76.69	224.0	44.97	3.97	216.53	66.35	200.01	42.36	7.63
UK20_04	3	354.83	3360.44	347.1	1546.29	2.17	354.34	2929.29	323.36	1919.35	8.74
UK20_05	3	325.59	258.29	317.4	360.26	2.53	312.87	370.71	292.12	219.98	6.63
UK20_06	3	349.35*	2124.82	365.02*	5637.66	-	339.50*	6701.12	347.27*	1520.50	
UK20_07	3	255.39	1456.06	246.93*	2394.83	-	223.1*	10800.40	223.4*	1091.46	-
UK20_08	3	307.47	575.73	298.3	54.03	3.00	288.17	232.23	274.10	83.39	4.88
UK20_09	3	inf.	54.36	345.0	169.47	-	inf.	32.64	321.26	119.14	-
UK20_10	3	291.58*	3977.5	310.9	1816.07	-	307.98	9120.02	287.15	2288.59	6.76
UK20_11	3	391	140.35	381.6	38.50	2.41	374.23	173.63	357.82	234.21	4.38
UK20_12	3	346.02	2253.71	334.6	463.88	3.29	322.48	1853.51	310.87	463.90	3.60
UK20_13	3	339.15	83.24	329.9	176.90	2.74	327.46	128.74	306.10	74.62	6.52
UK20_14	4	inf.*	10799.6	437.49*	1701.06	-	inf.*	2521.06	404.66*	1651.95	-
UK20_15	3	349.63	642.49	338.4	607.60	3.22	327.47	3105.17	313.94	800.90	4.13
UK20_16	3	358.16	741.31	346.4	170.18	3.30	331.72	895.87	322.60	149.28	2.75
UK20_17	3	inf.	905.97	379.72*	2170.39	-	inf.	2498.11	355.61	5864.80	-
UK20_18	3	inf.	445.71	367.5	1132.39	-	inf.	1357.34	343.71	685.20	-
UK20_19	3	351.16	1926.32	343.4	3405.90	2.22	349.63	253.10	319.60	2524.09	8.59
UK20_20	3	354.13	11.56	343.1	15.56	3.11	337.82	10.09	319.37	13.75	5.46

* Not solved to optimality.

Table 3.30 Computational results for 20-node instances with initial congestion period of 7200 seconds

		Drivers paid from the beginning of the planning horizon						Drivers paid from departure			
Instance	# of vehicles	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %	$TC(S_N)$ £	$t(S_N)$ s	$TC(S_D)$ £	$t(S_D)$ s	Saving %
UK20_01	3	inf.	416.29	362.44	286.10	-	inf.	212.49	314.90	673.14	-
UK20_02	3	inf.	295.04	378.75	207.29	-	inf.	321.04	331.20	541.94	-
UK20_03	3	264.38	76.69	247.53	158.97	6.37	245.9	66.35	200.00	100.15	18.66
UK20_04	3	inf.	3360.44	371.80	4318.09	-	inf.	2929.29	324.30	4647.32	-
UK20_05	3	356.9	258.29	340.60	894.88	4.57	inf.	370.71	293.10	940.19	-
UK20_06	3	349.35*	2124.82	412.04*	10799.80	-	339.50*	6701.12	inf.	-	-
UK20_07	3	285.35	1456.06	270.63*	7058.32	-	223.17*	10800.40	inf.*	4299.49	-
UK20_08	3	338.8	575.73	321.91	128.01	4.99	inf.	232.23	274.40	119.65	-
UK20_09	3	inf.	54.36	379.15	676.14	-	inf.	32.64	331.80	1761.27	-
UK20_10	3	291.58*	3977.50	335.73*	4271.10	-	inf.	9120.02	288.80	7355.26	-
UK20_11	3	inf.	140.35	414.64	2554.09	-	inf.	173.63	368.20	2471.30	-
UK20_12	3	inf.	2253.71	361.30	3523.84	-	inf.	1853.51	316.20	3076.75	-
UK20_13	3	inf.	83.24	360.09	2171.69	-	inf.	128.74	312.60	1884.26	-
UK20_14	3	inf.*	10799.60	inf.*	1954.92	-	inf.*	2521.06	inf.*	1779.78	-
UK20_15	3	inf.	642.49	366.01	3407.18	-	inf.	3105.17	318.50	5048.37	-
UK20_16	3	inf.	741.31	370.12	748.19	-	363.12	895.87	322.60	1811.35	11.16
UK20_17	3	inf.	905.97	410.747*	10800.80	-	inf.	2498.11	369.13*	10797.90	-
UK20_18	3	inf.	445.71	395.57	4054.01	-	inf.	1357.34	351.65*	10799.50	-
UK20_19	3	inf.	1926.32	371.63	9726.61	-	inf.	253.10	324.111*	10799.50	-
UK20_20	3	inf.	11.56	367.51	21.24	-	inf.	10.09	320.00	36.21	-

* Not solved to optimality

It is a mistake to think you can solve any major problems just with potatoes.
Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

4 A Metaheuristic Algorithm for the Time-Dependent Pollution-Routing Problem

In the previous chapter we introduced the Time-Dependent Pollution Routing Problem (TDPR), which consists of routing a number of vehicles to serve a set of customers and determining their speed on each route segment with the objective of minimizing driver wage and carbon dioxide-equivalent emissions. The vehicles face traffic congestion which, at peak periods, significantly restricts vehicle speeds and leads to more emissions. We presented a linear mixed integer mathematical model for the TDPRP which can be used to solve small scale problem instances, up to 25 customer nodes. In order to solve larger instances, in this chapter we propose a metaheuristic algorithm for the TDPRP. Our algorithm is based on an adaptive large neighborhood search heuristic and uses new insertion and removal operators which are tailored to this problem. Results from extensive computational experimentation demonstrate the good performance of our algorithm.

4.1 Introduction

In the past, the planning of freight transportation activities was mostly focused on cutting costs and increasing profitability by considering internal transportation costs only, that is, mainly fuel cost and drivers' wages (see, e.g., Forkenbrock, 1999, 2001). Nowadays, freight companies also need to consider their impact on the environment and particularly the amount of greenhouse gases (GHGs) generated by their vehicle fleet, as many cities have enacted new environmental legislation which restrict heavy freight vehicle traffic in certain urban areas (see, e.g., Demir et al., 2014b, 2015). Congestion, which is a major problem in many cities, increases greenhouse gases emissions and therefore should be taken into account when planning vehicle routing.

In this chapter we propose a metaheuristic algorithm to solve the TDPRP which uses the *Time-Dependent Departure Time and Speed Optimization Procedure* (TDDSOP) presented in Chapter 5 as a sub-routine. Other recent studies have developed metaheuristic algorithms for solving the PRP and its variants: Demir et al. (2012) a propose a metaheuristic which iterates between the solution of the Vehicle Routing Problem with Time Windows (VRPTW) and a speed optimization problem. The VRPTW is solved by an Adaptive Large Neighborhood Search (ALNS). The speed optimization problem is solved by means of a procedure that runs in polynomial time. In a related study, Demir et al. (2013) investigate the trade-offs between fuel consumption and driving time. The authors show that in order to achieve a considerable reduction in fuel consumption and CO_2e emissions, trucking companies need not compromise significantly in terms of driving time. Kramer et al. (2014) propose a method that combines a local search-based metaheuristic with an integer programming approach over a set covering formulation and a recursive speed optimization algorithm. Koç et al. (2014) introduce the *fleet size and mix PRP*, which considers a heterogenous vehicle fleet and develop a hybrid evolutionary algorithm. Dabia et al. (2014) obtain an exact solution based on a branch-and-price algorithm by formulating the master problem as a set-partitioning problem, and the pricing problem as a speed- and start time elementary shortest path problem with resource constraints. They solve the master problem by means of column generation, and the pricing problem by a tailored labeling algorithm.

The main contributions of our chapter are as follows. First we consider an extension of the PRP to a time-dependent setting with traffic congestion, namely the TDPRP, for which, to our knowledge, no metaheuristic has been proposed. Second, our algorithm is based on an adaptive large neighborhood search (ALNS) heuristic for which we develop new insertion and removal operators, which are tailored to this problem and shown to perform well. Third, we show numerically that our algorithm is fast and performs well compared to existing solution methods for the TDPRP and even gives good results for the special case of the PRP.

The remainder of the chapter is organized as follows. §4.2 recap the main feature of the TDPRP. §4.3 describes our metaheuristic algorithm. In §4.4, we present our numerical study. Conclusions are stated in §4.5.

4.2 Model

In this section we present the main features of the TDPRP (see Chapter 3 for a more in-depth description) and discuss the feasibility conditions of the problem.

4.2.1 Problem description

The TDPRP consists of routing a number of vehicles to make deliveries from a depot to a set of customers. It is defined on a complete graph $G = \{N, A\}$, where N is the set of nodes, and A is the set of arcs between every pair of nodes. Let 0 denote the depot, and $N_0 = N \setminus \{0\}$ denote the set of customer nodes. The distance between two nodes $(i \neq j \in N)$ is denoted by $d_{i,j}$. We consider a homogeneous fleet of vehicles, each with a capacity of Q units, which are initially located at the depot. Let h_i denote the service time at customer node $i \in N_0$, which corresponds to the time required to make the delivery. We set the service time at the depot equal to 0, i.e. $h_0 = 0$. Also let $[l_i, u_i]$ denote the hard time window at customer node $i \in N_0$ during which service must start: if a vehicle arrives at node i before the lower time window limit l_i, the driver must wait until time l_i to start serving the customer and the service must start before upper time limit u_i. After the service has been completed, the vehicle is allowed to wait idly at the customer node before leaving for the next customer node. This waiting time is referred to as the *post-service waiting time*. As shown in Chapter 3, in some cases waiting idly at the customer is an effective strategy to avoid traveling in congestion and may lead to a reduction in fuel consumption. Finally, let q_i denote the delivery quantity at customer node $i \in N_0$.

In line with the previous chapter we consider two methods to calculate the driver labour costs, referred to as *driver wage policies*: (a) the driver is paid from the beginning of the planning horizon, (b) the driver is paid from the instant he or she leaves the depot. The objective of the TDPRP is to determine: (i) the set of vehicle routes, each starting and ending at the depot, (ii) the vehicle speed on each arc, and (iii) the departure times from each node, so as to minimize emissions and labor costs.

Traffic congestion is modeled using a two-level speed function where there is an initial period of congestion, lasting a units of time, followed by a period of free-flow, during which the vehicle is allowed to drive at any speed up to a maximum value of $v^{max} > v_c$ as in Figure 3.1.

To calculate the amount of vehicle CO_2e emissions we use the *comprehensive modal emissions model* (CMEM) by Scora and Barth (2006) and Barth and Boriboonsomsin (2009). According to this model, the quantity of GHG emissions generated when traversing a distance d at a constant speed of v carrying a load of f is directly proportional to the amount of fuel consumed on this arc. For a complete description of the formulas used to model the TDPRP we refer to Chapter 3. The values of all vehicle and emission parameters are reported in Table 4.2 in Chapter 3.

4.2.2 Feasibility conditions

The TDPRP is feasible if it is possible to serve each customer with a separate vehicle without violating their upper time window, that is, if

$$\min\left\{a, d_{0,i}/v_c\right\} + ((d_{0,i} - av_c)^+)/v^{max} \leq u_i \qquad \forall i \in N_0.$$

In what follows we assume that these conditions are satisfied (and this is also true of all the problem instances we consider in our numerical experiment).

Next we discuss the feasibility conditions for a given vehicle route. Let $(0, 1, \ldots, n, 0)$ denote a fixed route where 0 is the depot and $i \in \{1, \ldots, n\}$ are customer nodes. Let $\underline{w}_i$ denote the earliest possible service completion time at node i, which is obtained by assuming the vehicle drives at the maximum speed v^{max} on every arc of the route and never waits at any node following the completion of service. The values of $\underline{w}_i$ can be computed recursively as:

$$\begin{aligned}
\underline{w}_0 &= 0\\
\underline{w}_i &= \begin{cases}
\max\left\{\underline{w}_{i-1} + \frac{d_{i-1,i}}{v_c}, l_i\right\} + h_i & \text{if } \underline{w}_{i-1} \leq \left(a - \frac{d_{i-1,i}}{v_c}\right)^+\\
\max\left\{a + \frac{d_{i-1,i} - (a - \underline{w}_{i-1})v_c}{v^{max}}, l_i\right\} & \text{if } \left(a - \frac{d_{i-1,i}}{v}{}_c\right)^+ \leq \underline{w}_{i-1}\\
+h_i & \text{and } \underline{w}_{i-1} \leq a\\
\max\left\{\underline{w}_{i-1} + \frac{d_{i-1,i}}{v^{max}}, l_i\right\} + h_i & \text{if } \underline{w}_{i-1} \geq a.
\end{cases} \qquad \text{for } i = 1, \ldots, n
\end{aligned}$$

The route is feasible if (i) the sum of delivery quantities does not exceed the vehicle capacity, i.e., $\sum_{i=1}^{n} q_i \leq Q$, and (ii) the vehicle arrives at each customer node before its upper time window limits when driving at the maximum speed without any post-service waiting time, i.e. $\underline{w}_i - h_i \leq u_i$ for $i = 1, \ldots, n+1$.

4.3 An Adaptive Large Neighborhood Search heuristic for the TDPRP

This section presents an Adaptive Large Neighborhood Search (ALNS) heuristic for the TDPRP. Pioneered by Pisinger and Ropke (2007) and Ropke and Pisinger (2006a), the ALNS heuristic is an extension of the Large Neighborhood Search (LNS) heuristic first introduced by Shaw (1998). Both methods are advanced techniques which aim at finding near-optimal solutions by repeatedly looking for a better solution in a large neighborhood around the *current* solution. Specifically, the neighborhood of the *current* solution is explored using a *removal* operator and a *insertion* operator in order to create an *incumbent* solution. The removal operator partially deconstructs the *current* solution and the insertion operator rebuilds it in a different way. Whether the *incumbent* solution is accepted as the new *current* solution is determined using a simulated annealing acceptance rule: the *incumbent* solution is always accepted if it has a lower cost than the *current* solution and is accepted with a certain positive probability otherwise. This probability is calculated using a *temperature* variable which decreases at the end of each iteration such that the probability of accepting the *incumbent* solution goes down over time. The ALNS heuristic extends the LNS heuristic by allowing the use of multiple removal and insertion operators. At each iteration, the ALNS heuristic selects one removal and one insertion operator using a roulette wheel mechanism, where the probability of choosing a certain operator is adjusted dynamically and depends on its past and current performance.

The ALNS heuristic has proved to be very efficient in solving a wide variety of transportation problems, (see, e.g., Ropke and Pisinger, 2006a; Hemmelmayr et al., 2012; Aksen et al., 2014). A pseudocode for the general formwork of our ALNS heuristic is shown in Algorithm 2. We use the following notation: S_{init} is the *initial* solution, S_{best} is the *best* solution encountered so far, S_c is the *current* solution and S_n is the *incumbent* solution.

All user-controlled parameters are denoted by Greek letters. A detailed description of each part of Algorithm 2 is provided in the following subsections.

4.3.1 Construction of the initial solution

An initial feasible solution is generated using a modified version of the sequential insertion heuristic (SIH) introduced by Solomon (1987). The SIH starts creating a first route from a "seed" customer which is the one closest to the depot. In each subsequent iteration, the SIH either adds one of the currently unassigned nodes to one of the existing (partial) routes or creates a new route with only that node (leaving from the depot and returning to it immediately afterwards).

Algorithm 2: Our implementation of the ALNS heuristic

Input: Set of removal operators Ω^-, set of insertion operators Ω^+, cooling rate ς and constants $\underline{\alpha}, \overline{\alpha}, \eta, \underline{\gamma}, \overline{\gamma}, \beta, \delta, k, \lambda, \sigma_1, \sigma_2, \sigma_3$ and Δ.
Output: A feasible solution S_{best}

$S_{init} \leftarrow$ Generate an initial solution using $\underline{\alpha}$ and $\overline{\alpha}$
For each removal operator $i \in \Omega^-$, initialize probability $\phi_i^- \leftarrow \frac{1}{|\Omega^-|}$
For each insertion operator $j \in \Omega^+$, initialize probability $\phi_j^+ \leftarrow \frac{1}{|\Omega^+|}$
$T \leftarrow \eta \cdot TC(S_{init})$
$S_{best} \leftarrow S_{init}$
$S_c \leftarrow S_{init}$
repeat
    Select a removal operator $i \in \Omega^-$ with probability ϕ_i^-
    Select a insertion operator $j \in \Omega^+$ with probability ϕ_j^+
    $S_n \leftarrow$ Obtain incumbent solution by applying operators i and j to S_c (possibly using $\underline{\gamma}, \overline{\gamma}, \beta, \delta$ and κ)
    if $TC(S_n) < TC(S_c)$ **then**
        $S_c \leftarrow S_n$
    else
        $r \leftarrow$ Generate a random number in $[0, 1]$
        if $r < e^{-(TC(S_n) - TC(S_c))/T}$ **then**
            $S_c \leftarrow S_n$
    if $TC(S_c) < TC(S_{best})$ **then**
        $S_{best} \leftarrow S_c$
    $T \leftarrow \varsigma \cdot T$
    Update: ϕ_i^-, ϕ_i^+ using constants $\lambda, \sigma_1, \sigma_2$ and σ_3.
until *The maximum number of iterations* Δ *is reached*;

Let $(j_0, \ldots, j_{n+1})$ be a current partial route, where j_0 and j_{n+1} are two copies of the depot, i.e., $j_0 = j_{n+1} = 0$. The insertion cost of unassigned customer i between adjacent nodes j_k and $j_k + 1$ for $k \in \{0, 1, \ldots, n\}$ is $C^i_{j_k, j_k+1}$, which is calculated as:

$$C^i_{j_k,j_k+1} = \begin{cases} d_{j_k,i} + d_{i,j_k+1} - \alpha d_{j_k,j_k+1} & \\ + \max\{0, u_i - u_{j_k+1}\} & \text{if route } (j_0, ..., j_k, i, j_{k+1}, ..., j_{n+1}) \text{ is feasible,} \\ \infty & \text{otherwise.} \end{cases} \quad (4.1)$$

In equation (4.1), α is a diversification parameter used to obtain different initial solutions in separate runs of the ALNS heuristic. Specifically, every time we compute the insertion cost, a new α value is randomly drawn from the interval $[\underline{\alpha}, \overline{\alpha}]$ assuming a uniform distribution. Note that the insertion cost is infinite if inserting customer i between nodes j_k and j_{k+1} would violate the route feasibility conditions of §4.2.2. In each iteration, the SIH considers each unassigned node i and calculates the cost of inserting it in each possible position in the current set of (partial) routes, as well as the cost of creating a new route to serve this customer, i.e., $C^i_{0,0}$. The unassigned node with the least insertion cost is then selected and inserted in the position where it minimizes the insertion cost. After all customers have been inserted into a feasible position, the TDDSOP is run to optimize the travel speeds and departure times on each route.

4.3.2 Adaptive weight adjustment procedure

The selection of the removal and insertion operators is regulated by a roulette-wheel mechanism in which a weight is assigned to each operator and the probability of selection is proportional to these weights. Let Ω^- and Ω^+ denote the set of removal and insertion operators, respectively and let ρ^-_j and ρ^+_j denote the weights of the j^{th} removal and insertion operator, respectively. At every iteration of the ALNS algorithm the probability ϕ^-_j of choosing operator j is calculated as $\phi^-_j = \rho^-_j / \sum_{i=1}^{|\Omega^-|} \rho^-_i$ and $\phi^+_j = \rho^+_j / \sum_{i=1}^{|\Omega^+|} \rho^+_i$ respectively for the removal and insertion operators. At the end of each iteration, the weights of the removal and insertion operators which were used in this iteration are updated; suppose removal operator j and insertion operator k were used, then their weights are recalculated as follows:

$$\rho^-_j \leftarrow \lambda \rho^-_j / n^-_j + (1-\lambda)\Psi \quad \text{or} \quad \rho^+_k \leftarrow \lambda \rho^+_k / n^+_k + (1-\lambda)\Psi, \tag{4.2}$$

where λ is the roulette wheel parameter, n^-_j or n^+_k is the number of times removal operator j and insertion operator k have been used since the start of the algorithm and Ψ is the score of the operators for this iteration of the ALNS algorithm, which is based on their joint performance, as follows:

$$\Psi = \begin{cases} \sigma_1 & \text{if a new best solution was found, i.e., if } TC(S_n) \leq TC(S_{best}); \\ \sigma_2 & \text{if the incumbent solution was better than the current solution,} \\ & \text{i.e., if } TC(S_n) \leq TC(S_c); \\ \sigma_3 & \text{if the incumbent solution was worse than the current solution,} \\ & \text{i.e., if } TC(S_n) > TC(S_c). \end{cases} \tag{4.3}$$

Note that, when the incumbent solution is worse than the current solution, the score of the operators is the same whether or not the incumbent solution is accepted as the new current solution.

4.3.3 Acceptance and stopping criteria

We use a simulated annealing heuristic as the acceptance rule for the incumbent solution. Given a current solution S_c with a total cost $TC(S_c)$, the incumbent solution S_n is always accepted if it has a lower cost than the current solution, i.e., if $TC(S_n) \leq TC(S_c)$. Otherwise, the incumbent solution S_n is accepted with probability $e^{-(TC(S_n)-TC(S_c))/T}$ where T is the current *temperature*. The temperature starts at a positive value then decreases over time as it gets multiplied by the *cooling rate* $\varsigma \in [0,1]$ in each iteration. This has the effect of making it less likely over time to accept an incumbent solution for a given difference with respect to the current solution. In line with Ropke and Pisinger (2006a) we set the initial temperature equal to $\eta \cdot TC(S_{init})$, where S_{init} denotes the initial solution. The ALNS heuristic stops when a number Δ of iterations has been reached.

4.3.4 Removal and insertion operators

Our proposed ALNS heuristic uses eight removal operators and four insertion operators. In the removal phase, a number of customer nodes are removed from the current solution and added to a removal list $\mathcal{L}$. Subsequently, in the insertion phase all nodes in $\mathcal{L}$ are reinserted in the partially destroyed solution, according to some insertion criteria to obtain the incumbent solution. A heuristic algorithm is then run to optimize the travel speeds and the departure times of the vehicles on each route. Such algorithm, called *Time-Dependednt Departure Time and Speed Optimization* procedure (TDDSOP) is described in Chapter 5.

4.3.4 Removal operators

We first provide a description of our eight removal operators. The first five, namely the RR, WNR, PSR, CR and NGR operators, are adapted from existing work (see Shaw (1998), Ropke and Pisinger (2006a), Demir et al. (2012), Ribeiro and Laporte (2012), Demir et al. (2013)). The last three, namely the MSR, HSR and LSR operators, are new operators which we propose. Note that of these three new operators, the MSR can be used for solving several types of VRPs, while the other two (HSR and LSR) were specifically designed for the TDPRP.

Most of the removal operators listed below operate by sorting the customer nodes according to a given metric. Then a fixed number of nodes are removed from the current solution, such that nodes with a smaller index in the sorted list are more likely to be chosen. In practice, this choice is implemented using a randomization process as follows. Consider a ranking of n customer nodes according to a given metric. For each node which is removed from the list, a random number, y, is drawn from a uniform distribution on $[0,1]$. The node which is chosen for removal is the $\lfloor y^{\delta} n \rfloor$-th one on the list, where δ is a positive fixed input parameter. Note that high (low) delta values leads to a higher (lower) probability of choosing nodes with smaller indices. Such a randomization process is done in order to diversify the search as in Ribeiro and Laporte (2012).

We now provide a detailed description of the removal operators used in the ALNS algorithm. In what follows, γ, which is the number of nodes removed from the current solution is a value randomly generated from a uniform distribution on $[\underline{\gamma}, \overline{\gamma}]$.

Random removal operator (RR)
: The RR operator randomly selects γ customer nodes and removes them from the current solution S_c.

Worst node removal operator (WNR)
: The WNR operator sorts the customer nodes according to their removal cost calculated as $TC(S_c) - TC(S_c \setminus i)$, where $TC(S_c \setminus i)$ is the cost of solution S_c after the removing node i, which is calculated after running the DSOP in order to re-optimize the speeds and the departure times in the route which included i. Given the ranking of customer nodes based on their removal cost (from highest to lowest), γ nodes are chosen using the randomization process described above.

Proximity-based Shaw removal operator (PSR)
: The PSR operator randomly selects a node in N_0 and adds it to the removal list $\mathcal{L}$. Then the customer nodes which are not in $\mathcal{L}$ are sorted based on their distance to this node j (from the closest to the furthest) and a node is chosen from this ranking using the randomization process described above. Next, a node in $\mathcal{L}$ is selected at random and the customer nodes which are in $N_0 \backslash \mathcal{L}$ are then sorted based on their distance to this node (from the closest to the furthest). The next node added to $\mathcal{L}$ is then chosen from this ranking using the randomization process described above. The process repeats until γ nodes have been removed from the current solution S_c.

Cluster removal operator (CR)
: The CR operator starts by randomly choosing a route r from the current solution, then divides its nodes into two subsets using a modified version of the Kruskal's algorithm for the *Minimum Spanning Tree Problem* (Kruskal, 1956) which stops as soon as two connected components are found. After the two clusters have been created, the CR operator randomly selects one of the two, removes all its nodes from the current solution and adds them to the removal list $\mathcal{L}$. If the total number of removed nodes is less than γ, the CR operator selects a random node $j \in \mathcal{L}$ and looks for a node i which is the closest one to j but belongs to a different route, say route r'. The route

r' is then partitioned into two clusters and the process is repeated until at least γ nodes have been removed from the current solution.

Neighbor graph removal operator (NGR)

The NGR operator chooses γ nodes to remove using some historical information saved in a *neighbor graph*. To each arc (i, j) in the original graph is associated a weight in the neighbor graph which corresponds to the total cost of the best solution found so far wherein node i is visited just before node j by the same vehicle. At the beginning of the ALNS heuristic all arcs have an infinite weight, then these values are updated at the end of each iteration. Given a current solution S_c the NGR operator calculates the cost of customer node i by summing the arcs weights in the *neighbor graph* of the arc going into node i and the arc leaving node i. The customer nodes are then sorted based on that cost metric (from the highest to the lowest) and γ nodes are chosen for removal using the randomization process described above.

Most scattered route removal operator (MSR)

For each route r in the current solution S_c, the MSR operator calculates a scatter index SI_r equal to the ratio of the length of the route to the number of nodes in the route, i.e., $SI_r = l_r/n_r$ where l_r is the sum of arc distances in route r, and n_r is the number of nodes in r, (including the depot). The routes with the greatest scatter index values are considered for removal. In order to avoid a myopic behavior and to diversify the search, we randomize the selection of the route by multiplying the current greatest index by $1 + y$, where y is a random variable between -0.5 and 0.

Highest speed removal operator (HSR)

Given the current solution S_c, the HSR operator assigns a value to every node $i \in N_0$ as follows: every customer node such that the travel speed on its incoming arc is larger than on its outgoing arc gets a value equal to the travel speed on its incoming arc. Other customer nodes gets a value of zero. The customer nodes are then sorted based on this value (from the highest to the lowest) and γ nodes are chosen from removal using the randomization process described above. The reasoning behind this newly developed operator is that changing the position of the nodes which mark a slow down in the vehicle speed might help even out the travel speed, and therefore reduce the emissions cost.

Lowest speed removal operator (LSR)

The LSR is very similar to the HSR operator except that, the value assigned to each node $i \in N_0$ is equal to the difference between the travel speed on the outgoing arc and the travel speed on the incoming, if this value is positive, otherwise it is equal to zero. The customer nodes are then sorted based on this value (from the highest to the lowest) and γ nodes are chosen from removal using the randomization process described above. The reasoning behind this newly developed operator is that re-positioning the nodes with such large speed difference might be beneficial as it might contribute to even out the travel speeds and reduce the average pre-service waiting time.

4.3.4 Insertion operators

Let S_p denote the partial solution obtained after removing a number of customer nodes from the current solution S_c using one of the removal operators listed above and let $\mathcal{L}$ be the removal list, wherein the nodes are listed in the order in which they were removed. We now describe the insertion operators which can be used to construct the incumbent solution S_n. Note that all the operators we present can be used for solving several types of VRPs, not just the TDPRP. The first three operators, namely the BGI, MGI and $k-$RIH operators, are adapted from Shaw (1998); Ropke and Pisinger (2006b); Demir et al. (2012); Ribeiro and Laporte (2012); Demir et al. (2013), the last one, namely the HIH operator is a new one we propose.

Best greedy insertion operator (BGI)

The BGI operator selects the nodes in $\mathcal{L}$ one at time (in the order they are listed). For each node, it calculates the cost of inserting it between every pair of adjacent nodes in the partial destroyed solution S_p, as well as in a new route from a depot and returning to it immediately afterwards. This insertion cost is equal to the increase in total cost after inserting the node (calculated after running the DSOP) if the resulting solution satisfies the route feasibility conditions from §4.2.2 and is infinite otherwise. The node is then inserted in the position with the least insertion cost. The process is repeated until all nodes in $\mathcal{L}$ have been inserted back.

Modified greedy insertion operator (MGI)

The MGI operator works in a similar way as the BGI operator except that the insertion cost of node $i \in \mathcal{L}$ between adjacent nodes j and k in S_p is calculated as $d_{ji} + d_{ik} - d_{jk}$ if the insertion satisfies the route feasibility conditions from §4.2.2 and is infinite otherwise. Compared to BGI operator, this operator is much faster since calculating the insertion cost does not require solving the DSOP. In this case the DSOP is run only once, after all nodes in $\mathcal{L}$ have been inserted.

κ-regret insertion operator (κ-RIH)

For each node in $\mathcal{L}$, the κ-RIH operator first calculates the cost of inserting it between every pair of adjacent nodes in the partial destroyed solution

S_p, as well as in a new route from a depot and returning to it immediately afterwards, in the same way as the BGI operator. Then, for each node $i \in \mathcal{L}$ the k-RIH operator calculates the *Regret Value* $RV_i = C_{i,\kappa} - C_{i,1}$, where $C_{i,\kappa}$ is the cost of inserting node i into the position with the κ^{th} lower cost, where κ is a fixed parameter.

The node with the maximum regret value is then removed from $\mathcal{L}$ and inserted where it generates the lowest insertion cost. The process is repeated until all nodes in $\mathcal{L}$ have been inserted in S_p.

Note that the DSOP is run whenever an insertion cost is calculated and whenever an unassigned node is inserted in the partial solution. The main advantage of this operator is that it improves the myopic behavior of the operators introduced previously (see Ropke and Pisinger, 2006b; Potvin and Rousseau, 1993).

Hybrid insertion operator (HIH)

If the length of the removal list $\mathcal{L}$ is shorter than a fixed parameter β, the HIH operator tries to insert the whole list of nodes between every pair of adjacent nodes and also considers creating a new route with these nodes only. The order of the nodes is the original order from the removal list with probability 1/2 or the reverse order with probability 1/2 (this is implemented using a uniform random variable on $[0, 1]$). The insertion cost of the sequence of nodes is equal to the increase in total cost if the resulting route satisfies the route feasibility conditions from §4.2.2 and is infinite otherwise. If the minimum insertion cost is finite, the sequence of nodes is inserted in the position of lowest insertion cost. If not, or if the removal list $\mathcal{L}$ is longer then β, the HIH operator randomly selects nodes one at time from $\mathcal{L}$, and inserts them each in their position with the lowest insertion cost. Note that the DSOP is run whenever an insertion cost is calculated and after an unassigned node is inserted in the partial solution.

An overview of all removal and insertion operator is reported in Table 4.1.

4.4 Computational experiments

This section presents the results of computational experiments to assess the performance of our metaheuristic on the TDPRP.

All our numerical experiments are based on instances from the PRP Library (PRPLIB), available at `http://www.apollo.management.soton.ac.uk/prplib.htm`. The PRPLIB contains 180 problem instances, grouped into nine sets of 20 instances, such that the instances in each set have the same number of customer nodes, which varies between 10 and 200 nodes. Each instance comes with the

Table 4.1 A summary of the removal and insertion operators

Operator	Newly proposed ?	Call DSOP as sub-routine
Removal		
RR	✘	✘
WNR	✘	✔
PSR	✘	✘
CR	✘	✘
NGR	✘	✘
MSR	✔	✘
HSR	✔	✘
LSR	✔	✘
Insertion		
BGI	✘	✔
MGI	✘	✘
k-RIH	✘	✔
HIH	✔	✔

following information: (i) the distances between each pair of nodes (these are based on actual geographical distances between randomly selected cities from the United Kingdom), (ii) the demand at each node, (iii) the service time window at each node, (iv) the service time at each node and (v) the maximum vehicle traveling speed. In our numerical experiments, we consider all 180 problem instances from the PRPLIB and use the same cost function parameters as in Chapter 3. These are reported in Table 4.2. For each instance we ran the ALNS heuristic five times.

Table 4.2 Used parameters in the objective function (Demir et al., 2012)

Notation	Description	Unit	Value
A	Weight module constant	$£m/(kJ\,s^2)$	0.000382
B	Engine module constant	$£/s^2$	0.00142
C	Speed module constant	$£kg/(kJ\,m)$	$1.98e^{-7}$
D	Driver wage	$£/s$	0.00222
μ	Curb-weight	kg	6,350

Our ALNS heuristic was coded in Java and run on a server with 2.4 GHz of CPU and 8 GB of RAM.

4.4.1 Parameter tuning

Our implementation of the ALNS heuristic contains fourteen user controlled parameters which are provided in the Table 4.3, along with the value we used in our numerical experiments. Similarly to Demir et al. (2012), we divide the parameters into four groups. Group (i) includes the parameters that control the generation of the initial solution. Group (ii) includes the parameters that control the roulette-wheel mechanism. Group (iii) includes the parameters that control the simulated annealing search framework and the ones that calibrate the initial temperature and cooling rate. Finally, group (iv) includes all parameters that control the removal and insertion operators.

Table 4.3 Parameters used in the ALNS heuristic

Group	Notation	Description	Value
(i)	$\underline{\alpha}$	Minimum α value	0.7
	$\overline{\alpha}$	Maximum α value	2
(ii)	λ	Roulette wheel parameter	0.2
	σ_1	Operators score when best solution found	3
	σ_2	Operators score when incumbent better than current solution	1
	σ_3	Operators score when incumbent worse than current solution	2
(iii)	Δ	Total number of iterations	$25,000$
	η	Initial temperature parameter	0.9985
	ς	Cooling rate	0.001
(iv)	$\underline{\gamma}$	Minimum number of nodes to remove	$\lfloor \log_{10}(N) \rfloor$
	$\overline{\gamma}$	Maximum number of nodes to remove	$\lfloor \log_{1,35}(N) \rfloor$
	δ	Randomization parameter	4
	κ	Insertion control parameter for κ-RIH operator	4
	β	Insertion control parameter for HIH operator	3

To arrive at the value of these parameters, we conducted some parameter tuning numerical experiments. In particular, we considered multiple sets of values for the $(\sigma_1, \sigma_2, \sigma_3)$ and η parameters. The results from these tuning experiments are shown in the following subsections. The values of all other parameters are set as in Demir et al. (2012). To ensure comparison fairness we used the same initial solution and seeds for the random numbers across runs with different parameter values.

4.4.1 Tuning of σ values

In order to tune the control parameters used in the roulette-wheel mechanism, i.e., σ_1, σ_2 and σ_3 defined in (4.3), we ran some numerical tests on five 100-node instances from the PRPLIB. We considered the following sets of values : (i) $\sigma_1 = 3, \sigma_2 = 1, \sigma_3 = 2$, (ii) $\sigma_1 = 1, \sigma_2 = 2, \sigma_3 = 3$, (iii) $\sigma_1 = 3, \sigma_2 = 2, \sigma_3 = 1$, and (iv) $\sigma_1 = 1, \sigma_2 = 1, \sigma_3 = 1$. In Table 4.4 we report the total cost of the best solution obtained across the 5 runs as well as the average total cost.

The results reported in Table 4.4 show that $(3, 1, 2)$ and $(3, 2, 1)$ overall perform the best and $(1, 2, 3)$ performs the worst. This suggests that it is a good idea to reward the operators with a high score when a new best solution was found. Also it may beneficial to encourage diversification by assigning a medium score when

Table 4.4 Tuning of the roulette wheel mechanism parameters (σ_1, σ_2 and σ_3)

Instance	$(3,1,2)$		$(1,2,3)$		$(3,2,1)$		$(1,1,1)$	
	Best	Average	Best	Average	Best	Average	Best	Average
UK100_01	**1314.47**	1321.85	1316.12	**1320.37**	1314.91	1321.25	1319.29	1321.78
UK100_02	1236.75	1243.99	1237.12	1245.76	**1235.21**	**1240.74**	1237.27	1242.52
UK100_03	**1180.24**	**1182.39**	1182.61	1186.76	1183.92	1186.23	1184.30	1186.63
UK100_04	1173.17	1183.47	1182.01	1185.99	1177.00	1184.32	**1170.30**	**1179.73**
UK100_05	1134.69	1147.71	1142.07	1149.34	**1130.29**	1151.12	1140.54	**1147.63**

Highest values for each instance are shown in bold.

the incumbent solution is worse than the current solution. Based on this analysis, we decided to use $\sigma_1 = 3, \sigma_2 = 1, \sigma_3 = 2$ for the rest of our numerical experiments.

4.4.1 Tuning of η value

In order to tune the parameter η, which controls the initial temperature, we ran some tests on the same five instances as in the previous subsection using three different parameter values: (*i*) 0.001, (*ii*) 0.01, and (*iii*) 0.1. A higher value of η means that the initial temperature is higher, leading to a higher probability of accepting the incumbent solution in each iteration, given a fixed cost difference value with the current solution. The results of the computational experiments are reported in Table 4.5.

Table 4.5 Tuning of the initial temperature parameter (η) for the simulated annealing acceptance rule

Instance	$\eta = 0.001$		$\eta = 0.01$		$\eta = 0.1$	
	Best	Average	Best	Average	Best	Average
UK100_01	**1314.47**	**1321.85**	1324.94	1330.44	1338.48	1350.80
UK100_02	**1236.75**	**1243.99**	1238.85	1253.40	1251.21	1265.13
UK100_03	1180.24	**1182.39**	**1178.21**	1189.29	1191.70	1210.00
UK100_04	**1173.17**	**1183.47**	1174.79	1188.06	1186.22	1213.44
UK100_05	**1134.69**	**1147.71**	1140.61	1158.68	1184.42	1188.72

Highest values for each instance are shown in bold.

In Table 4.5 we report the total cost of the best solution obtained across the 5 runs as well as the average total cost. Our results suggest that it is preferable to use a small values of η, since 0.001 generally leads to lower total cost values.

Figure 4.1 shows how the total cost of the best solution, i.e., $TC(S_{best}$, varies over time for the three different parameter values for a particular run of the ALNS heuristic on one instance, (namely UK100_03). We see that, not only does our heuristic find a over better solution with $\eta = 0.0001$ parameter, it also reaches better solutions sooner in the process.

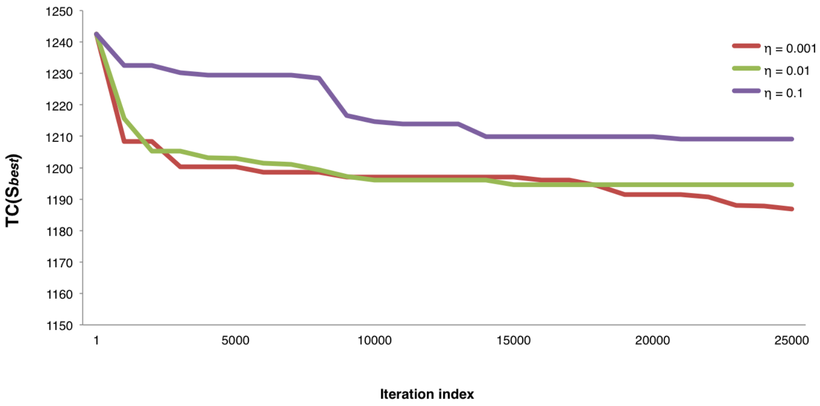


Figure 4.1 Total cost of the best solution for different initial temperature values (η)

Given this analysis, in our computational experiments, we decided to set the value of η equal to 0.001.

4.4.2 Computational time analysis

In this section we analyze the speed of our ALNS heuristic. In Table 4.6 we report average computational (CPU) time on a number problem instances with varying sizes (the number of customer nodes varies between 10 and 200). Each row reports the average results (over 5 runs) from a single instance. The first column entitled *Instance* reports the name of the instance (the first number in the name corresponds to the number of nodes in the instance), the column entitled *CPU time* reports the average CPU time (in minutes), the column entitled *RO time* reports the average CPU time (in minutes) collectively spent by all the removal operators, the column entitled *IO time* reports the average CPU time (in minutes) collectively spent by all the repair operators, finally the column entitled *DSOP time* reports the average CPU time (in minutes) spent in solving the DSOP. Next to the average values, we report the standard deviation values in brackets. Finally, we also report the percentage of the total CPU time spent in solving the DSOP in the last column.

The values in Table 4.6 show that, our ALNS heuristic can solve up to 100-node instances in less than 15 minutes with remarkably low volatility in the CPU time. Further, we observe that, on average, the insertion operators require more time than the removal operators. This is because three out of four insertion operators (namely BGI, k-RIH and HIH) run the DSOP as a sub-routine versus only one out of the eight removal operators (namely WNR). Running the DSOP is a time-

Table 4.6 Average computational time spent in each stage of the algorithm

Instance	CPU time (min)	RO time (min)	IO time (min)	DSOP time (min)
UK10_03	0.23 (0.02)	0.03 (0.01)	0.17 (0.07)	0.14 (0.01) 59.0%
UK15_03	0.46 (0.19)	0.04 (0.02)	0.37 (0.15)	0.30 (0.13) 67.0%
UK20_03	0.76 (0.05)	0.05 (0.01)	0.64 (0.04)	0.53 (0.04) 69.7%
UK25_03	1.05 (0.15)	0.06 (0.01)	0.89 (0.14)	0.72 (0.12) 68.6%
UK50_03	2.97 (0.40)	0.13 (0.02)	2.56 (0.38)	2.16 (0.32) 72.6%
UK75_03	6.67 (0.27)	0.29 (0.01)	5.78 (0.26)	4.92 (0.24) 73.8%
UK100_03	12.62 (0.47)	0.65 (0.03)	10.29 (0.62)	8.67 (0.48) 68.7%
UK150_03	23.37 (1.06)	1.24 (0.08)	19.77 (0.94)	16.68 (0.80) 71.4%
UK200_03	44.31 (1.63)	2.33 (0.14)	34.19 (1.28)	28.73 (1.08)
Average	10.27 (0.47)	0.54 (0.04)	8.30 (0.43)	6.98 (0.36) 68.4%

consuming process as evidenced by the fact that it requires more than 50% of the total CPU time.

We further analyze the time required by each operator in the next section.

4.4.3 Relative performance of the operators

In this section, we study the relative performance of the removal and insertion operators. Table 4.7 reports the average CPU time (in seconds) required by each operator per iteration in which they are used.

Table 4.7 Average CPU time (in seconds) for each operator per iteration

Instance	Removal Operators								Insertion Operators			
	RR	WNR*	PSR	CR	NGR	MSR	HSR	LSR	BGI*	MGI	k-RIH*	HIH*
UK10_03	0.0000	0.0002	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0003	0.0000	0.0007	0.0005
UK15_03	0.0000	0.0003	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0006	0.0000	0.0019	0.0010
UK20_03	0.0000	0.0004	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0009	0.0000	0.0035	0.0017
UK25_03	0.0000	0.0006	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0011	0.0000	0.0050	0.0027
UK50_03	0.0001	0.0020	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0031	0.0001	0.0147	0.0067
UK75_03	0.0001	0.0047	0.0001	0.0001	0.0002	0.0001	0.0001	0.0001	0.0071	0.0001	0.0338	0.0143
UK100_03	0.0001	0.0109	0.0002	0.0003	0.0004	0.0002	0.0002	0.0002	0.0146	0.0003	0.0627	0.0209
UK150_03	0.0001	0.0220	0.0003	0.0002	0.0005	0.0002	0.0002	0.0002	0.0251	0.0005	0.1152	0.0484
UK200_03	0.0002	0.0414	0.0005	0.0004	0.0009	0.0003	0.0004	0.0004	0.0442	0.0008	0.2037	0.0784
Average	0.0001	0.0092	0.0002	0.0002	0.0003	0.0001	0.0001	0.0001	0.0108	0.0002	0.0490	0.0194

* uses the DSOP as a sub-routine.

The numbers in Table 4.7 confirm what discussed in §4.4.2, that is, that the operators which use the DSOP as a sub-routine require a longer CPU time per iteration. These results also show that, on average, all operators require less than one second per iteration.

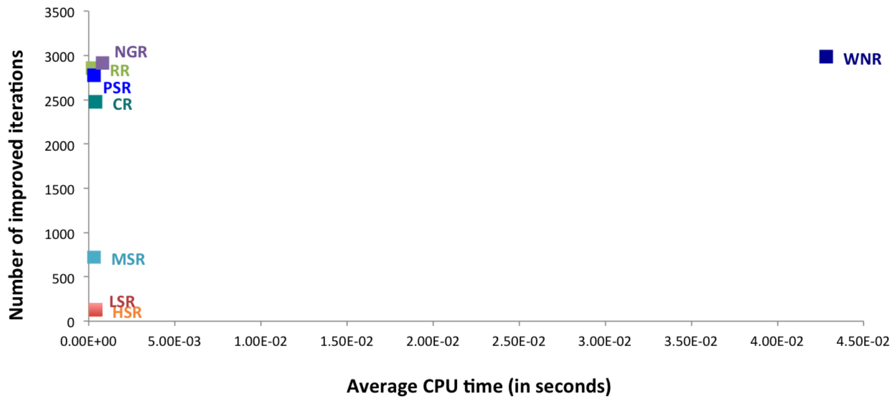
With Figure 4.2 we investigate the relationship between CPU time and performance of each operator. Specifically, the X-axis reports the average CPU time required by each operator per iteration and the Y-axis reports the number of iterations in which the total cost of the incumbent solution, which was found using this operator, is less that the total cost of the current solution. Figures 5.5 and 5.6b display the average values (out of five runs) obtained solving instance UK200_3.

From Figure 5.5 we see that, the NGR, RR, PSR and CR operators are fast and perform well, in the sense that they very often lead to an improved current solution. The WNR operator has the best improvement performance but it is much slower than the other operators due to the fact that it runs the DSOP as a sub-routine. Regarding the insertion operators, we see a clear tradeoff between performance and speed: the MGI operator is fast but does not improve the solution as much as the other three insertion operators which use the DSOP as a sub-routine.

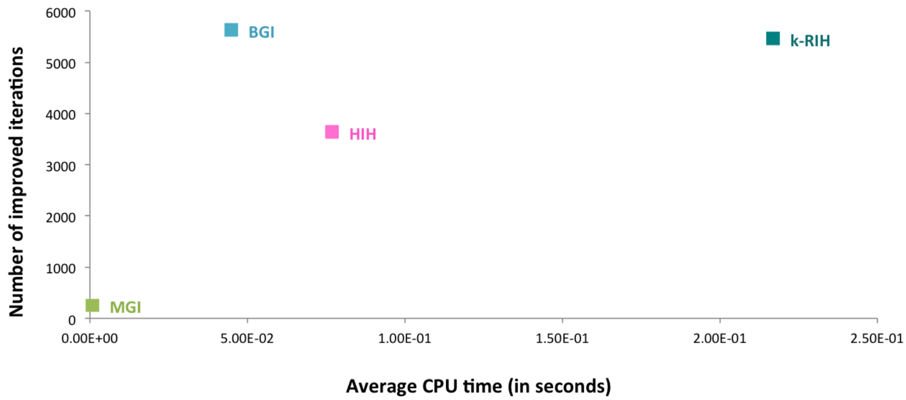
To emphasize the value of the operators we proposed versus the ones we borrowed from existing work, we ran different versions of our ALNS heuristic algorithm on a set of 100-node instances. In Version #1, we included all eight removal and all four insertion operators listed in §4.3.4. In Version #2, we included all four insertion operators but only the five removal operators borrowed from the literature (namely RR, WNR, PSR, CR and NGR). In Version #3, we included all eight removal operators but only the three insertion operators borrowed from the literature (namely BGI, MGI and k-RIH). Comparing the performance of Version #1 and Version #2 provides information on the value of the removal operators we developed, while comparing between Version #1 and Version #3 provides information on the value of the insertion operators we proposed.

The results are reported in Table 4.8. The columns entitled $Version\#1$, $Version\#2$ and $Version\#3$ report the total cost of the best solution value (out of five runs) for Version #1, Version #2, Version #3, respectively. The column *Savings RO* reports the percentage improvement obtained from using the new removal operators, which is calculated as $[TC(Version\#2) - TC(Version\#1)]/TC(Version\#1)$. Finally, the column *Savings IO* reports the percentage improvement obtained from using the new insertion operators which is calculated as $[TC(Version\#3) - TC(Version\#1)]/TC(Version\#1)$.

The results displayed in Table 4.8 show that, in most cases, adding the new operators we developed to the list of possible operators improves the solution quality. The same conclusion holds for both the removal and the insertion operators. While the operators inspired from the literature are mainly aimed at minimizing the total distance travelled by the vehicles, the new ones attempt to reduce the negative effects of fluctuating speeds in transport networks. As



(a) Removal operators



(b) Insertion operators

Figure 4.2 Number of "improved" iteration vs average CPU time per iteration (UK200_03)

discussed in Chapter 3, these negative effects becomes more relevant in the presence of traffic congestion and therefore they cannot be ignored.

4.4.4 Performance on TDPRP instances

In this section we study the performance of our ALSN heuristic on TDPRP instances. First we use a set of small-sized (i.e., 10-, 20-, and 25-node) instances

Table 4.8 Comparison of three different variants of the algorithm

Instance	Version #1 TC (£)	Version #2 TC (£)	Version #3 TC (£)	Savings RO (%)	Savings IO (%)
UK100_01	1314.5	1318.83	1315.24	0.33	0.06
UK100_02	1236.8	1235.59	1238.04	-0.09	0.10
UK100_03	1180.2	1181.92	1182.24	0.14	0.17
UK100_04	1173.2	1180.62	1177.52	0.63	0.37
UK100_05	1134.7	1149.16	1148.03	1.28	1.18
Average	1207.9	1213.2	1212.2	0.46	0.38

from the PRPLIB to compare our algorithm to the solutions presented in Chapter 3, which were ran using CPLEX. [1] The values of the parameters are set as follows: $a = 2$ hours, $v_c = 10$ km/h and $v^{max} = 90$ km/h. Table 4.9 presents the comparison for the case where the driver is paid from the beginning of the planning horizon and Table 4.10 presents the results for the case where the driver is paid from the departure time. The columns entitled *ALNS* report total cost, i.e. TC, of the best solution found by our metaheuristic. The columns entitled *CPLEX* report the total cost reported in Chapter 3. Finally, the columns entitled *Dev.* report the percentage difference between the total cost we obtain and the total cost from Chapter 3.

The results in Tables 4.9 and 4.10 show that most of the instances have very small deviations (in absolute value), which suggests that our ALNS heuristic performs very well. The maximum positive deviation is 1.59%, which is the worst case performance of our solution method. In a few cases the deviation is negative, implying that the solution obtained with our ALNS heuristic is actually better than the solution obtained inChapter 3. This is due to the fact that the MIP formulation from Chapter 3 uses a discrete set of values for the free-flow travel speeds while our ALNS heuristic allows for continuous values.

Next we use a set of large-size (100 and 200-node) instances from the PRPLIB to further analyze the performance of our ALNS heuristic, in terms of speed and performance volatility across separate runs. As before, for every instance, we ran the ALNS heuristic five times. The values of the parameters are set as follows: $a = 1$ hour, v_c =10 km/h and v^{max} = 90km/h. The driver is paid from the departure time.[2]

The results for the 100- and 200-node instances are displayed in Tables 4.11 and Table 4.12, respectively. The first column lists the instance name, the second, third and fourth ones report the cost of the best, worst and average solution

[1] Since the MIP formulation requires that the the number of vehicles is fixed at a value K, we first run our ALNS heuristic, then use the number of vehicles obtained in the best solution as the value of K when solving the MIP.

[2] Similar insights were obtained when the driver is paid from the beginning of the time horizon.

Table 4.9 Total cost comparison between ALNS and CPLEX (driver paid from the beginning of the planning horizon)

Instance	UK_10			UK_15			UK_20		
	ALNS $TC(£)$	CPLEX $TC(£)$	Dev. (%)	ALNS $TC(£)$	CPLEX $TC(£)$	Dev.	ALNS $TC(£)$	CPLEX $TC(£)$	Dev. (%)
1	183.975	183.980	-0.00	298.998	299.060	-0.02	337.859	337.860	-0.00
2	218.904	218.900	-0.00	219.359	219.360	-0.00	352.878	352.880	-0.00
3	213.341	213.340	-0.00	302.422	316.590	-4.68	224.008	224.010	-0.00
4	202.168	202.170	-0.00	318.416	318.500	-0.03	347.119	347.120	-0.00
5	188.072	188.070	-0.00	299.820	299.900	-0.03	317.360	317.360	-0.00
6	229.088	229.130	-0.02	244.047	244.050	-0.00	363.540	365.02*	-0.41
7	205.179	205.180	-0.00	269.440	269.440	-0.00	246.935	246.935*	-0.00
8	237.169	237.170	-0.00	178.965	178.970	-0.00	298.251	298.250	-0.00
9	189.729	189.730	-0.00	281.873	281.890	-0.01	344.918	345.020	-0.03
10	204.894	204.890	-0.00	227.714	227.710	-0.00	309.789	310.910	-0.36
11	277.118	277.120	-0.00	275.167	275.260	-0.03	381.579	381.580	-0.00
12	195.611	193.650	1.00	330.514	330.510	-0.00	334.628	334.630	-0.00
13	208.374	208.370	-0.00	264.996	265.090	-0.04	329.856	329.860	-0.00
14	179.835	179.840	-0.00	359.584	359.584.	-0.00	435.414	437.490*	-0.48
15	135.458	135.460	-0.00	232.812	232.810	-0.00	338.369	338.370	-0.00
16	198.858	198.860	-0.00	214.373	214.370	-0.00	346.357	346.360	-0.00
17	171.559	171.600	-0.02	302.040	302.040	-0.00	379.367	379.72*	-0.09
18	173.958	173.960	-0.00	332.357	332.400	-0.01	366.293	367.470	-0.32
19	181.197	181.280	-0.05	178.305	178.310	-0.00	343.361	343.360	-0.00
20	181.675	181.680	-0.00	218.783	220.570	-0.82	343.129	343.130	-0.00
Average	198.808	198.719	0.05	267.499	263.517	-0.28	337.050	332.386	-0.001

calculated with our ALNS,respectively. The fifth column reports the percentage range, calculated as *(worst-best)/worst*. Finally the last column reports the average computational time (in minutes).

We see from Tables 4.11 and Table 4.12 that the performance of the ALNS heuristic does not vary significantly from one run to the next. The maximum percentage range is 3.52% for 100-node instances and 1.23% for 200-node instances. This suggest that our solution method is quite robust so it is not necessary to run our algorithm a very large number of times. We also note that the average CPU time does not vary greatly from one instance to the next, with an average of 11.49 minutes for 100-node instances and 38.3 minutes for 200-node instances. The numbers from Table 4.12 suggest that our method is even more robust on larger problem instances.

4.4.5 Performance on PRP instances

In this section we compare the performance of our ALNS heuristic to other solution methods for the PRP, which, as discussed earlier, is a special case of the TDPRP where the congestion period is zero. We compare our results on 100-node instances to the results from Demir et al. (2012), Koç et al. (2014) and Kramer et al. (2014)

Table 4.10 Total cost comparison between ALNS and CPLEX (driver paid from departure time)

Instance	UK_10			UK_15			UK_20		
	ALNS $TC(£)$	CPLEX $TC(£)$	Dev. (%)	ALNS $TC(£)$	CPLEX $TC(£)$	Dev. (%)	ALNS $TC(£)$	CPLEX $TC(£)$	Dev. (%)
1	168.135	168.140	-0.00	283.158	283.220	-0.00	314.099	314.100	-0.00
2	203.064	203.060	-0.00	203.519	203.520	-0.00	329.118	329.120	-0.00
3	197.501	197.500	-0.00	278.662	300.750	-0.08	200.011	200.010	-0.00
4	186.328	185.880	-0.00	294.656	294.740	-0.00	323.359	323.360	-0.00
5	172.232	172.230	-0.00	283.980	284.060	-0.00	292.115	292.120	-0.00
6	213.248	213.290	-0.00	225.049	228.210	-0.01	340.332	347.270*	-2.04
7	189.339	189.340	-0.00	245.724	245.680	-0.00	223.175	223.400*	-0.10
8	221.329	221.330	-0.00	163.125	163.130	-0.00	274.096	274.100	-0.00
9	173.889	173.890	-0.00	260.491	258.110	0.01	321.158	321.260	-0.03
10	189.054	189.050	-0.00	211.874	211.870	-0.00	288.093	287.150	0.33
11	261.278	261.280	-0.00	259.327	259.420	-0.00	357.819	357.820	-0.00
12	179.771	177.810	0.01	306.754	306.750	-0.00	310.868	310.870	-0.00
13	192.534	192.530	-0.00	238.519	249.250	-0.04	306.096	306.100	-0.00
14	164.716	164.720	-0.00	335.824	335.824	-0.00	405.811	404.660	0.28
15	119.618	119.620	-0.00	215.993	216.970	-0.00	319.015	313.940	1.59
16	183.018	183.020	-0.00	198.533	198.530	-0.00	322.597	322.600	-0.00
17	155.719	155.760	-0.00	278.280	278.280	-0.00	355.607	355.610	-0.00
18	158.118	158.000	-0.00	308.597	308.650	-0.00	348.868	343.710	1.48
19	165.357	165.440	-0.00	162.465	162.470	-0.00	319.601	319.600	-0.00
20	165.835	165.840	-0.00	198.025	196.810	0.01	319.369	319.370	-0.00
Average	183.004	182.887	-0.00	247.628	249.312	-0.01	313.560	316.417	-0.00

Table 4.11 Performance of ALNS heuristic on 100-node instances

Instance	Best $TC(£)$	Worst TC(£)	Average $TC(£)$	Range (%)	CPU Time (min)
UK100_01	1313.10	1328.13	1320.33	1.13	11.13
UK100_02	1232.15	1277.09	1252.25	3.52	10.99
UK100_03	1181.02	1192.63	1186.09	0.97	10.60
UK100_04	1162.41	1203.46	1181.10	3.41	11.57
UK100_05	1140.17	1157.85	1149.64	1.53	9.84
UK100_06	1291.91	1314.72	1303.10	1.74	9.57
UK100_07	1134.63	1157.84	1146.45	2.00	12.21
UK100_08	1172.65	1213.32	1190.14	3.35	12.45
UK100_09	1070.62	1089.97	1080.49	1.78	13.60
UK100_10	1144.91	1185.75	1161.82	3.44	12.25
UK100_11	1297.52	1329.01	1312.21	2.37	11.38
UK100_12	1118.23	1151.84	1134.50	2.92	9.84
UK100_13	1220.87	1241.66	1228.84	1.67	11.77
UK100_14	1347.21	1359.43	1353.79	0.90	11.24
UK100_15	1404.48	1439.02	1422.95	2.40	10.94
UK100_16	1051.71	1073.60	1068.12	2.04	13.22
UK100_17	1368.04	1393.64	1382.14	1.84	10.74
UK100_18	1166.95	1191.58	1179.48	2.07	12.23
UK100_19	1101.73	1122.40	1110.73	1.84	11.07
UK100_20	1335.91	1355.84	1344.15	1.47	13.18
Average	1212.81	1238.94	1225.42	2.12	11.49

in Table 4.13. The first column lists the instance name, the second one reports the best solution calculated with our ALNS heuristic (out of five runs). The columns

Table 4.12 Performance of ALNS heuristic on 200-node instances

Instance	Best $TC(£)$	Worst TC(£)	Average $TC(£)$	Range (%)	CPU Time (min)
UK200_01	2255.05	2308.76	2278.63	0.01	35.01
UK200_02	2106.51	2128.93	2119.72	0.01	38.99
UK200_03	2183.89	2211.73	2193.10	0.00	34.66
UK200_04	2072.93	2099.32	2085.95	0.01	34.89
UK200_05	2321.12	2349.16	2339.81	0.01	34.16
UK200_06	2019.73	2052.19	2040.33	0.01	38.44
UK200_07	2161.31	2183.44	2169.98	0.00	37.29
UK200_08	2240.30	2270.53	2255.93	0.01	37.04
UK200_09	1984.03	2018.40	2000.04	0.01	37.53
UK200_10	2363.40	2407.45	2378.37	0.01	39.18
UK200_11	2069.41	2096.99	2083.01	0.01	36.21
UK200_12	2276.70	2327.79	2299.02	0.01	34.16
UK200_13	2282.64	2310.01	2296.27	0.01	34.26
UK200_14	2154.00	2204.75	2175.42	0.01	41.20
UK200_15	2245.66	2273.94	2261.96	0.01	37.90
UK200_16	2220.16	2248.00	2228.62	0.00	41.09
UK200_17	2333.95	2362.21	2347.32	0.01	38.91
UK200_18	2163.56	2188.39	2175.29	0.01	48.56
UK200_19	1950.87	1991.36	1975.16	0.01	46.69
UK200_20	2294.61	2338.14	2313.65	0.01	39.15
Average	2184.99	2218.58	2200.88	0.01	38.27

DBL12, *KBJL14* and *KSV14* provide the results reported in Demir et al. (2012), Koç et al. (2014) and Kramer et al. (2014), respectively. Finally, the columns entitled *Dev.* report the percentage deviation between our best results and the ones from the literature, namely (ALNS-DBL12)/ALNS, (ALNS-KBJL14)/ALNS and (ALNS-KBJL14)/ALNS .

Table 4.13 shows that our ALNS heuristic is able to compete with the best heuristics for the PRP, even though it was designed for a more general version of the problem, namely the TDPRP. It even provides the best solution out of all methods for four of the instances (namely UK100_02, UK100_06, UK100_16, and UK100_20).

4.5 Conclusions

We have described a metaheuristic algorithm to solve the TDPRP, which extends the PRP to settings with traffic congestion. This problem is very relevant since congestion is an important problem for many cities and the amount of greenhouse gasses emissions significantly increase at lower vehicle speeds.

Table 4.13 Computational results on 100-node PRP instances

Instance	ALNS	DBL12		KBJL14		KBJL14	
	$TC(£)$	$TC(£)$	Dev. (%)	TC (\$)	Dev. (%)	TC (\$)	Dev. (%)
UK100_01	1216.79	1240.79	-1.97	1212.72	0.33	**1209.11**	0.63
UK100_02	**1146.06**	1168.17	-1.93	1149.16	-0.27	1146.79	-0.06
UK100_03	1086.84	1092.73	-0.54	1080.87	0.55	**1078.75**	0.74
UK100_04	1095.07	1106.48	-1.04	1085.66	0.86	**1075.29**	1.81
UK100_05	1042.34	1043.41	-0.10	1033.19	0.88	**1028.86**	1.29
UK100_06	1193.05	1213.61	-1.72	**1192.67**	0.03	1193.38	-0.03
UK100_07	1054.24	1060.08	-0.55	**1044.58**	0.92	1045.02	0.87
UK100_08	1091.70	1106.78	-1.38	1092.67	-0.09	**1089.84**	0.17
UK100_09	991.45	1015.46	-2.42	992.36	-0.09	**988.41**	0.31
UK100_10	1069.66	1076.56	-0.65	1063.05	0.62	**1059.95**	0.91
UK100_11	1201.55	1210.25	-0.72	1200.53	0.08	**1196.50**	0.42
UK100_12	1043.34	1053.02	-0.93	1030.17	1.26	**1027.38**	1.53
UK100_13	1133.61	1154.83	-1.87	**1132.02**	0.14	1132.03	0.14
UK100_14	1247.86	1264.50	-1.33	**1241.31**	0.52	1242.68	0.41
UK100_15	1312.98	1315.50	-0.19	1311.36	0.12	**1300.13**	0.98
UK100_16	**981.03**	1005.03	-2.45	986.57	-0.57	981.86	-0.08
UK100_17	1272.21	1284.81	-0.99	**1257.44**	1.16	1258.16	1.10
UK100_18	1086.44	1106.00	-1.80	1088.89	-0.23	**1073.38**	1.20
UK100_19	1016.82	1044.71	-2.74	1024.17	-0.72	**1015.95**	0.09
UK100_20	**1237.93**	1263.06	-2.03	1249.84	-0.96	1240.00	-0.17
Average	1126.05	1141.29	-1.37	1123.46	0.23	1119.17	0.61

Our algorithm is based on an application of the classical ALNS heuristic and uses the departure- and speed-optimization procedure (TDDSOP) as a sub-routine. Our implementation of the ALNS combines newly developed with pre-existing removal and insertion operators and we show that the new operators we propose significantly improve the solution quality.

We conduct extensive numerical experiments to test the performance of our metaheuristic algorithm and compare it to existing solution methods for the TDPRP and one of its special cases, namely the PRP. Our results show that our algorithm performs extremely well, is robust and relatively fast on instances with up to 200 nodes.

The scheme in mathematical thinking is to divine and demonstrate. There are no set patterns of procedure. We try this and that. We guess. We try to generalize the result in order to make the proof easier. We try special cases to see if any insight can be gained in this way. Finally —who knows how? —a proof is obtained.

Joseph Buffington Roberts, *The Real Number System in an Algebraic Setting*

5 Algorithms for the Departure Times and Speed Optimization Problems

As shown in the previous chapters, travel speed is a key determinant of the amount of CO_{2e} produced by vehicles. In the recent years growing environmental concerns have shifted the focus of researchers towards the problem of optimizing the travel speed in order to reduce the amount of vehicles emissions and to improve the sustainability of transport activities. In this chapter we study the Departure Time and Speed Optimization Problem (DSOP), that is the problem of optimizing departure times and travel speeds of a vehicle visiting a finite number of customer locations according to a fixed sequence. The objective is to minimize the sum of labor and CO_{2e} emissions costs, while satisfying hard time windows at customer locations. In the first part of the chapter we present the DSOP problem and we propose an exact algorithm for solving the DSOP, which runs in quadratic time. In the second part of the chapter we study an extension of the DSOP, where there is a presence of traffic congestion limiting the vehicle speed during peak hours. This problem is referred to as the Time-Dependent Departure Time and Speed Optimization Problem (TDDSOP). Building upon some of the theoretical results of Chapter 3 we propose a heuristic algorithm for solving the TDDSOP which runs in polynomial time.

5.1 Introduction

These last years have witnessed a growing interest towards incorporating the environmental aspects (such as carbon dioxide equivalent (CO_{2e}) emissions) in the transportation decision for vehicle routing and scheduling. Whether they are motivated primarily by a desire to appear environmentally responsible or by economic concerns, companies are increasingly advertising the sustainability and green focus of their transportation strategies. As CO_{2e} emissions are directly proportional to fuel consumption and vehicle travel speed is a major determinant of fuel use, it is important for companies and logistics service providers to carefully plan out vehicle movements so as to minimize their financial and environmental impact.

The problem of optimizing the travel speed of a vehicle which is to provide service at a finite number of locations according to a fixed sequence given hard time windows was first introduced by Hvattum et al. (2010) and Hvattum et al. (2013) for ship scheduling and referred to as the *Speed Optimization Problem* (SOP). Demir et al. (2012) later adapted this problem to vehicle scheduling with the objective of minimizing the sum of labour cost and emission costs, as in the Pollution Routing Problem of Bektaş and Laporte (2011). They propose a polynomial time heuristic algorithm which builds upon the exact algorithm by Hvattum et al. (2010) and Hvattum et al. (2013). More recently Kramer et al. (2015) proposed a heuristic algorithm for a similar scheduling problem, namely the objective function is the same as in the Pollution Routing Problem and the driver is paid from the time he or she departs from the origin location.

In the first part of the chapter, we study the DSOP, that is, we optimize the departure times and travel speeds of a vehicle who visits a finite number of customers locations according to a fixed sequence in order to minimize the sum of labour and CO_{2e} emissions costs, while satisfying hard time windows. We assume there is no traffic congestion so the vehicle is able to drive at a constant speed between each pair of locations. We consider two practically motivated cases: either the driver is paid from the beginning of the planning horizon or from the start of service at the origin location. In the first case, the driver reports to the origin location at a fixed time and may be asked to do some administrative tasks before he or she starts using the vehicle. In the second, the driver is asked to report in time for his or her service and driving duties. We first establish that it is never optimal for the vehicle to wait at a customer location, following the completion of service. However it may be optimal to wait at the origin location when the driver is paid from the start of service at that location. We show that the DSOP can be formulated as a dynamic program from which we derive some key properties for the optimal travel speeds. We also show that the solution to the DSOP can be obtained by solving a shortest path problem on a network with $2n + 2$ nodes, where n is the number of customers locations. Hence, we provide an efficient

Figure 5.1 Example of a route with $n+1$ nodes and n arcs.

algorithm for solving the DSOP to optimality in quadratic time. To the best of our knowledge, our algorithm is the first exact algorithm for solving the DSOP problem. In the second part of the chapter, we extend the DSOP to the case where there is a presence of traffic congestion limiting the vehicle speed during peak hours. This problem is referred to as the *Time-Dependent Departure Time and Speed Optimization Problem* (TDDSOP). In line with Chapters 3 and 4 we assume that there is an initial period of traffic congestion lasting a units of time in which the vehicle is forced to drive at a given speed v_c, followed by a period of free-flow where the vehicle can drive at any speed level within the limits imposed by traffic regulation. We propose a heuristic algorithm to solve the TDDSOP which runs polynomial time and builds upon some of the theoretical results presented in Chapter 3 derived from the single-arc version of the problem. To the best of our knowledge, this is the first algorithm for solving the TDDSOP. The rest of the chapter is organized as follows. In §5.2 we present our model for the DSOP. In §5.3 we formulate the DSOP as a dynamic program and obtain some key properties of the value function. In §5.4 we show how to recast the problem as shortest path problem and in §5.5 we derive some managerial insights. In §5.6 we introduce the TDDSOP and we present a heuristic algorithm for solving the problem. Finally, we conclude in §5.7.

5.2 The Departure Time and Speed Optimization Problem (DSOP)

A single vehicle departs from an origin location to visit a number of customer locations according to a fixed route. For example, a delivery vehicle leaves from a central warehouse to visit retail locations and deliver merchandise, or a plumber leaves his office to visit customer homes in order to conduct plumbing repairs. Let 0 denote the origin location and let locations 1 to n denote the customer locations, so that the fixed route is $0, 1, \ldots, n$. In practice location n may be a copy of the origin location if the trip includes a return to the origin location. In Figure 5.1 we represent the route using a network of $n+1$ nodes, and n arcs. Let arc i be the arc between locations i and $i+1$, for $i = 0, \ldots, n$. Let d_i denote the distance on arc i for $i = 0, \ldots, n-1$.

Let v_i denote the speed used on arc $i = 0, \ldots, n-1$ and let v^{min} and v^{max} be the minimum and the maximum speed on every arc. We assume there is no traffic congestion so the vehicle is able to drive at a constant speed between each pair of locations. Let h_i denote the service time at location $i = 0, \ldots, n$. In the delivery vehicle example, the service time at the origin location would be the loading time and the service time at customer locations would be the delivery time, which may be proportional to the quantity of merchandise delivered. Each customer location $i = 1, \ldots, n$ has a hard time window $[l_i, u_i]$, where l_i is the lower time window limit, i.e. the earliest time service at location i can start and u_i is the upper time window limit, i.e., the latest time service at location i can start. The vehicle may arrive earlier than the lower time window limit but service may not start until that time. We refer to the time between the arrival time at a customer location and the start of service as the *pre-service waiting time*. This waiting time at customer location i can be divided between the *mandatory pre-service waiting time* and the *voluntary pre-service waiting time*, denoted by y_i for $i = 1, \ldots, n$. If the vehicle arrives at location i before the lower time window limit, then the *mandatory pre-service waiting time* is the difference between the lower time window limit and the arrival time at customer location i and the *voluntary pre-service waiting time* is equal to the difference between the start of service at location i and the lower time window limit at location i. If the vehicle arrives at location i after the lower time window limit, then the *mandatory pre-service waiting time* is zero and the *voluntary pre-service waiting time* is equal to the difference between the start of service at location i and the arrival time at location i. For the origin location, let y_0 denote the time between the start of service at the origin location and the beginning of the planning horizon.

Upon completion of service at a location, the vehicle may decide to wait before leaving; let z_i denote the *post-service waiting time* at location $i = 0, \ldots, n-1$. Let $\mathbf{y} = (y_0, \ldots, y_n)$, $\mathbf{z} = (z_0, \ldots, z_{n-1})$ and $\mathbf{v} = (v_0, \ldots, v_{n-1})$ be the voluntary pre-service waiting time vector, the post-service waiting time vector and speed vector, respectively. The trip ends with the completion of service at location n. Figure 5.2 shows an example for the sequence of events at a customer location. Note that on this picture, the vehicle arrives before the lower time window limit, so that there is a positive mandatory pre-service waiting time. Because the vehicle does not start service immediately at the lower time window limit, there is also a positive voluntary pre-service waiting time. Finally the vehicle does not leave the location immediately after the completion of service, so there is also a positive post-service waiting time.

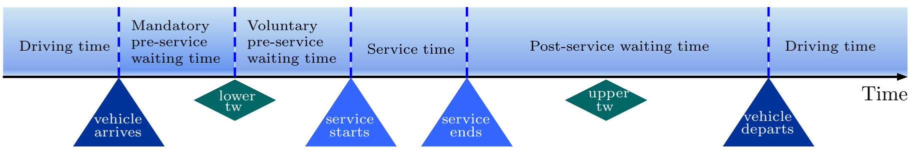


Figure 5.2 Sequence of events at a customer location.

5.2.1 Objective function

The firm which operates the vehicle aims at minimizing the cost of serving the customers, which is composed of (i) the cost of carbon dioxide equivalent (CO_{2e}) emissions, which is directly proportional to the amount of fuel used during the trip and, (ii) the labor cost, i.e. the cost of paying the driver. As such, our objective function is similar to that of Chapters 3 and 4, which is based on the formulation of emissions costs by Barth et al. (2005) and Scora and Barth (2006). In line with Chapters 3 and 4 we consider two ways of calculating the total time for which the driver is paid, referred as *driver wage policies*: (a) the driver is paid starting from the beginning of the planning horizon until the completion of service at location n, (b) the driver is paid starting from the start of the service at the origin location until the completion of service at location n. Under driver wage policy (a), the driver reports to the origin location at the start of the planning horizon. Differently, under driver wage policy (b), the driver arrives at the origin location right on time to start service.

Barth et al. (2005) and Scora and Barth (2006) assume that the amount of CO_{2e} emissions produced by a vehicle is directly proportional to the amount of fuel consumed. According to their model, the emissions cost, denoted by E_i, from traversing arc i, driving at a constant speed v_i is given by:

$$E_i(v_i) \quad = \quad Ad_i + B\frac{d_i}{v_i} + Cd_iv_i^2,$$

where A, B and C are non-negative constants, see Chapter 3 for how they are calculated. Note that this function is convex and minimized at speed $\underline{v} = \sqrt[3]{B/2C}$.

We assume that the labor cost on arc i is measured from the end of service time at location i until the end of service time at location $i+1$. Formally, the labor cost, denoted by L_i, from traversing arc i at a constant speed v_i, given a service completion time of w_i at location i, a post-service waiting time z_i at location i and a voluntary pre-service waiting time at location $i+1$ of y_{i+1} is given by:

$$\begin{aligned} & L_i(v_i, z_i, w_i, y_{i+1}) = \\ = \quad & D \begin{cases} l_{i+1} - w_i + y_{i+1} + h_{i+1} & \text{if } w_i + z_i + \frac{d}{v_i} \leq l_{i+1} \\ z_i + \frac{d_i}{v_i} + y_{i+1} + h_{i+1} & \text{if } l_{i+1} \leq w_i + z_i + \frac{d_i}{v_i} \\ & \text{and } w_i + z_i + \frac{d_i}{v_i} + y_{i+1} \leq u_{i+1} \\ \infty & \text{if } w_i + z_i + \frac{d_i}{v_i} + y_{i+1} \geq u_{i+1}, \end{cases} \end{aligned}$$

where D is the driver wage. In this expression we write that the cost is infinite when the solution is infeasible, i.e., does not satisfy the upper time window limit.

The total cost for traversing arc i is denoted by g_i and is given by:

$$
\begin{aligned}
& g_i(v_i, z_i, w_i, y_{i+1}) = \\
= \;& E_i(v_i) + L_i(v_i, z_i, w_i, y_{i+1}) \\
= \;& \begin{cases}
Ad_i + B\frac{d_i}{v_i} + Cd_i v_i^2 & \\
+D\left(l_{i+1} - w_i + y_{i+1} + h_{i+1}\right) & \text{if } w_i + z_i + \frac{d_i}{v_i} \le l_{i+1} \\
Ad_i + B\frac{d_i}{v_i} + Cd_i v_i^2 & \\
+D(z_i + \frac{d_i}{v_i} + y_{i+1} + h_{i+1}) & \text{if } l_{i+1} \le w_i + z_i + \frac{d_i}{v_i} \le u_{i+1} - y_{i+1} \\
\infty & \text{if } w_i + z_i + \frac{d_i}{v_i} + y_{i+1} > u_{i+1}
\end{cases} \\
= \;& \begin{cases}
Ad_i + B\frac{d_i}{v_i} + Cd_i v_i^2 & \\
+D\left(\max\left\{\frac{d_i}{v_i} + z_i, l_{i+1} - w_i\right\} + y_{i+1} + h_{i+1}\right) & \text{if } w_i + z_i + \frac{d_i}{v_i} + y_{i+1} \le u_{i+1} \\
\infty & \text{if } w_i + z_i + \frac{d_i}{v_i} + y_{i+1} > u_{i+1}.
\end{cases}
\end{aligned}
\tag{5.1}
$$

Note that the speed value which minimizes the first piece in (5.1) is $\underline{v}$ since the labor cost is a constant in that case. In contrast, the speed value which minimizes the second piece in (5.1) is $\overline{v} = \sqrt[3]{(B+D)/2C}$. Since $D \ge 0$, we always have $\underline{v} \le \overline{v}$. These two speed values will appear in many of our future derivations. Throughout this chapter we assume that $\underline{v} \ge v^{min}$ and $\overline{v} \le v^{max}$.

The firm's problem is to find $\mathbf{v}$, $\mathbf{z}$ and $\mathbf{y}$ to minimize the total cost incurred over the entire vehicle trip. When the driver is paid from the beginning of the planning horizon (driver wage policy (a)) the total cost, denoted by $C^{(a)}$ is equal to:

$$
(P) \qquad \min_{\mathbf{v},\mathbf{z},\mathbf{y}} \left\{ C^{(a)}(\mathbf{v}, \mathbf{z}, \mathbf{y}) \;=\; D(y_0 + h_0) + \sum_{i=0}^{n-1} g_i(v_i, z_i, w_i, y_{i+1}) \right\}
$$

such that:

$$
\begin{aligned}
& w_0 = y_0 + h_0 + z_0, & \\
& w_i = \max\left\{ w_{i-1} + z_{i-1} + \frac{d_{i-1}}{v_{i-1}}, l_i \right\} + y_i + h_i & \text{for } i = 1\ldots, n, \\
& w_{i-1} + z_{i-1} + \frac{d_{i-1}}{v_{i-1}} + y_i \le u_i & \text{for } i = 1\ldots, n, \\
& v^{min} \le v_i \le v^{max} & \text{for } i = 0, \ldots, n-1,
\end{aligned}
$$

where w_i denotes the end of service time at location i.

Similarly, let $C^{(b)}$ denote the total cost when the driver is paid from the start of service at the origin location (driver wage policy (b)). We have

$$
C^{(b)}(\mathbf{v}, \mathbf{z}, \mathbf{y}) = Dh_0 + \sum_{i=0}^{n-1} g_i(v_i, z_i, w_i, y_{i+1}) = C^{(a)}(\mathbf{v}, \mathbf{z}, \mathbf{y}) - Dy_0. \tag{5.2}
$$

5.2.2 Feasibility conditions

Let $\check{w}_i$ denote the earliest possible service completion time at location i, which is obtained by driving at the maximum speed up to that point and never waiting post-service at any location, i.e., setting $v_j = v^{max}$ and $z_j = 0$ for $j = 0, \ldots, i-1$ and $y_j = 0$ for $j = 0, \ldots, i$. We can calculate $\check{w}_i$ for $i = 0, \ldots, n$ recursively starting from location 0 as follows:

$$\begin{aligned} \check{w}_0 &= h_0 \\ \check{w}_i &= \max\left\{\check{w}_{i-1} + \frac{d_{i-1}}{v^{max}}, l_i\right\} + h_i \qquad \text{for } i = 1, \ldots, n. \end{aligned}$$

The problem is feasible if: $\check{w}_{i-1} + d_{i-1}/v^{max} \leq u_i$ for $i = 1, \ldots, n$. In what follows we assume that these conditions are satisfied.

For $i = 1, \ldots, n$, let U_i denote the latest possible arrival time into location i so it is possible to meet the upper time window at locations $i, \ldots, n$. We can calculate U_i recursively starting from location n assuming maximum speed and no post-service waiting time, as follows:

$$\begin{aligned} U_n &= u_n \\ U_i &= \min\left\{u_i, U_{i+1} - \frac{d_i}{v^{max}} - h_i\right\} \qquad \text{for } i = n, \ldots, 1. \end{aligned}$$

By definition, we have $U_i \leq u_i$ for $i = 1, \ldots, n$. Note that the feasibility of the problem implies that $l_i \leq U_i$ for $i = 1, \ldots, n$. We call U_i the *effective* upper time window limit at location i: if the vehicle was to arrive at location i between U_i and u_i then it would not violate the upper time window limit of location i but it would certainly violate the upper time window limit of at least one of the subsequent locations. Therefore, we use these effective upper time window limit U_i, rather than u_i, throughout our analysis. Similarly we refer to $[l_i, U_i]$ as the *effective* time window at location i.

Let $\hat{w}_i$ denote the latest possible service completion time at location i, so that it is still possible to meet all the effective upper time window limits at locations $i+1, \ldots, n$. We have $\hat{w}_i = U_{i+1} - d_i/v^{max}$, for $i = 0, \ldots, n-1$.

Hence, we have defined a lower and an upper bound on the service completion time at location i, namely $\check{w}_i$ and $\hat{w}_i$, for $i = 0, \ldots, n-1$.

5.3 Results

Our first result establishes that it is always optimal for the vehicle to leave a customer location immediately upon completion of service, i.e., the post-service waiting time is zero at every customer location.

Proposition 5.1 *There exists an optimal solution such that* $z_i = 0$ *for* $i = 0, \ldots, n-1$ *and* $y_i = 0$ *for* $i = 1, \ldots, n$.

Proof: First we show that, for every solution with positive voluntary pre-service waiting times at customer locations, there exists another solution with zero voluntary pre-service waiting times at customer locations which achieves the same total cost. Consider a solution S with $y_i > 0$ for $i \in \{1, ..., n\}$. There are two cases (i) either the vehicle arrives at location i before its lower time window limit, i.e., $w_{i-1} + z_{i-1} + d_{i-1}/v_{i-1} \leq l_i$ (ii) or it arrives after its lower time window limit $w_{i-1} + z_{i-1} + d_{i-1}/v_{i-1} > l_i$. In Case (i) consider an alternate solution S' with $\mathbf{v}' = \mathbf{v}, \mathbf{y}' = \mathbf{y}$ and $\mathbf{z}' = \mathbf{z}$ expect that $z'_{i-1} = l_i + y_i - w_{i-1} - d_{i-1}/v_{i-1}$ and $y'_i = 0$, which implies that the vehicle arrives at location i exactly at l_i and starts service immediately at that time. In both solutions S and S' the departure time from location i is $w_i = l_i + y_i + h_i = w_{i-1} + z'_{i-1} + d_{i-1}/v_{i-1} + h_i$ so the costs on arcs other than $i-1$ are the same. From (5.1), we can see that the cost on arc i is the same, that is, $g_{i-1}(v_{i-1}, z_{i-1}, w_{i-1}, y_i) = g_{i-1}(v'_{i-1}, z'_{i-1}, w_{i-1}, y'_i)$. In Case (ii) consider an alternate solution S' with $\mathbf{v}' = \mathbf{v}, \mathbf{y}' = by$ and $\mathbf{z}' = \mathbf{z}$ expect that $z'_{i-1} = y_i$ and $y'_i = 0$. In both solutions S and S' the departure time from location i is $w_i = w_{i-1} + z_{i-1} + d_{i-1}/v_{i-1} + y_i + h_i = w_{i-1} + z'_{i-1} + d_{i-1}/v'_{i-1} + h_i$ so the costs on arcs other than $i-1$ are the same. From (5.1), we can see that the cost on arc i is the same, that is, $g_{i-1}(v_{i-1}, z_{i-1}, w_{i-1}, y_i) = g_{i-1}(v'_{i-1}, z'_{i-1}, w_{i-1}, y'_i)$. Hence in both cases, solution S' has the same total cost at solution S, showing that there must exist an optimal solution with zero voluntary pre-service waiting time at all customers nodes.

Next we assume that every optimal solution has with positive post-service waiting times and show (contradiction) that it is always possible to find another solution with zero post-service waiting times which achieves the same or lower total cost. So suppose we have an optimal solution S with $y_i = 0$ for $i \in \{1, ..., n\}$ but $z_i > 0$ for some $i \in \{0, \ldots, n\}$. Again, there are two cases (i) either the vehicle arrives at location $i+1$ before its lower time window limit, i.e., $w_i + z_i + d_i/v_i \leq l_{i+1}$ (ii) or the it arrives after the lower time window limit, i.e., $w_i + z_i + d_i/v_i \geq l_{i+1}$. In Case (i), the vehicle has to wait at location $i+1$ until time l_{i+1} before starting service. So the sum of post-service waiting time at location i and pre-service waiting time at location $i+1$ is $z_i + (l_{i+1} - w_i - z_i - d_i/v_i) = l_{i+1} - w_i - d_i/v_i$. Now consider an alternate solution S' such that $\mathbf{v}' = \mathbf{v}, \mathbf{y}' = \mathbf{y}$ and $\mathbf{z}' = \mathbf{z}$ expect that $z'_i = 0$. In this case, the departure time from location i is $w'_i = w_i$ and the arrival time at location $i+1$ at time is $w'_i + d_i/v_i = w_i + d_i/v_i \leq l_{i+1}$, so that the vehicle has to wait $l_{i+1} - w_i - d_i/v_i$ at location $i+1$. Hence, in this alternate solution the sum of post-service waiting time at location i and pre-service waiting time at location $i+1$ is $0 + l_{i+1} - w_i - d_i/v_i$, which is the same as in the optimal solution. Since the driver is paid equally for any type of waiting time and the speed driven on all the arcs has not changed, both solutions have the same total cost. In Case (ii), the vehicle arrives at location $i+1$ at time $w_i + z_i + d_i/v_i$, which is greater than l_{i+1}, so it does not wait at location $i+1$ before starting service. Consider an alternative solution S' such that $\mathbf{v}' = \mathbf{v}, \mathbf{y}' = \mathbf{y}$ and $\mathbf{z}' = \mathbf{z}$ expect that with $z'_i = 0$ and z'_{i+1} is set as explained below. In this case, the departure time from location i is $w'_i = w_i$ and the arrival time at location $i+1$ is $w'_i + d_i/v_i = w_i + d_i/v_i$. If $i = n-1$, then the vehicle arrives at location n in solution S' at time $\max\{l_n, w_{n-1} + d_{n-1}/v_{n-1}\}$, which cannot be later than the arrival time of $\max\{l_n, w_{n-1} + z_{n-1} + d_{n-1}/v_{n-1}\}$ in the solution S. Since all waiting times and speed values are the same, it must have a lower cost, which is a contradiction. If $i < n-1$, there are 2 subcases: (a) $w_i + d_i/v_i \leq l_{i+1}$ and (b) $w_i + d_i/v_i > l_{i+1}$. In sub-case (a), the vehicle in solution S' waits for a duration of $l_{i+1} - w_i - d_i/v_i$ at location $i+1$ before starting service; then we set $z'_{i+1} = w_i + z_i + d_i/v_i + z_{i+1} - l_{i+1}$. In sub-case (b), the vehicle arrives at location $i+1$ before l_{i+1} so it does not wait before starting service. In this case we set a post-service waiting time of $z'_{i+1} = z_{i+1} + z_i$ at location $i+1$. In either sub-case, solution S' has no post-service waiting time at location i and it has the same total cost as the original solution since the total waiting time and the speed values are the same. The same argument can be used for the following locations:either we eliminate the post-service waiting time (as in Case (i)) or we transfer it to the next location (as in Case (ii)), etc., until we reach the last arc. $\square$

Note that this result contrast with that of Chapter 3 where we show that, in the presence of traffic congestion, it may be optimal for the vehicle to wait at the customer locations, following the completion of service.

Given this result we can simplify our problem formulation as follows:

$$(P2) \qquad \min_{\mathbf{v}, y_0} \left\{ C^{(a)}(\mathbf{v}, y_0) \quad = \quad D(y_0 + h_0) + \sum_{i=0}^{n-1} g_i(v_i, w_i) \right\}$$

such that

$$\begin{aligned}
& w_0 = y_0 + h_0, \\
& w_i = \max\left\{ w_{i-1} + \frac{d_{i-1}}{v_{i-1}}, l_i \right\} + h_i && \text{for } i = 1, \ldots, n, \\
& w_{i-1} + \frac{d_{i-1}}{v_{i-1}} \leq u_i && \text{for } i = 1, \ldots, n+1, \\
& v^{min} \leq v_i \leq v^{max} && \text{for } i = 1, \ldots, n.
\end{aligned}$$

when the driver is paid from the start of the planning horizon. If the driver is paid from the start of service at the origin location, the constraints are identical and

$$C^{(b)}(\mathbf{v}, y_0) = Dh_0 + \sum_{i=0}^{n-1} g_i(v_i, w_i) = C^{(a)}(\mathbf{v}, y_0) - Dy_0.$$

Note that we have dropped the reference to the z_i variables in the one-arc cost function $g_i(v_i, w_i)$. Also, from now on instead of referring to w_i as the *service completion time* at location i, we call it the *departure time* from location i, as the two can be assumed to be equal given Proposition 5.1.

When the driver is paid from the start of the planning horizon we can prove a stronger result: we can show that there is an optimal solution without voluntary pre-service wait time at the origin location also. In other words, it is never optimal to voluntarily delay the start of service at any location.

Proposition 5.2 *When the driver is paid from the start of the planning horizon, there exists an optimal solution which satisfies the conditions of Proposition 5.1 and also has $y_0 = 0$.*

Proof: The proof is by contradiction. Suppose all optimal solutions have a positive voluntary pre-service waiting time at the origin location. Let S be one such solution with voluntary pre-service waiting time at the origin location $y_0 > 0$. By Proposition 5.1, we can assume that $y_1 = \ldots = y_n = 0$ and $z_0 = \ldots = z_{n-1} = 0$. Therefore, the departure time from the origin location is $w_0 = y_0 + h_0$ and the departure time from location i can be computed recursively as $w_i = \max\{w_{i-1} + d_{i-1}/v_{i-1}, l_i\} + h_i$ for $i = 1, \ldots, n$. Consider an alternate solution S' with $\mathbf{v}' = \mathbf{v}, \mathbf{y}' = \mathbf{y}$ and $\mathbf{z}' = \mathbf{z}$ expect that $y'_0 = 0$. In this solution, we have $w'_0 = h_0$ and $w'_i = \max\left\{w'_{i-1} + d_{i-1}/v_{i-1}, l_i\right\} + h_i$ for $i = 1, \ldots, n$. It is easy to see that $w'_i \leq w_i$ for $i = 0, \ldots, n$. Since the total time the driver is paid for is w_n in S and w'_n in S', and the arc speeds are the same in both solutions, the total cost in S' must be lower or equal to that in S. Hence we have a contradiction. □

When the driver is paid from the start of the planning horizon, any waiting time at the origin location is paid for, therefore there is no benefit of postponing the start of service at the origin location. In contrast, when the driver is paid from the start of the service at the origin location, it may be optimal to delay his or her arrival to the origin location if this delay translates into a decrease in mandatory pre-service waiting time at some customer locations. We provide a more in-depth discussion of the difference between the two driver wage policies in §5.4.

5.3.1 The one-arc problem

In this section we solve the one-arc (two-locations) problem, i.e., $n = 1$. Let $v^l_{0,1}(w_0) = d_0/(l_1 - w_0)$ and $v^u_{0,1}(w_0) = d_0/(U_1 - w_0)$ denote the speed on the arc such that the vehicle arrives at location 1 exactly at time l_1 and $U_1 = u_1$ respectively, when leaving location 0 at time w_0. Since $l_1 \leq U_1$, we have $v^l_{0,1}(w_0) \geq v^u_{0,1}(w_0)$. Remember the definition of $\underline{v}$ and $\overline{v}$ from §5.2.1. We now present a full characterization of the optimal solution for the one-arc problem under both driver wage policies.

Lemma 5.1 *If the driver is paid from the beginning of the planning horizon then the optimal speed on arc* 0 *is a function of* h_0 *as depicted on Figure 5.3.*

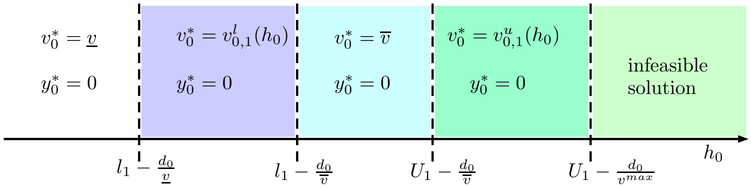


Figure 5.3 Optimal solution for the one-arc problem when the driver is paid from the beginning of the planning horizon.

Also the minimum total cost $C^{(a)*}$ *is given by:*

$$C^{(a)*}(h_0) \quad = \quad Dh_0 + \begin{cases} g_0(\underline{v}, h_0) & \textit{if } h_0 \leq l_1 - d_0/\underline{v} \\ g_0(v_{0,1}^l(h_0), h_0) & \textit{if } l_1 - d_0/\underline{v} \leq h_0 \leq l_1 - d_0/\overline{v} \\ g_0(\overline{v}, h_0) & \textit{if } l_1 - d_0/\overline{v} \leq h_0 \leq U_1 - d_0/\overline{v} \\ g_0(v_{0,1}^u(h_0), h_0) & \textit{if } U_1 - d_0/\overline{v} \leq h_0 \leq U_1 - d_0/v^{max} \\ \infty & \textit{if } h_0 > U_1 - d_0/v^{max}. \end{cases}$$

Proof: If $h_0 > u_1 - d_0/v^{max}$ the feasibility condition is not satisfied so the cost is infinite. Otherwise, we can rewrite (5.1) as:

$$g_0(v_0, h_0) \quad = \quad \begin{cases} \infty & \text{if } v_0 \leq v_{0,1}^u(h_0) \\ \begin{array}{c} Ad_0 + Bd_0/v_0 + Cd_0v_0^2 \\ +D(d_0/v_0 + h_1) \end{array} & \text{if } v_{0,1}^u(h_0) \leq v_0 \leq v_{0,1}^l(h_0) \\ \begin{array}{c} Ad_0 + Bd_0/v_0 + Cd_0v_0^2 \\ +D\left(l_1 - h_0 + h_1\right) \end{array} & \text{if } v_0 \geq v_{0,1}^l(h_0). \end{cases}$$

It follows that g_0 is continuous and convex in v_0. The second piece in this expression is minimized at $\overline{v}$ and the third piece is minimized at $\underline{v}$. So there are four sub-cases depending on how $\underline{v}, \overline{v}, v_{0,1}^u(h_0)$ and $v_{0,1}^l(h_0)$ compare. Note that we always have $v_{0,1}^u(h_0) < v_{0,1}^l(h_0)$.

- If $v_{0,1}^u(h_0) \leq \underline{v}$, which is equivalent to $0 \leq h_0 \leq l_1 - d_0/\underline{v}$, the second piece of g_0 is decreasing and the third piece reaches a minimum at $v_0 = \underline{v}$ where total cost is equal to $g_0(\underline{v}, h_0)$;

- If $\underline{v} \leq v^l_{0,1}(h_0) \leq \overline{v}$ which is equivalent to $l_1 - d_0/\underline{v} \leq h_0 \leq l_1 - d_0/\overline{v}$, then second piece of g_0 is decreasing and the third piece is increasing so that the minimum is achieved at $v_0 = v^l_{0,1}(h_0)$ where total cost is equal to $g_0\left(v^l_{0,1}(h_0), h_0\right)$;
- If $v^u_{0,1}(h_0) \leq \overline{v} \leq v^l_{0,1}(h_0)$ which is equivalent to $l_1 - d_0/\overline{v} \leq h_0 \leq u_1 - d_0/\overline{v}$, then second piece of g_0 reaches a minimum at $v_0 = \overline{v}$ where total cost is equal to $g_0(\overline{v}, h_0)$ and the third piece is increasing;
- If $\overline{v} \leq v^u_{0,1}(h_0) \leq v^{max}$ which is equivalent to $u_1 - d_0/\overline{v} \leq h_0 \leq u_1 - d_0/v^{max}$, then the second and third pieces of g_0 are increasing so that the minimum is achieved at $v_0 = v^u_{0,1}(h_0)$, where total cost is equal to $g_0\left(v^u_{0,1}(h_0), h_0\right)$.

□

Lemma 5.1 shows that, in order to minimize cost in a one-arc problem when the driver is paid from the beginning of the planning horizon, it is either optimal to (i) drive at the speed which minimizes emissions ($\underline{v}$) and arrive at the customer location before the lower time window limit, (ii) arrive exactly at the lower time window limit, (iii) drive at speed $\overline{v}$ and arrive within the time window or (iv) arrive exactly at the effective upper time window limit.

Lemma 5.2 *If the driver is paid from the start of service time at the origin location, then the optimal waiting time at location 0 and optimal speed on arc* 0 *are a function of* h_0 *as depicted on Figure 5.4.*

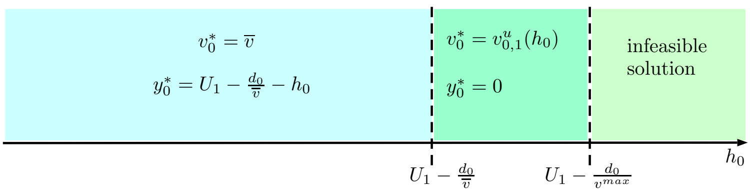


Figure 5.4 Optimal solution for the one-arc problem when the driver is paid from the start of service time at the origin location.

Also the minimum total cost $C^{(b)*}$ *is given by:*

$$C^{(b)*}(h_0) \quad = \quad Dh_0 \begin{cases} g_0(\overline{v}, U_1 - d_0/\overline{v}) & \textit{if } h_0 \leq U_1 - d_0/\overline{v} \\ g_0(v^u_{0,1}(h_0), h_0) & \textit{if } U_1 - d_0/\overline{v} \leq h_0 \leq U_1 - d_0/v^{max} \\ \infty & \textit{if } h_0 > U_1 - d_0/v^{max}. \end{cases}$$

Proof: We have

$$C^{*(b)}(h_0) = \min_{y_0 \geq 0} \min_{\mathbf{v}} C^a(\mathbf{v}, h_0 + y_0) - Dy_0 = \min_{y_0 \geq 0} C^{(*)a}(h_0 + y_0) - Dy_0.$$

From Lemma 5.1, we have:

$$C^{(*)a}(h_0 + y_0) - Dy_0 =$$

$$= \quad Dh_0 + \begin{cases} g_0(\underline{v}, h_0 + y_0) & \text{if } h_0 + y_0 \leq l_1 - d_0/\underline{v} \\ g_0(v^l_{0,1}(h_0 + y_0), h_0 + y_0) & \text{if } l_1 - d_0/\underline{v} \leq h_0 + y_0 \leq l_1 - d_0/\overline{v} \\ g_0(\overline{v}, h_0 + y_0) & \text{if } l_1 - \frac{d_0}{\overline{v}} \leq h_0 + y_0 \leq U_1 - d_0/\overline{v} \\ g_0(v^u_{0,1}(h_0 + y_0), h_0 + y_0) & \text{if } U_1 - d_0/\overline{v} \leq h_0 + y_0 \leq U_1 - d_0/v^{max} \\ \infty & \text{if } h_0 > U_1 - d_0/v^{max}. \end{cases} - Dy_0$$

$$= \begin{cases} Bd_0/\underline{v} + Cd_0\underline{v}^2 + D(l_1 - h_0 - y_0 + h_1) & \text{if } y_0 \leq l_1 - d_0/\underline{v} - h_0 \\ B(l_1 - h_0 - y_0) + Cd_0 \left(v^l_1(h_0 + y_0)\right)^2 & \\ +D(l_1 - y_0 - h_0 + h_1) & \text{if } l_1 - d_0/\underline{v} - h_0 \leq y_0 \leq l_1 - d_0/\overline{v} - h_0 \\ B\left(d_0/\overline{v}\right) + Cd_0\overline{v}^2 + D\left(d_0/\overline{v} + h_1\right) & \text{if } l_1 - d_0/\overline{v} - h_0 \leq y_0 \leq U_1 - d_0/\overline{v} - h_0 \\ B(U_1 - y_0 - h_0) + Cd_0 \left(v^u_1(w_0)\right)^2 & \\ +D(U_1 - y_0 - h_0 + h_1) & \text{if } U_1 - d_0/\overline{v} - h_0 \leq y_0. \end{cases}$$

The first piece is linearly decreasing, the second piece is convex decreasing and minimized at $l_1 - d_0/\overline{v} - h_0$, the third piece is flat and the fourth one is convex increasing and minimized at $U_1 - d_0/\overline{v} - h_0$.

It follows that the minimum of $C^{(*)a}(h_0 + y_0) - Dy_0$ is obtained at any value of $y_0 \in [(l_1 - d_0/\overline{v} - h_0)^+, (U_1 - d_0/\overline{v} - h_0)^+]$. In particular it is optimal to set $y_0^* = (U_1 - d_0/\overline{v} - h_0)^+$. This means that, if $h_0 > U_1 - d_0/\overline{v}$, we have $y_0^* = 0$ and $C^{*(b)}(h_0) = C^{*(a)}(h_0) = g_0\left(v^u_{0,1}(h_0), h_0\right)$. Otherwise, we have $y_0^* = U_1 - d_0/\overline{v} - h_0$ and $C^{*(b)}(h_0) = C^{*(a)}(U_1 - d_0/\overline{v}) - D\left(U_1 - d_0/\overline{v} - h_0\right) = Dh_0 + g_0\left(\overline{v}, U_1 - d_0/\overline{v}\right)$. □

Lemma 5.2 shows that, in order to minimize cost in a one-arc problem when the driver is paid from the start of service at the origin location, it is either optimal to (i) postpone the vehicle departure from location 0 until it is possible to drive at speed $\overline{v}$ and arrive at the customer location exactly at the upper time window limit or (ii) arrive exactly at the effective upper time window limit. In particular, the optimal speed in the optimal solution cannot be less than $\overline{v}$, which is the speed which minimizes the sum of the labor and emissions costs.

5.3.2 Dynamic programming formulation

In this section we show that the general problem with n arcs can be written as a dynamic program. First we extend the notation defined in the previous section.

For $i = 0, \ldots, n-1$ and $k = i+1, \ldots, n$, we define:

$$v^u_{i,k}(w_i) = \begin{cases} \frac{\sum_{j=i}^{k-1} d_j}{U_k - \sum_{j=i+1}^{k-1} h_j - w_i} & \text{if } U_k > \sum_{j=i+1}^{k-1} h_j + w_i \\ \infty & \text{otherwise} \end{cases}$$

and

$$v^l_{i,k}(w_i) = \begin{cases} \frac{\sum_{j=i}^{k-1} d_j}{l_k - \sum_{j=i+1}^{k-1} h_j - w_i} & \text{if } l_k > \sum_{j=i+1}^{k-1} h_j + w_i \\ \infty & \text{otherwise.} \end{cases}$$

Here, $v^l_{i,k}(w_i)$ and $v^u_{i,k}(w_i)$ correspond to the speeds required to leave location i at time w_i and arrive at location k at time l_k and U_k respectively, in the absence of time windows at the intermediate locations $i+1, \ldots, k-1$, that is if $l_j = 0$ and $u_j = \infty$ for $j = i+1, \ldots, k-1$. An infinite value means that doing so is not possible.

Let $V_i(w_i)$ be the i-th location value function which is equal to the minimum cost on arcs i to $n-1$ given that the vehicle departs location i at time w_i. We have $V_n(w) = 0$ for all $w \geq 0$ and for $i = 0, ..., n-1$, we have:

$$\begin{aligned} & V_i(w_i) = \\ = \quad & \min_{\substack{v^{min} \leq v_i \leq v^{max} \\ w_i + d_i/v_i \leq U_{i+1}}} \left\{ g_i(v_i; w_i) + V_{i+1}\left(\max\left\{ w_i + \frac{d_i}{v_i}, l_{i+1} \right\} + h_{i+1} \right) \right\} \\ = \quad & \min_{v_i \in \left[\max\{v^u_{i,i+1}(w_i), v^{min}\}, v^{max} \right]} \left\{ g_i(v_i; w_i) + V_{i+1}\left(\max\left\{ w_i + d_i/v_i, l_{i+1} \right\} + \right.\right. \\ & \left.\left. + \; h_{i+1} \right) \right\}. \end{aligned}$$

Given Lemma 5.2, when the driver is paid from the start of the planning horizon, we have $C^{(a)*} = Dh_0 + V_0(h_0)$ so solving problem $(P2)$ amounts to calculating $V_0(h_0)$. When the driver is paid from the start of service at the origin location, we have $C^{(b)*} = Dh_0 + \min_{y_0 \in [0, \hat{w}_0 - h_0]} V_0(y_0 + h_0)$.

We first show that the value function has a piecewise structure with at most 4 *pieces*. Let $G_{i,k}(v; w_i) = \sum_{j=i}^{k-1} g_j(v, w_j)$ where $w_j = \max\left\{ w_{j-1} + d_{j-1}/v, l_j \right\} + h_j$ for $j = i+1, \ldots, k-1$ which corresponds to the the cost of driving from location i to location k at speed v, leaving location i at time w_i assuming no voluntary pre-service and no post-service waiting time, i.e., $y_j = 0$ for $j = i+1, \ldots, k$ and $z_j = 0$ for $j = i, \ldots, k-1$. If V_i has exactly 4 pieces, there exist values $p_i, k_i \in \{i+1, \ldots, n\}$ such that, it can be written as follows for $w_i \in [0, \hat{w}_i]$:

$$V_i(w_i) = \begin{cases} G_{i,p_i}(\underline{v}; w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } 0 \leq w_i \leq b^1 \\ G_{i,p_i}(v^l_{i,p_i}(w_i); w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \leq w_i \leq b^2 \\ G_{i,n}(\overline{v}; w_i) & \text{if } b^2 \leq w_i \leq b^3 \\ G_{i,k_i}(v^u_{i,k_i}(w_i); w_i) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^3 \leq w_i \leq \hat{w}_i, \end{cases} \tag{5.3}$$

where breakpoint b^1 is such that $\underline{v} = v^l_{i,p_i}(b^1)$, b^2 is such that $v^l_{i,p_i}(b^2) = \overline{v}$ and b^3 is such that $\overline{v} = v^u_{i,k_i}(b^3)$.

Some of the pieces in (5.3) may be missing; for example, if the third piece is missing, we can write $V_i(w_i)$ as:

$$V_i(w_i) = \begin{cases} G_{i,p_i}(\underline{v}; w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } 0 \leq w_i \leq b^1 \\ G_{i,p_i}(v^l_{i,p_i}(w_i); w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \leq w_i \leq b^2 \\ G_{i,k_i}(v^u_{i,k_i}(w_i); w_i) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^2 \leq w_i \leq \hat{w}_i, \end{cases}$$

where breakpoint b^1 is such that $\underline{v} = v^l_{i,p_i}(b^1)$ and b^2 is such that $v^l_{i,p_i}(b^2) = v^u_{i,k_i}(b^2)$.

Proposition 5.3 *For $i = 0, \ldots, n-1$, $V_i(w_i)$ is continuous in w_i and has the piecewise structure described above.*

Proof: For use in this proof, we define the following threshold values:

$$\begin{aligned} \overline{w}^u_{i,k} &= U_k - \sum_{j=i}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j, \\ \overline{w}^l_{i,k} &= l_k - \sum_{j=i}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j, \\ \underline{w}^l_{i,k} &= l_k - \sum_{j=i}^{k-1} d_j/\underline{v} - \sum_{j=i+1}^{k-1} h_j. \end{aligned}$$

and the following speed values:

$$\begin{aligned} \overline{v}^u_k(w_i) &= \frac{d_j}{U_k - \sum_{j=i+1}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j - w_i} \\ \overline{v}^l_k(w_i) &= \frac{d_j}{l_k - \sum_{j=i+1}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j - w_i} \end{aligned}$$

$$\underline{v}_k^l(w_i) \quad = \quad \frac{d_n}{l_k - \sum_{i=n+1}^{k-1} d_j/\underline{v} - \sum_{i=n+1}^{k-1} h_j - w_n}.$$

To simplify the exposition of the proof we assume, without loss of generality, that $A = 0$. The proof is by induction. First we show the structure is true for $V_{n-1}(w_{n-1})$. For $w_{n-1} \in [0, \hat{w}_{n-1}]$, we have:

$$\begin{aligned} V_{n-1}(w_{n-1}) &= \min_{v_{n-1} \in \left[\max\{v_{n-1,n}^u(w_{n-1}), v^{min}\}, v^{max}\right]} g_{n-1}(v_{n-1}; w_{n-1}) \\ &= \begin{cases} g_{n-1}(\underline{v}; w_{n-1}) & \text{if } 0 \le w_{n-1} \le \underline{w}_{n-1,n}^l \\ g_{n-1}(v_{n-1,n}^l(w_{n-1}); w_{n-1}) & \text{if } \underline{w}_{n-1,n}^l \le w_{n-1} \le \overline{w}_{n-1,n}^l \\ g_{n-1}(\overline{v}; w_{n-1}) & \text{if } \overline{w}_{n-1,n}^l \le w_{n-1} \le \overline{w}_{n-1,n}^u \\ g_{n-1}(v_{n-1,n}^u(w_{n-1}); w_{n-1}) & \text{if } \overline{w}_{n-1,n}^u \le w_{n-1} \le \hat{w}_{n-1} \end{cases} \end{aligned}$$

Since $v_{n-1,n}^l(\underline{w}_{n-1,n}^l) = \underline{v}$, $v_{n-1,n}^l(\overline{w}_{n-1,n}^l) = \overline{v}$ and $v_{n-1,n}^u(\overline{w}_{n-1,n}^u) = \overline{v}$, we can write it as:

$$= \begin{cases} G_{n-1,n}(\underline{v}; w_{n-1}) + V_n(l_n + h_n) & \text{if } 0 \le w_{n-1} \le b^1 \\ G_{n-1,n}(v_{n-1,n}^l(w_{n-1}); w_{n-1}) + V_n(l_n + h_n) & \text{if } b^1 \le w_{n-1} \le b^2 \\ G_{n-1,n}(\overline{v}; w_{n-1}) & \text{if } b^2 \le w_{n-1} \le b^3 \\ G_{n-1,n}(v_{n-1,n}^u(w_{n-1}); w_{n-1}) + V_n(U_n + h_n) & \text{if } b^3 \le w_{n-1} \le \hat{w}_{n-1} \end{cases}$$

This corresponds to (5.3) with $p_{n-1} = k_{n-1} = n$, $b^1 = \underline{w}_{n-1,n}^l$, $b^2 = \overline{w}_{n-1,n}^l$ and $b^3 = \overline{w}_{n-1,n}^u$. Hence, the structure holds for $V_{n-1}(w_{n-1})$. Now let us assume the structure holds for $V_i(w_i)$ and prove that it holds also for $V_{i-1}(w_{i-1})$. We have:

$$\begin{aligned} & \mathcal{L}_{i-1}(v_{i-1}; w_{i-1}) \\ = \; & g_{i-1}(v_{i-1}; w_{i-1}) + V_i(w_i) \\ = \; & \begin{cases} B\left(d_{i-1}/v_{i-1}\right) + C d_{i-1} v_{i-1}^2 \\ +D\left(d_{i-1}/v_{i-1} + h_i\right) \\ +V_i\left(w_{i-1} + d_{i-1}/v_{i-1} + h_i\right) & \text{if } v_{i-1} \le v_{i-1,i}^l(w_{i-1}) \\ B\left(d_{i-1}/v_{i-1}\right) + C d_{i-1} v_{i-1}^2 \\ +D(l_i - w_{i-1} + h_i) + V_i(l_i + h_i) & \text{if } v_{i-1,i}^l(w_{i-1}) \le v_{i-1} \end{cases} \end{aligned} \tag{5.4}$$

Using the induction hypothesis, and assuming that V_i has all 4 pieces from (5.3), we can write:

$$V_i\left(w_{i-1}+d_{i-1}/v_{i-1}+h_i\right)=$$

$$=\begin{cases} G_{i,p_i}(\underline{v}, w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i) & \\ +V_{p_i}(l_{p_i}+h_{p_i}) & \text{if } w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i \leq b^1 \\ G_{i,p_i}(v^l_{i,p_i}(w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i), w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i) & \\ +V_{p_i}(l_{p_i}+h_{p_i}) & \text{if } b^1 \leq w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i \leq b^2 \\ G_{i,n}(\overline{v}, w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i) & \text{if } b^2 \leq w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i \leq b^3 \\ G_{i,k_i}(v^u_{i,k_i}(w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i), w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i) & \\ +V_{k_i}(U_{k_i}+h_{k_i}) & \text{if } b^3 \leq w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i \end{cases}$$

$$=\begin{cases} B(U_{k_i}-w_{i-1}-\frac{d_{i-1}}{v_{i-1}}-\sum_{j=i}^{k_i-1}h_j) & \\ +C\sum_{j=i}^{k_i-1}d_j\left(v^u_{i,k_i}\left(w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i\right)\right)^2 & \\ +D(U_{k_i}-w_{i-1}-\frac{d_{i-1}}{v_{i-1}}-h_i+h_{k_i}) & \\ +V_{k_i}(U_{k_i}+h_{k_i}) & \text{if } v_{i-1} \leq \overline{v}^u_{i-1,k_i}(w_{i-1}) \\ B\left(\sum_{j=i}^{n-1}\frac{d_j}{\overline{v}}\right)+C\left(\sum_{j=i}^{n-1}d_j\right)\overline{v}^2 & \text{if } \overline{v}^u_{i-1,k_i}(w_{i-1}) \leq v_{i-1} \\ +D\left(\sum_{j=i}^{n-1}\frac{d_j}{\overline{v}}+\sum_{j=i+1}^{n}h_j\right) & \text{and } v_{i-1} \leq \overline{v}^l_{i-1,p_i}(w_{i-1}) \\ B(l_{p_i}-w_{i-1}-\frac{d_{i-1}}{v_{i-1}}-\sum_{i=i}^{p_i-1}h_j) & \\ +C\sum_{j=i}^{p_i-1}d_j\left(v^l_{i,p_i}\left(w_{i-1}+\frac{d_{i-1}}{v_{i-1}}+h_i\right)\right)^2 & \\ +D(l_{p_i}-w_{i-1}-\frac{d_{i-1}}{v_{i-1}}-h_i+h_{p_i}) & \text{if } \overline{v}^l_{i-1,p_i}(w_{i-1}) \leq v_{i-1} \\ +V_{p_i}(l_{p_i}+h_{p_i}) & \text{and } v_{i-1} \leq \underline{v}^l_{i-1,p_i}(w_{i-1}) \\ B\sum_{j=i}^{p_i-1}\frac{d_j}{\underline{v}}+C\sum_{j=i}^{p_i-1}d_j\underline{v}^2 & \\ +D(l_{p_i}-w_{i-1}-\frac{d_{i-1}}{v_{i-1}}-h_i+h_{p_i}) & \\ +V_{p_i}(l_{p_i}+h_{p_i}) & \text{if } \underline{v}^l_{i-1,p_i}(w_{i-1}) \leq v_{i-1} \end{cases} \tag{5.5}$$

The resulting expression for $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ depends on how the values of the breakpoints in (5.5) compare to the breakpoint in (5.4). For the sake of brevity, we consider only one case in this proof, such that $\underline{v}^l_{i-1,p_i}(w_{i-1}) \leq v^l_{i-1,i}(w_{i-1})$ and $v^u_{i-1,i}(w_{i-1}) \leq \overline{v}^u_{i-1,k_i}(w_{i-1})$. This case is the one which results in the greatest number of breakpoints in the expression for $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$. All other cases can be proven in a similar way. In this case, we have:

$$
\mathcal{L}_{i-1}(v_{i-1};w_{i-1}) =
$$

$$
= \begin{cases}
B\left(U_{k_i} - w_{i-1} - \sum_{j=i}^{k_i-1} h_j\right) + Cd_{i-1}v_{i-1}^2 & \\
+C\left[\sum_{j=i}^{k_i-1} d_j \left(v_{i,k_i}^u\left(w_{i-1} + \frac{d_{i-1}}{v_{i-1}} + h_i\right)\right)^2\right] & \\
+D\left(U_{k_i} - w_{i-1} + h_{k_i}\right) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } v_{i-1} \leq \overline{v}_{i-1,k_i}^u(w_{i-1}) \\
B\left(\sum_{j=i}^{n} \frac{d_j}{\overline{v}} + \frac{d_{i-1}}{v_{i-1}}\right) + Cd_{i-1}v_{i-1}^2 & \\
+C\left[\left(\sum_{j=i}^{n} d_j\right)\overline{v}^2\right] & \text{if } \overline{v}_{i-1,k_i}^u(w_{i-1}) \leq v_{i-1} \\
+D\left(\sum_{j=i}^{n-1} \frac{d_j}{\overline{v}} + \frac{d_{i-1}}{v_{i-1}} + \sum_{j=i}^{n} h_j\right) & \text{and } v_{i-1} \leq \overline{v}_{i-1,p_i}^l(w_{i-1}) \\
B\left(l_{p_i} - w_{i-1} - \sum_{i=i}^{p_i-1} h_i\right) + Cd_{i-1}v_{i-1}^2 & \\
+C\left[\sum_{j=i}^{p_i-1} d_j \left(v_{i,p_i}^l\left(w_{i-1} + \frac{d_{i-1}}{v_{i-1}} + h_i\right)\right)^2\right] & \text{if } \overline{v}_{i-1,p_i}^l(w_{i-1}) \leq v_{i-1} \\
+D\left(l_{p_i} - w_{i-1} + h_{p_j}\right) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{and } v_{i-1} \leq \underline{v}_{i-1,p_i}^l(w_{i-1}) \\
B\left(\sum_{j=i}^{p_i-1} \frac{d_j}{\underline{v}} + \frac{d_{i-1}}{v_{i-1}}\right) + Cd_{i-1}v_{i-1}^2 & \\
+C\left[\sum_{j=i}^{p_i-1} d_j \underline{v}^2\right] & \text{if } \underline{v}_{i-1,p_i}^l(w_{i-1}) \leq v_{i-1} \\
+D(l_{p_i} - w_{i-1} + h_{p_i}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{and } v_{i-1} \leq v_{i-1,i}^l(w_{i-1}) \\
B\frac{d_{i-1}}{v_{i-1}} + Cd_{i-1}v_{i-1}^2 + D(l_i - w_{i-1} + h_i) & \\
+V_i(l_i + h_i) & \text{if } v_{i-1} \geq v_{i-1,i}^l(w_{i-1})
\end{cases} \tag{5.6}
$$

All the pieces in this expression are convex in v_{i-1}. Moreover, the first piece reaches a minimum at $v_{i-1,k_i}^u(w_{i-1})$. The second piece reaches a minimum at $\overline{v}$. The third piece reaches a minimum at $v_{i-1,p_i}^l(w_{i-1})$. And the last 2 pieces reach a minimum at $\underline{v}$. The shape of the $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ function (and therefore where its minimum point is located) depends on how the values of $\underline{v}$ and $\overline{v}$ compare to the breakpoints in (5.3.2). We list the possible cases next and show that, in each of them, the $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ function accepts a unique minimum.

- If $\underline{v} \geq \underline{v}_{i-1,i}^l(w_{i-1})$, which is equivalent to $w_{i-1} \leq \underline{w}_{i-1,i}^l$, $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ is decreasing in the first 4 pieces and achieves a minimum at $\underline{v}$ in the last piece.
- If $\underline{v}_{i-1,p_i}^l(w_{i-1}) \leq \underline{v} \leq \underline{v}_{i-1,i}^l(w_{i-1})$, which is equivalent to $\underline{w}_{i-1,i}^l \leq w_{i-1} \leq \underline{w}_{i-1,p_i}^l$, $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ is decreasing in the first 3 pieces, reaches a minimum at $\underline{v}$ in the fourth piece and is increasing in the last piece.
- If $\underline{v} \leq \underline{v}_{i-1,p_i}^l(w_{i-1})$ and $\overline{v} \geq \overline{v}_{i-1,p_i}^l(w_{i-1})$, which is equivalent to $\underline{w}_{i-1,p_i}^l \leq w_{i-1} \leq \overline{w}_{i-1,p_i}^l$, $\mathcal{L}_{i-1}(v_{i-1};w_{i-1})$ is decreasing in the first 2 pieces, reaches

a minimum at $v^l_{i-1,p_i}(w_{i-1})$ in the third piece and is increasing in the last 2 pieces.

- If $\overline{v}^u_{i-1,k_i}(w_{i-1}) \leq \overline{v} \leq \overline{v}^l_{i-1,p_i}(w_{i-1})$, which is equivalent to $\overline{w}^l_{i-1,p_i} \leq w_{i-1} \leq \overline{w}^u_{i-1,k_i}$, $\mathcal{L}_{i-1}(v_{i-1}; w_{i-1})$ is decreasing in the first piece, reaches a minimum at $\overline{v}$ in the second piece and is increasing in the last 3 pieces.

- If $\overline{v} \leq \overline{v}^u_{i-1,k_i}(w_{i-1})$, which is equivalent to $w_{i-1} \geq \overline{w}^u_{i-1,k_i}$, $\mathcal{L}_{i-1}(v_{i-1}; w_{i-1})$ reaches a minimum in the first piece and is increasing in the last 4 pieces. The minimum in the first piece is achieved at $v^u_{i-1,k_i}(w_{i-1})$ if $v^u_{i-1,k_i}(w_{i-1}) \geq v^u_{i-1,i}(w_{i-1})$ (i.e., if it is feasible given the lower bound on v_{i-1} given in (5.3)) and at $v^u_{i-1,i}(w_{i-1})$ otherwise. In what follows, we assume that the maximum is reached at $v^u_{i-1,k_i}(w_{i-1})$, the other case can be handled in a similar way.

After some simplifications, we obtain:

$$
\begin{aligned}
& V_{i-1}(w_{i-1}) = \\
= & \begin{cases}
B\frac{d_{i-1}}{\underline{v}} + Cd_{i-1}\underline{v}^2 + D(l_i - w_{i-1} + h_i) & \\
+V_i(l_i + h_i) & \text{if } w_{i-1} \leq \underline{w}^l_{i-1,i} \\
B\sum_{j=i-1}^{p_i-1} \frac{d_j}{\underline{v}} + C\left(\sum_{j=i-1}^{p_i-1} d_j\right)\underline{v}^2 & \\
+D(l_{p_i} - w_{i-1} + h_{p_i}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } \underline{w}^l_{i-1,i} \leq w_{i-1} \leq \underline{w}^l_{i-1,p_i} \\
B\left(l_{p_i} - w_{i-1} - \sum_{i=i}^{p_i-1} h_i\right) & \\
+C\sum_{j=i-1}^{p_i-1} d_j \left(v^l_{i-1,p_i}(w_{i-1})\right)^2 & \\
+D\left(l_{p_i} - w_{i-1} + h_{p_i}\right) + V_{p_i}\left(l_{p_i} + h_{p_i}\right) & \text{if } \underline{w}^l_{i-1,p_i} \leq w_{i-1} \leq \overline{w}^l_{i-1,p_i} \\
B\left(\sum_{j=i-1}^{n} \frac{d_j}{\overline{v}}\right) + C\left(\sum_{j=i-1}^{n} d_j\right)\overline{v}^2 & \\
+D\left(\sum_{j=i-1}^{n-1} \frac{d_j}{\overline{v}} + \sum_{j=i}^{n} h_j\right) & \text{if } \overline{w}^l_{i-1,p_i} \leq w_{i-1} \leq \overline{w}^u_{i-1,k_i} \\
B\left(U_{k_i} - w_{i-1} - \sum_{j=i}^{k_i-1} h_j\right) & \\
+C\sum_{j=i-1}^{k_i-1} d_j \left(v^u_{i-1,k_i}(w_{i-1})\right)^2 & \\
+D\left(U_{k_i} - w_{i-1} + h_{k_i}\right) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } w_{i-1} \geq \overline{w}^u_{i-1,k_i}
\end{cases}
\end{aligned}
$$

Using the induction hypothesis to expand $V_i(l_i + h_i)$ in the first piece (based on the fact that $l_i + h_i \leq l_{p_i} - \sum_{j=i}^{p_i-1} \frac{d_j}{\underline{v}} - \sum_{j=i+1}^{p_i-1} h_j = \underline{w}^l_{i,p_i}$) , we get:

$$
\begin{aligned}
&V_{i-1}(w_{i-1}) = \\
&= \begin{cases}
B\sum_{j=i-1}^{p_i-1}\frac{d_j}{\underline{v}} + C\left(\sum_{j=i-1}^{p_i-1} d_j\right)\underline{v}^2 & \\
+D(l_{p_i} - w_{i-1} + h_{p_i}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } 0 \le w_{i-1} \le \underline{w}^l_{i-1,p_i} \\
B\left(l_{p_i} - w_{i-1} - \sum_{i=i}^{p_i-1} h_i\right) & \\
+C\left(\sum_{j=i-1}^{p_i-1} d_j\right)\left(v^l_{i-1,p_i}(w_{i-1})\right)^2 & \\
+D\left((l_{p_i} - w_{i-1} + h_{p_i}\right) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } \underline{w}^l_{i-1,i} \le w_{i-1} \le \underline{w}^l_{i-1,p_i} \\
B\left(\sum_{j=i-1}^{n}\frac{d_j}{\overline{v}}\right) + C\left(\sum_{j=i-1}^{n} d_j\right)\overline{v}^2 & \\
+D\left(\sum_{j=i-1}^{n}\frac{d_j}{\overline{v}} + \sum_{j=i}^{n+1} h_j\right) & \text{if } \overline{w}^l_{i-1,p_i} \le w_{i-1} \le \overline{w}^u_{i-1,k_i} \\
B\left(U_{k_i} - w_{i-1} - \sum_{j=i}^{k_i-1} h_j\right) & \\
+C\left(\sum_{j=i-1}^{k_i-1} d_j\right)\left(v^u_{i-1,k_i}(w_{i-1})\right)^2 & \\
+D\left(U_{k_i} - w_{i-1} + h_{k_i}\right) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } w_{i-1} \ge \overline{w}^u_{i-1,k_i}.
\end{cases} \\
&= \begin{cases}
G_{i-1,p_i}(\underline{v}, w_{i-1}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } 0 \le w_{i-1} \le b^1 \\
G_{i-1,p_i}(v^l_{i-1,p_i}(w_{i-1}), w_{i-1}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \le w_{i-1} \le b^2 \\
G_{i-1,p_i}(\overline{v}, w_{i-1}) & \text{if } b^2 \le w_{i-1} \le b^3 \\
G_{i-1,p_i}(v^u_{i-1,k_i}(w_{i-1}), w_{i-1}) + V_{k_i}(U_{k_i} + h_{p_i}) & \text{if } w_{i-1} \ge b^3.
\end{cases}
\end{aligned}
$$

This corresponds to the structure from (5.3) where $k_{i-1} = k_i$, $p_{i-1} = p_i$, $b^1 = \underline{w}^l_{i-1,p_i}$, $b^2 = \underline{w}^l_{i-1,p_i}$ and $b^3 = \overline{w}^u_{i-1,k_i}$. □

Our next result establishes that the value function achieves a minimum either at a single point or over an interval where it is constant.

Proposition 5.4 *For $i = 0, \ldots, n-1$, the set of minimizers of $V_i(w_i)$ on $[0, \hat{w}_i]$ is a convex set.*

PROOF:

From Proposition 5.3 there are four cases: (i) only p_i exists, (ii) only k_i exists, (iii) both p_i and k_i exist, (iv) none of them exists. In case (i) the $V_i(w_i)$ function has at most three pieces and can be written as:

$$
V_i(w_i) = \begin{cases}
G_{i,p_i}(\underline{v}, w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } w_i \le b^1 \\
G_{i,p_i}(v^l_{i,p_i}(w_i), h_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \le w_i \le b^2 \\
G_{i,n}(\overline{v}, w_i) & \text{if } b^2 \le w_i \le \hat{w}_i
\end{cases} \tag{5.7}
$$

where $b_1 = \underline{w}^l_{i,p_i}$ and $b_2 = \overline{w}^l_{i,p_i}$. The first piece is decreasing linearly in w_i, the second one is convex decreasing (since it is minimized at $b_2 = \overline{w}^l_{i,p_i}$) and the third

one is constant. Therefore, the set of minimizers of V_i is $[\overline{w}^l_{i,p_i}, \hat{w}_i]$.
In case (ii) the $V_i(w_i)$ function has at most two pieces:

$$V_i(w_i) = \begin{cases} G_{i,n}(\overline{v}, w_i) & \text{if } w_i \leq b^1 \\ G_{i,k_i}(v^u_{i,k_i}(w_i), w_i) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^1 \leq w_i \leq \hat{w}_i \end{cases} \tag{5.8}$$

where $b_1 = \overline{w}^u_{i,k_i}$. The first piece is constant and the second one is convex increasing (since it is minimized at $b_1 = \overline{w}^u_{i,k_i}$). Therefore, the set of minimizers of V_i is $[0, \overline{w}^u_{i,k_i}]$.
In case (iii) we distinguish two cases: (a) $\overline{w}^l_{i,p_i} \leq \overline{w}^u_{i,k_i}$, (b) $\overline{w}^l_{i,p_i} > \overline{w}^u_{i,k_i}$.
In case (iii.a) the $V_i(w_i)$ function has at most four pieces:

$$V_i(w_i) = \begin{cases} G_{i,p_i}(\underline{v}, w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } w_i \leq b^1 \\ G_{i,p_i}(v^l_{i,p_i}(w_i), w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \leq w_i \leq b^2 \\ G_{i,n}(\overline{v}, w_i) & \text{if } b^2 \leq w_i \leq b^3 \\ G_{i,k_i}(v^u_{i,k_i}(w_i), w_i) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^3 \leq w_i \leq \hat{w}_i \end{cases} \tag{5.9}$$

where $b_1 = \underline{w}^l_{i,p_i}$, $b_2 = \overline{w}^l_{i,p_i}$, and $b_3 = \overline{w}^u_{i,k_i}$. The first piece is decreasing linearly in w_i, the second one is convex decreasing (since it is minimized at $b_2 = \overline{w}^l_{i,p_i}$), the third one is constant and the last one is convex increasing (since its is minimized at $b_3 = \overline{w}^u_{i,k_i}$). Therefore, the set of minimizers of V_i is $[\overline{w}^l_{i,p_i}, \overline{w}^u_{i,k_i}]$.

In case (iii.b), $V_i(w_i)$ function has at most three pieces:

$$V_i(w_i) = \begin{cases} G_{i,p_i}(\underline{v}, w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } w_i \leq b^1 \\ G_{i,p_i}(v^l_{i,p_i}(w_i), w_i) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \leq w_i \leq b^2 \\ G_{i,k_i}(v^u_{i,k_i}(w_i), w_i) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^2 \leq w_i \leq \hat{w}_i \end{cases} \tag{5.10}$$

where $b_1 = \underline{w}^l_{i,p_i}$ and $b_2 = \tilde{w}^l_{i,p_i,k_i} = l_{p_i} - \frac{\sum_{j=i}^{p_i-1} d_j}{v^u_{p_i,k_i}(l_{p_i}+h_{p_i})} - \sum_{j=i+1}^{p_i-1} h_j$ if $k_i > p_i$, otherwise $b_2 = \tilde{w}^u_{i,k_i,p_i} = U_{k_i} - \frac{\sum_{j=i}^{k_i-1} d_j}{v^l_{k_i,p_i}(U_{k_i}+h_{k_i})} - \sum_{j=i+1}^{k_i-1} h_j$. The first piece is decreasing linearly in w_i, the second one is convex and minimized at $\overline{w}^l_{i,p_i}$, and the last piece is convex and minimized at $\overline{w}^u_{i,k_i}$. We distinguish two sub-cases: (iii.b.1) $k_i > p_i$, and (iii.b.2) $k_i < p_i$. In case (iii.b.1), given $\overline{w}^l_{i,p_i} > \overline{w}^u_{i,k_i}$ it follows $\overline{w}^u_{i,k_i} < \overline{w}^l_{i,p_i} < \tilde{w}^l_{i,p_i,k_i}$. Hence, the first piece is decreasing, the second one is minimized at $\overline{w}^l_{p_i}$ and the third one is increasing. Therefore, in this case, V_i has a unique minimum located at $\overline{w}^l_{p_1}$.
In case (iii.b.2), given $\overline{w}^l_{i,p_i} > \overline{w}^u_{i,k_i}$ it follows that $\tilde{w}^u_{i,k_i,p_i} < \overline{w}^u_{i,k_i} < \overline{w}^l_{i,p_i}$. Hence, the first piece is decreasing, the second one is decreasing (since its minimum is to

the right), and the last one is minimized at $\overline{w}^u_{i,k_i}$. Therefore, in this case, V_i has a unique minimum located at $\overline{w}^u_{i,k_i}$.

In case (iv), which can only occur if $l_j = 0$ and $u_j = \infty$ for $j = i+1, \ldots, n$, $V_i(w_i)$ function has only one piece and is equal to $G_{i,n}(\overline{v}, w_i)$ for $w_i \in [0, \hat{w}_i)$, which is constant in w_i. Therefore, the set of minimizers of V_i is $[0, \hat{w}_i)$. □

Let $\tilde{w}_0$ denote a value which minimizes $V_0(w_0)$ on $[0, \hat{w}_0]$. When the driver is paid from the start of the service at the origin location, it is optimal to delay the arrival of the driver to the origin location if the length of service at the origin, i.e., h_0, is less or equal than $\tilde{w}_0$. Otherwise, it is optimal for the driver to start service at the origin location at the start of the planning horizon. In other words, we set $y_0^* = (\tilde{w}_0 - h_0)^+$. Proposition 5.6 in §5.A provides a more detailed characterization of the optimal solution when the driver is paid from the start of service at the origin location.

Direct observation of the piecewise structure of the value function gives us the following important corollary:

Corollary 5.1 *Given a departure time from location i equal to w_i, the optimal speed on arc i can only take one of the following values: (i) $\underline{v}$, (ii) $v^l_{i,p_i}(w_i)$ for some $p_i \in \{i+1, \ldots, n\}$, (iii) $\overline{v}$, or (iv) $v^u_{i,k_i}(w_i)$ for some $k_i \in \{i+1, \ldots, n\}$.*

- *In case (i), there exists a location $q_i \in \{i+1, \ldots, n\}$ such that the vehicle keeps the same speed on arcs i to $q_i - 1$, arrives at locations $i+1, \ldots, q_i - 1$ before their effective upper time window limit then reaches location q_i before or exactly at time l_{q_i};*

- *In case (ii), the vehicle keeps the same speed on arcs i to $p_i - 1$, arrives at locations $i+1, \ldots, p_i - 1$ within their effective time window then reaches location p_i exactly at time l_{p_i}.*

- *In case (iii) the vehicle keeps the same speed on all remaining arcs, i.e., i to $n-1$, and arrives at each remaining location within their effective time window;*

- *In case (iv), the vehicle keeps the same speed on arcs i to $k_i - 1$, arrives at locations $i+1, \ldots, k_i - 1$ within their effective time window, then reaches location k_i exactly at time U_{k_i}.*

PROOF: (5.3) can be rewritten as:

$$V_i(w_i) =$$

$$= \begin{cases} B\left(\sum_{j=i}^{p_i-1} \frac{d_j}{\underline{v}}\right) + C\sum_{j=i}^{p_i-1} d_j \underline{v}^2 & \\ +D(l_{p_i} - w_i + h_{p_i}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } w_i \leq b^1 \\ B\left(l_{p_i} - w_i - \sum_{j=i+1}^{p_i-1} h_i\right) + C\sum_{j=i}^{p_i-1} d_j \left(v_{i,p_i}^l(w_i)\right)^2 & \\ +D(l_{p_i} - w_i + h_{p_i}) + V_{p_i}(l_{p_i} + h_{p_i}) & \text{if } b^1 \leq w_i \leq b^2 \\ B\left(\sum_{j=i}^{n} \frac{d_j}{\overline{v}}\right) + C\sum_{j=i}^{n} d_j \overline{v}^2 & \\ +D\left(\sum_{j=i}^{n} \frac{d_j}{\overline{v}} + \sum_{j=i+1}^{n+1} h_j\right) & \text{if } b^2 \leq w_i \leq b^3 \\ B\left(U_{k_i} - w_i - \sum_{j=i+1}^{k_i-1} h_j\right) + C\sum_{j=i}^{k_i-1} d_j \left(v_{i,k_i}^u(w_i)\right)^2 & \\ +D\left(U_{k_i} - w_i + h_{k_i}\right) + V_{k_i}(U_{k_i} + h_{k_i}) & \text{if } b^3 \leq w_i \leq \hat{w}_i. \end{cases} \tag{5.11}$$

The proof follows directly from a careful analysis of the terms in (5.11). From the definition of the g_i function, we know that the multiplier of the B constant corresponds to the driving time and that the multiplier of the D constant is the sum of the the driving time, service time and pre-service waiting time. Hence the difference between the multipliers of the D and B constant corresponds to the sum of the service time and pre-service waiting time.

Let us first look at the first piece of (5.11). The first three terms correspond to the cost of driving on arcs i to $p_i - 1$ at speed $\underline{v}$, possibly with some pre-service waiting time at these locations. The last term, i.e. $V_{p_i}(l_{p_i} + h_{p_i})$ is, by definition, the optimal cost of serving locations $p_i, \ldots, n$ when leaving location p_i at time $l_{p_i} + h_{p_i}$, which means that the vehicle arrived at location p_i before or exactly at time l_{p_i}. Hence this is equivalent to this statement with $p_i = q_i$.

Next let us look at the second piece of (5.11). The difference between the multiplier of D and B is equal to $\sum_{j=i+1}^{p_i} h_i$, which means that there is no pre-service waiting time at locations $i+1$ to p_i. Hence, the first three terms correspond to the cost of driving on arcs i to $p_i - 1$ at speed $v_{i,p_i}^l(w_i)$ without any pre-service waiting time. The last term, i.e. $V_{p_i}(l_{p_i} + h_{p_i})$ is, by definition, the optimal cost of serving locations $p_i, \ldots, n$ when leaving location p_i at time $l_{p_i} + h_{p_i}$, which means that the vehicle arrived at location p_i exactly at time l_{p_i}.

Next let us look at the third piece of (5.11). The difference between the multiplier of D and B is $\sum_{j=i+1}^{n} h_i$, which means that that there is no pre-service waiting time at locations $i+1$ to n and the vehicle drives on arcs i to $n-1$ at speed $\overline{v}$.

Finally, let us look at the fourth piece of (5.3). The difference between the multiplier of D and B is equal to $\sum_{j=i+1}^{k_i} h_i$, which means that there is no pre-service waiting time at locations $i+1$ to k_i. Hence, the first three terms correspond to the cost of driving on arcs i to $k_i - 1$ at speed $v_{i,k_i}^u(w_i)$ without any pre-service waiting time. The last term, i.e. $V_{k_i}(U_{k_i} + h_{p_i})$ is, by definition, the optimal cost of serving locations $k_i, \ldots, n$ when leaving location k_i at time $U_{p_i} + h_{p_i}$, which means that the vehicle arrived at location k_i exactly at time U_{p_i}.

No other speed value is possible since there are only (a maximum) of 4 pieces in (5.3). □

This result implies that the optimal driving schedule can be broken down into segments, made out of adjacent arcs, on which the vehicle travels at the same speed. Further, at the customer locations which are common to two adjacent segments, the driver always starts service either exactly at the lower time window limit or exactly at the effective upper time window limit (note that arrival at these locations could be before the lower time window, in which case there is positive pre-service waiting time). In contrast, at the final customer location, the driver can start service at anytime within the time window. Because of this property, the problem of finding the optimal solution can be solved by solving a shortest path problem, as we show in the next section.

5.4 Shortest path formulation

In this section we show how to re-cast the problem into a shortest path (SP) problem by building a new network of arcs and nodes. From the original network with $n+1$ locations (see Figure 5.1), we construct an SP network with $2n+2$ nodes. In this SP network, there is one node for the origin location, denoted 0, two nodes for each intermediate location, denoted i_l and i_u for $i = 1, \ldots, n-1$ and three nodes for the final location, denoted n_l, n_u and n. So let V denote the set of nodes in the SP network, i.e., $V = \{0, 1_l, 1_u, 2_l, \ldots, n_l, n_u, n\}$. For $i = 1, \ldots, n$, node i_l corresponds to the event "the vehicle arrives at location i before or at the lower time window limit l_i and, for $i < n$, leaves exactly at time $l_i + h_i$" and node i_u corresponds to the event "the vehicle arrives at location i exactly at the effective upper time window limit U_i and, for $i < n$, leaves exactly at time $U_i + h_i$". Finally, node n corresponds to the event "the vehicle arrives at the final location within its effective time window". In the SP network, there are arcs between every pair of nodes such that the number is strictly increasing (e.g., arcs between 1_u and 2_l and arcs between 0 and 1_l but no arc between 2_u and 2_l and no arc between 2_u and 1_u), plus two extra arcs: (n_l, n) and (n_u, n). Hence the SP network is an acyclic graph. Figure 5.5 shows the SP network with $n = 3$.

An arc in the SP network means that the vehicle travels at a constant speed between the corresponding locations, arriving at each intermediate location within its effective time window and the length of the arc is set equal to the minimum cost of doing so (or infinity if it is not possible). So, for example, an arc between nodes i_l and j_u means that the vehicle leaves location i at time $l_i + h_i$, travels at the same speed on arcs i to $j-1$, arriving at locations k within $[l_k, U_k]$ for

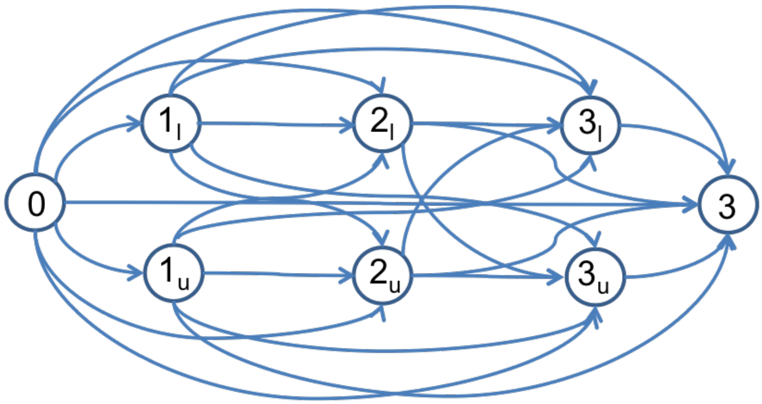


Figure 5.5 Shortest path network

$k = i+1, \ldots, j-1$ then reaches location j exactly at time U_j. Let $T_{i,k}(v,w)$ denote the arrival time into location k when leaving location i at time w, driving on arcs i to $k-1$ at speed v assuming no voluntary pre-service or post-service waiting time, i.e., with $y_j = 0$ for $j = i+1, \ldots, k$, $z_j = 0$ for $j = i, \ldots, k-1$. These values can be computed recursively as $T_{i,i+1} = w + d_i/v$ and $T_{i,j}(v,w) = \max\{T_{i,j-1}(v,w), l_{j-1}\} + h_{j-1} + d_j/v$ for $j = i+2, \ldots, k$. The arcs lengths in the SP network are calculated as follows.

- The length of arcs (n_l, n) and (n_u, n) is zero.
- For $0 < i < j \leq n$, an arc from a node i_l to a node j_l has a length equal to $C(i_l, j_l) = \min\{C^1(i_l, j_l), C^2(i_l, j_l)\}$ where:

$$C^1(i_l, j_l) = \begin{cases} G_{i,j}\left(v^l_{i,j}(l_i+h_i); l_i+h_i\right) & \text{if } T_{i,k}\left(v^l_{i,j}(l_i+h_i); l_i+h_i\right) \in [l_k, U_k] \\ & \text{for } k = i+1, \ldots, j \\ \infty & \text{otherwise,} \end{cases}$$

and

$$C^2(i_l, j_l) = \begin{cases} G_{i,j}\left(\underline{v}; l_i+h_i\right) & \text{if } T_{i,k}\left(\underline{v}; l_i+h_i\right) \leq U_k \\ & \text{for } k = i+1, \ldots, j \text{ and } T_{i,j}\left(\underline{v}; l_i+h_i\right) \leq l_j \\ \infty & \text{otherwise.} \end{cases}$$

- For $0 < i < j \leq n$, an arc from a node i_l to a node j_u has a length equal to

$$C(i_l, j_u) = \begin{cases} G_{i,j}\left(v^u_{i,j}(l_i+h_i); l_i+h_i\right) & \text{if } T_{i,k}\left(v^u_{i,j}(l_i+h_i); l_i+h_i\right) \in [l_k, U_k] \\ & \text{for } k = i+1, \ldots, j \\ \infty & \text{otherwise.} \end{cases}$$

- For $0 < i < j \leq n$, an arc from a node i_u to a node j_l has a length equal to $C(i_u, j_l) = \min\{C^1(i_u, j_l), C^2(i_u, j_l)\}$ where:

$$C^1(i_u, j_l) \quad = \quad \begin{cases} G_{i,j}\left(v^l_{i,j}(U_i+h_i); l_i+h_i\right) & \text{if } T_{i,k}\left(v^l_{i,j}(U_i+h_i); l_i+h_i\right) \in [l_k, U_k] \\ & \text{for } k=i+1,\ldots,j \\ \infty & \text{otherwise,} \end{cases}$$

and

$$C^2(i_u, j_l) \quad = \quad \begin{cases} G_{i,j}\left(\underline{v}; U_i+h_i\right) & \text{if } T_{i,k}\left(\underline{v}; U_i+h_i\right) \leq U_k \\ & \text{for } k=i+1,\ldots,j \text{ and } T_{i,j}\left(\underline{v}; U_i+h_i\right) \leq l_j \\ \infty & \text{otherwise.} \end{cases}$$

- For $0<i<j\leq n$, an arc from a node i_u to a node j_u has a length equal to

$$C(i_u, j_u) \quad = \quad \begin{cases} G_{i,j}\left(v^u_{i,j}(U_i+h_i); U_i+h_i\right) & \text{if } T_{i,k}\left(v^u_{i,j}(U_i+h_i); U_i+h_i\right) \in [l_k, U_k] \\ & \text{for } k=i+1,\ldots,j \\ \infty & \text{otherwise.} \end{cases}$$

- For $i>0$, an arc from node i_l to node n has a length equal to

$$C(i_l, n) \quad = \quad \begin{cases} G_{i,j}\left(\overline{v}; l_i+h_i\right) & \text{if } T_{i,k}\left(\overline{v}; l_i+h_i\right) \in [l_k, U_k] \text{ for } k=i+1,\ldots,n \\ \infty & \text{otherwise.} \end{cases}$$

- For $i>0$, an arc from node i_u to node n has a length equal to

$$C(i_u, n) \quad = \quad \begin{cases} G_{i,j}\left(\overline{v}; U_i+h_i\right) & \text{if } T_{i,k}\left(\overline{v}; U_i+h_i\right) \in [l_k, U_k] \text{ for } k=i+1,\ldots,j \\ \infty & \text{otherwise.} \end{cases}$$

- For $j>0$ an arc from node 0 to node j_l has a length equal to $C(0, j_l) = \min\{C^1(0, j_l), C^2(0, j_l)\}$ where:

$$C^1(0, j_l) \quad = \quad \begin{cases} G_{i,j}\left(v^l_{i,j}(h_0); h_0\right) & \text{if } T_{0,k}\left(v^l_{i,j}(h_0); h_0\right) \in [l_k, U_k] \text{ for } k=1,\ldots,j \\ \infty & \text{otherwise,} \end{cases}$$

and

$$C^2(0, j_l) \quad = \quad \begin{cases} G_{i,j}\left(\underline{v}; h_0\right) & \text{if } T_{0,k}\left(\underline{v}; h_0\right) \leq U_k \text{ for } k=1,\ldots,j \text{ and } T_{0,j}\left(\underline{v}; h_0\right) \leq l_j \\ \infty & \text{otherwise,} \end{cases}$$

if the driver is paid from the beginning of the planning horizon and

$$C(0, j_l) \quad = \quad \begin{cases} G_{0,j}\left(v^l_{0,j}(w_0); w_0\right) & \text{if } T_{0,k}\left(v^l_{0,j}(w_0); w_0\right) \in [l_k, U_k] \text{ for } k=1,\ldots,j \\ \infty & \text{otherwise,} \end{cases}$$

where $w_0 = \max\left\{h_0, l_j - \sum_{k=0}^{j-1} d_k/\overline{v} - \sum_{k=1}^{j-1} h_k\right\}$, if the driver is paid from the start of service at the origin location.

- For $j > 0$, an arc from node 0 to node j_u has a length equal to

$$C(0, j_u) = \begin{cases} G_{0,j}\left(v^u_{0,j}(h_0); h_0\right) & \text{if } T_{0,k}\left(v^u_{0,j}(h_0); h_0\right) \in [l_k, U_k] \text{ for } k = 1, \ldots, j, \\ \infty & \text{otherwise,} \end{cases}$$

  if the driver is paid from the beginning of the planning horizon and

$$C(0, j_u) = \begin{cases} G_{0,j}\left(v^u_{0,j}(w_0); w_0\right) & \text{if } T_{0,k}\left(v^u_{0,j}(w_0); w_0\right) \in [l_k, U_k] \text{ for } k = 1, \ldots, j \\ \infty & \text{otherwise,} \end{cases}$$

  where $w_0 = \max\left\{h_0, U_j - \sum_{k=0}^{j-1} d_k/\overline{v} - \sum_{k=1}^{j-1} h_k\right\}$, if the driver is paid from the start of service at the origin location.

- The arc from node 0 to node n has a length equal to

$$C(0, n) = \begin{cases} G_{0,n}\left(\overline{v}; h_0\right) & \text{if } T_{0,k}\left(\overline{v}; h_0\right) \in [l_k, U_k] \text{ for } k = 1, \ldots, n \\ \infty & \text{otherwise,} \end{cases}$$

  if the driver is paid from the beginning of the planning horizon and

$$C(0, n) = \begin{cases} G_{0,j}\left(\overline{v}; w_0\right) & \text{if } T_{0,k}\left(\overline{v}; w_0\right) \in [l_k, U_k] \text{ for } k = i+1, \ldots, n \\ \infty & \text{otherwise,} \end{cases}$$

  where $w_0 = \max\left\{h_0, l_n - \sum_{k=0}^{n-1} d_k/\overline{v} - \sum_{k=1}^{n-1} h_k\right\}$, if the driver is paid from the start of service at the origin location.

Proposition 5.5 *The problem (P2) reduces to a shortest path from node* 0 *to node n. The minimum cost in (P2) is equal to the length of the shortest path, plus Dh_0.*

Proof: Corollary 5.1 implies that the optimal driving schedule can be broken down into *segments*, made of adjacent arcs, on which the vehicle travels at the same speed. Let us refer to the customers locations between two such segments as *transition locations*. Corollary 5.1 also implies that, at these transition locations, the driver should always start service either exactly at its lower time window limit or exactly at its effective upper time window limit (note that arrival at the location could be before the lower time window, in which case there is positive pre-service waiting time). In contrast, the driver can start service at the final customer location anytime within the time window.

By Proposition 5.1, there is no post-service waiting time at any location hence, it is optimal for the vehicle to leave transition locations directly upon completion of service, that is, at time $l_j + h_j$ if service started at time l_j and at time $U_j + h_j$ if service started at time U_j. These two possibilities are consistent with the definitions of nodes j_l and j_u in the SP network.

Consider a segment from node i to node $j < n$, such that the vehicle leaves node i at time $l_i + h_i$ (or $U_i + h_i$). If the vehicle starts service at node j exactly at time l_j, by Corollary 5.1, there are two possibilities: either the vehicle drove at speed $\underline{v}$ or at speed $v^l_{i,j}(l_i + h_i)$ (or $v^l_{i,j}(U_i + h_i)$) for some $i \in \{0, \dots, j-1\}$ on this segment. The calculation of the lengths for arcs (i_l, j_l) (and (i_u, j_l)) for $0 < i < j < n$ reflect these two possibilities. If the vehicle starts service at node j exactly at time U_j then it must have arrived there driving at speed $v^u_{i,j}(l_i + h_i)$ (or $v^u_{i,j}(U_i + h_i)$) for some $i \in \{0, \dots, j-1\}$ on this segment. The calculation of the lengths for arcs (i_l, j_u) (and (i_u, j_u)) for $0 < i < j < n$ reflects these possibilities.

Now consider a segment from node i to node n, such that the vehicle leaves node i at time $l_i + h_i$ (or $U_i + h_i$). Here there are four possibilities: (i) the vehicle travels at speed $\overline{v}$ and arrives at the final node within its time window, (ii) the vehicle travels at speed $v^l_{i,n}(l_i + h_i)$ (or $v^l_{i,n}(U_i + h_i)$) and arrives at the final node at time l_n, (iii) the vehicle travels at speed $\underline{v}$ and arrives at the final destination at or before time l_n. (iv) the vehicle travels at speed $v^u_{i,n}(l_i + h_i)$ (or $v^u_{i,n}(U_i + h_i)$) and arrives at the final node at time U_n. The calculation of the lengths for arcs (i_l, n) (and (i_u, n)), (i_l, n_l) (and (i_u, n_l)) and (i_l, n_u) (and (i_u, n_u)) account for these four possibilities.

Now consider segments from node 0 to node $j \leq n$. From Proposition 5.1 we get that if the driver is paid from the beginning of the planning horizon the optimal departure time from the origin location is h_0. In this case, the calculation of the length for arcs $(0, j_l)$, $(0, j_u)$, and $(0, n)$ is similar to the previous cases.

In contrast, if the driver is paid from the start of the service at the origin location, we distinguish three cases: (i) the vehicle leaves the origin location and arrives at location j exactly at time l_j, (ii) the vehicle leaves the origin location and arrives at location j exactly at time U_j or (iii) the vehicle leaves the origin location and arrives at location n within the $[l_n, U_n]$ time window. From a careful analysis of the proof of Proposition 5.6 we derive the following results. In Case (i) the optimal departure time from the origin location is $\max\{\overline{w}^l_{0,j}, h_0\}$ and the optimal speed value on arcs 0 to $j-1$ is $v^l_{0,j}(\max\{\overline{w}^l_{0,j}, h_0\})$. The calculation of the length of arcs $(0, j_l)$ reflects this fact. Similarly, in Case (ii) the optimal departure time from the origin location is $\max\{\overline{w}^u_{0,j}, h_0\}$ and the optimal speed value on arcs 0 to $j-1$ is $v^u_{0,j}(\max\{\overline{w}^u_{0,j}, h_0\})$. The calculation of the length of arcs $(0, j_u)$ reflects this fact. In Case (iii) the optimal departure time from the origin location is $\max\{\overline{w}^l_{0,n}, h_0\}$ and optimal speed is $\overline{v}$. The calculation of the length of arcs $(0, n)$ reflects this fact.

Hence, the length of each arc in the SP network correspond to the minimum cost of a segment between two transition locations. It follows that solving the SP amounts to calculating $V_0(h_0)$. Therefore, the minimum cost from problem $(P2)$ is equal to the length of the shortest path plus Dh_0. $\square$

For most problems, the following result can be used to simplify the SP network since it can be used to set a greater number of arc lengths equal to infinity.

Lemma 5.3 *Given a departure time from location* $i \in \{0, \ldots, n-1\}$ *equal to* w_i*, it is optimal to travel at speed* $\overline{v}$ *on all remaining arcs, i.e.,* i *to* $n-1$*, if and only if, by doing so the vehicle arrives at each remaining location within their time window, which is equivalent to* $\max_{j=i+1,\ldots,n} v^u_{i,j}(w_i) \leq \overline{v} \leq \min_{j=i+1,\ldots,n} v^l_{i,j}(w_i)$.

PROOF: Consider the relaxed problem of minimizing the total cost of travelling from location i to location n assuming there are no time windows at locations $i+1$ to n, i.e., $l_{i+1} = \ldots = l_n = 0$ and $u_{i+1} = \ldots = u_n = +\infty$ (which implies that $U_{i+1} = \ldots = U_n = \infty$). The total cost of driving at speeds v_j on arcs $j = i, \ldots, n-1$ is given by:

$$Dh_0 + \sum_{j=i}^{n-1} \left[Ad_j + B\frac{d_j}{v} + Cd_j v_j^2 + D\left(\frac{d_j}{v_j} + h_{j+1}\right)\right]$$

which is minimized at $v_i = \ldots = v_{n-1} = \overline{v}$. Hence, if this solution is feasible for the original problem with time windows, it must also be optimal.

Next we show that the vehicle, leaving location i at time w_i and driving at speed $\overline{v}$ on arcs i to $n-1$, arrives at locations $i+1$ to n within their time windows if and only if $\max_{j=i+1,\ldots,n} v^u_{i,j}(w_i) \leq \overline{v} \leq \min_{j=i+1,\ldots,n} v^l_{i,j}(w_i)$, which is equivalent to $v^u_{i,j}(w_i) \leq \overline{v}$ and $v^l_{i,j}(w_i) \geq \overline{v}$ for $j = i+1, \ldots, n$. Let a_j denote the arrival time of the vehicle into location j. We show by induction that $a_j \in [l_j, U_j]$ for $j = i+1, \ldots, n$. First, for $j = i+1$, we have:

$$\begin{aligned} v^u_{i,i+1}(w_i) \leq \overline{v} \leq v^l_{i,i+1}(w_i) \quad &\Leftrightarrow \quad \frac{d_{i+1}}{U_{i+1} - w_i} \leq \overline{v} \leq \frac{d_{i+1}}{l_{i+1} - w_i} \\ &\Leftrightarrow \quad l_{i+1} \leq a_{i+1} = w_i + \frac{d_i}{\overline{v}} \leq U_{i+1}. \end{aligned}$$

Therefore the arrival time in location $i+1$ is within its time window. Now assume that this is true for locations $i+1$ to $j-1$. This means that $a_j = w_i + \sum_{k=i}^{j-1} d_k/\overline{v} + \sum_{k=i+1}^{j-1} h_k$. Then we have:

$$\begin{aligned} v^u_{i,j}(w_i) \leq \overline{v} \leq v^l_{i,j}(w_i) \quad &\Leftrightarrow \quad \frac{\sum_{k=i}^{j-1} d_k}{U_j - \sum_{k=i+1}^{j-1} h_k - w_i} \leq \overline{v} \leq \frac{\sum_{k=i}^{j-1} d_k}{l_j - \sum_{k=i+1}^{j-1} h_k - w_i} \\ &\Leftrightarrow \quad l_j \leq a_j = w_i + \frac{\sum_{k=i}^{j-1} d_k}{\overline{v}} + \sum_{k=i+1}^{j-1} h_k \leq U_j. \end{aligned}$$

Therefore the arrival into location j is also within its time window. □

Using Lemma 5.3, we can modify the arc lengths in the shortest path network as follows: if the arc from node 0, i_l or i_u for some $i = 1, \ldots, n-1$ to node n has finite length for some $i = 1, \ldots, n-1$, then all other arcs from this node should have a length set equal to infinity.

We now provide an numerical example to illustrate the workings of the shortest path.

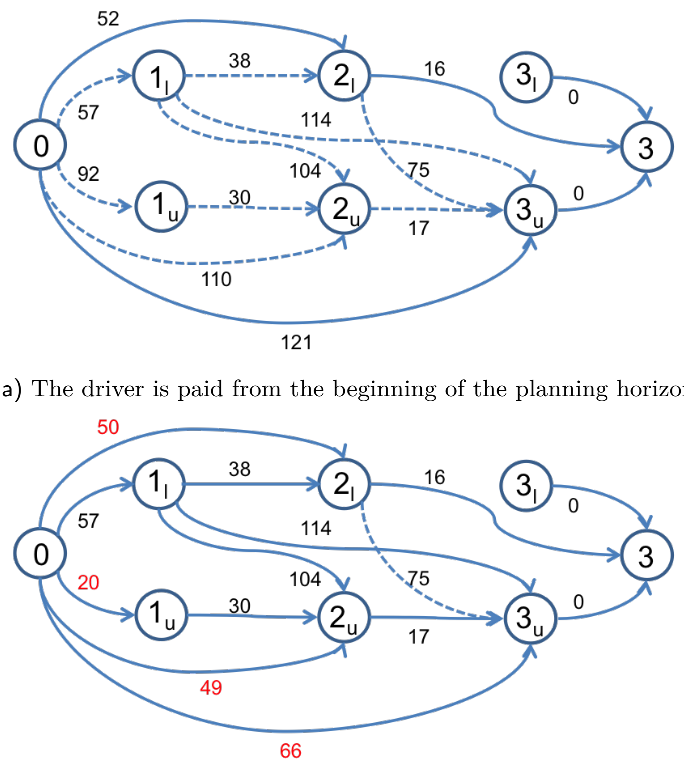
Example 5.1 Suppose $n = 3$. Let the distances (in meters) between locations be $d_0 = 77,200$, $d_1 = 113,620$ and $d_2 = 64,720$. The time windows (in seconds) are $[1313; 32,400]$ at node 1, [11,640; 32,400] at node 2 and $[0; 32400]$ at node 3. The service time is equal to zero at all nodes, i.e., $h_i = 0$ for $i = 0, \ldots, 3$. Let $v^{min} = 30$ km/h and $v^{max} = 90$ km/h. Without loss of generality we set $A = 0$. Finally we use $B = 0.00142$, $C = 1.98e^{-7}$ and $D = 0.00222$ (as in Chapter 3 which implies that $\underline{v} = 55.19$ km/h and $\overline{v} = 75.48$ km/h.

Figure 5.6a shows the SP network when the driver is paid from the beginning of the planning horizon and Figure 5.6b shows the SP network when the driver is paid from the start of service at the origin node (the arcs whose length differs are marked in red). To simplify the exposition, all arcs with infinite costs have been removed from these pictures.

When the driver is paid from the beginning of the planning horizon, the shortest path from node 0 to node n goes from node 0 to node $2l$ then to node 3 and the corresponding total cost is $52 + 16 = 68$. This means that it is optimal for the vehicle to travel at speed $v_{0,2}(0) = 59.02$ km/h on arcs 0 and 1 to arrive at location 2 exactly at the lower time window limit $l_2 = 11640$. Then the vehicle travels from location 2 to location 3 at speed $\overline{v} = 75.48$ km/h. Note that, since the length of arc $(2_l, 3)$ is finite, arc $(2_l, 3_u)$ could be removed from the SP network, i.e., set its cost infinite thanks to Lemma 5.3.

When the driver is paid from the start of the service at the origin location, there are two shortest paths from node 0 to 3. The first one goes through arcs $(0, 2_l)$ and $(2_l, 3)$ and the second one goes through arcs $(0, 3_u)$ and $(3_u, 3)$. In both cases the associated cost is 66. The first shortest path corresponds to a solution in which service at the origin location is postponed until $\max\left\{0, l_2 - \frac{d_0+d_1}{\overline{v}} - h_0 - h_1\right\} = 2538.94$ seconds into the planning horizon and the second shortest path corresponds to a solution wherein the service at the origin location is postponed until $\max\left\{0, U_3 - \frac{d_0+d_1+d_2}{\overline{v}} - h_0 - h_1 - h_2\right\} = 20212.16$ seconds into the planning horizon. In both cases, it is optimal for the vehicle to travel at speed $\overline{v}$ on arcs 0, 1 and 2. In practice, any solution wherein the voluntary pre-service waiting time at the origin location is between 2538.94 and 20212.16 and the speed on all arcs is $\overline{v}$ is optimal in this example.

Our next result establishes that our SP solution method is quadratic in the number



(a) The driver is paid from the beginning of the planning horizon

(b) The driver is paid from the start of service at the origin location

Figure 5.6 Shortest path networks in Example 5.1

of customer locations in the original network.

Corollary 5.2 *The optimal solution to Problem (P) can be found in* $O(n^2)$.

PROOF: The complexity of a shortest path problem in an acyclic network is bounded by the number of arcs (see K et al., 1988). Our SP network has $2n + 2$ nodes and at most $2n^2 + 2n + 1$ arcs with finite costs. Hence the complexity of our shortest path is quadratic in n. □

5.5 Insights

5.5.1 Impact of the driver wage policy

In Example 5.1, we see that the total cost decreases from 68 to 66, i.e. by 3%, when the driver is paid from the start of service at the origin location (driver wage policy (b)) as opposed to from the beginning of the planning horizon (driver wage policy (a)). In general we always have $C^{*(b)} \leq C^{*(a)}$ since it is always possible to make the start of service at the origin location coincide with the beginning of the planning horizon; therefore any delay must lead to a decrease in total costs. To further investigate the benefits of paying the driver from the start of service, we conducting some numerical tests on 1000 problem instances. The parameters were generated by randomly drawing parameters values from the following sets: $n = 3$, $d_i \in \{30000, 50000, 80000, 100000, 130000, 150000\}$ for $i = 0, 1, 2$, $h_i = 0$ for $i = 0, \ldots, 3$, $l_i \in \{0, 2000, 5000, 10000, 15000\}$ and $u_i \in \{10000, 15000, 35000\}$. We found that the percentage cost difference between the two driver wage policies, measured as $(C^{*(a)} - C^{*(b)})/C^{*(b)}$, ranged from 0 to 88.34% with an average of 9.25%. This suggests that paying drivers from the start of service at the origin location can lead to a very significant decrease in cost. In practice, this requires advanced planning on the part of the logistics provider and greater flexility on the part of its drivers whose schedule may change from day to day, based on the parameters of the delivery route.

5.5.2 Impact of time windows

In this section we explore the cost implications of having time windows at the customer locations. Specifically we compare the minimum cost obtained under each driver wage policy to the lower bound $\underline{C}$ which would be obtained in the absence of time windows. Using the same set of problem instances described in the previous section, we calculate the percentage cost difference between the solution with time windows and without time windows, that is we compute $(C^{*(a)} - \underline{C})/\underline{C}$ and $(C^{*(b)} - \underline{C})/\underline{C}$ for the two driver wages policies. We find that the percentage increase in costs due to the presence of time windows varies between 0% and 80.33% with an average of 12.28% when the driver is paid from the beginning of the planning horizon and varies between 0% and 17.88% with an average of 1.43% when the driver is paid from the start of service at the origin location. This suggests that the existence of time windows can have a negative impact on costs but that much of this negative effect can be offset by postponing the start of service at the origin location and paying the driver from that point on only.

5.6 The Time-Dependent Departure Times and Speed Optimization Problem

In this section we consider an extension of the DSOP, called *Time-Dependent Departure Times and Speed Optimization problem* (TDDSOP), which explicitly account for the presence of traffic congestion, which at peak periods constrains the speed of the vehicle increasing the amount of fuel consumed. Similar to the DSOP, the objective is to determine the optimal departure time from each location and the travel speed on each arc so as to minimize the total cost which encompasses emissions and labour costs. In line with Chapters 3 and 4, we assume that there is an initial period of traffic congestion which lasts a unit of time, followed by a period of free-flow in which the vehicle is allowed to travel at any speed level within the limits imposed by traffic regulation, i.e. v^{min} and v^{max}. As such, the TDDSOP problem boils down to a special case of the TDPRP where there is only one vehicle and a fixed sequence in which the customer locations are to be visited. For more information about how traffic congestion is modeled and how the vehicle emissions are calculated we refer to Chapter 3.

5.6.1 A heuristic algorithm for the TDDSOP

In this section we present a heuristic algorithm to solve the TDDSO. The proposed algorithm builds upon the solution to the Speed Optimization Problem (SOP) proposed by (e.g., Hvattum et al., 2010, 2013) for ship scheduling, which was then adapted to the PRP byDemir et al. (2012). These authors propose an algorithm to compute the optimal solution by recursively adjusting the travel speed for segments of the route until a feasible solution is found. Their method optimizes the travel speed only and is exact provided the total cost function is convex (Hvattum et al., 2010, 2013). In contrast, our algorithm is more general because it optimizes two sets of decision variables, namely the departure times and free flow speeds and the total cost function is no longer convex. As a consequence, the solution methods proposed for the SOP cannot be used to solve the TDDSOP. Specifically our algorithm maintains the recursive nature of the algorithm proposed for the SOP and uses some of the analytical properties presented in Chapter 3 for a single arc version of the problem. The TDDSOP algorithm operates as follows. It first solves a *relaxed* problem without any time windows at intermediary locations, that is, with only the time window at the end location maintained. This solution is calculated by reducing the problem to a single-arc TDPRP which is solved using Theorem 3.1 in Chapter 3. Once the solution to the relaxed problem has been calculated, the algorithm checks whether there are any time window violations at intermediate nodes, i.e., whether the arrival time at locate i is lower than l_i or higher than u_i. In case of multiple violations, the algorithm selects the location p with the largest violation. The solution is calculated by calling the algorithm recursively on each side of location p, that is, by calling the function for $(s, \ldots, p)$ and for $(p, \ldots, e)$ separately. A pseudocode of the algorithm is provided in Algorithm 3.

The function SOLVE_RELAXED(s, e, ϵ_s) calculates the optimal departure times, i.e., $w_s, \ldots, w_{e-1}$, and free flow speeds, i.e., $v_s, \ldots, v_{e-1}$, between nodes s and e assuming that the earliest departure time from location s is ϵ_s and that time window limits at nodes $s, \ldots, e-1$ are relaxed, i.e., that $l_s = \ldots = l_{e-1} = 0$ and $u_s = \ldots = u_{e-1} = \infty$. Only the time window limits at location e are maintained. Let

$$
\begin{aligned}
& TC_{s,e}(w_r, v_r; \epsilon_s) = \qquad (5.12)\\
= \;& A \sum_{i=s}^{e-1} d_i + B\left(\sum_{i=s}^{r-1} \frac{d_i}{v_c} + (a - w_r)^+\right) + \\
+ \;& B\left(\frac{(d_r - (a - w_r)^+ v_c)^+}{v_r} + \sum_{i=r+1}^{e-1} \frac{d_i}{v_r}\right) + \\
+ \;& C\left[v_c^3\left(\sum_{i=s}^{r-1} \frac{d_i}{v_c} + \min\left\{\frac{d_r}{v_c}, (a - w_r)^+\right\}\right) + v_r^2\left((d_r - (a - w_r)^+ v_c)^+ + \sum_{i=r+1}^{e-1} d_i\right)\right] + \\
+ \;& D\left(\max\left\{a + \frac{(d_r - (a - w_r)^+ v_c)^+}{v_r} + \sum_{i=r+1}^{e-1}\left(h_i + \frac{d_i}{v_r}\right), l_e\right\} + h_e - \epsilon_s\right).
\end{aligned}
$$

The function *SINGLE_ARC_TDPRP* calculates the optimal departure time, i.e. w, and free flow speed, i.e. v, for a single-arc TDPRP with parameters $(a, v_c, v_{max}, d, \epsilon, l, u)$ using Theorem 3.1 in Chapter 3 .

A pseudocode of the SOLVE_RELAXED function is provided in Algorithm 4.

5.6.2 Performance of the TDDSOP algorithm

We have performed several computational experiments in order to evaluate the performance of our TDDSOP algorithm. We compare the solutions obtained by our TDDSOP algorithm (denoted S_A) with the value obtained with the MIP formulation from Chapter 3 (denoted S_{IP}). The tests were run on three sets of instances from the PRPLIB. For each set of instances, the time window limits were relaxed by a factor δ, i.e. $l_i' = l_i - \delta(u_i - l_i)$ and $u_i' = u_i + \delta(u_i - l_i)$. In order to solve the MIP formulation, three sets (5, 10, and 15) of free-flow speed levels were considered. The results are reported in Table 5.1. The first column, entitled *Instances*, reports the name of the instance set (each set is made of 20 instances). The second column, entitled a, reports the duration of the congestion period.

Algorithm 3: TDDSOP

```
function [w*_s, ..., w*_{e-1}, v*_s, ..., v*_{e-1}] ← TDDSOP (s, e, ε_s, a)
    [r, w_s, ..., w_{e-1}, v_s, ..., v_{e-1}] ← SOLVE_RELAXED (s, e, ε_s, a);
    violation ← 0, p ← 0;
for i ← r+1 to e−1 do
    g_i ← max{0, l_i − w_{i−1} − T_{i−1}(w_{i−1}, v_{i−1}), w_{i−1} + T_{i−1}(w_{i−1}, v_{i−1}) − u_i};
    if g_i ≥ violation then
        violation ← g_i, p ← i;
if violation > 0 and w_{p−1} + T_{p−1}(w_{p−1}, v_{p−1}) < l_p then
    u_p ← l_p;
    [w*_s, ..., w*_{p−1}, v*_s, ..., v*_{p−1}] ← TDDSOP(s, p, ε_s, a);
    ε_p ← max{w*_{p−1} + T_{p−1}(w*_{p−1}, v*_{p−1}), l_p} + h_p;
    ã_p ← max{ε_p, a};
    [w*_p, ..., w*_{e−1}, v*_p, ..., v*_{e−1}] ← TDDSOP(p, e, ε_p, ã_p);
if violation > 0 and w_{p−1} + T_{p−1}(w_{p−1}, v_{p−1}) > u_p then
    l_p ← u_p;
    [w*_s, ..., w*_{p−1}, v*_s, ..., v*_{p−1}] ← TDDSOP(s, p, ε_s, a);
    ε_p ← max{w*_{p−1} + T_{p−1}(w*_{p−1}, v*_{p−1}), l_p} + h_p;
    ã_p ← max{ε, a};
    [w*_p, ..., w*_{e−1}, v*_p, ..., v*_{e−1}] ← TDDSOP(p, e, ε_p, ã_p);
end function
```

The third column, entitled v_c, reports the congestion speed. The fourth column, entitled *Average Dev (%)*, reports the average percentage deviation in total costs between S_A and S_{IP}, which is calculated as $100(TC(S_A) - TC(S_{IP}))/TC(S_A)$, where $TC(S)$ denotes the total cost of a solution S. Table 5.1 shows that in all cases, the deviations are negative, implying that the solution computed with our TDDSOP algorithm is better than the solution obtained with CPLEX, i.e., $TC(S_{IP}) > TC(S_A)$. This is because the MIP model optimizes the free-flow speed over a finite set of 15 speed levels, whereas our algorithm considers speed as a continuous variable. These findings are consistent with our TDDSOP algorithm reaching the optimal solution in all the problem instances we considered.

5.7 Conclusion

In the first part of the chapter, we study the problem of optimizing the departure time and the travel speeds of a vehicle visiting and serving a fixed sequence of customer locations. The objective is the minimization of a total cost function

Algorithm 4: SOLVE_RELAXED

```
function [r, w*_s, ..., w*_{e-1}, v*_s, ..., v*_{e-1}] ← SOLVE_RELAXED( s, e, ε_s, a)
    Let k̂ denote the last location that can be reached before the end congestion period,
    if no post-service waiting is performed;
    for i ← s to e − 1 do
        t̲_i ← a − ε_s − Σ_{j=s+1}^{i} h_j − Σ_{j=s}^{i} d_j/v_c, t̄_i ← a − ε_s − Σ_{j=s+1}^{i} h_j − Σ_{i=s}^{i-1} d_j/v_c;
    k̂ ← s;
    while k̂ < e − 1 and t̄_{k̂+1} > 0 do
        k̂ + +;
    for i ← s; i ≤ k̂; i + + do
        Calculate ã_i, d̃_i, l̃_i, ũ_i;
        d̃_i ← Σ_{j=i}^{e-1} d_j, ã_i ← max{ε_i, a}, h̃_i ← Σ_{j=i+1}^{e-1} h_j, ũ_i ← u_e − h̃_i, l̃_i ← l_e − h̃_i;
        (w̃_i, ṽ_i) ← SINGLE_ARC_TDPRP (ã_i, v_c, v_max, d̃_i, ε_i, l̃_i, ũ_i);
        if ã_i − d_i/v_c ≤ w̃_i ≤ ã_i then
            c_i ← TC_{s,e}(w̃_i, ṽ_i; ε_s);                          ▷ use Equation (5.12)
            K ← K ∪ {i};
        ε_{i+1} ← ε_i + d_i/v_c + h_{i+1};
    r ← arg min_{i∈K} c_i;
    if t̲_{e-1} > 0 and c_r > TC_{s,e}(ε_{e-1}, v_c; ε_s) then
        r ← e, v*_r ← v_c;
    else if t̲_k̂ ≤ 0 and c_r > TC_{s,e}(ε_k̂, ṽ_k̂; ε_s) then
        r ← k̂, w*_r ← ε_k̂, v*_r ← ṽ_k̂;
    else if t̲_k̂ > 0 then
        ε_{k̂+1} ← ε_k̂ + d_k̂/v_c + h_{k̂+1}, d̃_{k̂+1} ← Σ_{j=k̂+1}^{e-1} d_j, ã_{k+1} ← max{ε_{k̂+1}, a};
        h̃_{k̂+1} ← Σ_{j=k̂+1}^{e-1} h_j, ũ_{k̂+1} ← u_e − h̃_{k̂+1}, l̃_{k̂+1} ← l_e − h̃_{k̂+1};
        (w̃_{k̂+1}, ṽ_{k̂+1}) ← SINGLE_ARC_TDPRP (ã_{k̂+1}, v_c, v_max, d̃_{k̂+1}, ε_{k̂+1}, l̃_{k̂+1}, ũ_{k̂+1});
        if c_r > TC_{s,e}(ε_{k̂+1}, ṽ_{k̂+1}; ε_s) then
            r ← k̂ + 1, w*_r ← ε_{k̂+1}, v*_r ← ṽ_{k̂+1};
    for i ← s to r − 1 do
        w*_i ← ε_i v*_i ← v_c;
    for i ← r + 1 to e − 1 do
        Calculate T_{i-1}(w*_{i-1}, v*_{i-1});                         ▷ use Equation (3.1)
        w*_i ← w*_{i-1} + T_{i-1}(w*_{i-1}, v*_{i-1}) + h_i, v*_i ← v*_r ;
        w*_i ← a + (d_r − (a − w_r)v_c)/v_r + Σ_{j=r+1}^{i-1} d_j/v_r + Σ_{j=r+1}^{i} h_j, v*_i ← v*_r ;
end function
```

encompassing driver wage and CO_{2e} emissions cost. We considered two driver wage policies: (a) the driver is paid from the beginning of the planing horizon and (b) the driver is paid from the beginning of the service at the initial location.

First we show that it is never optimal for the vehicle to wait idly at a customer location upon completion of service. Using this result we formulate the problem as a dynamic programming problem and we show that the optimal driving schedule

Table 5.1 Average *Dev* (%) for three sets of instances

Instances	δ	a (s)	v_c (km/h)	Average *Dev* (%)
UK10	0.2	0	-	-0.005
UK10	0.3	3000	15	-0.005
UK10	0.5	3600	10	-0.002
UK15	0.7	3000	15	-0.004
UK20	1.0	3000	15	-0.008

can be broken down in segments on which the vehicle travels at a constant speed. By exploiting some structural properties of the optimal solution we show how to recast the problem into a shortest path problem. To the best of our knowledge, our study is the first one to obtain an exact solution to the DSOP. Our solution method, which has a complexity which is quadratic in the number of customer locations, has a very appealing structure and a nice visual representation. Moreover, we present some insights on how the driver wage policy ant the presence of time window at customer locations affect the optimal solution. In the second part of the chapter we propose a heuristic algorithm to solve the TDDSOP, an extension of the DSOP where there is the presence of traffic congestion, which at peaks periods, limits the travel speed increasing the amount of CO_{2e} produced by the vehicle. The procedure proposed in this section builds upon the analytical results for the single-arc version of the problem presented in Chapter 3. The proposed algorithm was empirically shown to run very quickly and consistently provide highly accurate solutions on realistic instances. Our procedure can be embedded within algorithms for the TDPRP, or can be used as a stand-alone routine when vehicle routes have already been fixed.

5.A Extra results

For use in the following proposition we introduce the following notation: threshold values:

$$\begin{aligned}
\overline{w}^u_{i,k} &= U_k - \sum_{j=i}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j, \\
\overline{w}^l_{i,k} &= l_k - \sum_{j=i}^{k-1} d_j/\overline{v} - \sum_{j=i+1}^{k-1} h_j, \\
\tilde{w}^l_{i,p,k} &= l_p - \frac{\sum_{j=i}^{p-1} d_j}{v^u_{p,k}(l_p + h_p)} - \sum_{j=i+1}^{p-1} h_j.
\end{aligned}$$

Proposition 5.6 *Suppose the driver is paid from the start of service at the origin location. Given the piecewise expression for $V_0(w_0)$, the optimal solution is as follows.*

- *If there exist some values of b and b' such that $V_0(w_0) = G_{0,n}(\overline{v}, w_0)$ for $w_0 \in [b, b']$, then it is optimal to set $y_0^* = (b - h_0)^+$.*
  - *if $h_0 \leq b'$, the vehicle travels at speed $\overline{v}$ on arcs 0 to n, reaching each location within their effective time windows.*
  - *If $h_0 > b'$, there exists a location $k_0 \in \{1, \dots, n\}$ such that the vehicle travels at speed $v^u_{0,k_0}(h_0)$ on arcs 0 to $k_0 - 1$, reaching location k_0 exactly at time U_{k_0}.*
- *If there does not exist some values of $w_0 \in [0, \hat{w}_0]$ such that $V_0(w_0) = G_{0,n}(\overline{v}, w_0)$, then there exists $p_0, k_0 \in \{1, \dots, n\}$ such that $p_0 \neq k_0$ and*
  - *If $k_0 > p_0$, then it is optimal to set $y_0^* = (\overline{w}^l_{0,p_0} - h_0)^+$.*
    - *If $h_0 \geq \tilde{w}^l_{i,p_0,k_0}$, the vehicle travels at speed $v^u_{0,k_0}(h_0)$ on arcs 0 to $k_0 - 1$, reaching location k_0 exactly at time U_{k_0}.*
    - *If $h_0 < \tilde{w}^l_{i,p_0,k_0}$, the vehicle travels at speed $v^l_{0,p_0}\left(\max\left\{h_0, \overline{w}^l_{0,p_0}\right\}\right)$ on arcs 0 to $p_0 - 1$, reaching location p_0 exactly at time l_{p_0}.*
  - *If $k_0 < p_0$, then it is optimal to set $y_0^* = (\overline{w}^u_{0,k_0} - h_0)^+$ and the vehicle travels at speed $v^u_{0,k_0}\left(\max\left\{h_0, \overline{w}^u_{0,k_0}\right\}\right)$ on arcs 0 to $k_0 - 1$, reaching location k_0 exactly at time U_{k_0}.*

Proof: When the driver is paid from the start of service time at the origin location, the optimal solution is found by solving $\min_{y_0 \in [0, \hat{w}_0} - h_0] V_0(y_0 + h_0)$, or equivalently $\min_{w_0 \in [h_0, \hat{w}_0]} V_0(w_0)$. In this proof we refer to the cases described in the proof of Proposition 5.4. In Cases (i), (ii), (iii.a) and (iv) there exists a piece of V_0 where the function is equal to $G_{0,n}(\overline{v}, w_0)$ and, in each case, this is where the minimum value of the function is reached. In particular, in Case (i) the minimum for $w_0 \geq h_0$ is achieved at $\overline{w}^l_{0,p_0}$ if $h_0 \leq \overline{w}^l_{0,p_0}$ and at h_0 otherwise, therefore we should set $y^*_0 = (\overline{w}^l_{0,p_0} - h_0)^+$ and the optimal speed is $\overline{v}$. In Cases (ii) and (iii.a) the minimum for $w_0 \geq h_0$ is achieved at $\overline{w}^u_{0,k_0}$ if $h_0 \leq \overline{w}^u_{0,k_0}$ and at h_0 otherwise, therefore we should set $y^*_0 = (\overline{w}^u_{0,k_0} - h_0)^+$. Moreover in these two cases, the optimal speed if $\overline{v}$ if $h_0 \leq \overline{w}^u_{0,k_0}$ and $v^u_{k_i}(h_0)$ otherwise. Finally, in Case (iv) the function is constant for $w_0 \geq h_0$ and the optimal speed on arcs 0 to n is $\overline{v}$.

Case (iii.b) is the only one where there does not exist a piece of V_0 where the function is equal to $G_{0,n}(\overline{v}, w_0)$. In sub-case (iii.b.1) the minimum for $w_0 > h_0$ is achieved at $\overline{w}^l_{0,p_0}$ if $h_0 < \overline{w}^l_{0,p_0}$ and at h_0, otherwise. If $\overline{w}^l_{0,p_0} > h_0$ the optimal speed id $\overline{v}$, if $\overline{w}^l_{0,p_0} < h_0 < l_{p_i} - \frac{\sum_{j=i}^{p_i-1} d_j}{v^u_{p_i,k_i}(l_{p_i}+h_{p_i})} - \sum_{j=i+1}^{p_i-1} h_j$ the optimal speed is $v^l_{0,p_0}(h_0)$, otherwise it is $v^u_{0,k_0}(h_0)$. In sub-case (iii.b.2) the minimum for $w_0 > h_0$ is achieved at $\overline{w}^u_{0,k_0}$ if $h_0 < \overline{w}^u_{0,k_0}$ and at h_0, otherwise. Specifically, if $h_0 > \overline{w}^u_{0,k_0}$ the optimal speed is $v^u_{0,k_0}(h_0)$, otherwise it is $\overline{v}$. $\square$

One's destination is never a place, but rather a new way of looking at things.
Henry Miller, *Big Sur and the Oranges of Hieronymus Bosch*

6 Conclusion

This thesis focuses on sustainable planning for city logistics, with particular attention to fleet management, routing and scheduling problems. In Chapter 2 we study the strategic problem of managing a heterogeneous fleet of vehicles operating in a urban area where access restrictions are applied to certain categories of vehicles. In Chapters 3 and 4 we study the problem of determining the optimal set of routes, the travel speed on each arc of a route and the departure time from each node for a homogeneous fleet of vehicles serving a set of customer nodes with hard time window limits. The travel speed of the vehicles is limited at peak hours due to the presence of traffic congestion. The objective is to minimize a total cost function encompassing driver wage and emissions cost. Finally, in Chapter 5 we study the scheduling problem of optimizing the travel speed and the departure times of a vehicle visiting a given sequence of customer locations. The objective is the minimization of a cost function including emissions cost and labour cost.

6.1 Research objectives revisited

In the introduction of this thesis we state five research objectives. In the follow we briefly explain how we addressed each research objective, and we summarize the main findings.

6.1.1 Sustainability in fleet management

Research objective 1 Develop a fleet management model to manage a (possibly heterogenous) fleet of vehicles to serve a city in the presence of access restrictions.

In Chapter 2 we develop a MIP model for the problem of managing a heterogeneous fleet of vehicles operating in a urban area where access restrictions are applied to certain types of vehicle. We represent the city as a rectangular service area and we recast the fleet management problem into an area partitioning problem. This latter problem consists of partitioning the urban area into rectangular service sectors, each served by a single vehicle. The length of the tour to serve each sector is calculated using a continuous approximation formula. The dimension of a sector depends on the position of the sector in the urban area, on the capacity of the vehicle serving the sector and on the vehicles access restrictions. The objective is to minimize the cost of fleet ownership or leasing, the fuel cost and the labor cost. Additionally, we formulate the problem as a dynamic programming problem and we study the properties of the value function. By exploiting some key structural properties we develop an efficient method to compute an optimal solution for a problem with two vehicle types, e.g. electric and diesel. We perform an extensive numerical analysis which shows that on average, operating a heterogenous fleet only leads to a small decrease in cost, which may not be outweighed by the increase in logistical complexity resulting from operating several vehicle types.

Research objective 2 Investigate the impact of traffic restrictions on urban fleet planning.

In a numerical study in Chapter 2 we investigate how time access restrictions affect the optimal fleet composition for a problem with two vehicle types, e.g. electric and diesel. We found that limiting the access or banning the use of large vehicles may in some cases be counterproductive, as it might actually increase the number of diesel vehicles on the streets, further it might also contribute to increase the traffic congestion.

6.1.2 Sustainability in vehicle routing and scheduling

Research objective 3 Study the problem of routing and scheduling a homogeneous fleet of vehicles in a presence of traffic congestion which, at peak periods, limits the vehicles travel speed and increases the amount of emissions produced. The objective is the minimization of a total cost function including labour and emissions cost. Formulate the problem as a mathematical model and develop heuristic algorithm to solve to solve medium and large size instances in a reasonable amount of time.

In Chapter 3 we provide an integer linear programming formulation (MILP) for a vehicle routing problem where CO_{2e} emissions and traffic congestion are accounted for. The problem consists on determining the optimal set of routes, the travel speed on each leg of a route and the departure time from each node for a homogeneous fleet of vehicles, serving a given set of customers with hard time window limits, in a presence of traffic congestion. The objective is the minimization of the total travel cost encompassing emissions costs and labour cost. We refer to this problem as the Time-Dependent Pollution Routing Problem (TDPRP) as it extends the Pollution Routing Problem (PRP) from Bektaş and Laporte (2011) by including traffic congestion. We consider two ways of calculating the total time for which the driver is paid, which we call driver wage policies: (i) the driver of each vehicle is paid from the beginning of the time horizon until returning back to the depot, or (ii) the driver is paid only for the time spent away from the depot, i.e., either en-route or at a customer. Computational results confirm that the proposed formulation computationally outperforms the formulation proposed for the PRP.

In Chapter 4 we propose an adaptive large neighbourhood search heuristic (ALNS) for the TDPRP. In addition to some removal and insertion operators adapted from the literature, the algorithm uses some new operators inspired by the analytical insights derived in Chapter 3. Computational results have shown that our algorithm is able to compete with the best heuristics for the PRP, even though it was not specifically designed for this specific problem type.

Research objective 4 Study how idle waiting either at the depot or at a customer node affects the emissions and the labour costs, in a presence of traffic congestion.

In Chapter 3 we derive a complete characterization of the optimal solution for a single-arc version of the TDPRP, identifying conditions under which it is optimal to wait idle at the depot and the associated amount of time. These analytical results show that in some cases adding idle waiting time at the depot can be used as an efficient strategy to avoid traveling in congestion, and therefore reduce the total travel cost. We also present several examples that motivate idle waiting time, either pre- or post-service, both at the depot and at customer nodes.

Research objective 5 Study the scheduling problem of a vehicle visiting a given sequence of locations. The objective is to determine the optimal departure times and the travel speed on each leg of the route so as to minimize the sum of labour and emissions costs. Formulate the problem in mathematical terms and develop an exact algorithm for solving the problem. Extend the study to the case where traffic congestion limits the vehicle speed during peak periods. Develop a heuristic algorithm to solve this latter problem.

In the first part of Chapter 5 we propose an exact method for optimizing the travel speed and the departure times of a vehicle visiting a given sequence of customer nodes as to minimize the sum of emissions cost and labour cost. We refer to this problem as the Departure Time and Speed Optimization Problem (DSOP). We formulate the problem as a dynamic programming problem, and we study the structure of the value function. By exploiting some key properties of the value function we obtain an efficient solution method that simplify to a shortest path method.

In the second part of Chapter 5, we describe a heuristic algorithm to solve the DSOP in a presence of peak hours traffic congestion. Such problem is referred as the Time-Dependent Speed Optimization Problem (TDDSOP). The proposed algorithm is inspired by the Speed Optimization Problem (SOP) algorithm first proposed by (e.g., Hvattum et al., 2010, 2013) for ship scheduling, then adapted to the PRP by Demir et al. (2012) and builds on the analytical properties for the single-arc TDPRP presented in Chapter 3.

6.2 Future research directions

In this section we briefly discuss some future research directions regarding the underlying concepts presented in this thesis.

6.2.1 Sustainability in fleet management

In the context of sustainability in fleet management a challenging extension at the conceptual level could be to allow a non uniform distribution of customers within the service area. Such an extension would be particularly relevant for those companies operating in rural areas or in large areas where customers are not evenly distributed throughout. However, since the continuous approximation formulas that we used to model the problem in Chapter 2 are based on the assumption of uniform distribution extending the problem in such a direction is not straightforward as it might require redefining the model and redesigning a new solution methodology.

6.2.2 Sustainability in vehicle routing and scheduling

In the context of sustainability in vehicle routing and scheduling, considering more real-life congestion patterns with multiple congestion periods alternating to free-flow periods during the planning horizon is a natural extension to the work presented in Chapters 3, 4 and 5. In the contest of the TDPRP, an interesting extension could be considering a heterogeneous fleet of vehicles and accounting for the presence of vehicles access restrictions. This way we would strengthen the connection between the strategic decision level, i.e managing the composition of the vehicles fleet, and the operational decision level, i.e. planning the routing and the scheduling of the vehicles. A more challenging extension could be extending the problems studied in Chapters 3, 4 and 5 so as to account for unforeseen traffic congestion. Such an extension implies redefining the problem in order to allow dynamic routing and scheduling decisions.

Bibliography

D. Aksen, O. Kaya, F. S. Salman, and Ö. Tüncel. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research*, 239(2):413–426, 2014.

N. Anand, H.J. Quak, R. Van Duin, and L. Tavasszy. City logistics modeling efforts: Trends and gaps-a review. *Procedia-Social and Behavioral Sciences*, 39:101–115, 2012a.

N. Anand, M. Yang, R. Van Duin, and L. Tavasszy. Genclon: An ontology for city logistics. *Expert Systems with Applications*, 39(15):11944–11960, 2012b.

M. Barth and K. Boriboonsomsin. Real-world CO_2 impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, 2058(1):163–171, 2008.

M. Barth and K. Boriboonsomsin. Energy and emissions impacts of a freeway-based dynamic eco-driving system. *Transportation Research Part D*, 14(6):400–410, 2009.

M. Barth, T. Younglove, and G. Scora. Development of a Heavy-Duty Diesel Modal Emissions and Fuel Consumption Model. Technical report, UC Berkeley: California Partners for Advanced Transit and Highways (PATH), 2005.

T. Bektaş and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

M. Browne, J. Allen, T. Nemoto, D. Patier, and J.s Visser. Reducing social and environmental impacts of urban freight transport: A review of some major cities. *Procedia-Social and Behavioral Sciences*, 39:19–33, 2012.

R.G. Conrad and M.A. Figliozzi. Algorithms to quantify impact of congestion on time-dependent real-world urban freight distribution networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2168:104–113, 2010.

J.-F. Cordeau, G. Laporte, and M.W.P. Potvin, J.-Y.and Savelsbergh. Transportation on demand. *Handbooks in operations research and management science*, 14:429–466, 2007a.

J.-F. Cordeau, G. Laporte, M.W.P. Savelsbergh, and D. Vigo. Vehicle Routing. In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, chapter 6, pages 367–428. Elsevier, Amsterdam, The Netherlands, 2007b.

J. Couillard and A. Martel. Vehicle fleet planning the road transportation industry. *Engineering Management, IEEE Transactions on*, 37(1):31–36, 1990.

T.G. Crainic, N. Ricciardi, and G. Storchi. Models for evaluating and planning city logistics systems. *Transportation science*, 43(4):432–454, 2009.

S. Dabia, E. Demir, and T. Van Woensel. An exact approach for the pollution-routing problem. Technical report, Beta Research School for Operations Management and Logistics, 2014.

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

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

C.F. Daganzo. Modeling distribution problems with time windows: Part I. *Transportation Science*, 21(3):171–179, 1987a.

C.F. Daganzo. Modeling distribution problems with time windows: Part II: Two customer types. *Transportation Science*, 21(3):180–187, 1987b.

C.F. Daganzo. *Logistics Systems Analysis*. Springer, Berlin, 2005.

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

E. Demir, T. Bektaş, and G. Laporte. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 6(5):347–357, 2011.

E. Demir, Bektaş T., and Laporte G. An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. *European Journal of Operational Research*, 223 (2):346–359, 2012.

E. Demir, T. Bektaş, and G. Laporte. The bi-objective Pollution-Routing Problem. *European Journal of Operational Research*, 232(3):464–478, 2013.

E. Demir, T. Bektaş, and G. Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014a.

E. Demir, Bektaş T., and Laporte G. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237:775–793, 2014b.

E. Demir, Y. Huang, S. Scholts, and T. Van Woensel. A selected review on the negative externalities of the freight transportation: Modeling and pricing. *Forthcoming in*

Transportation Research Part E: Logistics and Transportation Review, 2015.

J.F. Ehmke. *Integration of Information and Optimization Models for Routing in City Logistics*, volume 177. Springer New York, 2012a.

J.F. Ehmke. *Integration of information and optimization models for routing in city logistics*, volume 177. Springer Science & Business Media, 2012b.

ENCLOSE project report. Electric fleets in urban logistics. improving urban freight efficiency in small and medium-sized historic towns. http://www.bmvit.gv.at/verkehr/elektromobilitaet/downloads/emobil_urbanlogistics_annex.pdf, 2014. (accessed on March 2015).

European Commission. Technical report. Together towards competitive and resource-efficient urban mobility. Available at http://ec.europa.eu/transport/themes/urban/doc/ump/com(2013)913_en.pdf, 2013. (accessed on March 2015).

M.A. Figliozzi. Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board*, 2197:1–7, 2010.

M.A. Figliozzi. The impacts of congestion on time-definitive urban freight distribution networks CO_2 emission levels: Results from a case study in Portland, Oregon. *Transportation Research Part C*, 19(5):766–778, 2011.

Transport for London (TfL). Sustainable fleet management guide. http://www.tfw.org.uk/documents/fuel-and-fleet-management-guide.pdf, 2012. (accessed on March 2015).

D.J. Forkenbrock. External costs of intercity truck freight transportation. *Transportation Research Part A*, 33(7-8):505–526, 1999.

D.J. Forkenbrock. Comparison of external costs of rail and truck freight transportation. *Transportation Research Part A*, 35(4):321–337, 2001.

B.L. Golden, A.A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

J. Gonzalez-Feliu, F. Semet, and J-L. Routhier. *Sustainable urban logistics: Concepts, methods and information systems*. Springer, 2014.

I. Gray. Green your fleet: Thoughts from energy saving trust. https://blog.business.tomtom.com/green-your-fleet/, 2013. (accessed on March 2015).

S. Hansen, A. Byrd, A. Delcambre, A. Rodriguez, S. Matthews, and R.L. Bertini. PORTAL: An on-line regional transportation data archive with transportation system management applications. 2005 CITE Quad/WCTA Regional Conference Vancouver, BC, Canada April 7-9, 2005, 2005.

V.C. Hemmelmayr, J-F. Cordeau, and T.G. Gabriel Crainic. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. *Computers & operations research*, 39(12):3215–3228, 2012.

J. Hickman, D. Hassel, R. Joumard, Z. Samaras, and S. Sorenson. Technical Report. MEET Methodology for calculating transport emissions and energy consumption. http://www.transport-research.info/Upload/Documents/200310/meet.pdf, 1999. (accessed on March 2015).

M. Huang, K.R. Smilowitz, and B. Balcik. A continuous approximation approach for assessment routing in disaster relief. *Transportation Research Part B: Methodological*, 50:20–41, 2013.

L.M. Hvattum, I. Norstad, K. Fagerholt, and G. Laporte. An exact algorithm for the vessel speed optimization problem. Technical report, Norwegian University of Science and Technology, Trondheim, Norway, 2010.

L.M. Hvattum, I. Norstad, K. Fagerholt, and G. Laporte. Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2):132–135, 2013.

International Road Transport Union. Congestion is responsible for wasted fuel. http://www.iru.org/en_policy_co2_response_wasted, 2012. (accessed on March 2015).

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

O. Jabali, T. Van Woensel, and A.G. de Kok. Analysis of travel times and CO_2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6): 1060–1074, 2012b.

Ravindra A. K, T. L. Magnanti, and Orlin J. B. Network flows. Technical report, DTIC Document, 1988.

D. Kirby. Is your fleet the right size? *Operational Research Quarterly*, 10(4):252–252, 1959.

R. Knight (Editor). Technical Report. Mobility 2030: Meeting the challenges to sustainability, 2004. Available at: http://http://www.wbcsd.org/web/publications/mobility/mobility-full.pdf (accessed on March 2015).

C. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

R. Kramer, A. Subramanian, T. Vidal, and L. Cabral. A matheuristic approach for the Pollution-Routing Problem. Technical report, Cornell University, New York, 2014. Available at: http://arxiv.org/abs/1404.4895 (01.10.2014).

R. Kramer, N. Maculan, A. Subramanian, and T. Vidal. A speed and departure time optimization algorithm for the pollution-routing problem. Working Paper. Università degli Studi di Modena e Reggio Emilia, Italy, 2015.

J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

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

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

R. Loxton and Q. Lin. Optimal fleet composition via dynamic programming and golden section search. *Journal of Industrial and Management Optimization*, 7(4):875 – 890, 2011.

R. Loxton, Q. Lin, and K.L. Teo. A stochastic fleet composition problem. *Computers & Operations Research*, 39(12):3177–3184, 2012.

W. Maden, R. W. Eglese, and D. P. Black. Vehicle Routing and Scheduling with Time Varying Data: A Case Study. 2009.

J. Muñuzuri, J. Larrañeta, L. Onieva, and P. Cortés. Solutions applicable by local administrations for urban logistics improvement. *Cities*, 22(1):15–28, 2005.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435, 2007.

J.-Y. Potvin and J.-M. Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*, 66(3):331–340, 1993.

H.J. Quak and M.B.M. de Koster. Urban distribution: The impacts of different governmental time-window schemes. Technical report, 9th TRAIL Congress 2006, Research School TRAIL., Delft, The Netherlands, 2006.

H.J. Quak and M.B.M. de Koster. Delivering goods in urban areas: how to deal with urban policy restrictions and the environment. *Transportation Science*, 43(2):211–227, 2009.

G.M. Ribeiro and G. Laporte. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. *Computers & Operations Research*, 39(3):728–735, 2012.

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

S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–775, 2006b.

A. Roumboutsos, S. Kapros, and V. Thierry. Green city logistics: Systems of innovation to assess the potential of e-vehicles. *Research in Transportation Business & Management*, (11), 2014.

J. Roy. Recent trends in logistics and the need for real-time decision tools in the trucking industry. In *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, pages 9–pp. IEEE, 2001.

A. Sbihi and R. W. Eglese. The Relationship between Vehicle Routing & Scheduling and Green Logistics-A Literature Survey. *Lancaster University Management School*, 2007b.

G. Scora and M. Barth. Comprehensive modal emission model (CMEM), version 3.01, user's guide. Technical report, 2006. Available at: http://www.cert.ucr.edu/cmem/docs/CMEM_User_Guide_v3.01d.pdf (accessed on March 2015).

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Lecture Notes in Computer Science*, volume 1520, pages 417–431, Springer, Berlin, 1998.

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

A. Sorensen, P. Marcotullio, and J. Grant. *Towards sustainable cities: East Asian, North American, and European perspectives on managing urban regions.* Gower Publishing, Ltd., 2004.

A. Stewart. Technical report. Ultra low emission vans study. 2012. Available at https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/4550/ultra-low-emission-vans-study.pdf. (accessed on March 2015).

E. Taniguchi. Concepts of city logistics for sustainable and liveable cities. *Procedia-Social and Behavioral Sciences*, 151:310–317, 2014.

E. Taniguchi and R.G. Thompson. *City Logistics: Mapping The Future.* CRC Press, 2014.

E. Taniguchi, R.G. Thompson, T. Yamada, and R. Van Duin. *City Logistics. Network modelling and intelligent transport systems.* 2001.

E. Taniguchi, T.F. Fwa, and R.G. Thompson. *Urban Transportation and Logistics: Health, Safety, and Security Concerns.* CRC Press, 2013.

E. Taniguchi, R.G. Thompson, and T. Yamada. Recent trends and innovations in modelling city logistics. *Procedia-Social and Behavioral Sciences*, 125:4–14, 2014.

R.S.J. Tol. The marginal damage costs of carbon dioxide emissions: an assessment of the uncertainties. *Energy Policy*, 33(16):2064–2074, 2005.

P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*, volume 18. SIAM, 2014.

T. Van Woensel, R. Creten, and N. Vandaele. Managing the environmental externalities of traffic logistics: The issue of emissions. *Production and Operations Management*, 10(2):207–223, 2001.

T. Van Woensel, L. Kerbache, H. Peremans, and N. Vandaele. Vehicle routing with dynamic travel times: a queueing approach. *European Journal of Operational Research*, 186(3):990–1007, 2008.

S. Wheeler. Planning sustainable and livable cities. *The city reader*, 2:434–445, 1998.

Wikipedia. Carbon dioxide equivalent. http://en.wikipedia.org/wiki/Carbon_dioxide_equivalent, 2013a. (accessed on March 2015).

Wikipedia. Rush hour. http://en.wikipedia.org/wiki/Rush_hour, 2013b. (accessed

on March 2015).

Summary

Sustainable city logistics. Fleet planning, routing and scheduling

The issue of sustainability has emerged over the last few decades as an important problem in the transportation field. A major challenge for city logistics is to minimize the harmful effects of transportation activities, while guaranteeing a high service quality.

City Logistics has been defined as the process of optimizing the logistics and transport activities by private companies with the support of advanced information systems in urban areas considering the traffic environment, its congestion, safety and energy savings within the framework of a market economy.

There is a growing consensus on the view that significant benefits can be achieved by an appropriate mix of different measures such as optimized delivery plans, electric and low emissions vehicles and public incentive policies. The primary target of such measures is to improve the safety and the livability of the urban areas by reducing the amount of emissions produced by freight vehicles and the traffic congestion on the roads. For this purpose, a growing number of municipalities have started implementing restrictions policies which prioritize the usage of green vehicles in the city center. As a consequence, the carrier companies operating in such areas are forced to redefine their strategies and update their decision models in order to improve their sustainability and meet the requirements imposed by the new regulations.

The research in this thesis introduces a number of decision models where environmental issues and real world operational constraints are included on top of economical aspects. Specifically, Chapter 2 addresses sustainability issues in

urban logistics at the strategic decision level, while Chapters 3-5, focus on city logistics problems at a operational decision level.

Chapter 2 studies the problem of managing a heterogeneous fleet of vehicles to serve a urban area where access restrictions are applied to certain types of vehicles. The aim is to determine the optimal fleet composition in order to minimize the sum of ownership or leasing, transportation and labor costs. In this chapter we present a mixed integer program formulation and a dynamic program formulation. In both formulations the problem is modeled as an area partitioning problem where a rectangular service region has to be divided into sectors, each served by a single vehicle. The length of the tour to serve a sector depends on the dimension of the sector and on the customer density, and it is calculated using a continuous approximation formula. The objective is to partition the area into service sectors and to assign the vehicle types to the sectors, in order to minimize the total cost function. Furthermore, we present some structural properties of the optimal partition of the service region and we develop efficient algorithms to obtain an optimal solution. Finally, we derive some valuable insights on the effects of traffic restrictions and on the benefits of operating a heterogeneous fleet of vehicles.

Chapter 3 addresses sustainability in urban logistics at an operational decision level by introducing the Time-Dependent Pollution-Routing Problem (TDPRP), a vehicle routing problem where CO_2e emissions and traffic congestion are accounted for. The presence of traffic congestion during the first period of the day limits the vehicles speed and increases the amount of vehicle emissions per kilometre. The objective of the problem is to determine the assignment and the scheduling of customers to vehicles, the travel speed on each arc of the route and the departure time from each node, so as to minimize a total cost function encompassing labour cost and emissions cost. In this chapter we present an integer linear programming formulation of the TDPRP and we provide an analytical characterization of the optimal solution for a single-arc version of the problem. The theoretical results show that in some cases it is optimal to wait idly at certain locations in order to avoid congestion and to reduce the cost of emissions.

In Chapter 4 we propose a metaheuristic algorithm for the TDPRP based on Adaptive Large Neighborhood Search procedure. In this chapter, a number of destruction and construction operators specifically design for the TDPRP are presented. Extensive computational experiments are conducted for tuning the algorithm parameters and for assessing the quality of the new operators. The resulting solutions are then compared to (i) those reported in Chapter 3 and (ii) to those reported in the literature for the TDPRP where there is no congestion period, this problem is known in the literature as Pollution Routing Problem (PRP). Finally, extensive computational experiments on large TDPRP instances are presented.

Chapter 5 introduces the Departure Time and Speed Optimization Problem

(DSOP), namely a vehicle scheduling problem where CO_2e emissions are accounted for. The objective is to determine the departure times and the travel speed of a vehicle traveling on a fixed route, so as to minimize a total cost function composed of labour cost and emissions cost. In this chapter, we formulate the problem as a dynamic programming problem, and we study the structure and the properties of the value function. Based on the theoretical results, we derive an efficient solution method which simplifies into a shortest path problem. Finally, we propose a heuristic algorithm to solve the DSOP in a presence of traffic congestion which limits the vehicle speed during a period of the day, increasing the amount of emissions per kilometre.

About the author

Anna Franceschetti was born on December 16, 1984 in Legnago, Italy. She received her B.Sc. (2006) and M.Sc. (2009) in Management Engineering from the University of Bologna. Her master's thesis on container storage allocation was supervised by Alberto Caprara and Marco Lübbecke. After her studies she worked for two years in industry and in a research centre.

In 2011, she started her PhD at the Eindhoven University of Technology under the supervision of Tom Van Woensel, Dorothée Honhon and Jan Fransoo. During her PhD she cooperated with Tolga Bektaş from the University of Southampton, UK. Moreover she collaborated with Emrah Demir and Mark Stobbe from the Eindhoven University of Technology. Part of her PhD research was conducted at CIRRELT (Interuniversity Research Center on Enterprise Network, Logistics and Transportation), Canada, in collaboration with Gilbert Laporte. On September 22, 2015 Anna defends her PhD thesis at Eindhoven University of Technology.

Beta
Research School for Operations
Management and Logistics

Eindhoven University of Technology,
School of Industrial Engineering