UNIVERSITY OF SOUTHAMPTON
FACULTY OF SOCIAL SCIENCES
SOUTHAMPTON BUSINESS SCHOOL

Route and Speed Optimization Problems under Uncertainty and Environmental Concerns

by

Moncef Ilies Nasri

A thesis submitted for the degree of
Doctor of Philosophy in Management Science

October 2018

UNIVERSITY OF SOUTHAMPTON
FACULTY OF SOCIAL SCIENCES
SOUTHAMPTON BUSINESS SCHOOL

Route and Speed Optimization Problems under Uncertainty and Environmental Concerns

by

Moncef Ilies Nasri

Main Supervisor: Professor Tolga Bektaş
Supervisor: Professor Gilbert Laporte
Internal Examiner: Professor Jörg Fliege
External Examiner: Dr Dimitris Paraskevopoulos

A thesis submitted for the degree of
Doctor of Philosophy in Management Science

October 2018

UNIVERSITY OF SOUTHAMPTON

Abstract

FACULTY OF SOCIAL SCIENCES
SOUTHAMPTON BUSINESS SCHOOL

Doctor of Philosophy in Management Science

Route and Speed Optimization Problems under Uncertainty and Environmental Concerns

by Moncef Ilies Nasri

This thesis studies logistics problems with the overall aim to reduce the emission of greenhouse gases. These problems are formalized, modeled and solved to derive useful insight for both logistics companies and policy makers. Chapter 1 introduces the background, presents the research aims and objectives as well as the research context. Chapter 2 studies The Pollution-Routing Problem under traffic uncertainty. The problem assumes uncertain traffic conditions and aims at reducing the cost of emissions, fuel consumption and travel times. Stochastic programming has been used to propose new mathematical models capable of considering traffic conditions as a discrete set of random scenarios. Extensive computational experiments are carried out, to quantify the savings yielded by the stochastic approach over a deterministic approach, and by controlling speed. Chapter 3 reconsiders the problem defined in Chapter 2. However, instead of solving it with commercial solvers, new solution techniques based on decomposition, and more precisely integer L-shaped algorithm that uses cuts, lower-bounds and local-branching are proposed. Chapter 4 focuses on the speed optimization problem that consists of choosing the optimal speed on each leg of a given vehicle route represented by a fixed sequence of customers. The objective function accounts also for the pollution emitted by the vehicles. Each customer in the sequence has a service time window. Early and late starts of service are allowed, but at the expense of penalties. A natural model of the problem in the form of a non-linear program is presented, which is then linearized in several ways. Several algorithms are described based on the use of time-space networks. Managerial insight is derived for maritime and road transportation. Chapter 5 concludes by summarizing the key findings and contributions of this thesis, discusses the limitations of this work and suggests future directions of research.

UNIVERSITY OF SOUTHAMPTON

مختصر

FACULTY OF SOCIAL SCIENCES
SOUTHAMPTON BUSINESS SCHOOL

Doctor of Philosophy in Management Science

تحسين مسار وسرعة مركبات توزيع السلع مع أخذ بعين الاعتبار غموض حركة السّير

بغاية تقليل التلوث

من إعداد: منصف إلياس ناصري

هذه الأطروحة تُعنى بدراسة المشاكل اللّوجستيّة وتهدف لتقليل الانبعاثات الغازية المؤدية للاحتباس الحراري. هذه المشاكل تُدرَس، تُنَمذَج وتُحَلّ للوصول لاستنتاجات تفيد الشركات اللوجستية وصُنَّاع القرار. يتناول الفصل الأوّل مقدمة عن خلفيّة البحث، أهدافه، غاياته وسياقه البحثّي. الفصل الثانيّ يدرُس ويُناقش كيفية توجيه المركبات لتوزيع سلع على زبائن مع الحد من التلوُّث البيئيّ (The Pollution-Routing Problem) مفترضاً عدم اتّضاح حالة السَّير وهادفاً إلى تقليل التَّكلفة النَّاتجة عن الانبعاثات الغازية، استهلاك الوقود واليد العاملة. في هذه الأطروحة استُخدمت البرمجة العشوائيّة (Stochastic Programming) لإنتاج نماذج رياضية قادرة على تمثيل مجموعة سيناريوهات من حالات السَّير الغامضة. تم إجراء مجموعة من التجارب البرمجيَّة المكثَّفة لتقييم التَّوفير الحاصل من استخدام البرمجة العشوائيّة والتَّحكم في سرعة المركبات. في الفصل الثَّالث، تم اقتراح وتطوير طرق برمجية مخصَّصة لحل المشكلة التي تمَّ مناقشتها في الفصل السَّابق. تعتمد هذه الطُّرق على خوارزميَّات (L-shaped) والتي تهدف إلى تقليل مجال الحلول الممكنة باستخدام (cutting-planes)، استراتيجيَّة الحدود الدنيا (Lower Bounds) وخوارزميَّة البحث المحليّ (Local-Branching). يُركِّز الفصل الرَّابع على تحسين اختيار سرعة المركبات على كلِّ جزء من المسارات المحدَّدة للمَركبات وتؤدِّي إلى تحقيق نفس الأهداف المطروحة في الفصل الثَّاني. المسارات تتكوَّن من طرق مؤدية إلى زبائن، حيث لكل زبون وقت محدَّد للخدمة. يحِقُّ للمَركبة أن تصل قبل أو بعد الوقت المحدد من قبل الزبون ولكن بتكلفة إضافيَّة على المركبة. وفي هذا الفصل أيضاً، تمّ تقديم المشكلة في صفتها الغير خطيَّة، وتقديم نماذج خطيَّة وخوارزميَّات مخصّصة لحلِّ المشكلة، الوصول لاستنتاجات مفيدة لشركات النقل البرّي والبحريّ. يَختتم الفصل الخامس بتلخيص النتائج الرئيسية ومساهمات هذه الأطروحة، ويناقش قيود هذا العمل ويقترح اتجاهات مستقبلية للبحث.

Contents

Abstract v

Abstract in Arabic vii

List of Figures xiii

List of Tables xv

Declaration of Authorship xvii

Acknowledgements xix

List of Abbreviations xxiii

1 Introduction 1
- 1.1 Research context . . . 2
- 1.2 Potential application for autonomous trucks . . . 3
- 1.3 Research aims . . . 4
- 1.4 Research objectives . . . 4
- 1.5 A schematic representation of the thesis . . . 5

2 The Pollution-Routing Problem under Traffic Uncertainty 7
- 2.1 Introduction . . . 8
- 2.2 Literature review . . . 9
  - 2.2.1 The Pollution-Routing problem . . . 10
  - 2.2.2 Stochastic vehicle routing problems . . . 11
  - 2.2.3 Relationship with the state of knowledge . . . 13
- 2.3 Technical background information . . . 14
  - 2.3.1 Calculation of fuel consumption and emissions . . . 14
  - 2.3.2 The impact of travel speed uncertainty on emissions . . . 15
  - 2.3.3 Decision time-lines and modeling of stochasticity . . . 18
- 2.4 Problem description and mathematical models . . . 19
  - 2.4.1 Notations common to the two models . . . 20
  - 2.4.2 One-route, one-speed strategy . . . 20
    - 2.4.2.1 Discrete speeds and discrete recourse . . . 23
    - 2.4.2.2 Discrete speeds and continuous restricted recourse . . . 25
  - 2.4.3 One-route, several-speeds strategy . . . 29
  - 2.4.4 Discussion about the values of constants M . . . 31

2.5 Computational experiments . . . 31
2.5.1 Probability distribution and scenario generation . . . 32
2.5.2 Comparative results for the discretized and continuous recourse strategies . . . 33
2.5.3 Comparative results for the OR-OS strategy with the deterministic model and OR-SS strategy . . . 34
2.5.4 Problem size and difficulty induced by the number of scenarios . . . 36
2.5.5 Correlation between total cost and fuel cost savings . . . 37
2.5.6 The value of optimizing speed . . . 38
2.5.7 The effect of variability in traffic speed . . . 40
2.6 Conclusions . . . 40

3 Integer L-shaped Algorithms for the Pollution-Routing Problem under Traffic Uncertainty 43
3.1 Introduction . . . 44
3.2 Literature review . . . 45
3.3 Methodology . . . 46
3.3.1 The original integer L-shaped method . . . 46
3.3.2 Adaptation to the PRP . . . 47
3.3.3 Integer L-shaped with lower bound . . . 49
3.3.4 Integer L-shaped with lower bound and local branching . . . 51
3.4 Computational experiments . . . 51
3.5 Conclusions . . . 60

4 Speed Optimization under Soft Time Window Constraints 61
4.1 Introduction . . . 62
4.2 Problem description and non-linear model . . . 65
4.3 Discretization schemes and algorithms . . . 66
4.3.1 Discretized speed model (DSM) . . . 67
4.3.2 Discretized arrival times: time-space networks . . . 68
4.3.2.1 Fixed time step (FTS) . . . 68
4.3.2.2 Fixed size graph (FSG) . . . 70
4.3.2.3 Fixed size graph with zooming (FSG-Z) . . . 70
4.3.2.4 Theoretical complexity of time-space algorithms . . . 71
4.3.2.5 Illustrative example . . . 72
4.4 Computational experiments . . . 74
4.4.1 Road transportation . . . 75
4.4.1.1 Instance generation . . . 75
4.4.1.2 Parameters and results . . . 75
4.4.1.3 Methods performance: trade-off between accuracy and time . . . 76
4.4.1.4 Soft time window, penalty and fuel consumption . . . 77
4.4.1.5 Waiting times at customers . . . 80
4.4.2 Maritime transportation . . . 80
4.4.2.1 Methods performance on maritime instances . . . 81
4.4.2.2 Soft time windows vs hard time windows . . . 81
4.5 Conclusions . . . 83

5 Conclusions **85**
5.1 Short summary 86
5.2 Main findings 86
5.3 Research outputs 87
5.4 Research limitations and future research directions 88

Bibliography **91**

List of Figures

1.1 Main themes considered in this thesis 6

2.1 Two solutions of the four-node instance 17
2.2 Example of speed variability 18
2.3 Time-line of the first strategy decision making process 18
2.4 Time-line of the second strategy decision making process 19

3.1 Upper and lower bound as obtained on 20-node instance during solution process by Lshaped_O and Lshaped_A 59
3.2 Upper and lower bound as obtained on 25-node instance during solution process by Lshaped_O and Lshaped_A 59

4.1 Time-space networks example with different discretization techniques . . . 70
4.2 Distance and time window data 72
4.3 Time-space network created using FTS 73
4.4 Time-space network created using FSG 73

List of Tables

2.1 Comparison of the problem characteristics studied in this chapter and related papers . . . 14
2.2 Demand and service time data . . . 16
2.3 Distance matrix for the example of Table 2.2 . . . 16
2.4 Values of the parameters used in the emission model . . . 32
2.5 Comparison between the solutions of the discretized and continuous recourse models . . . 34
2.6 Performance of deterministic approach compared with OR-OS . . . 35
2.7 Performance of OR-SS approach compared with OR-OS . . . 36
2.8 Eight scenarios considered . . . 37
2.9 Problem size and solution time change depending on the number of scenarios 37
2.10 Comparison between deterministic and OR-OS solution values . . . 38
2.11 Savings achieved by considering speed as a decision variable . . . 39
2.12 Savings achieved by considering speed as a decision variable without driver 40
2.13 Comparison between network with two segments per arc and with one segment per arc . . . 41

3.1 Fifteen scenarios considered in instance . . . 53
3.2 Results on 20-node instances . . . 55
3.3 Results on 25-node instances . . . 56
3.4 Results on 50-node instances . . . 57
3.5 L-shaped algorithms robustness against uncertainty . . . 58

4.1 Algorithms run statistics . . . 74
4.2 FSG-Z iterations analysis . . . 74
4.3 Comparison of all algorithms in terms of cost and CPU time . . . 78
4.4 Comparison of costs under three penalty values . . . 79
4.5 Comparison of costs for three time window types . . . 79
4.6 Comparison of costs with disallowed or allowed waiting time at customers 80
4.7 Methods performance on maritime instances . . . 82
4.8 Cost and speed variability with time window length, type and penalty values . . . 83

Declaration of Authorship

I, Moncef Ilies Nasri, declare that this thesis titled, "Route and Speed Optimization Problems under Uncertainty and Environmental Concerns" and the work presented in it are my own and has been generated by me as the result of my own original research original research. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.
- Part of this work has been published as:
  - Nasri, M I. Bektaş, T., Laporte, G., 2018. "Route and Speed Optimization for Autonomous Trucks", *Computers & Operations Research*. (Chapter 2).

Signed : **Moncef Ilies Nasri**

Date : **July 2018**

Acknowledgements

Here, I would like to show my gratitude to a number of people, without whom this work wouldn't have been what it is today. But before that, I would like to thank Allah, the most high, for bringing me where I am, and allowing me doing this work and surrounding me with amazing people that helped me evolve as a student and as a person.

I would like first to express my utmost and sincere gratitude to my main supervisor Professor Tolga Bektaş who has been supporting me since my first day. I would like to thank him for his insightful comments, bright ideas, availability and for always directing me in the right directions. He did all of that, not because I told him that I was originally from the city of *Tolga*, but because of his professional integrity and his love for research. I would like then to thank my second supervisor Professor Gilbert Laporte, whom I always considered as a legend, especially when I was working on my master's project on logistics in Algiers. Having a personal meeting with him years was merely a dream. I would like to thank him for giving my ideas a chance, and for being always present when I needed advice, and for replying to my email faster than I replied to his. I will never be able to thank my supervisors enough for what they did for me!

I would like to thank Professor Jörg Fliege and Dr Dimitris Paraskevopoulos for accepting to be my examiners, and for taking time to read this present manuscript carefully.

I want also to thank the University of Southampton for funding this thesis, and its staff for always taking my needs into consideration.

Last but not least, I would like to thank my colleagues and fellow PhD students at the business school, and more particularly those in my office as they helped me in creating a wonderful and productive atmosphere.

I could have said "To my family and friends" and passed on, but people I dedicate my work to are just more important than that!

This thesis marks the end of four years of research globe-trotting that took me from Algiers to Southampton, passing by Toulouse and Montreal. Cities that will remain in heart for ever.

First and foremost, I would like to dedicated this work to my family. To my beloved parents, for their constant support. To my wonderful sister, for always being here to listen to my thoughts and moaning. To my dear wife, for her unconditional love and support, for taking away PhD loneliness and for creating a peaceful environment for us, which cannot be called "home" if she is not present. To my brothers for always being available to give me advice. To all my family members for their continuous support.

Last but not least, to all my friends from the four cities mentioned above. They are so far from each other geographically that my heart is the only place that can bring them together!

List of Abbreviations

ALNS	Adaptive large neighborhood search
AS	Average speed
AT	Autonomous truck
ATVC	Average time window violation per customer
BCP	Branch-cut-and-price
CO_2	Carbone dioxide
DC	Driver cost
DSM	Discretized speed model
EA	Emission controlled area
FC	Fuel cost
FSG	Fixed size graph
FTS	Fixed time step
GHG	Greenhouse gases
HWC	Hard-worst case
IMO	International maritime organization
IPOPT	Interior point optimizer
LB	Lower bound
NLM	Non-linear model
OR-OS	One-route, one-speed
OR-SS	One-route, several-speed
PRP	Pollution-routing problem
RHH	Rolling horizon heuristic
SOA	Speed optimization algorithm
SOP	Speed optimization problem
SOPSTW	Speed optimization problem with soft time windows
SSOP	Stochastic speed optimization problem
SWC	Soft-worst case
TDVRP	Time-dependent VRP
UB	Upper bound
VRP	Vehicle routing problem
VSS	Value of the stochastic solution

Chapter 1

Introduction

1.1 Research context

Road transportation is one of the most common means of transport for freight (Spielmann and Scholz, 2005), but is also one of the most polluting. A report published by the U.S. Environmental Protection Agency in April 2016 (U.S. Environmental Protection Agency, April, 2016) shows that the transportation sector was the second largest contributor of greenhouse gases (GHG) emissions in 2014, with a 26% share of the total. GHG emissions have increased by 17% between 1990 and 2014, in which period a sharp increase of 76% was also observed in heavy- and medium-duty trucks emissions. Continuously deteriorating air quality is believed to result in millions of deaths each year across the world, and these numbers will double by year 2050 if the current trend holds (Lelieveld et al., 2015). Many logistics companies and policy makers have started taking actions to reduce the amount of GHG emissions.

Distribution activities represent a significant share of the cost incurred by companies. It is for this reason that the Vehicle Routing Problem (VRP) and its variants are among the most studied optimization problems in the literature (Toth and Vigo, 2014). However, ever since it was introduced by Dantzig and Ramser (1959), the models and solution techniques proposed for the problem assumed objective functions representing economic measures (e.g, distance or cost), and ignored the impact on the environment (Bektaş et al., 2016).

The amount of GHG emitted by a vehicle is directly related to the quantity of fuel consumed, which in turn is affected by a number of parameters including speed and load. Bektaş and Laporte (2011) introduced the Pollution-Routing Problem (PRP), an extension to the classical VRP where the traditional, profit maximization objective function is extended to explicitly minimize fuel consumption. The PRP aims at optimizing vehicle routes and speeds to reduce GHG emissions.

Selecting the optimal travel speed of a vehicle depends on the traffic conditions in the network, which was not taken into account in the original version of the PRP. Vehicle speed is often affected by unforeseen and random factors such as weather-related events or traffic congestion. These events deteriorate traffic conditions, which will prevent vehicles from traveling at desired speeds and force them to travel at the traffic speed. This may result in a cost increase, which may render optimal solutions computed by ignoring the uncertainty sub-optimal. In addition, unpredictable delays may be experienced at customers, which reduces the quality of the service provided. Therefore, incorporating uncertainty in the optimization process may increase the robustness of planned solutions against speed variations.

Although several studies have looked at VRP variants under uncertainty and environmental concerns (Hwang and Ouyang, 2015; Ehmke et al., 2016a; Eshtehadi et al., 2017;

Huang et al., 2017), no approach, to the best of the author's knowledge, has considered route and speed optimization to tackle the PRP under uncertain traffic conditions. In this thesis, we investigate this problematic and provide mathematical models and formulations, methods, algorithms and tools to bridge identified research gaps in this context.

1.2 Potential application for autonomous trucks

Autonomous trucks (ATs) are an emerging technology with the potential to revolutionize the transportation sector. In Europe, a convoy of more than 12 ATs completed a European cross-border journey in April 2016 (The Guardian, 2016). The ATs formed platoons of two or three connected trucks that drove close to each other, using the same speed, acceleration and steering profile. This was achieved by the use of an existing technology of adaptive cruise control where the trucks were connected via wireless technologies with a driver sitting in the lead truck. In such a setting, fuel savings of around 5% can be achieved by the lead truck, and between 10–15% for the follower trucks (Tsugawa, 2012). In the US, the first freight shipment with an AT was made on the 25th of October 2016 (Wired.com, 2016). The truck was able to drive without driver assistance on the highway. It was able to assess traffic conditions, choose acceleration and speed, and had the ability to self steer to stay in the driving lane. The truck, however, needed a driver to control it in complex driving areas, such as cities.

The emergence of ATs can also address the problem of truck drivers shortage. In the US, there was a shortage of 48,000 truck drivers in 2015, which is predicted to be 175,000 by 2024 under the current trend (Costello and Suarez, 2015). ATs can help increase the overall traffic safety. Traffic crashes including trucks represent 10% of all crashes in the US, and more than 90% of accident causes are due to driver errors (Singh, 2015).

The introduction of ATs is likely to significantly improve transport efficiency and reduce pollution. To capture these two aspects, in addition to determining routes, optimal speeds at which ATs should travel on each leg of a route should be computed. However, selecting the optimal travel speed depends on the traffic conditions in the network. In practice, vehicle speed is often affected by factors that are not known with certainty in advance, such as weather-related events or traffic congestion. In addition, customers may experience large and unpredictable delays. Consequently, it makes sense to incorporate uncertainty in the optimization process to ensure that the planned solutions are robust against speed variations.

There have been several strategies proposed for the implementation of ATs, including lane reservation (Fang et al., 2013) and exclusive assignment of existing infrastructures (Wu et al., 2017), although both are deemed to be expensive and restrictive with respect

to the applicability of this new technology (Vanholme et al., 2013). A preferred feasible strategy is for ATs to share the same infrastructure with conventional vehicles (driven by humans) (Vanholme et al., 2013), at least for the foreseeable future. The latter option would introduce uncertain traffic conditions for ATs, which is precisely what our research studies.

1.3 Research aims

As highlighted in the analysis of the research context, the lack of studies with regards to vehicle route and speed optimization problem under uncertainty represents a clear research gap. In Chapters 2 and 3, we aim to address this gap and contribute to the state-of-art with new models and solution techniques for pollution-routing problems where the uncertainty is caused by the traffic conditions. Additionally, in Chapter 4, we study the problem of optimizing vehicle speed over a fixed customer sequence with soft time windows. We propose generic solution methods with application in different transportation modes, including road and maritime transportation. The thesis adopts the three-paper structure, where the details of research aims of each paper are given below.

The first paper first reviews the existing literature on (green) vehicle routing problems under uncertainty. We then study the Pollution-Routing Problem under stochastic traffic conditions. We aim to quantify the fuel consumption savings that can be achieved by explicitly considering traffic uncertainty. Managerial insights are derived by comprehensive computational results.

The second paper builds on the encouraging findings of the first paper, where only small scale instances were solved. In the second paper we develop specialized techniques to solve larger instances of the problem. These techniques exploit the structure of two-stage stochastic programs.

The third paper studies the speed optimization problem with soft time-windows that arises as a subproblem in the first and second paper. The problem is also encountered in several logistics applications. We provide an overview of existing techniques to solve similar problems. We then propose algorithms capable of optimally solving large instances of the problem faster than commercial solvers. Managerial insight is derived for road and maritime transportation.

1.4 Research objectives

The research objectives of Chapter 2 are:

- to explore the state-of-the art and provide an overview of stochastic green vehicle routing problems,
- to propose modeling concepts and stochastic mathematical programming models for the PRP with stochastic travel times and to overcome modeling challenges such as non-linearity,
- to provide comprehensive computational analysis of the method proposed and provide insight.

The research objectives of Chapter 3 are:

- to review the literature on implementations of decomposition methods for VRPs,
- to develop an integer L-shaped based algorithm and variations to solve medium to large-scale instances of the problem introduced in Chapter 2,
- to provide extensive computational analysis and comparison of the algorithms developed.

The research objectives of Chapter 4 are:

- to review the literature on speed optimization problems in different transportation modes,
- to overcome the non-linearity of the mathematical model,
- to develop time-space network approaches for the problem,
- to run extensive computational result and compare with non-linear available solvers,
- to provide insight for road and maritime transportation.

1.5 A schematic representation of the thesis

In Figure 1.1 we present a schematic representation of the overall research conducted, where each rectangle represents a subject treated in this thesis. The overall theme of this thesis is to reduce greenhouse gases emissions and is represented by the green rectangle. To achieve this aim, we consider in the Chapter 2 route and speed optimization under uncertainty, represented by the black rectangle. In Chapter 3 we consider the same problem, but we develop an Integer L-shaped method to solve the problem for larger instances, represented by the yellow rectangle. Chapter 4 focuses on speed optimization over a fixed sequence of customers, a general problem that arises in different transportation modes, represented by the red rectangle.

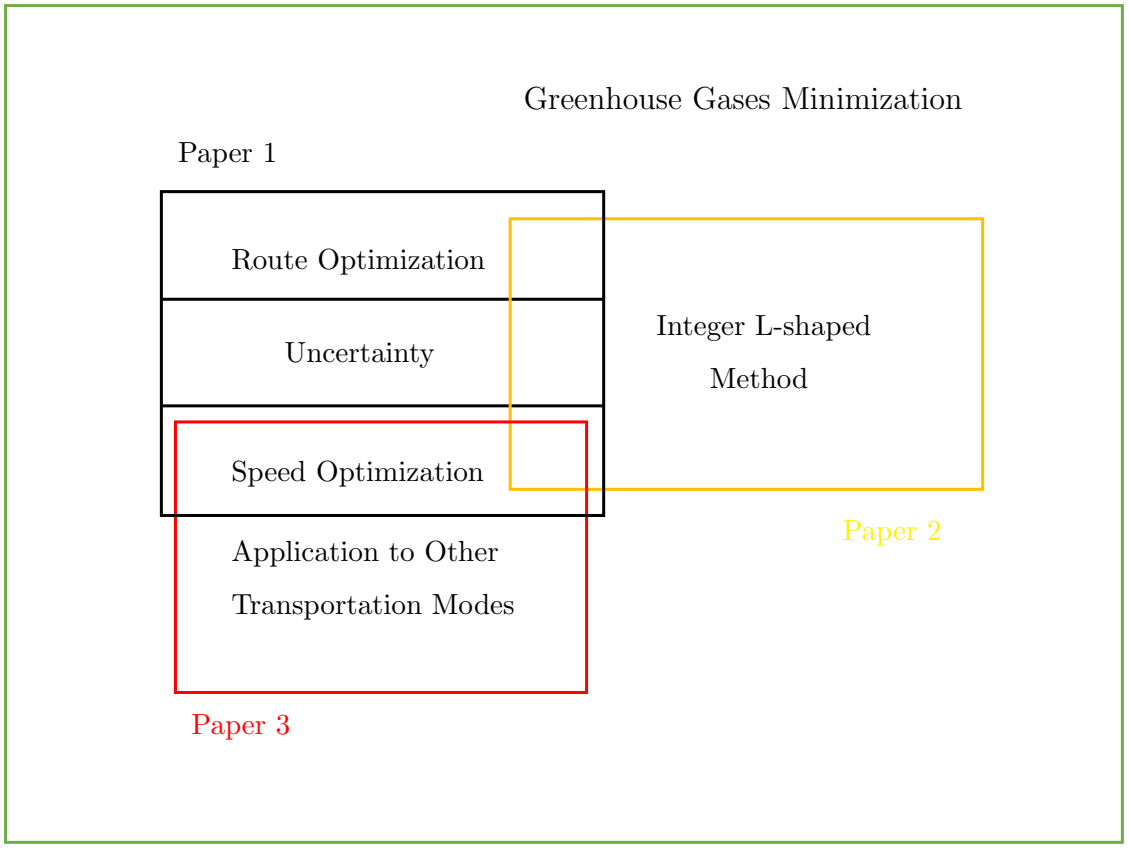


Figure 1.1: Main themes considered in this thesis

Chapter 2

The Pollution-Routing Problem under Traffic Uncertainty

Abstract

The Pollution-Routing Problem (PRP) is an extension of the classical Vehicle Routing Problem with a more comprehensive objective function that accounts for the amount of greenhouse gases emissions, fuel, travel times and their costs. This chapter introduces a new variant of the PRP that takes into account travel time uncertainty due to traffic congestion. Uncertainty is modeled through stochastic traffic speeds on the arcs which are represented by a discrete set of scenarios. Using two different strategies, this study provides three mathematical formulations in the form of two-stage stochastic programming models. The two strategies differ in the amount of information available during the decision making process and in the recourse policies. The objective is to minimize the expected total cost of drivers, fuel costs and greenhouse gases emissions. Computational results show the added value of stochastic modeling over a deterministic approach.

Keywords: pollution-routing; stochastic programming; stochastic traffic speed; recourse.

2.1 Introduction

The Pollution-Routing Problem (PRP) introduced by Bektaş and Laporte (2011) is a variant of the classical Vehicle Routing Problem (VRP) that explicitly considers driver, fuel and pollution costs. In addition to determining vehicle routes, the PRP computes optimal speeds at which vehicles should travel on each leg of the route, so as to minimize a comprehensive cost function that includes fuel consumption. In its original version, the PRP includes time windows and imposes limits on vehicle speeds on every leg of each route, which are assumed to be constant and known with certainty at the time of planning.

Constant traffic speeds imposed in the PRP correspond to the minimum and maximum speeds at which vehicles may legally travel on a given road segment. In practice, however, vehicle speed is often affected by factors that are not known with certainty in advance, such as weather-related events or traffic congestion. Such events are likely to reduce the speed of vehicle and cause deterioration in the general traffic speed. In this case, the vehicle has to travel at a speed less or equal to the traffic speed. Ignoring this reality at the planning stage may render optimal solutions computed using deterministic parameters sub-optimal or even infeasible, and in turn may result in cost increases, depending on the speeds that are achievable in practice. In addition, customers may experience large and unpredictable delays. Consequently, it makes sense to incorporate uncertainty in the optimization process to ensure that the planned solutions are robust against speed variations.

Considerable work has been carried out on the stochastic VRP Gendreau et al. (2014, 2016), in particular and of relevance to this study, in what regards stochastic travel times Laporte et al. (1992); Ehmke et al. (2015). However, vehicle speed was only recently introduced as a decision variable, only few studies are yet available on the effect of stochastic traffic speed.

In order to bridge this gap, this chapter models uncertainty as stochastic input parameters. We describe two stochastic programming models that differ with respect to the time at which the routing decisions are made. We consider a finite number of scenarios, each with a different speed limit for each arc of the network. The first formulation is a two-stage stochastic programming model with complete recourse. In this case, the recourse variables correspond to the delays experienced in servicing the customers, and to the speed reduction needed when the planned speed is higher than the realized speed. The second formulation is also a two-stage stochastic programming model in which travel speeds are computed for every scenario separately, assuming that the realizations of the scenarios are revealed before the start of routing, in the expectation that further cost minimization will be achieved.

The three key contributions of this chapter are as follows. (1) We describe two recourse strategies to mitigate the adverse effects of speed uncertainty on fuel consumption and GHG, where speed reductions, to the best of our knowledge, have been made an integral part of the recourse decisions for the first time, (2) using a stochastic programming methodology, we describe several two-stage stochastic programming formulations with recourse, and present results that numerically compare the recourse strategies, and (3) we offer insights into the relative performance of each strategy and discuss implications of the strategies on fuel consumption and total cost.

The remainder of this chapter is structured as follows. Section 2.2 presents a brief review of the relevant literature on the PRP and on the VRP with stochastic travel times. Section 2.3 introduces some technical background information. Mathematical models are described in Section 2.4. Section 2.5 presents computational experiments, and conclusions follow in Section 2.6.

2.2 Literature review

In this section, we review some of the relevant existing literature of the PRP and its variants, given the similarities to the problem studied in this chapter. Our study contains also stochastic travel times within a VRP framework, therefore it can also be viewed as a natural extension of the VRP with stochastic travel times. Therefore, we also review the relevant literature on this problem. For recent reviews of the literature on green road freight transportation, we refer the reader to Demir et al. (2014b) where the authors

review gas emissions models, and to Lin et al. (2014), Eglese and Bektaş (2014) and Bektaş et al. (2016) which are surveys of green VRPs. For a wider and comprehensive overview of the green transportation, including air and maritime transportation, we refer the reader to Psaraftis (2016).

2.2.1 The Pollution-Routing problem

In the original paper of Bektaş and Laporte (2011), the PRP makes use of the comprehensive fuel consumption model proposed by Scora and Barth (2006), and is modeled through a non-linear mathematical formulation in which the decision variables correspond to the selection of routes and speeds. Non-linear constraints are linearized using speed discretization. The authors presented extensive computational results on instances involving up to 20 customers and showed that CO_2 emissions could be reduced by up to 8% by solving a comprehensive model that accounts for pollution, which is significant since the transportation sector has an important share in the global CO_2 emissions (U.S. Environmental Protection Agency, April, 2016). Instead of discretizing speed, Fukasawa et al. (2016b) tackled the PRP with continuous speed and put forward two arc-based mixed-integer convex optimization models that can be solved as disjunctive convex programs, and can handle instances with up to 25 customers. Dabia et al. (2016) described a branch-and-price algorithm for the PRP based on a column generation mechanism in which the master problem is a set partitioning problem. The pricing problem is solved by means of modification of a labeling algorithm for the elementary shortest path problem with resource constraints. The algorithm can optimally solve all instances with 10 customers, as well as some instances from the PRP-lib with 15 and 20 customers.

Demir et al. (2012) developed an adaptive large neighborhood search (ALNS) metaheuristic for larger PRP instances which embeds an adaptation of the speed optimization algorithm of Norstad et al. (2011). The algorithm provided solutions for realistic instances with up to 200 customers within just over 10 minutes. Kramer et al. (2015) later proposed a matheuristic that combines an iterative local search, based on the randomized variable neighborhood descent heuristic of Subramanian et al. (2010), with a speed optimization algorithm and an exact method for a set partitioning problem. When compared with the ALNS of Demir et al. (2012), their matheuristic yielded better quality solutions on all instances with 100 and 200 customers.

Franceschetti et al. (2013) studied a version of the PRP with time-dependent speeds which assumes two periods, one characterized by a congestion speed, and the other when a vehicle can travel at an optimized free-flow speed. The authors used the emissions model of Scora and Barth (2006) to compute fuel consumption. In this problem, service at customers must start within specified time windows, and it is possible for a vehicle to wait at customer *after* completing service to avoid congestion further along the route. The authors also studied a special case of the problem with a single customer

and analytically characterized optimal solutions. They described a speed optimization algorithm for the problem and solved to optimality instances with up to 20 customers.

Qian and Eglese (2014) tackled the problem of minimizing fuel consumption under time-varying speeds. The authors considered both vehicle routes and speeds as decision variables. They proposed two algorithms based on dynamic programming, one of which is exact and the other is heuristic, and tested both on a realistic 14-customer instance. Their results showed that the heuristic is much faster than the exact method, taking five minutes instead of 12 hours. The same authors (Qian and Eglese, 2016) later proposed a combined tabu search and column generation algorithm for this problem, based on an earlier algorithm by Prescott-Gagnon et al. (2009) for the VRP with time windows. The algorithm was tested on a real instance with 60 customers and one depot, and the results indicated about three percent savings in GHG emissions compared with a time-minimizing objective.

Maden et al. (2010) considered the VRP with time windows and time-dependent speeds, and developed a tabu search heuristic in which the objective is to minimize the total time spent by the vehicles. The authors showed that up to 7% of CO_2 emissions can be saved on the instances tested.

2.2.2 Stochastic vehicle routing problems

The most common sources of uncertainty in stochastic vehicle routing are customer availability, customer demand, service times and travel times (Gendreau et al., 2016). Here we focus on VRPs with stochastic travel times. In the case of stochastic customer demands, the recourse policies are decided before starting the routing. A recourse action is needed if vehicle capacity is not always sufficient to satisfy customer demand. Laporte and Louveaux (1990), Dror et al. (1993) and Yang et al. (2000) used a preventive return trips to the depot when vehicle load falls below a certain threshold. This policy prevents vehicles from being unable to meet a customer demand, but may result in unnecessary travel. Dror et al. (1989); Secomandi (2001) Described policies by which the remaining route is reoptimized after each customer delivery, but this tends to be computationally very expensive and therefore rarely used.

Laporte et al. (1992) considered stochastic travel and service times in a VRP in which the route durations are limited. They described a chance-constrained model and two two-stage models with simple recourse, i.e. the cost of recourse is proportional to the travel time in excess of the imposed route duration limit. The authors implemented a branch-and-cut algorithm and presented computational results on instances with up to 20 customers. Ehmke et al. (2015) also used a chance-constrained model in which the aim is to ensure a certain service level for each customer by computing the random distributions of both arrival and start-service times. They developed a tabu search based

algorithm which was applied to large instances. The authors pointed out that given wide time windows, service improvements can be achieved without a significant increase in the cost of the solution.

Taş et al. (2013) described a tabu search algorithm for a variant of the problem where customer service can start either before or after the time windows with a penalty. The objective is to minimize the cost incurred by the total distance, the expected delay, and the expected drivers' overtime. Taş et al. (2014) later described a branch-and-price algorithm based on column generation for the same problem. Agra et al. (2013) considered a similar problem in a maritime context. The authors proposed a robust optimization approach that determines a set of routes feasible for all realizations of stochastic travel times. Lee et al. (2012) developed a robust optimization algorithm based on a column generation technique considering uncertainty in both travel times and customer demands.

Hwang and Ouyang (2015) considered the full-truck delivery problem under uncertainty, defined on a road network where nodes represent major intersections and directed arcs represent route segments between them. Speed congestion is modeled as an independent random variable on each arc of the network. The problem is to select the sequence of arcs to traverse in order to minimize an objective function based on expected total travel time, GHG emissions and penalties for late or early arrivals. The authors described a stochastic dynamic approach and a deterministic shortest path heuristic to solve problem instances with up to 416 nodes. The heuristic approach provided solutions quickly, but of lower quality as compared to the dynamic approach. Savings between 4% and 8% were obtained when compared to the same approach without considering emissions.

Ehmke et al. (2016a) developed two algorithms to find expected emissions-minimizing paths in urban areas with stochastic travel times. They developed an A^* algorithm based on sampling travel times to estimate the expected emissions on each arc of the graph and a second heuristic based on Dijkstra's algorithm (Dijkstra, 1959) to solve a time-dependent but deterministic version of the problem, using expected emissions on each arc. This simplification was made to reduce the total time required to solve the problem. The heuristic method showed similar results but the running times were significantly faster than for the A^* algorithm.

Ehmke et al. (2016b) considered the VRP in urban areas, with the objective of minimizing gas emissions. Using the same model as in Bektaş and Laporte (2011), a heuristic is used to estimate the emissions-minimizing paths, thus transforming random traffic speeds to deterministic time-dependent speeds. Solutions obtained with a tabu search heuristic on real-world instances showed that emissions can be reduced significantly without a significant increase in cost.

Eshtehadi et al. (2017) considered the PRP under demand and travel time uncertainty. The authors focused on stochastic customer demand and proposed three mathematical models based on a robust optimization approach. The authors defined a hard-worst case (HWC), a soft-worst case (SWC) scenario and a chance-constrained robust model. The difference between these approaches is their robustness to uncertainty. Travel time uncertainty is handled by considering a deterministic congestion level. The results showed that considering robust optimization technique to provide reliable solutions may result in 30, 50 and 60 liters of additional fuel consumption for 10-, 15- and 20-node instances respectively.

Huang et al. (2017) considered the Time-Dependent VRP (TDVRP) with path flexibility. As opposed to considering a simple path between a pair of customers, the authors start first by determining the set of time-dependent shortest paths on the network using a modified version of Dijkstras algorithm. The authors describe a mathematical model to the TDVRP with path-flexibility and a two-stage stochastic program to solve a stochastic variant in which speeds on arcs are represented by discrete scenarios. The first-stage decisions relate to customer sequences and the second-stage decisions relate to the selection of paths. Instances with 30 nodes were solved by the deterministic version, and instances with 10 nodes and three discrete scenarios were solved by the stochastic program. The deterministic equivalent of the stochastic PRP with path flexibility yielded better results than solving a standard TDVRP.

2.2.3 Relationship with the state of knowledge

Although the papers just reviewed share some similarities with the problem we introduce here, they exhibit two fundamental differences with respect to the types of decisions and the cost components. The first of these is vehicle speed. The authors of the above papers argue that speed cannot be controlled in urban areas and vehicles must follow the traffic speed. In our study, we consider an intercity travel setting, such as highways, where vehicles are more likely to have control over the speed. From a methodological point of view, modeling speed as decision variable is challenging due to the non-linearity it entails. The second fundamental difference relates to the cost. In particular, we consider driver wage as part of the objective function. While this may seem a relatively straightforward modification of the objective function, it does make a difference since it has been shown that the largest component of the PRP objective function is the drivers wage (Bektaş and Laporte, 2011). Table 2.1 summarizes the differences between the papers cited above and our paper.

Our work builds on and contributes to the existing body of work in a number of aspects. First, we continue to treat speed as a decision variable, as was done in the PRP and other relevant studies reviewed in Section 2.1 (e.g. (Demir et al., 2012), (Fukasawa et al., 2016b)). However, we break away from the main assumption made in those references

Paper	Objective function		Time	Speed	Path	Time	Solution method	
	Cost	Fuel	dependency	decisions	flexibility	windows	Exact	Heuristic
Ehmke et al. (2016a)	-	✓	✓	-	-	-	-	✓
Hwang and Ouyang (2015)	✓	✓	-	-	✓	-	✓	✓
Eshtehadi et al. (2017)	-	✓	-	✓	-	✓	✓	-
Huang et al. (2017)	-	✓	✓	-	✓	-	✓	✓
This chapter	✓	✓	-	✓	-	✓	✓	-

Table 2.1: Comparison of the problem characteristics studied in this chapter and related papers

in that the vehicle speed is bounded by a constant value. One novel aspect of our contribution is to restrict the vehicle speed on each arc by a stochastic parameter, the value of which depends on a future realization of a scenario that prescribes the traffic speeds. This makes for a more realistic treatment of the problem by reflecting part of the uncertainty of the traffic conditions in the relevant models. Second, we assume that the travel times are stochastic by following a line of analysis similar to that of the references reviewed in this section. However, whereas the references cited above treat the travel time on an arc as an uncertain input *parameter*, we model it as a function of the speed chosen to traverse that arc, implying that travel time itself is effectively a *decision variable*, but one that is subject to uncertainty. The latter aspect of our contribution is therefore different and unique. While the robust approach proposed by Eshtehadi et al. (2017) for the pollution-routing problem considers speed as a decision variable, it tends to be over-conservative since it assumes the realization of a worst-case scenario. Robust solutions are therefore costlier than those produced by the stochastic programming methods we advocate in this chapter. Finally, we resort to two-stage stochastic programming formulations with recourse as the modeling framework, for the very reason that this approach is shown to be a suitable way to formulate such problems (e.g. (Laporte et al., 1992)), and with an objective function that minimizes the expected cost (e.g. (Laporte et al., 1992), (Taş et al., 2013)).

2.3 Technical background information

In our study, we use the same consumption model as in Bektaş and Laporte (2011), which we now describe. From this model and by using an illustrative example, we show also the impact that stochastic travel times may have on the amount of emissions. We then propose two ways of modeling our problem.

2.3.1 Calculation of fuel consumption and emissions

There exist numerous emission models, which were reviewed and compared in Demir et al. (2014b). These models vary with respect to the input parameters and level of

detail, but most use non-linear functions of vehicle speed. Such functions generally have a convex U-shaped form, which can be optimized to yield a speed minimizing fuel consumption and emissions. Our emissions model is an instantaneous model proposed by Barth and Boriboonsomsin (2008), which estimates the fuel consumed F_r per second based on the function

$$F_r = \varepsilon(k\Omega V + P/\eta)/\kappa \text{ (L/s)}, \tag{2.1}$$

where η and κ are constants related to diesel fuel typically used by delivery vehicles, ε is fuel-to-air mass ratio, k, Ω and V are the engine friction factor, engine speed and engine displacement respectively. Furthermore, P is the engine power output per second (in kW) and is calculated as

$$P = P_{\text{tract}}/\eta_{tf} + P_{\text{acc}} \text{ (kW)}, \tag{2.2}$$

where η_{tf} is the vehicle drive train efficiency, and P_{acc} is the engine power demand associated with running losses of the engine and the operation of vehicle accessories (i.e., air conditioning). In our study, we assume that P_{acc} is equal to zero. The parameter P_{tract} is the total tractive power requirements (in kW) placed on the wheels:

$$P_{\text{tract}} = (\Lambda\tau + \Lambda g \sin\theta + 0.5 C_d \rho \Gamma v^2 + \Lambda g C_r \cos\theta)v/1000 \text{ (kW)}, \tag{2.3}$$

with Λ and v are the vehicle weight (kg) and speed (m/s) respectively, τ is the acceleration (m/s^2), θ is the road angle, g is the gravitational constant (m/s^2), and C_d and C_r are the coefficients of the aerodynamic drag and rolling resistance, respectively. Finally, ρ is the air density (kg/m^3), and Γ is the frontal surface area of the vehicle (m^2). For a given arc (i, j) of length d and assuming that all parameters remain constant except for the vehicle speed v, we can express fuel consumption in liters (L) as

$$F(v) = k\Omega V \lambda d/v \tag{2.4}$$

$$+ (w\alpha v + \alpha f v + \beta v^3)\lambda\gamma d/v \text{ (L)}, \tag{2.5}$$

where λ and γ are constants defined as $\lambda = \varepsilon/\kappa\Psi$, where Ψ is the conversion factor of fuel from (g/s) to (L/s), and $\gamma = 1/(1000\eta_{tf}\eta)$. Furthermore, if Λ is the total weight of vehicle between node i and j, then $\Lambda = w + f$, where w is the weight of an empty vehicle (curb weight), and f is the vehicle load. Let $\alpha = \tau + g\sin\theta + gC_r\cos\theta$ be a vehicle-arc specific constant and $\beta = 0.5 C_d \rho \Gamma$ be a vehicle specific constant. The introduction of d/v in Equations (2.4) and (2.5) is done to express F(v) in liters (L) instead of liters/seconds (L/s). We omit indices (i, j) on the variables v, d, f, and α to simplify the presentation.

2.3.2 The impact of travel speed uncertainty on emissions

Traffic congestion is one of the main factors preventing vehicles from driving at an optimal speed. It can be caused by accidents, bad weather, or simply the traffic level

in the network. Congestion is typically characterized by low speeds and regular start-and-stops. As a result, it drastically increases fuel consumption compared with the non-congestion case.

We now illustrate how congestion may impact the routing decisions on a four-node instance corresponding to four British cities, where the fictional demand of each customer and the service times for this instance are given in Table 2.2. The time windows of the customers (in minutes) are shown in Figure 2.1a. The distance matrix of Table 2.3 shows the distances (m) between the nodes of the network.

Node index	City	Demand (kg)	Service time (min)
0	Kingston upon Hull (depot)	–	–
1	Pocklington	721	24
2	Brough	814	27
3	Selby	620	21

Table 2.2: Demand and service time data

Node index	0	1	2	3
0	0	41150	25680	54200
1	40660	0	51980	32800
2	25010	51780	0	61520
3	54270	32750	61560	0

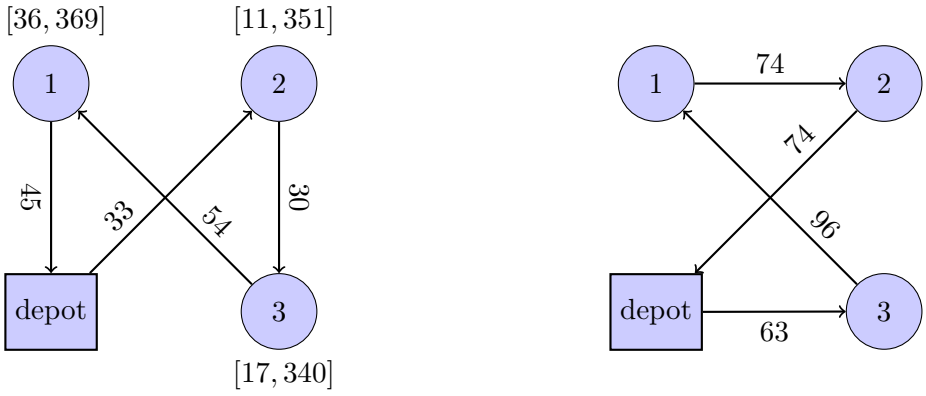
Table 2.3: Distance matrix for the example of Table 2.2

We assume that the traffic speeds on all the arcs of the network is not known in advance, but is instead a discrete random variable referred to as a scenario. In this example, we consider three scenarios S_1, S_2 and S_3 with respective probabilities 33%, 33% and 34%. Below, we show the matrices representing the three scenarios S_1,S_2 and S_3 for this example where each entry (i, j) in a matrix represents the traffic speeds (km/h) on each arc (i, j). The scenarios S_1, S_2 and S_3 are derived from a probability distribution discussed further in Section 2.5.1.

$$S_1 = \begin{pmatrix} 0 & 22 & 28 & 35 \\ 21 & 0 & 40 & 23 \\ 67 & 56 & 0 & 59 \\ 46 & 37 & 29 & 0 \end{pmatrix} \quad S_2 = \begin{pmatrix} 0 & 59 & 48 & 60 \\ 31 & 0 & 53 & 28 \\ 43 & 46 & 0 & 33 \\ 41 & 70 & 30 & 0 \end{pmatrix} \quad S_3 = \begin{pmatrix} 0 & 22 & 21 & 50 \\ 39 & 0 & 60 & 23 \\ 37 & 21 & 0 & 70 \\ 33 & 29 & 58 & 0 \end{pmatrix}.$$

The problem is to find an optimal route and the optimal speeds on each arc, where the chosen speeds are constrained by the traffic conditions. One way to solve the problem is by creating an average scenario S_{av} where the traffics speeds on each arc (i, j) is the average of all traffic speeds over all scenarios, taking into account their respective probability and solve a deterministic PRP. S_{av} is represented by a matrix where each item $S_{av}(i, j) = 0.33S_1(i, j) + 0.33S_2(i, j) + 0.34S_3(i, j)$.

Figure 2.1 shows two optimal solutions obtained by using £1.4 as the combined unit cost of fuel (per litre) and CO_2, and £10 as the hourly driver cost. The solution shown in Figure 2.1a is obtained by solving the deterministic PRP where the traffic speeds are prescribed by the matrix S_{av}, and are imposed as upper limits on the speeds that can be chosen on each arc. The optimal speeds are shown on the arcs. The cost of this solution is £99.96, and its *expected* value when calculated under the realization of S_1, S_2 and S_3, is equal to £121.58. However, this approach is suboptimal as it does not fully utilize all the available information. In contrast, Figure 2.1b shows the solution generated using stochastic programming where the three scenarios S_1, S_2 and S_3 are explicitly taken into account during the solution process. This solution has an expected cost equal to £106.70, which represents a saving of 12.26% over the deterministic. The stochastic program used to produce the latter solution will be further explained in Section 2.4.2.1.



(a) The solution obtained by considering S_{av}

(b) The solution obtained by considering the scenarios (S_1, S_2 and S_3)

Figure 2.1: Two solutions of the four-node instance

Variability of traffic speed on a given intercity road can be better represented by dividing an arc between two customers into smaller segments. The traffic speed will then be defined for every segment, where for each segment an optimal speed can be prescribed. To illustrate, we consider the example in Figure 2.2, which shows an arc $(1, 2)$ of 100km length, on which we assume the traffic speed to be 50 km/h. Suppose now that further information is available about the variability of traffic speeds on this arc such that it can be divided into three segments of 50km, 20km and 30km, where the traffic speed on each segment is 40 km/h, 60 km/h and 50 km/h, respectively. If we consider a vehicle traveling at 50 km/h, assuming a constant traffic speed throughout the arc, the combined cost of fuel, CO_2 and driver is £34. If however, we consider the three segments and that the vehicle travels at the traffic speed on each of the segments, the total cost is £37, which is more accurate, but is at the expense of a 9% increase.

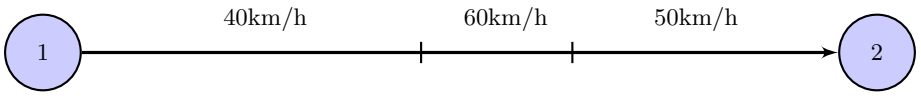


Figure 2.2: Example of speed variability

2.3.3 Decision time-lines and modeling of stochasticity

Planning vehicle routes without taking the congestion effect into account may lead to poor quality solutions when implemented. In addition, the assumption that one has perfect information on congestion data is not realistic, due to the difficulty associated with accurately predicting travel times. In this study, we use a strategy based on stochastic programming that is able to represent the variability in the traffic speeds. Stochastic programming has been extensively used in a variety of contexts, and has yielded good results on routing problems (Gendreau et al., 2016).

Traffic speeds are modeled as discrete random variables. They are represented by a finite set S of scenarios, where each scenario $s \in S$ has an associated probability $p_s > 0$ of occurring. In our context, a scenario s is characterized by a matrix of traffic speeds in which every entry (i, j) corresponds to an upper bound u_{ij}^s on the maximum speed that can be attained on arc (i, j) which depends on the traffic conditions on the corresponding route leg.

We now describe two ways of modeling the problem as a stochastic program, as well as the possible recourse actions. To clarify the difference between the proposed strategies, we use the following notation. We define a planning horizon of D units of time. The time at which the routing decision are made and the time where speed decisions are made are denoted T^p and T^s, respectively, where $T^p \leq T^s < D$. The random variables realization are revealed at time T^r, where $T^r \geq T^s$. We denote by T^e the time at which the solution is executed (i.e., when vehicles leave the depot to visit the customers), where $T^e \geq T^r$.

Under the first strategy, illustrated in Figure 2.3, we assume that the realization of the random variables will be revealed after the routing plan has been finalized, which also includes the speeds of travel on each arc of the routes. No further action can be taken to change the routes and speeds to improve the solution after this point. We will refer to this strategy as "one-route, one-speed" (OR-OS).

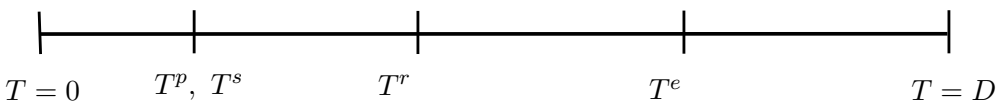


Figure 2.3: Time-line of the first strategy decision making process

Under the second strategy, we also assume that the scenarios are revealed after the routes have been designed. However, before executing the routes, further actions can be taken to reoptimize the speeds, as shown in Figure 2.4. This situation occurs in practice when the vehicles must be loaded the day before the routing plans are operationalized, and the information about traffic speeds is revealed before the start of routing. In this case, the routes cannot be redesigned assuming that the goods are already loaded in the vehicles. We will refer to this strategy as "one-route, several speeds" (OR-SS). This strategy is unlikely to be applied in practice given that the information required is almost impossible to acquire. However, the aim of considering such a strategy is to assess the value of the added information and the savings that it can provide.

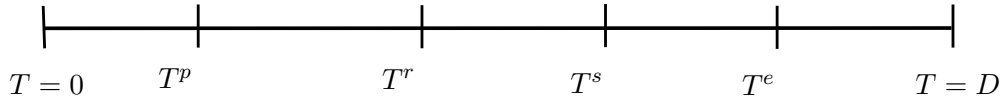


Figure 2.4: Time-line of the second strategy decision making process

While allowing to reoptimize routes in a "several-route, several speeds" after visiting customers may yield further savings, we do not consider this setting in our work for two reasons. First, the goods in a delivery vehicle are usually loaded in a specific order so that the driver will unload goods of the first customers first. Therefore, re-routing is not always feasible in this case as the driver may be required to unload goods that are not accessible. The second reason is that redesigning vehicle routes is known to be computationally intractable when exact methods, same as the methods used in this chapter, are used (Secomandi, 2001).

2.4 Problem description and mathematical models

The problem studied in this chapter is similar to the PRP, where the traffic speeds constitute the upper bounds on the maximum speed that can be attained on each leg of the route are not known with certainty at the time of route planning. Instead, they are described by a discrete and finite set of scenarios. The problem involves determining a set of vehicle routes and speeds on each leg of the routes, as in the PRP. However, upon realization of the random variables, the optimal speeds may no longer be feasible with respect to the traffic speeds, in which case delays may occur in arriving at customer locations. The objective is to minimize the expected cost of the total fuel consumption, emissions, drivers and delays. The problem is cast as a two-stage stochastic programming problem (Birge and Louveaux, 2011), for which we define two variants. The first variant corresponds to the OR-OS strategy, where the first-stage decisions involve determining the routes and the arc speeds. The second-stage decisions are the speed reductions inferred by the traffic speeds, which translate into possible delays. The second variant is

based on the OR-SS strategy, where the first stage decisions are restricted to planning a route for each vehicle, and the second stage variables correspond to computing the speeds on each route, separately for each scenario. In what follows, we describe mathematical models associated with the two problem variants just described.

2.4.1 Notations common to the two models

The stochastic models presented in this chapter are based on the deterministic mathematical model of the classical PRP proposed by Demir et al. (2012). The problem is defined on a complete graph $G = (N, A)$, where $N = \{0, 1, \ldots, n\}$ is the node set, and $A = \{(i, j); i, j \in N, i \neq j\}$ is the arc set, where each arc (i, j) has $|L_{ij}|$ segments of length d_{ijl}, $L_{ij} = \{1, \ldots, l, \ldots, |L_{ij}|\}$. Node 0 represents the depot at which m vehicles of capacity Q are based, and $N_0 = N \backslash \{0\}$ is the set of customers. Each customer $i \in N_0$ has a demand $q_i > 0$, which should start to be delivered within the time window $[a_i, b_i]$, and incurs a service time c_i. The decision variables are defined as follows: x_{ij} is a binary variable equal to 1 if and only if a vehicle visits customer j immediately after customer i. A vehicle travels at speed v_{ijl} on segment $l \in L_{ij}$ of arc (i, j), subject to a lower bound v^{lb} and an upper bound v^{ub} that correspond to legal limits on speed. Under uncertainty, a traffic speed u^s_{ijl} is defined on segment $l \in L_{ij}$ of arc (i, j) for each scenario $s \in S$ which is the maximum attainable speed on that arc. The continuous variable f_{ij} is the amount of commodity carried on the vehicle on arc $(i, j) \in A$. The continuous variable y_i is the time at which service begins at customer $i \in N_0$, and the continuous variable δ_i is the the cumulative time spent on the route whose last customer is i.

2.4.2 One-route, one-speed strategy

Under the OR-OS strategy, we assume that the decisions concerning routes and speeds are made simultaneously, and prior to the realization of the random variables. These two sets of decisions are made at the first stage and cannot subsequently be changed. However, if $v^*_{ijl} > u^s_{ijl}$, a vehicle will not be able to drive at speed v^*_{ijl} chosen on a specific segment l of arc (i, j) for a particular scenario $s \in S$. In this case, the recourse action is to reduce the speed by at least $v^*_{ijl} - u^s_{ijl}$ in order for the solution to be feasible. To this end, we introduce a speed reduction recourse variable SR^s_{ijl} to ensure the feasibility of solutions independently of random variables realizations. The scenarios also imply that some customers may not be served within their time window. Therefore, the time window requirements will be treated as soft constraints, in which case, the delay for the start of service at customer i will be represented by a recourse variable l^s_i in the second stage, and penalized by using a unit delay cost e.

It is important to observe that speed reduction recourse variables are needed in the computation of delays and routes durations, but are not explicitly penalized in the objective

function. They are only implicitly penalized because the vehicle will have to travel at a speed different from the optimal speed, which will increase the cost accordingly.

The speed chosen on an arc will affect the time y_i at which service starts at node i, and hence it will be different from one scenario to another. For this reason, this time will be represented by a scenario-dependent variable y_i^s. This variable captures the fact that a vehicle can be late to serve a customer in one scenario, but on time in a another one. The same reasoning applies to the variables representing the overall time spent on a route δ_i which will now be δ_i^s. We define M as a large enough constant. The mathematical model is described as follows:

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{l\in L_{ij}} w f_c \gamma \lambda \alpha_{ij} d_{ijl} x_{ij} + \sum_{(i,j)\in A} \sum_{l\in L_{ij}} f_c \gamma \lambda \alpha_{ij} f_{ij} d_{ijl} \tag{2.6}$$

$$+ \sum_{s\in S} p_s \Big(f_c k \Omega V \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl}/(v_{ijl} - SR_{ijl}^s) \tag{2.7}$$

$$+ f_c \beta \gamma \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl}(v_{ijl} - SR_{ijl}^s)^2 \tag{2.8}$$

$$+ \sum_{j\in N_0} f_d \delta_j^s \tag{2.9}$$

$$+ \sum_{j} el_j^s \Big) \tag{2.10}$$

subject to

$$\sum_{j \in N} x_{0j} = m \tag{2.11}$$

$$\sum_{j \in N} x_{ij} = 1 \qquad i \in N_0 \tag{2.12}$$

$$\sum_{i \in N} x_{ij} = 1 \qquad j \in N_0 \tag{2.13}$$

$$\sum_{j \in N} f_{ji} - \sum_{j \in N} f_{ij} = q_i \qquad i \in N_0 \tag{2.14}$$

$$q_j x_{ij} \leq f_{ij} \leq (Q - q_i) x_{ij} \qquad (i,j) \in A \tag{2.15}$$

$$y_i^s - y_j^s + c_i + \sum_{l \in L_{ij}} d_{ijl}/(v_{ijl} - SR_{ijl}^s) \leq M(1 - x_{ij}) \qquad i \in N, j \in N_0, i \neq j, s \in S \tag{2.16}$$

$$y_j^s - c_j - \delta_j^s + \sum_{l \in L_{ij}} d_{j0l}/(v_{j0l} - SR_{j0l}^s) M(1 - x_{j0}) \qquad j \in N_0, s \in S \tag{2.17}$$

$$a_j \leq y_j^s \leq b_j + l_j^s \qquad j \in N_0, s \in S \tag{2.18}$$

$$v_{ijl} - SR_{ijl}^s \leq u_{ijl}^s \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.19}$$

$$v_{ijl} - SR_{ijl}^s \geq v^{lb} x_{ij} \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.20}$$

$$f_{ij} \geq 0 \qquad (i,j) \in A \tag{2.21}$$

$$y_i^s \geq 0 \qquad i \in N_0, s \in S \tag{2.22}$$

$$SR_{ij}^s \geq 0 \qquad (i,j) \in A, s \in S \tag{2.23}$$

$$v^{lb} x_{ij} \leq v_{ijl} \leq v^{ub} x_{ij} \qquad (i,j) \in A, l \in L_{ij} \tag{2.24}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A. \tag{2.25}$$

The first term of the objective function computes the cost of the solution that can be attributed to vehicle weight and load. Similarly, the terms (2.7)–(2.10) represent the expected cost as a function of speeds, delays at customers, and driving time. Constraints (2.11) ensure that all vehicles leave the depot. Constraints (2.12)–(2.13) state that each customer must be visited exactly once. Constraints (2.14) and (2.15) define the flow on each arc. Constraints (2.16)–(2.17) compute the arrival time at each customer $j \in N_0$ in each scenario $s \in S$, (2.18) calculate the delay at customers, (2.19) ensure that the speed used in a scenario after any reduction is lower than the traffic speed in that scenario. Constraints (2.20) ensure that the speeds chosen are at least v^l if the arc is in the route of a vehicle.

Due to the terms (2.7), (2.8), (2.16) and (2.17), the model described above is non-linear. One way to linearize linear-fractional models is to use Charnes-Cooper transformation (Charnes and Cooper, 1962). However, due to the fact that v_{ijl} and SR_{ijl}^s variables are

continuous, this decomposition method cannot be applied here. In the following, we present two different ways of linearizing these terms.

2.4.2.1 Discrete speeds and discrete recourse

In order to linearize the non-linear terms, we adopt the same strategy as in Bektaş and Laporte (2011), which consists of discretizing the speed variables. We will also discretize the recourse variables. The speeds are represented by a set $R = \{1, \ldots, r, \ldots\}$ of discrete levels. A binary variable z_{ijl}^r is equal to 1 if and only if the r^{th} speed level v^r is chosen for the vehicle traveling on segment $l \in L_{ij}$ of arc (i, j). We note that $v^1 = v^{lb}$ and $v^{|R|} = v^{ub}$. The same reasoning applies to the discretized recourse. We define a set $|T|$ of recourse levels $T = \{1, \ldots, t, \ldots\}$, and a binary variable α_{ijl}^{ts} is equal to 1 if and only if the level t of speed reduction level ψ^t is chosen for the vehicle traveling on segment $l \in L_{ij}$ of arc (i, j) in scenario $s \in S$, with $\psi^1 = 0$ and $\psi^{|T|} = v^{ub} - v^{lb}$. We also introduce M as a large constant. The discretized mathematical model is described as follows.

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{l\in L_{ij}} w f_c \gamma \lambda \alpha_{ij} d_{ijl} x_{ij} + \sum_{(i,j)\in A} \sum_{l\in L_{ij}} f_c \gamma \lambda \alpha_{ij} d_{ijl} f_{ij} \tag{2.26}$$

$$+ \sum_{s\in S} p_s \Big(f_c k \Omega V \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl} \Big(\sum_{r\in R} \sum_{t\in T} z_{ijl}^r \alpha_{ijl}^{ts} / (v^r - \psi^t) \Big) \tag{2.27}$$

$$+ f_c \beta \gamma \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl} \sum_{r\in R} \sum_{t\in T} z_{ij}^r \alpha_{ijl}^{ts} \big(v^r - \psi^t \big)^2 \tag{2.28}$$

$$+ \sum_{j\in N_0} f_d \delta_j^s \tag{2.29}$$

$$+ \sum_{j} e l_j^s \Big) \tag{2.30}$$

subject to

(2.11)–(2.15), (2.18), (2.21), (2.22)

$$y_i^s - y_j^s + c_i + \sum_{l \in L_{ij}} d_{ijl} \sum_{r \in R} \sum_{t \in T} z_{ijl}^r \alpha_{ijl}^{ts} / (v^r - \psi^t) \leq M(1 - x_{ij}) \qquad i \in N, \; j \in N_0, i \neq j, s \in S \tag{2.31}$$

$$y_j^s - c_j - \delta_j^s + \sum_{l \in L_{ij}} d_{j0l} \sum_{r \in R} \sum_{t \in T} z_{j0l}^r \alpha_{j0l}^{ts} / (v^r - \psi^t) \leq M(1 - x_{j0}) \qquad j \in N_0, s \in S \tag{2.32}$$

$$\alpha_{ijl}^{1s} \geq (u_{ijl}^s - \sum_{r \in R} z_{ijl}^r v^r)/M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.33}$$

$$\alpha_{ijl}^{1s} \leq (u_{ijl}^s - \sum_{r \in R} z_{ijl}^r v^r)/M + 1 \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.34}$$

$$\sum_{r \in R} \sum_{t \in T} z_{ijl}^r \alpha_{ijl}^{ts} (v^r - \psi^t) \leq u_{ijl}^s \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.35}$$

$$\sum_{r \in R} \sum_{t \in T} z_{ijl}^r \alpha_{ijl}^{ts} (v^r - \psi^t) \geq v^1 x_{ij} \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.36}$$

$$\sum_{r \in R} z_{ijl}^r = x_{ij} \qquad (i,j) \in A, l \in L_{ij} \tag{2.37}$$

$$\sum_{t \in T} \alpha_{ijl}^{ts} = x_{ij} \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.38}$$

$$\alpha_{ijl}^{ts}, z_{ijl}^r, x_{ij} \in \{0,1\} \qquad (i,j) \in A, l \in L_{ij}, \; s \in S, r \in R, t \in T. \tag{2.39}$$

Constraints (2.31)–(2.32) compute the arrival time at each customer $j \in N_0$ in each scenario $s \in S$. Constraints (2.33)–(2.34) ensure that the value of the speed reduction is 0 if the speed chosen is lower than the traffic speed in scenario s. Constraints (2.35)–(2.36) play the same role as (2.19)–(2.20). Finally, constraints (2.38) ensure that only one recourse is chosen if arc (i,j) is selected.

This model is still non-linear due to the presence of products of two binary variables in constraints (2.35) and (2.36). To linearize these constraints, we introduce a new binary variable τ_{ijl}^{rts} equal to 1 if and only if a vehicle travels on segment l of arc (i,j) at speed

v^r and applies a speed recourse ψ^t in scenario $s \in S$. The variables τ_{ijl}^{rts} will replace the products $z_{ijl}^r \alpha_{ijl}^{ts}$ through the use of the following constraints:

$$\tau_{ijl}^{rts} \leq z_{ijl}^r \qquad (i,j) \in A, l \in L_{ij}, r \in R, s \in S \tag{2.40}$$

$$\tau_{ijl}^{rts} \leq \alpha_{ijl}^{ts} \qquad (i,j) \in A, l \in L_{ij}, t \in T, s \in S \tag{2.41}$$

$$\tau_{ijl}^{rts} \geq z_{ijl}^r + \alpha_{ijl}^{ts} - 1 \qquad (i,j) \in A, l \in L_{ij}, r \in R, t \in T, s \in S \tag{2.42}$$

$$\tau_{ijl}^{rts} \in \{0,1\} \qquad (i,j) \in A, l \in L_{ij}, s \in S, r \in R, t \in T. \tag{2.43}$$

2.4.2.2 Discrete speeds and continuous restricted recourse

In the previous model, we used discretization to linearize the mathematical program. However, due to the uncertainty about the traffic speeds, using discrete speed reductions may result in poor quality recourse actions. We therefore introduce an alternative modeling scheme based on the use of continuous recourse variables, using same notation as in Section 2.4.2.1. However, here we make an additional assumption that restricts the value of the recourse action. Recall that SR_{ijl}^s is the speed reduction needed on segment l of arc $(i,j) \in A$ to travel at a speed not exceeding u_{ijl}^s in scenario $s \in S$. We now assume that if a speed recourse is needed, then SR_{ijl}^s, will be exactly equal to $\sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s$. This assumption is needed to linearize the model, and to have continuous recourse variables. We introduce a binary activation variable a_{ijl}^s equal to 1 if and only if a speed reduction is needed on segment l of arc $(i,j) \in A$ in scenario $s \in S$. The mathematical model is described as follows:

$$\text{Minimize} \sum_{(i,j)\in A} \sum_{l \in L_{ij}} w f_c \gamma \lambda \alpha_{ij} d_{ijl} x_{ij} + \sum_{(i,j)\in A} \sum_{l \in L_{ij}} f_c \gamma \lambda \alpha_{ij} d_{ijl} f_{ij} \tag{2.44}$$

$$+ \sum_{s \in S} p_s \Big(f_c k \Omega V \lambda \sum_{(i,j)\in A} \sum_{l \in L_{ij}} d_{ijl} \big(\sum_{r \in R} z_{ijl}^r / (v^r - SR_{ijl}^s) \big) \tag{2.45}$$

$$+ f_c \beta \gamma \lambda \sum_{(i,j)\in A} \sum_{l \in L_{ij}} d_{ijl} \sum_{r \in R} z_{ijl}^r \big(v^r - SR_{ijl}^s \big)^2 \tag{2.46}$$

$$+ \sum_{j \in N_0} f_d \delta_j^s \tag{2.47}$$

$$+ \sum_j e l_j^s \Big) \tag{2.48}$$

subject to

(2.11)–(2.15) , (2.18), (2.21)–(2.23)

$$y_i^s - y_j^s + c_i + \sum_{l \in L_{ij}} \sum_{r \in R} d_{ijl} z_{ijl}^r / (v^r - SR_{ijl}^s) \leq M(1 - x_{ij}) \qquad i \in N, j \in N_0, i \neq j, s \in S \tag{2.49}$$

$$y_j^s - c_j - \delta_j^s + \sum_{l \in L_{ij}} \sum_{r \in R} d_{j0l} z_{j0l}^r / (v^r - SR_{ijl}^s) \leq M(1 - x_{j0}) \qquad j \in N_0, s \in S \tag{2.50}$$

$$SR_{ijl}^s \geq \sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.51}$$

$$SR_{ijl}^s \leq \sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s + (1 - a_{ijl}^s)M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.52}$$

$$SR_{ijl}^s \leq a_{ijl}^s M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.53}$$

$$a_{ijl}^s \geq (\sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s)/M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.54}$$

$$a_{ijl}^s \leq (\sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s)/M + 1 \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.55}$$

$$a_{ijl}^s, z_{ijl}^r, x_{ijl} \in \{0,1\} \qquad (i,j) \in A, r \in R \tag{2.56}$$

$$\sum_{r \in R} z_{ij}^r = x_{ij} \qquad (i,j) \in A. \tag{2.57}$$

In the objective function, the term (2.44) computes the cost of the solution associated to the vehicle weight and load. The terms (2.45)–(2.48) compute the expected cost as a function of speeds, delays at customers and drivers' wages. Constraints (2.49)–(2.50) define the arrival time at each customer $j \in N_0$ in each scenario $s \in S$. Constraints (2.51)–(2.55) force SR_{ijl}^s to be equal to $\sum_{r \in R} z_{ijl}^r v^r - u_{ijl}^s$ if and only if there is a speed reduction.

Due to the presence of the terms (2.45) and (2.46) in the objective function, the model is non-linear. In order to linearize these terms, we introduce two sets of new variables SF_{ijl}^s and SS_{ijl}^s. These new variables are such that $SF_{ijl}^s = 1/u_{ijl}^s - \sum_{r \in R} z_{ijl}^r / v^r$ if and only if there is a speed reduction. Using the same reasoning, $SS_{ijl}^s = \sum_{r \in R} z_{ijl}^r (v^r)^2 - (u_{ijl}^s)^2$.

Components (2.45) and (2.46) then become

$$f_c k \Omega V \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl} \Big(\sum_{r\in R} (z^r_{ijl}/v^r) + SF^s_{ijl} \Big) \tag{2.58}$$

$$f_c \beta \gamma \lambda \sum_{(i,j)\in A} \sum_{l\in L_{ij}} d_{ijl} \Big(\sum_{r\in R} z^r_{ijl} (v^r)^2 - SS^s_{ijl} \Big). \tag{2.59}$$

We also replace constraints (2.49)–(2.53) with constraints (2.60)–(2.69):

$$y^s_i - y^s_j + c_i + \sum_{l\in L_{ij}} \sum_{r\in R} d_{ijl} \Big(\sum_{r\in R} (z^r_{ijl}/v^r) + SF^s_{ijl} \Big) \leq M(1 - x_{ij}) \qquad i \in N, j \in N_0, i \neq j, \; s \in S \tag{2.60}$$

$$y^s_j - c_j - \delta^s_j + \sum_{l\in L_{ij}} \sum_{r\in R} d_{j0l} \Big(\sum_{r\in R} (z^r_{j0l}/v^r) + SF^s_{j0l} \Big) \leq M(1 - x_{j0}) \qquad j \in N_0, s \in S \tag{2.61}$$

$$SF^s_{ijl} \geq 1/u^s_{ijl} - \sum_{r\in R} z^r_{ijl}/v^r - (1 - x_{ij})M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.62}$$

$$SF^s_{ijl} \leq 1/u^s_{ijl} - \sum_{r\in R} z^r_{ijl}/v^r + (2 - a^s_{ijl} - x_{ij})M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.63}$$

$$SF^s_{ijl} \leq a^s_{ijl} M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.64}$$

$$SS^s_{ijl} \geq \sum_{r\in R} z^r_{ijl} (v^r)^2 - (u^s_{ijl})^2 - (1 - x_{ij})M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.65}$$

$$SS^s_{ijl} \leq \sum_{r\in R} z^r_{ijl} (v^r)^2 - (u^s_{ijl})^2 + (2 - a^s_{ijl} - x_{ij})M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.66}$$

$$SS^s_{ijl} \leq a^s_{ijl} M \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.67}$$

$$SF^s_{ijl} \geq 0 \qquad (i,j) \in A, l \in L_{ij}, s \in S \tag{2.68}$$

$$SS^s_{ijl} \geq 0 \qquad (i,j) \in A, l \in L_{ij}, s \in S. \tag{2.69}$$

We now present a proposition that establishes the equivalence between the non-linear and the linearized formulations of the discrete speed and continuous recourse. The example presented in Section 2.3.2 was obtained by solving this mathematical model with the input parameters of Tables 2.2 and 2.3.

Proposition 1. *Let v^*_{NLP} be the optimal value and (x^*, z^*, k^*, SR^*) be an optimal solution of the non-linear formulation. Let v^*_{LP} be the optimal value and $(x^*, z^*, k^*, SS^*, SF^*)$ be an optimal solution of the linearized formulation. If (2.45)–(2.46) and (2.49)–(2.53) are replaced by (2.58)–(2.59) and (2.60)–(2.69) respectively, then $v^*_{NLP} = v^*_{LP}$ and the optimal solutions coincide with respect to (x^*, z^*, k^*).*

Proof. To prove the validity of Proposition 1, we compute the value of the two original terms in the objective functions, and the terms that replace them in the linearized version. The value of these terms needs to be the same. For a given arc $(i, j) \in A$, there are four cases:

1. There is a no speed violation on (i, j):
   (a) non-linear model: Due to (2.54)–(2.55) $a^s_{ijl} = 0$, hence, due to (2.53), $SR^s_{ijl} = 0$ for all $s \in S$. Therefore the values of (2.45)–(2.46) are as follows :
      - $d_{ijl} \sum_{r\in R} z^r_{ijl}/(v^r - SR^s_{ijl}) = d_{ijl} \sum_{r\in R} z^r_{ijl}/v^r = d_{ijl}/v^r$ if $z^r_{ijl} = 1$, and 0 otherwise,
      - $d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r - SR^s_{ijl})^2 = d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r)^2 = d_{ijl}(v^r)^2$ if $z^r_{ijl} = 1$, and 0 otherwise.

   (b) linearized model: Due to (2.54)–(2.55) $a^s_{ijl} = 0$, hence, due to (2.64), $SF^s_{ijl} = 0$ and due to (2.67) $SS^s_{ijl} = 0$ for all $s \in S$. Therefore the value of (2.58)–(2.59) are as follows:
      - $d_{ijl} \sum_{r\in R}(z^r_{ijl}/v^r) + SF^s_{ijl} = d_{ijl} \sum_{r\in R} z^r_{ijl}/v^r = d_{ijl}/v^r$ if $z^r_{ijl} = 1$, and 0 otherwise,
      - $d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r)^2 - SS^s_{ijl} = d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r)^2 = d_{ijl}(v^r)^2$ if $z^r_{ijl} = 1$, and 0 otherwise.

2. There is a speed violation on (i, j) (i.e. $\sum_{r\in R} z^r_{ijl} > u^s_{ijl}$) :
   (a) non-linear model: Due to (2.54)–(2.55) $a^s_{ijl} = 1$, hence, due to (2.51)–(2.52), $SR^s_{ijl} = \sum_{r\in R} z^r_{ijl} v^r - u^s_{ijl}$ for all $s \in S$. Therefore:
      - $d_{ijl} \sum_{r\in R} z^r_{ijl}/(v^r - SR^s_{ijl}) = d_{ijl} \sum_{r\in R} z^r_{ijl}/(v^r - \sum_{r\in R} z^r_{ijl} v^r + u^s_{ijl}) = d_{ijl}/u^s_{ijl}$ if $z^r_{ijl} = 1$, and 0 otherwise,
      - $d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r - SR^s_{ijl})^2 = d_{ijl} \sum_{r\in R} z^r_{ijl}(v^r - \sum_{r\in R} z^r_{ijl} v^r + u^s_{ijl})^2 = d_{ijl}(u^s_{ijl})^2$ if $z^r_{ijl} = 1$, and 0 otherwise.

   (b) linearized model: Due to (2.54)–(2.55) $a^s_{ijl} = 1$, hence, due to (2.62)–(2.63), $SF^s_{ijl} = 1/u^s_{ijl} - \sum_{r\in R} z^r_{ijl}/v^r$ and due to (2.65)–(2.66) $SS^s_{ijl} = \sum_{r\in R} z^r_{ijl}(v^r)^2 - (u^s_{ijl})^2$ for all $s \in S$. Therefore:
      - $d_{ijl} \sum_{r\in R}(z^r_{ijl}/v^r) + SF^s_{ijl} = d_{ijl}(\sum_{r\in R} z^r_{ijl}/v^r + 1/u^s_{ijl} - \sum_{r\in R} z^r_{ijl}/v^r) = d_{ijl}/u^s_{ijl}$ if $z^r_{ijl} = 1$, and 0 otherwise,

- $d_{ijl}\sum_{r\in R} z^r_{ijl}(v^r)^2 - SS^s_{ijl} = d_{ijl}(\sum_{r\in R} z^r_{ijl}(v^r)^2 - \sum_{r\in R} z^r_{ijl}(v^r)^2 + (u^s_{ijl})^2) = d_{ijl}(u^s_{ijl})^2$ if $z^r_{ijl} = 1$, and 0 otherwise.

We have shown that in all possible cases, the linearized and non-linear models are equivalent. Therefore, $v^*_{NLP} = v^*_{LP}$ and the optimal solutions of the two models coincide with respect to (x^*, z^*, k^*). □

2.4.3 One-route, several-speeds strategy

As explained in Section 2.3.3, the OR-SS strategy assumes that the actual maximum speeds are revealed prior to executing the routes, but after the routing decisions have been made. Here the decisions involve determining a set of routes that will remain intact regardless of the scenario, but will include a set of optimal speeds for each scenario. This problem is now modeled as a two-stage stochastic program in which the first-stage variables correspond to the routes. The variables z^r_{ijl} that were used to choose a speed level v^r for the segment l of the arc (i,j) in Section 2.4.2.1 will now be second-stage decision variables and redefined separately for each scenario as z^{rs}_{ijl} to denote the speed level v^r. The recourse variables are potential delays experienced when serving customers. The objective is to minimize the expected cost by taking into account the scenario probabilities. The second-stage problem is a speed optimization problem and is independently studied in Chapter 4. The two-stage stochastic program is defined as follows, where $\mathbf{E}_s W(x,s)$ represents the expected cost of the objective function of second-stage problem $W(x,s)$:

$$\text{Minimize} \quad \sum_{(i,j)\in A}\sum_{l\in L_{ij}} w f_c \gamma \lambda \alpha_{ij} d_{ijl} x_{ij} \tag{2.70}$$

$$+ \sum_{(i,j)\in A}\sum_{l\in L_{ij}} f_c \gamma \lambda \alpha_{ij} d_{ijl} f_{ij} \tag{2.71}$$

$$+ \sum_{s\in S} p_s \Big(\sum_{(i,j)\in A}\sum_{l\in L_{ij}} f_c k \Omega V \lambda d_{ijl} \sum_{r\in R} z^{rs}_{ijl}/v^r \tag{2.72}$$

$$+ \sum_{(i,j)\in A}\sum_{l\in L_{ij}} f_c \beta \gamma \lambda d_{ijl} \sum_{r\in R} z^{rs}_{ijl}(v^r)^2 \Big) \tag{2.73}$$

$$+ \mathbf{E}_s W(x,s) \tag{2.74}$$

subject to

$$\sum_{j\in N} x_{0j} = m \tag{2.75}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad i \in N_0 \tag{2.76}$$

$$\sum_{j\in N} x_{ij} = 1 \qquad j \in N_0 \tag{2.77}$$

$$\sum_{j\in N} f_{ij} - \sum_{j\in N} f_{ji} = q_i \qquad i \in N_0 \tag{2.78}$$

$$q_j x_{ij} \leq f_{ij} \leq (Q - q_i)x_{ij} \qquad (i,j) \in A \tag{2.79}$$

$$\sum_{r\in R} z_{ijl}^{rs} = x_{ij} \qquad (i,j) \in A, l \in L_{ij}, \quad s \in S \tag{2.80}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{2.81}$$

$$f_{ij} \geq 0 \qquad (i,j) \in A, \tag{2.82}$$

where the second stage problem is defined as

$$W(x,s) = \text{minimize}\{\sum_{j\in N_0} f_d \delta_j^s + \sum_{j\in N} el_j^s\}, \tag{2.83}$$

subject to

$$y_i^s - y_j^s + c_i + \sum_{r\in R}\sum_{l\in L_{ij}} d_{ijl} z_{ijl}^{rs}/v^r \leq M(1 - x_{ij}) \qquad i \in N, j \in N_0, i \neq j, \quad s \in S \tag{2.84}$$

$$y_j^s - c_j - \delta_j^s + \sum_{l\in L_{ij}}\sum_{r\in R} d_{j0l} z_{j0l}^{rs}/v^r \leq M(1 - x_{j0}) \qquad j \in N_0, \quad s \in S \tag{2.85}$$

$$a_j \leq y_j^s \leq b_j + l_j^s \qquad i \in N, j \in N_0, i \neq j, s \in S \tag{2.86}$$

$$y_i^s \geq 0 \qquad i \in N_0, s \in S \tag{2.87}$$

$$z_{ijl}^{rs} \in \{0,1\} \qquad (i,j) \in A, l \in L_{ij}, \quad r \in R \quad s \in S. \tag{2.88}$$

The value of optimal solution yielded by the OR-SS strategy is at least as good as the optimal value of the OR-OS strategy. The relative difference between these values can be interpreted as the value of the additional information provided by the OR-SS strategy. Indeed, the gain in cost is due to the knowledge of traffic speeds before the start of the routing.

2.4.4 Discussion about the values of constants M

Big-M constraints are particularly useful in mathematical programming to represent conditional constraints. While the theoretical value of the constants M is large enough, a value has to be chosen to do the experiments. To avoid longer solution times, these values have to be chosen as small as possible. In this section we give typical values that can be used for these constants in the previous models.

In constraints (2.31), (2.32), (2.49), (2.50), (2.60), (2.61), (2.84), (2.85) the M constant is used in a time-window constraints, therefore the sum of the $n+m$ longest leg durations plus the sum of the service-times can be used as an upper bound. In constraints (2.33), (2.34), (2.52), (2.53), (2.54) the largest value of the u^s_{ijl} can be used. In constraints (2.62)–(2.64) we can use the largest value of $(z^{|R|}_{ijl})^2$ and (2.65)–(2.67) we can use the largest of the values $1/(z^{|1|}_{ijl})$.

2.5 Computational experiments

The purpose of the computational experimentation carried out in this section is fourfold. First, we aim to compare the two linear models described in Section 2.4.2, which are the discrete speed and discrete recourse model, and the discrete speed and continuous recourse model. This comparison will show which of the linearization techniques is more suitable to solve the problem. Second, once the best linear stochastic model is identified, it will be compared with a deterministic model. This model considers only one scenario, where on each arc the traffic speed is the average of traffic speeds of each scenario. This comparison will highlight the advantage of using a stochastic strategy rather than a simple deterministic strategy. The relative difference between the two solutions values can be interpreted as the value of a stochastic solution (Birge, 1982; Birge and Louveaux, 2011). Third, in order to assess the benefit of having perfect knowledge of the traffic speeds, we will compare the OR-OS and OR-SS strategies. The difference between the solution values yielded by these two strategies can be interpreted as the value of complete information about congestion and random events. Finally, we quantify the benefits of optimizing speeds as compared with an approach where the vehicles simply travel at the fixed traffic speed.

We have conducted experiments on the 20 benchmark instances of PRP-lib with 10 customers. All tests were performed on a computer equipped with an Intel Core i7-3770 processor, 3.4 GHz and a RAM of 8GB. The mathematical models were solved by CPLEX 12.6.0.1 using default options. A maximum CPU time of three hours was allowed for the solution of any instance. The typical values of the parameters used in fuel emission model as described in Bektaş and Laporte (2011) are shown in Table 2.4,

with exception of the driver wage for which we use £10 per hour as per the 2017 rate reported in (National Careers Service, 2017).

Notation	Description	Typical values
w	Curb-weight (kg)	6350
ε	Fuel-to-air mass ratio	1
k	Engine friction factor (kJ/rev/L)	0.2
Ω	Engine speed (rev/s)	33
V	Engine displacement (L)	5
g	Gravitational constant (m/s^2)	9.81
C_d	Coefficient of aerodynamic drag	0.7
ρ	Air density (kg/m^3)	1.2041
Γ	Frontal surface area (m^2)	3.912
C_r	Coefficient of rolling resistance	0.01
η_{tf}	Vehicle drive train efficiency	0.4
η	Efficiency parameter for diesel engines	0.9
f_c	Fuel and CO_2 emissions cost (£/L)	1.4
f_d	Driver wage (£/s)	0.0028
e	Unit delay cost (£/s)	0.0028
κ	Heating value of a typical diesel fuel (kJ/g)	44
Ψ	Conversion factor (g/s to L/s)	737
θ	Road angle	0
v^{lb}	Speed lower bound (km/h)	20
v^{ub}	Speed upper bound (km/h)	96

Table 2.4: Values of the parameters used in the emission model

2.5.1 Probability distribution and scenario generation

As mentioned in Section 2.3.3, the traffic speeds are the random variables of the problem. To generate traffic speeds, we need to generate different traffic conditions on the network, which can be achieved through the use of distributions to generate vehicle speeds. Hofleitner et al. (2012) and Rakha et al. (2006) suggest that compared with other distributions, the log-normal distribution provides a good fit for vehicle speeds. This is therefore the distribution we have used to model traffic speeds on arcs.

In the computational experiments carried out, we considered three scenarios S_1, S_2 and S_3. Each scenario was defined using a log-normal distribution with parameters μ and σ from which we drew traffic speeds for each arc of the graph. This distribution describes the general characteristics of the speeds for that scenario. For example, if μ is low, the traffic is low in general, although some arcs may have higher traffic speeds due to the long tail of the distribution, particularly if σ is large. The mean and variance of a sample generated using a log-normal distribution is $\hat{\mu} = e^{\mu+\sigma^2/2}$ and its variance is $\hat{\sigma}^2 = (e^{\sigma^2}-1)e^{2\mu+\sigma^2}$. In the remainder of the paper, we will refer to the log-normal distribution

by its mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ defined above. In the first part of the computational experiments, we consider only one segment per arc (i.e. $L_{ij} = \{1\}, \forall (i,j) \in A$).

To generate realistic scenarios, we used the *Mobile Century Data* (Herrera et al., 2010) that provides vehicles speed collected using phones GPSs during different times. The first scenario S_1 is extracted from the data where the average speed is relatively low but has a large variance. It represents a relative congestion on some arcs while others have larger traffic speeds. The scenario S_1 is generated using a mean $\hat{\mu} = 40$ km/h and a variance $\hat{\sigma}^2 = 64$. The second scenario S_2 is extracted from data where the average speed is high but values over the arcs have high fluctuation. It is generated with $\hat{\mu} = 70$ and $\hat{\sigma}^2 = 144$. The last scenario is generated using $\hat{\mu} = 55$ and $\hat{\sigma}^2 = 100$ The traffic speed generated u_{ij}^s for all arcs $(i,j) \in A$ and all scenarios $s \in S$ must lie within the interval $20 \leq u_{ij}^s \leq 96$ required by law for delivery vehicles. Below is an example of a complete graph defined with a single depot and three customers, where on each arc the traffic speed are drawn from scenario S_2:

$$\begin{bmatrix} 0 & 87 & 20 & 45 \\ 58 & 0 & 26 & 51 \\ 29 & 59 & 0 & 29 \\ 58 & 24 & 61 & 0 \end{bmatrix} .$$

For all comparisons provided in this section, every instance was solved on three different, and randomly generated realizations of S_1, S_2 and S_3, the results displayed on the tables represents the average of all these tests. This methodology allow the results to be meaningful and do not rely on a particular realization of the scenarios. It also enables the assessment of the robustness of the solution algorithms with respect to changes in traffic conditions.

2.5.2 Comparative results for the discretized and continuous recourse strategies

We now present the results of experiments in which each of the 20 instances was solved once by the discretized recourse model, and once by the continuous recourse model, both described in Section 2.4.2. The results are presented in Table 2.5 which displays the optimal values for the instances (in £) and solution times in seconds to optimality. The gap (%) is the percentage of the continuous recourse optimal solution value saved with respect to the optimal solution value of the discrete recourse model. We can see from Table 2.5 that the fixed continuous recourse model outperforms the discretized recourse model by an average of 4.95% in terms of optimal solution value. This table also shows that the continuous recourse model is more than twice faster on average than the discretized recourse model. On instance UK10_15, the discretized recourse model

could not yield a proven optimal solution within the time limit of three hours; in this case, we report the best known solution value. Given this result, OR-OS, will henceforth refer to the discretized speed and continuous recourse model.

Instance	Discrete recourse		Continuous recourse		
	Optimal value	Seconds	Optimal value	Seconds	Gap (%)
UK10_01	261.6	4799.0	251.3	1119.7	4.1
UK10_02	327.6	2505.3	308.4	1844.7	6.2
UK10_03	295.6	2682.0	282.3	818.7	4.7
UK10_04	284.4	1736.3	270.3	711.0	5.2
UK10_05	240.4	3061.0	232.1	719.0	3.6
UK10_06	343.4	4352.0	328.9	2092.3	4.4
UK10_07	288.5	2250.3	277.7	1004.7	3.9
UK10_08	330.0	1607.5	313.2	487.0	5.4
UK10_09	259.6	763.7	246.4	213.3	5.4
UK10_10	293.1	2319.7	278.0	587.3	5.4
UK10_11	439.0	1206.3	409.8	1083.3	7.1
UK10_12	260.2	849.7	248.4	829.3	4.8
UK10_13	289.5	1408.3	279.7	1431.7	3.5
UK10_14	282.3	5305.7	267.0	2963.0	5.7
UK10_15	210.7	10800.0	199.5	3381.0	5.6
UK10_16	252.0	4616.0	243.8	1447.0	3.4
UK10_17	255.6	1235.3	246.5	347.7	3.7
UK10_18	242.5	3574.0	225.8	1062.0	7.4
UK10_19	266.3	2974.3	251.6	573.3	5.8
UK10_20	244.1	4259.0	235.8	1373.0	3.6
Average	283.32	3115.28	269.82	1204.45	4.95

Table 2.5: Comparison between the solutions of the discretized and continuous recourse models

2.5.3 Comparative results for the OR-OS strategy with the deterministic model and OR-SS strategy

Here, we present the results of experiments in which each of the 20 instances is solved by the deterministic and the OR-SS approaches, both described in Section 2.4.2 and 2.4.3. The relative difference in cost between the OR-OS and deterministic approach is the *Value of the Stochastic Solution (VSS)* (Birge, 1982). It represents the percentage cost saved when solving a stochastic model rather than a deterministic one. The VSS is computed as follows:

$$VSS = \frac{\text{Deterministic solution cost} - \text{Stochastic solution cost}}{\text{Stochastic solution cost}} \times 100.$$

In Table 2.6, we show the optimal solutions values (in £), the solution times in seconds, and the VSS. We see that the stochastic solutions are on average better than the deterministic ones by 7.48%. The largest saving 17.7%, obtained by the OR-OS strategy is achieved on instance UK10_15. The average solution time to optimality of the deterministic model is 47.03 seconds, while the same statistic for the OR-OS model is 1204.45 seconds.

Instance	Optimal value	Seconds	VSS (%)
UK10_01	265.4	33.0	5.6
UK10_02	332.6	37.7	7.8
UK10_03	297.3	64.3	5.3
UK10_04	288.7	43.3	6.8
UK10_05	241.5	52.5	4.1
UK10_06	352.3	78.7	7.1
UK10_07	295.5	35.3	6.4
UK10_08	332.4	16.0	6.1
UK10_09	256.5	10.7	4.1
UK10_10	294.0	18.0	5.8
UK10_11	444.6	41.7	8.5
UK10_12	263.5	45.3	6.1
UK10_13	290.6	71.3	3.9
UK10_14	282.1	105.3	5.7
UK10_15	234.9	58.7	17.7
UK10_16	254.9	49.0	4.5
UK10_17	281.4	13.7	14.2
UK10_18	259.3	53.7	14.8
UK10_19	277.7	30.0	10.4
UK10_20	246.7	82.3	4.6
Average	289.59	47.03	7.48

Table 2.6: Performance of deterministic approach compared with OR-OS

In Table 2.7 we see that The OR-SS strategy reduces the total cost obtained by the OR-OS on average by 1.27%. This improvement is the saving achieved when perfect information about the traffic speeds is available, but this information is in practice very difficult to acquire. However, the results of these tests show that the use OR-OS strategy can provide solutions that are on average less than two percent more expensive than those based on perfect information. Comparing the times required to solve the instances to optimality, we see that OR-SS strategy requires 191.05 seconds on average, while the OR-OS strategy is slower, with an average solution time of 1204.45 seconds. This difference is justified by the fact that no speed recourse decisions are needed under the OR-SS strategy, and hence fewer solutions need to be checked to prove the optimality of the current solution.

Instance	Optimal value	Seconds	Gap (%)
UK10_01	246.3	154.3	−2.0
UK10_02	303.5	138.0	−1.6
UK10_03	278.8	158.3	−1.3
UK10_04	267.4	132.3	−1.1
UK10_05	230.4	187.0	−0.7
UK10_06	325.2	358.7	−1.1
UK10_07	274.5	144.3	−1.2
UK10_08	309.5	57.0	−1.2
UK10_09	243.3	30.0	−1.3
UK10_10	273.5	84.7	−1.6
UK10_11	406.3	218.0	−0.9
UK10_12	245.9	185.3	−1.0
UK10_13	275.0	333.0	−1.7
UK10_14	263.3	427.7	−1.4
UK10_15	196.0	498.0	−1.8
UK10_16	242.0	113.3	−0.7
UK10_17	244.1	58.0	−1.0
UK10_18	223.1	143.7	−1.2
UK10_19	248.4	101.0	−1.3
UK10_20	232.5	298.3	−1.4
Average	266.45	191.05	−1.27

Table 2.7: Performance of OR-SS approach compared with OR-OS

2.5.4 Problem size and difficulty induced by the number of scenarios

In this section we look at the number of variables, number of constraints and solution times evolution with the increase in the number of scenarios in the OR-OS strategy. To asses the impact of the number of scenarios, we solve Instance UK10_01 several times with a number of scenarios varying from one to eight shown in Table 2.8. This instance has been chosen as it has a solution time close to the average solution time of 10-node instances described in Table 2.5. In addition to showing the number of variables, constrains and solution time (seconds), we report in Table 2.9 the change in these values of each additional scenario considered. We see from Table 2.9 that numbers of scenarios has a significant impact on the size and solution time of the problem. We can see that considering one additional scenario introduces 246 variables and 1051 constraints on average, and increase the computational time of 1243 seconds. While the number of additional variables and constraints increase steadily, it is not the case of the solution time. The latter depends on the difficulty of solving an instance, which in turn depend on the size of the instance and the values of traffic speeds generated by the scenarios. This means that two instances with the same number of scenarios have the same number of variables and constrains, but differ in the solution time needed.

Scenario	$\hat{\mu}$	$\hat{\sigma}^2$
S_1	40	64
S_2	70	144
S_3	55	100
S_4	35	49
S_5	80	121
S_6	60	100
S_7	70	144
S_8	29	64

Table 2.8: Eight scenarios considered

Scenarios	Variables	Change	Constraints	Change	Time (s)	Change
1	1349	-	1490	-	126	-
2	1592	243	2614	1124	714	588
3	1839	247	3578	964	1282	568
4	2089	250	4604	1026	2773	1491
5	2334	245	5747	1143	4525	1752
6	2576	242	6731	984	5165	640
7	2824	248	7789	1058	7488	2323
8	3072	248	8849	1060	8830	1342
Average	2209	246	5175	1051	3863	1243

Table 2.9: Problem size and solution time change depending on the number of scenarios

2.5.5 Correlation between total cost and fuel cost savings

To gain more insight, we provide a pairwise comparison of six scenarios, two with low means equal to 30 km/h and 40 km/h, two with higher means equal to 55 km/h and 70 km/h, and with variances equal to either 64 or 144, yielding eight combinations. The tests show how the traffic conditions affect both the fuel consumption and the total cost of the solutions. The tests are run on the instance UK10_01 and the results are reported in Table 2.10, where each entry corresponds to a pair of scenarios, and contains two rows. The first row reports the percentage reduction in fuel consumption achieved by the OR-OS strategy over the deterministic approach, and it is displayed in the form (minimum, average, maximum). A negative value indicates that the deterministic approach performed better than the OR-OS strategy. The second row reports similar statistics for the total solution cost.

The results shown in Table 2.10 suggest that the savings in total cost and in fuel consumption are not necessarily correlated. In fact, a large reduction in total cost does not imply a large reduction in fuel consumption. In this respect, Demir et al. (2014a) pointed

	$\hat{\mu} = 40,\ \hat{\sigma}^2 = 144$	$\hat{\mu} = 70,\ \hat{\sigma}^2 = 144$
$\hat{\mu} = 30,\ \hat{\sigma}^2 = 64$	(−0.2%, 0.4%, 1.2%)	(−1.8%, 0.1%, 2.2%)
	(9.3%, 12.9%, 15.6%)	(7.6%, 15.8%, 20.8%)
$\hat{\mu} = 30,\ \hat{\sigma}^2 = 144$	(−1.9%, −1.3%, −0.8%)	(−1.2%, −0.1%, 1.3%)
	(3.2%, 7.6%, 13.3%)	(6.7%, 12.7%, 16.2%)
$\hat{\mu} = 55,\ \hat{\sigma}^2 = 64$	(−3.7%, −1.1%, 1.1%)	(−2.0%, −0.7%, 1.4%)
	(3.9%, 5.7%, 9.3%)	(10.1%, 11.8%, 13.2%)
$\hat{\mu} = 55,\ \hat{\sigma}^2 = 144$	(−0.5%, 2.1%, 5.6%)	(−1.7%, 0.0%, 2.7%)
	(4.4%, 11.4%, 20.1%)	(7.2%, 13.5%, 24.7%)

Table 2.10: Comparison between deterministic and OR-OS solution values

out that the objective function is composed of two cost components: fuel consumption and driver cost. These are often conflicting, and the latter plays a more important role in achieving reductions in the overall cost. The results indicate that the largest savings both in fuel and in driver cost are achieved when the average speed across all scenarios is less than the the globally optimal speed v^* that minimizes fuel consumption (which depends on the type of vehicle and is between 50 and 60 km/h in our case), otherwise the savings in fuel consumption are less than 0.6% on average. The results also show that the deterministic approach was able to save fuel on some instances, but this saving is achieved at the expense of the overall cost. Finally, when the average speed of one scenario is lower than v^* and the other is higher than v^*, the two approaches yield similar fuel consumptions, but the OR-OS strategy performs better in terms of the overall cost.

2.5.6 The value of optimizing speed

Most papers relevant to the stochastic VRP, such as those reviewed in Section 2.2.2, do not consider speed as a decision variable. This means that if arc (i, j) is chosen in the first stage, then it is assumed that the vehicle will travel on that arc at a traffic speed that is scenario dependent. From a decision-making point of view, this does not allow any flexibility in changing vehicle speeds and does not take into account the flexibility of reducing speed to save on fuel cost. From a methodological point of view, this is a special case of our approach where the speed decision variables are fixed to the traffic speeds on all arcs.

To quantify the value of optimizing speed, we present results using the first 10 instances of the testbed (namely UK_10_01 to UK_10_10), on which we compare the percentage savings afforded by optimized speeds against the case where speeds are fixed to the traffic speed value prescribed by the scenarios. Table 2.11 shows the results over 3 realizations of each scenario in terms of the minimum, average and maximum savings, presented under the three main columns. For each of the columns, we again provide the minimum, average and maximum savings obtained. For example, the column titled Minimum and sub-column titled Av shows the average minimum saving achieved across the 10 instances and the 3 realizations of the scenarios. The values are expressed in percentage of the objective value which includes only fuel costs. Each row represents a setting, numbered 1–5, that are generated using the scenarios where settings 1 and 2 include one, settings 3, 4 and 5 include two scenarios. The scenarios are defined by the following distributions $(\hat{\mu} = 40,\ \hat{\sigma}^2 = 64)$; $(\hat{\mu} = 70,\ \hat{\sigma}^2 = 144)$; $(\hat{\mu} = 70,\ \hat{\sigma}^2 = 144), (\hat{\mu} = 30,\ \hat{\sigma}^2 = 64)$; $(\hat{\mu} = 70,\ \hat{\sigma}^2 = 144), (\hat{\mu} = 80,\ \hat{\sigma}^2 = 164)$; $(\hat{\mu} = 40,\ \hat{\sigma}^2 = 64), (\hat{\mu} = 30,\ \hat{\sigma}^2 = 64)$.

Setting	Minimum (%)			Average (%)			Maximum (%)		
	Min	Av	Max	Min	Av	Max	Min	Av	Max
1	0.0	0.0	0.0	0.0	0.1	0.3	0.0	0.3	0.4
2	0.0	0.1	0.3	0.1	0.3	0.8	0.2	0.7	1.5
3	0.0	0.0	0.0	0.1	0.1	0.3	0.1	0.3	0.5
4	0.0	0.0	0.2	0.1	0.2	0.4	0.1	0.4	0.7
5	0.0	0.0	0.0	0.0	0.1	0.2	0.1	0.2	0.4

Table 2.11: Savings achieved by considering speed as a decision variable

Table 2.11 shows that considering speed as a decision variable can result in average savings of up to 0.8% in terms of the total cost. Low savings are due to the fact that driver cost is the largest cost in the objective function (Bektaş and Laporte, 2011). With the introduction of autonomous trucks, the driver is not in control of the vehicle on the highways, and hence driver cost can be removed from the objective function to minimize exclusively fuel consumption and GHG emissions.

In Table 2.12, we repeat the experiment, with the same scenarios described above. This time however, we do not consider driver wage in the objective function. We can see that, on average, considering speed as a decision variable can achieve savings on fuel consumption and GHG emissions between 0.7% and 3.8%. The average minimum saving is achieved in setting 1, whereas the average maximum is 4.4% is made on setting 2, where the maximum saving achieved is 8.5%. This shows that the savings made are highly correlated with general trend of the traffic speed on the network. Indeed, when the traffic speed is high, the fuel consumption is also high. Therefore, if the vehicle has to travel on all arcs at the traffic speed, it will not choose some arcs that have high speed, which in turn will increase the fuel consumption. This situation is avoided by

considering speed as a decision variable, since the vehicle then has the ability to travel at speed that is lower than the traffic speed.

Setting	Minimum (%)			Average (%)			Maximum (%)		
	Min	Av	Max	Min	Av	Max	Min	Av	Max
1	0.0	0.6	1.4	0.7	1.5	3.8	1.1	2.5	7.9
2	0.0	1.3	2.1	1.7	2.7	3.6	2.5	4.4	8.4
3	0.5	1.1	1.6	1.2	1.8	2.5	1.7	2.6	4.2
4	0.5	1.3	2.6	1.5	2.2	3.3	1.9	3.2	4.5
5	0.2	0.8	1.8	0.8	1.4	1.9	1.5	2.0	2.4

Table 2.12: Savings achieved by considering speed as a decision variable without driver

2.5.7 The effect of variability in traffic speed

This section reports results when more than one segment is assumed for each arc in the network to be able to evaluate the effect of any speed variability.

We consider a road network where we generate a traffic speed for two segments per arc and then solve the problem using the OR-OS approach. To compare this approach, we consider another network where the traffic speed on one arc is the average of its two segments and solve using the same approach. From this test we see the difference in cost of the two approaches which represents the accuracy in handling speed variability brought by considering more segments.

Table 2.13 shows the results of these two settings. We see that, on average, considering two segments increased the accuracy by 8% compared to only one segment per arc. We also notice that ignoring speed variability may result in an over-optimistic approach that underestimates the real cost. From a computational complexity point of view, considering two segments slows down the solution time by a factor of almost six.

2.6 Conclusions

We have introduced, modeled and solved the Pollution-Routing Problem with stochastic traffic speed. We have used a comprehensive cost function and considered stochastic speeds to represent the uncertainty in traffic conditions. Since, the traffic speed on each arc is stochastic, the travel times were also treated as random variables. We first modeled the problem considering discrete uncertainty representation, where the traffic speeds are represented by a finite set of scenarios. A two-stage non-linear stochastic program was developed. We linearized the resulting model by using a discretization of both speed variables and speed recourse variables. The use of a discrete recourse has an significant impact on the performance of the algorithm. We therefore proposed another linearization

Instance	Two segments		One segment		
	Optimal value	Seconds	Optimal value	Seconds	Gap (%)
UK10_01	247.3	7500.0	230.3	804.0	7.4
UK10_02	309.3	8634.0	284.4	797.0	8.8
UK10_03	285.4	6475.0	268.2	1008.0	6.4
UK10_04	275.4	6385.0	253.9	826.0	8.5
UK10_05	252.1	8690.0	237.2	3852.0	6.3
UK10_06	322.5	7916.5	292.1	1347.0	10.4
UK10_07	278.2	3826.0	254.8	410.5	9.2
UK10_08	325.6	1813.0	299.1	223.3	8.9
UK10_09	250.3	3283.0	232.0	239.7	7.9
UK10_10	272.6	2659.0	254.7	274.3	7.0
Average	281.91	5718.15	260.67	978.18	8.0

Table 2.13: Comparison between network with two segments per arc and with one segment per arc

technique that uses a continuous recourse instead of a discretized one. When 10-node instances from the PRP-lib were solved using both formulations, the continuous recourse model outperformed the discretized recourse model, yielding an average cost reduction of 4.95%. The continuous recourse strategy was also compared with its deterministic counterpart in which traffic speeds are the averages across all scenarios. This enabled us to estimate the value of solving a stochastic model as opposed to a deterministic one. The continuous recourse model yielded average cost savings of 7.48%.

We have also studied a decision making process in which the routing decisions are made before the information about stochastic events becomes available, but speeds can be chosen after the realizations of the scenarios. While this scenario is less likely to occur in practice, it was implemented to evaluate the value of perfect information on traffic condition. When this model was solved on the same instances, only 1.27% of the total cost was saved with respect to the continuous recourse model.

We also proposed a way to represent traffic speeds variability by considering several segments on a given arc. We showed that considering two segments per arc can increase the accuracy of the solution by 8.0% on average.

Chapter 3

Integer L-shaped Algorithms for the Pollution-Routing Problem under Traffic Uncertainty

Abstract

The previous chapter defined the Pollution-Routing problem with stochastic traffic speed. It showed that by considering traffic uncertainty, 7.48% of the total cost could be saved on average on 10-node instances. To model this uncertainty, the PRP was modeled as a two-stage stochastic program, which is known to be difficult to solve. As a result, the commercial solver used was not able to solve instances with more than 10 nodes. With the aim to solve larger instances of the problem, this chapter introduces new solution techniques based on decomposition. Two-stage stochastic programs have a particular structure as the first stage problem and the second stage problem can be solved separately with the relaxation of some constraints. Several techniques based on the integer L-shaped algorithm are proposed in this chapter. These techniques differ in the way in which the stochastic program is relaxed and on how the first stage and the second stage problems interact. Several mathematical formulation, cuts and lower-bounds are proposed together with local search procedures. The techniques proposed are tested and compared on instances from the PRP-lib with sizes between 20 and 50 customers.

Keywords: pollution-routing; stochastic programming; Integer L-shaped method; decomposition.

3.1 Introduction

Stochastic mixed-integer program have been extensively used in the literature to tackle uncertainty in optimization problems. Two-stage stochastic programs are of particular interest as they have the ability to model decisions and uncertainty realizations over a time-line (Grass and Fischer, 2016). However, these problems combine two difficult decision problems, connected by a set of constraints, which makes them computationally intractable for large problem sizes (Laporte and Louveaux, 1993).

Although two-stage stochastic programs are difficult to solve in their original form, they have a special structure that can be exploited by dedicated methods. Several decomposition methods have been proposed to take advantage of that special structure, by decomposing the problem in a set of smaller and often relaxed subproblems. The decomposition of stochastic programs is generally done either on scenarios or on decision stages (Guo et al., 2015).

We model the PRP under uncertain traffic condition using the two-stage stochastic program described in Section 2.4.2.2 (2.44)–(2.57), which was proved in Chapter 2 to provide the best solutions. The first stage of this model consists of finding a set of feasible routes, whereas the second stage consists of optimizing speed of the vehicle on

those routes. The first stage uses binary decision variables, and the second stage is a mixed-integer mathematical program. To decompose this problem we use the integer L-shaped method of Laporte and Louveaux (1993). We decompose the problem in a master and a subproblem. The master problem consists of solving the routing part of the problem, while the subproblem addresses the speed decisions on each leg of the routes found by solving the master problem. We propose several techniques based on the integer L-shaped method, together with strategies and lower bounds to improve the techniques.

The remainder of this chapter is structured as follows. Section 3.2 presents a brief review of the literature of applications of the integer L-shaped method to stochastic problems. Section 3.3 explains the methodology of the methods and mathematical models described. Section 3.4 presents computational experiments, and conclusions follow in Section 3.5

3.2 Literature review

Since the introduction of the PRP (Bektaş and Laporte, 2011), a number of authors have investigated a range of variants, see (Koç et al., 2014; Franceschetti et al., 2013; Demir et al., 2014a; Tajik et al., 2014). A review of these papers is available in Section 2.2.2. For a wider and comprehensive overview of the green transportation, including air and maritime transportation, we refer the reader to Psaraftis (2016). Here, We review relevant studies that used the integer L-shaped method since its introduction by Laporte and Louveaux (1993).

Laporte and Louveaux (1993) proposed the integer L-shaped method to solve two-stage stochastic problems with binary first stage decisions. The method was called integer L-shaped due to its similarity with the L-shaped method of Van Slyke and Wets (1969). The method consists of relaxing some constraints in a given formulation, and re-integrating them gradually until an optimal solution is found. The authors defined feasibility cuts and optimality cuts that are derived after every iteration.

The integer L-shaped method was first adapted to the VRP with stochastic demands and customers by Gendreau et al. (1995). The authors found that stochastic customers are more difficult to handle than stochastic demands. The problem was modeled as a two-stage stochastic program, where the first stage consists of finding a set of routes, and the second stage consists of skipping absent customers, and returning to the depot to unload and resume the route if demand is exceeded. The algorithm could solve instances with 10 to 70 customers, depending on the difficulty of the instance, and the values of some input parameters. Salavati-Khoshghalb et al. (2017) considered the same problem allowing the vehicle to return the depot in anticipation of potential failures when the capacity becomes less than a threshold. The authors also defined lower bound

to approximate the cost of the recourse strategy. Instances with four vehicles and 60 customers were solved optimally.

Recent applications of the integer L-shaped method include the integrated problem of staffing and scheduling nurses by Kim and Mehrotra (2015). The authors model the problem as a two-stage stochastic program where the first stage finds a feasible solution to the staffing and scheduling problem. The second stage consists of adjusting the first stage decisions according to the realizations of the stochastic demand. The authors developed several cuts for the problem, and showed that significant cost savings were achieved over the deterministic version of the problem.

Angulo et al. (2016) proposed several strategies to improve the integer L-shaped method. A first strategy consists of computing the cost of the second stage problem by alternating between linear and integer programing. In addition, the authors developed a cut-generating linear program, that finds cuts based on previously explored solutions. The techniques performance depend on the convexity and complexity of the objective function of the problem. The method reduces the solution time when tried on two different problems, the stochastic multiple knapsack and the stochastic server location problem.

3.3 Methodology

In this section, we describe an integer L-shaped decomposition algorithm for the mathematical model described in Section 2.4.2.2 (2.44)–(2.57) that results in a master and a subproblem, and how the interaction between these two problems is operated in the different methods proposed.

3.3.1 The original integer L-shaped method

A two-stage stochastic problem can be modeled as follows:

$$\min\ z = cx + Q(x) \tag{3.1}$$

$$\text{s.t. } Ax = b, x \in X, \tag{3.2}$$

where

$$Q(x) = E_{\xi}[\min_{y}\{q(\omega)y \mid Wy = h(\omega) - T(\omega)x, y \in Y\}], \tag{3.3}$$

where c and b are two known vectors in $\mathbb{R}^{n1}$ and $\mathbb{R}^{n2}$, respectively. The vector $x \in \{0,1\}^n$ represents the first stage binary decision variables. $Q(x)$ is a function of input x and random parameters q, W, h and T represented by the set ξ. Finally, the vector $y \in Y$

represents the second stage decisions. The problem (3.1)–(3.2) can be reformulated as follows:

$$\min z \quad = cx + \theta \tag{3.4}$$

$$\text{s.t. } Ax = b, x \in X \tag{3.5}$$

$$Q(x) \quad - \theta \leq 0, x \in X. \tag{3.6}$$

In order to solve (3.4)–(3.6), the integer L-shaped method consists of relaxing the set of constraints (3.6), which results in solving a simplified master problem. In this formulation, θ is an underestimate of $Q(x)$. Constraints (3.6) are reincorporated iteratively for fixed vectors $\hat{x} \in X$ obtained by solving the master problem, until optimality is reached. The subproblem $Q(x)$ is solved for every optimal solution of the master problem $\hat{x}$ in every iteration. If a master problem solution is not feasible for the subproblem, a feasibility cut is derived and added to the master problem in the next iteration.

3.3.2 Adaptation to the PRP

In the stochastic PRP, the first stage decisions, which are made in the master problem, consist of selecting routes for all vehicles. The objective function of the master problem includes cost functions that only depend on routing. All expressions in the objective function of the stochastic PRP that depends on speed or travel times, are scenario-dependent, and are therefore in the objective function of the subproblem. The master problem is as follows:

$$\text{Minimize} \sum_{(i,j)\in A} w f_c \gamma \lambda \alpha_{ij} d_{ij} x_{ij} + \sum_{(i,j)\in A} f_c \gamma \lambda \alpha_{ij} d_{ij} f_{ij} + \theta \tag{3.7}$$

subject to

$$\text{Constraints (2.75)–(2.82).}$$

The output of the master problem is the optimal solution $\hat{x}$ for the current iteration that represents a set of routes. We define S as the set of variables x_{ij} that are equal to 1 in $\hat{x}$. The subproblem consists of finding optimal speeds for vehicles to drive at on each leg of the routes. This problem is a stochastic version of the speed optimization problem described by Norstad et al. (2011), which we name here the stochastic speed optimization problem (SSOP). Here, the arrival times at customers are scenario-dependent, and time-windows can be violated at the expense of penalties. Therefore, the algorithm designed

by Demir et al. (2012) for the deterministic version of the SSOP fails to generalize for our problem. The mathematical model of the subproblem is described as follows:

$$\text{Minimize } Q(\hat{x}) = \sum_{s\in S} p_s \Big(f_c K \Omega V \lambda \sum_{(i,j)\in A} d_{ij} \Big(\sum_{r\in R} (z_{ij}^r / v^r) + SF_{ij}^s \Big) \tag{3.8}$$

$$+ f_c \beta \gamma \lambda \sum_{(i,j)\in A} d_{ij} \Big(\sum_{r\in R} z_{ij}^r (v^r)^2 - SS_{ij}^s \Big) \tag{3.9}$$

$$+ \sum_{j\in N_0} f_d \delta_j^s \tag{3.10}$$

$$+ \sum_{j} el_j^s \Big) \tag{3.11}$$

subject to

$$\text{Constraints (2.54)–(2.57), (2.60)–(2.69).}$$

The optimal speeds found by solving the subproblem provide the total cost $Q(\hat{x})^*$ of the routes defined by S, which is used to add a cut to the master problem. The cuts such as defined by Laporte and Louveaux (1993) for the problem are

$$\theta \geq Q(\hat{x})^* \Big(\sum_{x_{ij}\in S} x_{ij} - (|S| - 1) \Big) \tag{3.12}$$

The solution found by the master problem is evaluated by solving the subproblem. The algorithm iterates in this manner and converges if solving the master problem returns a solution already visited. A pseudo-code of this method, that we denote Lshaped_O, is provided in Algorithm 1.

Algorithm 1 Lshaped_O

```
Optimal ← False
θ ← 0
while not Optimal do
    Solve the master problem
    x̂ ← Optimal Solution of the master problem
    if θ > 0 then
        Optimal = True
        x* ← x̂
    else
        Minimize Q(x̂)
        Add cut (3.12) to the master problem
    end if
end while
return x*
```

3.3.3 Integer L-shaped with lower bound

The idea behind the L-shaped method is to relax a set of constraints, and to reincorporate a subset in an iterative fashion. By relaxing those constraints, no information about the subproblem is available in the master problem, which may lead the method to explore non-promising solutions. To guide the master problem in finding promising solution for the main problem, and without losing optimality, we add a lower bound on the subproblem in the objective function of the master problem. This lower bound provides an optimistic estimate of the second stage cost. We refer to this method as Lshaped_A.

The lower bound on the second stage problem is obtained with the following reasoning. The objective function of the deterministic PRP decomposes in two parts, fuel consumption and driver wage, Demir et al. (2014a). In the stochastic PRP, an additional term representing the cost of delays is also present in the objective function. The fuel consumption expression is also divided into two expressions that depend on load and speed. The load-dependent expression is determined by the routing decisions, which are made in the master problem. The speed-dependent expression is minimized in the subproblem. For a given scenario, the objective function is the sum of the following expressions

$$f_c K \Omega V \lambda \sum_{(i,j)\in A} d_{ij} \Big(\sum_{r\in R} (z_{ij}^r / v^r) + SF_{ij}^s \Big) \tag{3.13}$$

$$f_c \beta \gamma \lambda \sum_{(i,j)\in A} d_{ij} \Big(\sum_{r\in R} z_{ij}^r (v^r)^2 - SS_{ij}^s \Big) \tag{3.14}$$

$$\sum_{j\in N_0} f_d \delta_j^s \tag{3.15}$$

$$\sum_{j} el_j^s. \tag{3.16}$$

Terms (3.13) + (3.14) represent the fuel consumption related to speed variables z_{ij}^r and the speed recourse variables. It is also a convex function and its global minimum can be computed. Terms (3.15) + (3.16) represent the driver wage and the cost of delays, and is monotonously decreasing as speed increases and has a global minimum which can be computed analytically. A lower bound for (3.13) + (3.14) + (3.15) + (3.16) is the sum of a lower bound for (3.13) + (3.14) and one for (3.15) + (3.16).

We define V_f^* as the speed that minimizes the expression (3.13) + (3.14), and V_{ijs}^{max} as the maximum achievable speed across all scenarios $s \in S$ on an arc $(i,j) \in A$ and $V_{ij}^* = \min(V_f^*, V_{ijs}^{max})$.

A lower bound for (3.13) + (3.14) can be found by computing the cost of emissions based on V^*_{ij} as speed on the arc $(i,j) \in A$,

$$f_c K \Omega V \lambda \sum_{(i,j)\in A} d_{ij}/V^*_{ij} + f_c \beta \gamma \lambda \sum_{(i,j)\in A} d_{ij} V^{*2}_{ij}. \tag{3.17}$$

The lower bound on (3.15) + (3.16) is obtained by computing the arrival times at customer assuming that the vehicle travels on all route legs at speed V^{max}_{ijs}. This results in the vehicle arriving at the earliest possible time at each customer and minimizes the total duration of the routes, yielding minimized driver costs and late arrivals penalties. We add a new decision variable ζ to the master problem to represent the lower bound on the cost of the subproblem. The master problem is described as follows.

$$\text{Minimize} \sum_{(i,j)\in A} w f_c \gamma \lambda \alpha_{ij} d_{ij} x_{ij} + \sum_{(i,j)\in A} f_c \gamma \lambda \alpha_{ij} d_{ij} f_{ij} + \zeta + \theta \tag{3.18}$$

subject to

Constraints (2.75)–(2.82)

$$y^s_i - y^s_j + t_i + \sum_{r\in R} d_{ij}/V^{max}_{ijs} \leq M(1 - x_{ij}) \qquad i \in N, j \in N_0, i \neq j, s \in S \tag{3.19}$$

$$y^s_j - t_j - \delta^s_j + \sum_{r\in R} d_{j0}/V^{max}_{j0s} \leq M(1 - x_{j0}) \qquad j \in N_0, s \in S \tag{3.20}$$

$$a_j \leq y^s_j \leq b_j + l^s_j \qquad j \in N_0, s \in S \tag{3.21}$$

$$y^s_i \geq 0 \qquad i \in N_0, s \in S \tag{3.22}$$

$$\zeta \geq \Big(f_c K \Omega V \lambda \sum_{(i,j)\in A} d_{ij}/V^*_{ij} + f_c \beta \gamma \lambda \sum_{(i,j)\in A} d_{ij} V^{*2}_{ij} + \sum_{j\in N_0} f_d \delta^s_j + \sum_j e l^s_j \Big). \tag{3.23}$$

Using this definition of the lower bound ζ, we can compute a lower bound on all the solutions of the subproblem. We define the constant ι which is equal to the lower bound on the subproblem over all the master problem solutions. The value of ι is computed by solving the following problem.

$$\iota = \text{Min} \;\; \zeta \tag{3.24}$$

subject to

Constraints (2.75)–(2.82),(3.19)–(3.22).

The value of ι is then used to develop strengthened cuts as follows:

$$\theta \geq (Q(\hat{x})^* - \zeta - \iota)\Big(\sum_{x_{ij}\in S} x_{ij} - (|S| - 1)\Big) + \iota \tag{3.25}$$

3.3.4 Integer L-shaped with lower bound and local branching

In this section, we propose to include the local branching approach of Fischetti and Lodi (2003) within the integer L-shaped method. The idea behind the local branching technique is to explore neighborhoods of a given solution during the solution process. Commercial MIP solvers have a range of tools and built-in heuristics that makes them extremely fast in solving relatively small MIPs (Fischetti and Lodi, 2003). These tools can be effectively used to explore neighborhoods, which are a smaller and a restricted version of the problem.

In our implementation, after finding a solution that is better than the current best known solution, we explore its neighborhood until a stopping criterion is reached, following which we prune the neighborhood from the feasible region of the master problem. If the stopping criterion verifies that no neighboring solution is better than the best solution obtained so far, than the algorithm is an exact method. However, if the stopping criterion does not guarantee this condition, the algorithm becomes a heuristic.

We define a neighborhood of a solution by a set of solutions that has at least g similar arcs. Fixing the value of g impacts the solution time and the number of solutions discarded from the master problem. Too small a value of g yields a large neighborhood that is difficult to explore, whereas too large a value of g reduces the information extracted from the neighborhood and has smaller impact on the convergence of algorithm. The constraints added to obtain a neighborhood of a known solution $\hat{x}$, with S as the set of variables x_{ij} that are equal to 1 in $\hat{x}$ are

$$\sum_{x_{ij} \in S} \hat{x_{ij}} \geq g, \tag{3.26}$$

after reaching the stopping criterion, the neighborhood is pruned by adding the following constraint

$$\sum_{x_{ij} \in S} \hat{x_{ij}} < g, \tag{3.27}$$

The pseudo-code of the integer L-shaped with local branching, which we refer to as Lshaped_LB algorithm, is provided in Algorithm 2.

3.4 Computational experiments

In this section, we computationally compare the performance of the solution techniques proposed on instances from the PRP-lib with 20, 25 and 50 nodes. Every instance is solved three times, over different realizations of the traffic speed scenarios. The maximum time allowed for the resolution of one instance is 30 minutes. We also compare these results with the ones obtained by solving the original model (2.44)–(2.57) using

Algorithm 2 Lshaped_LB

```
Converged ← False
θ ← 0
while not Converged do
    Solve the master problem
    x̂ ← Optimal Solution of the master problem
    if θ > 0 then Converged = True
    else
        Solve Q(x̂)
        if x̂ is the best solution so far then
            Add cut (3.26) to the master problem
            while not stopping criterion of local branching do
                x̂^lb ← Optimal Solution of the restricted master problem
                if θ = 0 then
                    Solve Q(x̂^lb)
                    Add cut (3.12) to the master problem
                else
                    Neighbourhood fully examined
                end if
            end while
            Add cut (3.27) to prune the neighborhood from the master problem
        end if
    end if
end while
```

CPLEX 12.6.0.1 with a maximum number of four threads, and allowing the same maximum solution time. The tests were performed on a computer equipped with eight processors Intel Xeon E7-8837 and a RAM of 1 To.

In the computational experiments carried out, we considered fifteen equiprobable scenarios denoted S_1 to S_{15} generated using a log-normal distribution as explained in Section 2.5.1. The scenarios are defined by their mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ are displayed in Table 3.1

Scenario	$\hat{\mu}$	$\hat{\sigma}^2$
S_1	26	64
S_2	65	144
S_3	55	100
S_4	35	49
S_8	60	100
S_5	40	121
S_6	70	144
S_7	29	64
S_9	36	49
S_{10}	90	121
S_{11}	67	144
S_{12}	21	64
S_{13}	50	100
S_{14}	34	49
S_{15}	46	121

Table 3.1: Fifteen scenarios considered in instance

The results are shown in Tables 3.2, 3.3 and 3.4, where the first row displays the name of the corresponding approach. The column *Inst* shows the instance name contains the number of customers and the number of the instance. The best upper bound *(UB)* and lower bound *(LB)* are reported for each of the methods. For the L-shaped based methods, the iteration *Fit* and time *Ft* in seconds at which the best solution was found, together with the number of cuts *Nc*, the average master and subproblem solution time in seconds, *AMT* and *AST*, respectively are reported. The best solutions value found for each instances is displayed in bold in all the tables.

Within both the Lshaped_LB method and Lshaped_A, the lower bounding cut (3.23) in the master problem slows down the solution process considerably, as a result of which the algorithm cannot iterate after the first iteration. Therefore, we simplify the constraints and instead we use a lower bounding cut that considers the fuel consumption only, as shown below:

$$\zeta \geq f_c K \Omega V \lambda \sum_{(i,j)\in A} d_{ij}/V_{ij}^* + f_c \beta \gamma \lambda \sum_{(i,j)\in A} d_{ij} V_{ij}^{*2}. \tag{3.28}$$

The performance of the Lshaped_LB method depends on the number of solutions to visit in the neighborhood. To tune this parameter, we solved 30 instances with 20 and 25 nodes. The values that were tested are 5, 10 and 15 solutions per neighborhood. The results provided by the algorithms were very close, however, selecting 10 solutions

yielded better result with an average cost of £460 , while selecting 5 and 15 solutions yielded a cost of £461 and £460.3, respectively.

Table 3.2 shows that the Lshaped_LB method provided the best solutions on average, and found most of the best solutions on 20-node instances. All the L-shaped based method found lower-cost solutions than CPLEX. On 25-node instances, we can see from Table 3.3 that the Lshaped_O found the lowest-cost solutions, on average, and Lshaped_LB and Lshaped_A performed almost identically. Results on 50-node instances are reported in Table 3.4. CPLEX did not provide any feasible solution on 50-node instances within the 1800 seconds time limit, therefore, we allowed it to solve the instances in three hours. The L-shaped based methods for these instances did not perform more than one iteration, and therefore the Lshaped_LB and Lshaped_A provide the same results which are reported under column named Lshaped_A & _LB. CPLEX could not provide a feasible solution for five instances, and provided the best upper bound on only one instance. The results provided by all the L-shaped based methods were similar on average.

It is shown in all the three tables that, even though all the variants of the L-shaped based method performed similarly on average, the gap between the approaches can be significant on some instances. This is due to the specificity of each method, which gives it advantage depending on the instance solved. However, on all the instances, the method outperformed CPLEX by an average gap in the cost between 5% and 14%. CPLEX provided the best lower bounds on all instances, followed by Lshaped_A then Lshaped_O. Lshaped_LB does not provide valid lower bounds. We can also see from the tables that most of the solution time is spent on solving the master problem, and this time increases drastically with the size of the instance.

We can also see from the three tables that the Lshaped_LB provides the best upper bounds on 12 instances with 20 nodes, followed by Lshaped_A with 7 and Lshaped_O with 3 instances. On 25-node instances Lshaped_LB still provides the best upper bounds on 10 instances, followed by Lshaped_O with 7 and Lshaped_A with 4 instances. The Lshaped_O yielded to 13 best upper bounds on 50-node instances, while only 7 were provided by Lshaped_A and Lshaped_LB which provided the same results on these instances. We can also see that the number of best upper bounds of the Lshaped_O increases with the number of nodes in the instances. Whereas Lshaped_LB performs better on smaller instances.

	CPLEX		Lshaped_O							Lshaped_A							Lshaped_LB					
Inst	UB	LB	UB	LB	Fit	Ft	Nc	AMT	AST	UB	LB	Fit	Ft	Nc	AMT	AST	UB	Fit	Ft	Nc	AMT	AST
20_01	560.6	246.4	**507.5**	141.6	3	57	71	25	0.3	507.8	182.3	12	205	79	23	3.1	507.9	10	121	82	22	0.9
20_02	568.4	261.3	555.2	150.2	6	50	101	18	0.3	**551.0**	193.9	38	676	83	21	3.1	551.8	65	1034	102	17	1.1
20_03	307.5	158.9	303.0	84.0	43	683	93	19	0.3	**300.5**	108.0	59	766	107	17	3.6	**300.5**	23	178	99	18	1.1
20_04	593.2	253.7	525.7	144.2	8	860	17	102	0.3	525.9	185.0	10	804	19	93	3.3	**521.2**	18	984	29	61	1.1
20_05	529.7	218.9	485.0	131.2	12	262	69	26	0.3	**468.7**	167.5	52	1161	74	24	2.8	469.7	41	679	87	21	0.9
20_06	706.4	258.7	634.3	153.1	4	991	6	254	0.3	634.1	194.2	3	419	10	174	2.7	**633.0**	8	434	20	81	0.9
20_07	369.7	176.5	374.1	95.9	2	639	5	322	0.3	367.0	122.1	3	756	4	196	2.9	**359.6**	7	488	14	98	1.0
20_08	459.6	213.2	436.1	120.2	68	1147	100	18	0.3	427.3	153.5	74	1136	106	17	3.5	**426.8**	81	1065	121	15	1.2
20_09	588.6	269.4	615.1	147.3	47	1136	64	28	0.3	678.2	190.6	40	170	152	12	1.7	**595.9**	106	1050	142	13	0.6
20_10	476.3	210.7	462.0	126.7	9	786	18	94	0.3	458.8	162.5	17	1103	27	65	3.4	**455.8**	19	614	34	52	1.1
20_11	638.7	286.3	650.5	166.6	26	806	49	37	0.3	**642.7**	211.3	19	222	81	22	2.4	**642.7**	11	71	78	23	0.7
20_12	580.2	246.7	544.5	138.3	41	1104	60	30	0.3	547.8	179.6	40	1459	47	38	3.1	**543.9**	48	1189	63	28	1.0
20_13	575.6	231.9	567.7	139.2	29	652	71	25	0.3	**558.6**	179.0	62	981	99	18	2.1	564.2	35	394	108	17	0.7
20_14	575.6	231.9	587.5	181.3	7	395	25	68	0.3	**556.4**	235.2	27	938	43	41	2.7	564.0	29	781	51	35	0.8
20_15	607.4	302.8	507.5	140.5	25	1216	32	55	0.3	518.9	181.3	26	1476	30	58	2.5	**506.7**	31	1115	44	39	1.0
20_16	571.5	262.3	545.8	146.7	6	572	16	108	0.3	531.8	185.5	10	485	31	58	2.9	**530.8**	22	858	39	46	0.9
20_17	N/A	N/A	**631.8**	165.5	10	815	20	88	0.3	637.6	208.2	15	1261	19	88	3.2	633.2	23	997	35	50	1.1
20_18	603.4	276.6	**611.9**	155.7	29	881	55	32	0.3	681.0	198.0	31	590	78	23	2.3	691.5	20	169	83	21	0.8
20_19	592.0	242.7	538.6	143.4	6	387	20	86	0.3	538.3	183.5	6	448	20	88	3.0	**507.9**	10	121	82	22	0.9
20_20	537.7	268.0	543.6	145.9	88	926	126	14	0.3	**534.1**	189.7	69	293	186	10	2.5	**534.1**	50	175	162	11	0.9
Av	549.6	243.0	531.4	140.9	23.4	718.2	51.0	72.4	0.3	533.3	180.5	30.5	767.4	64.8	54.2	2.8	**527.1**	32.9	625.8	73.7	34.5	0.9

Table 3.2: Results on 20-node instances

	CPLEX		Lshaped_O							Lshaped_A							Lshaped_LB					
Inst	UB	LB	UB	LB	Fit	Ft	Nc	AMT	AST	UB	LB	Fit	Ft	Nc	AMT	AST	UB	Fit	Ft	Nc	AMT	AST
25_01	548.7	194.5	**475.5**	121.4	32	1385	39	45	0.2	482.9	155.7	15	406	49	36	1.9	477.0	21	1120	32	55	1.3
25_02	593.9	267.2	556.3	146.1	1	1128	3	**544**	0.2	550.7	188.0	2	1199	4	428	2.1	568.1	1	1041	4	443	1.1
25_03	370.7	145.3	328.7	86.9	1	868	3	518	0.3	**319.4**	109.9	6	1686	8	237	3.0	319.6	9	1594	12	269	1.5
25_04	479.4	164.3	451.4	56.0	0	1420	1	1420	0.2	446.8	142.0	2	729	4	248	1.8	**443.7**	3	642	13	105	1.1
25_05	553.4	263.6	**442.8**	141.2	7	484	20	86	0.2	442.9	181.9	4	692	10	161	3.4	**442.8**	4	614	11	146	1.2
25_06	500.7	207.9	**486.7**	121.3	5	831	13	129	0.2	520.2	155.3	23	1077	35	50	1.7	520.0	2	161	17	86	0.9
25_07	632.9	222.4	**615.5**	98.4	0	1427	1	1445	0.2	688.1	111.5	0	1365	1	1365	1.4	688.1	0	1365	1	1365	1.4
25_08	725.7	234.8	675.1	155.5	1	1127	3	488	0.2	**606.0**	196.0	8	1025	13	130	1.8	607.5	7	578	15	106	1.2
25_09	481.6	225.1	466.3	121.0	22	1175	29	54	0.8	449.0	157.2	16	848	22	68	1.9	**455.9**	18	1240	26	64	1.0
25_10	551.7	301.3	545.1	157.3	10	423	43	41	0.7	543.5	200.0	25	857	44	40	2.2	**543.3**	20	840	39	45	1.2
25_11	654.4	284.8	**570.7**	80.5	0	1800	0	1800	0.2	619.8	124.2	0	1290	0	1290	1.1	619.8	0	1290	0	1290	1.1
25_12	689.4	306.4	640.0	175.5	38	1087	62	29	0.4	**632.9**	226.3	23	680	59	30	4.1	633.2	24	501	73	25	1.1
25_13	383.7	170.9	363.6	92.8	43	1411	53	33	0.2	351.7	119.4	3	250	18	98	3.3	**348.3**	10	640	26	64	1.1
25_14	607.4	302.8	578.4	166.6	32	753	69	26	0.2	578.4	212.8	23	790	46	39	3.4	**577.4**	32	944	56	32	1.1
25_15	871.1	269.1	723.6	163.3	30	1118	44	40	0.2	738.5	208.4	41	1209	58	30	2.8	**723.2**	43	1179	60	29	0.9
25_16	601.8	256.1	**508.0**	148.3	36	1290	48	37	0.3	591.3	190.5	39	709	87	21	1.6	594.8	44	979	73	24	0.7
25_17	847.0	345.8	776.7	206.5	19	915	33	52	0.2	776.7	260.8	7	495	20	82	2.7	**776.1**	14	880	26	63	1.0
25_18	1276.2	264.5	758.5	171.3	3	346	15	122	0.2	**754.5**	216.8	5	203	35	50	1.5	**754.5**	2	96	31	56	0.8
25_19	724.0	322.8	695.1	178.7	42	1023	71	25	0.3	669.9	231.0	52	1384	65	27	3.6	**662.8**	54	1191	74	24	0.9
25_20	678.4	267.5	**572.3**	76.8	0	1801	0	1800	0.4	587.1	115.2	0	1801	0	1801	1.2	587.1	0	1801	0	1801	1.2
Av	638.6	250.9	**561.5**	133.3	16.1	1090.5	27.4	436.8	0.3	567.5	175.1	14.7	934.9	29.0	311.6	2.3	567.2	15.4	934.7	29.3	304.7	1.1

Table 3.3: Results on 25-node instances

	Lshaped_O		Lshaped_A & _LB		CPLEX	
Inst	UB	LB	UB	LB	UB	LB
50_01	919.5	126.1	**914.7**	190.6	1608	444
50_02	**937.9**	129.5	953.4	195.6	1290	441
50_03	**1012.6**	133.2	1018.7	203	1588	449
50_04	**1169.7**	161.1	1170.5	248.1	1535	584
50_05	1180.3	142.9	**1143.7**	210.2	-	419
50_06	**909.3**	118.9	911.3	181.3	1725	412
50_07	**856.6**	113.2	862.4	172	1588	385
50_08	**955.3**	116.8	961.5	178.6	-	401
50_09	**1100.7**	148.1	1110.6	226.4	1251	510
50_10	**1122.6**	145.6	1123.3	224.6	-	475
50_11	**1062.9**	135.5	1064	202.6	1073	464
50_13	940.8	119.1	**933.8**	185.9	957	414
50_12	886.5	120.8	**881.8**	187.5	1028	445
50_14	1082.9	143.6	**1068.0**	219.9	1210	505
50_15	1089.1	135.6	**1027.5**	194.2	-	410
50_16	898.1	124.4	899	191	**894**	441
50_17	**714.3**	86.9	742.6	148.6	1816	311
50_18	**1096.3**	143.5	1107.3	220.1	1170	521
50_19	985.1	126.8	**972.1**	191.8	1021	422
50_20	**1123.9**	147.4	1127.1	225.5	-	489
Average	1002.2	131	**999.7**	199.9	1512	453

Table 3.4: Results on 50-node instances

In Tables 3.2 and 3.3, we report the average results obtained by solving the instances three times, over different realizations of the 15 traffic speed scenarios described above. To assess the robustness of the L-shaped algorithms against uncertainty, we show in Table 3.5 the difference between the best and the worst solution obtained on these three different realizations. The results are displayed as percentage of the best solution value found by the algorithm. We see from Table 3.5 that on average Lshaped_A has the lowest variability on both 20 and 25-node instances, followed by Lshaped_LB and then Lshaped_O. We can also observe a significant increase in the variability for 25-node instances compared with 20-node instances for all approaches.

	20 Nodes			25 Nodes		
Instance	Lshaped_O (%)	Lshaped_A (%)	Lshaped_LB (%)	Lshaped_O (%)	Lshaped_A (%)	Lshaped_LB (%)
01	2.5	2.8	2.8	7.4	8.7	6.5
02	4.4	4.2	4.0	2.0	4.1	10.9
03	1.0	1.0	1.0	3.8	1.4	0.9
04	2.4	1.5	3.5	5.7	8.4	6.2
05	3.4	1.3	1.2	0.7	0.7	0.7
06	1.6	1.6	1.1	0.5	1.2	1.5
07	8.3	2.6	3.1	4.2	1.7	1.7
08	3.3	2.4	2.6	8.7	11.1	11.9
09	5.8	5.4	6.4	1.3	1.5	4.2
10	2.3	2.5	1.6	4.8	5.3	5.1
11	6.0	2.5	2.5	6.7	6.7	6.7
12	5.6	7.9	8.1	1.2	0.2	1.1
13	4.6	7.3	7.0	4.7	5.7	5.6
14	6.3	4.1	5.8	1.7	1.8	1.7
15	2.2	4.3	2.2	6.2	0.5	6.2
16	0.5	3.1	2.7	3.0	1.1	0.6
17	8.2	7.7	8.6	5.9	4.2	4.2
18	4.1	5.1	4.3	8.1	6.6	6.6
19	5.7	5.6	5.6	9.8	7.0	6.8
20	1.0	1.8	1.8	6.7	4.4	4.4
Average	4.0	3.7	3.8	4.7	4.1	4.7

Table 3.5: L-shaped algorithms robustness against uncertainty

In Figure 3.1 we display the upper provided by the Lshaped_O and Lshaped_A on the instance 20_01 at every iteration. We can see that the initial upper bound found by the Lshaped_O, UB_O was better than the one provided by Lshaped_A, UB_A. During the solution process the upper bound were similar, however after the 78th iteration Lshaped_A improved the upper bound. In Figure 3.2 we report the same results for the instance 25_01. We see that the best upper bound provided by Lshaped_A was found initially and was not changed. Lshaped_O improves the upper bound two times, however it does not provide a solution better than the Lshaped_A.

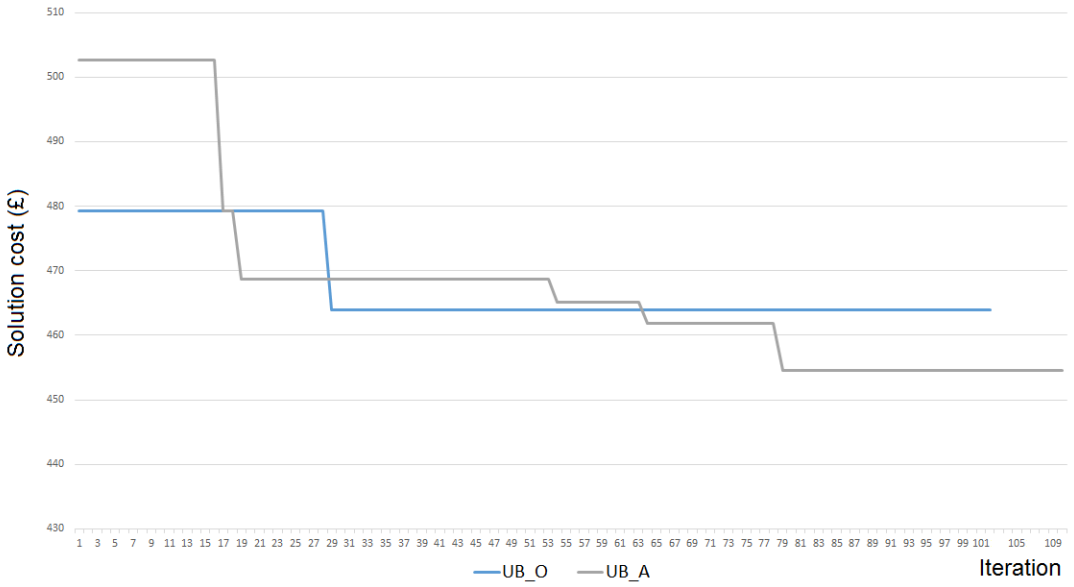


Figure 3.1: Upper and lower bound as obtained on 20-node instance during solution process by Lshaped_O and Lshaped_A

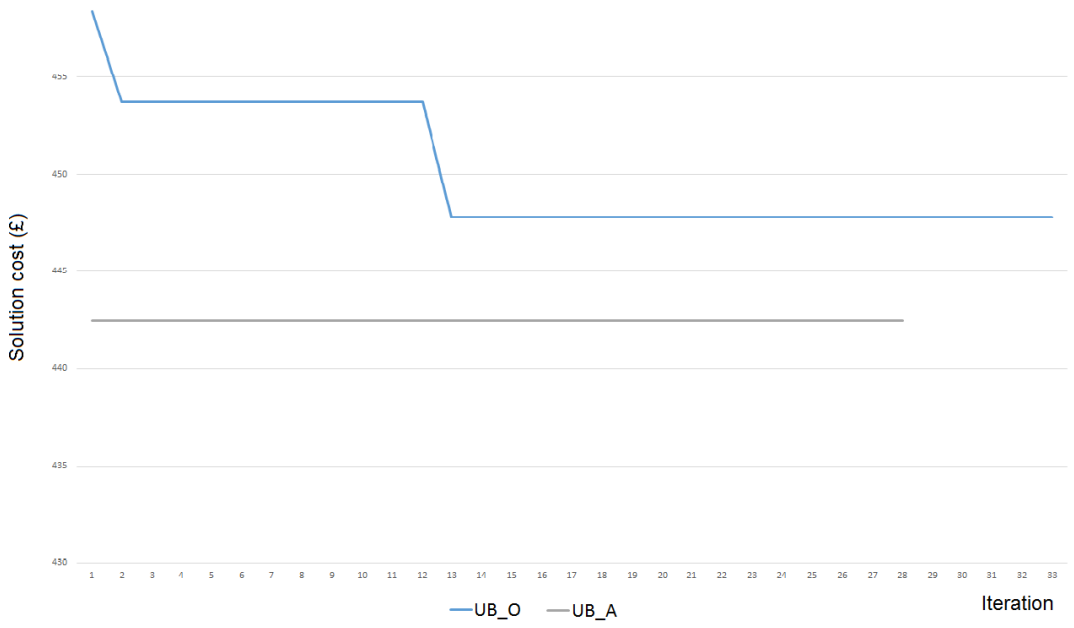


Figure 3.2: Upper and lower bound as obtained on 25-node instance during solution process by Lshaped_O and Lshaped_A

3.5 Conclusions

We aimed in this chapter to tackle the Pollution-Routing Problem under uncertain traffic conditions. We proposed techniques to solve the problem using the Integer L-shaped method adapted to a stochastic MIP where the first stage has only binary variables. We developed three approaches, which have different cuts and strategies. The first method was the adaptation of the original integer L-shaped, whereas the second method was an adaptation of the previous method, where we defined a lower bound on the cost of the second stage, which allowed us to strengthen the cuts and approximate better the subproblem cost in the master problem. The third approach proposed included the use of local branching technique. It consisted of exploring neighborhood of solution found during the solution process. The three method were tested on 20, 25 and 50-node instances from the PRP-lib. No method could provide a proof of optimality for any instance. However, the L-shaped based method yielded solutions of better quality in terms of the total cost as compared to CPLEX, which on 50-node instances could find any feasible solutions except for one instance. On average, the three L-shaped methods provided similar results.

Chapter 4

Speed Optimization under Soft Time Window Constraints

Abstract

Reducing pollution is increasingly a concern for policy makers and companies in the transportation sector. The amount of pollution emitted is directly related to the quantity of fuel consumed by the vehicles. One way of reducing the fuel consumption is to optimize speed on a given journey. This chapter studies a speed optimization problem that consists of choosing the optimal speed on each leg of a given route, which is represented by a fixed visit sequence of customers. Each customer has a delivery time window. Early and late arrivals to customers are allowed, but at the expense of penalties. The problem generally arises within a broader set of decisions within maritime and road transportation planning. The objective function in maritime transportation accounts for fuel consumption and penalties for early and late arrivals, whereas driver cost is also taken into account in road transportation. This chapter describes a non-linear program for the problem, which is linearized in several ways, including the use of time-space networks. Extensive computational experiments are carried out on road and maritime instances to assess the performance of the methods and to derive managerial insights for both modes of transportation.

Keywords: Speed optimization; pollution minimization; time-space networks; non-linear programming.

4.1 Introduction

The amount of pollution emitted by vehicles is related to the quantity of fuel consumed (Demir et al., 2014b; Fagerholt et al., 2010), which in turn depends on speed, among other factors. Optimizing speed is therefore relevant to reducing the fuel consumption of vehicles. Reducing fuel consumption is a problem faced in many logistics applications, particularly within maritime and road transportation. Emissions from maritime transportation were estimated to be 400 million tons in 2014, which contributed to both air pollution and ocean acidification (i.e, reducing the pH levels of the sea) (Organization, 2018). The International Maritime Organization (IMO), which is responsible for international policies to reduce maritime pollution, has limited the amount of emissions of certain greenhouse gases, introduced emission controlled areas (EAs), and started in 2018 to track the fuel consumption of ships of at least 5,000 gross tonnage in an attempt to reduce the overall maritime fuel consumption. Reducing fuel consumption in maritime transportation is also of economic importance for maritime companies. In 2012, for example, it was estimated that in some modes of maritime transportation, fuel cost accounts for up to 60% of the total cost (Wang and Meng, 2012). The importance of reducing fuel consumption in road transportation have been highlighted in Chapter 2.

In this chapter, we investigate the problem of computing an optimal speed for each leg of a given route, characterized by a fixed sequence of customers, each of which should ideally be visited within a specified time window. This is known as the Speed Optimization Problem (SOP) (Fagerholt et al., 2010). The SOP is particularly relevant to fuel consumption reduction and its application to different modes of transportation, including maritime, where it is a core problem for shipping applications with a fixed sequence of ports to visit (Norstad et al., 2011) and road, where it arises as a subproblem within the Pollution-Routing Problem (PRP) (Demir et al., 2012). Due to the non-linear nature of SOP formulations, the problem is usually solved either by a non-linear optimization software or by specialized algorithms, as in Norstad et al. (2011), who described a Speed Optimization Algorithm (SOA), under hard time window constraints.

Corbett et al. (2009) explored the effect of speed reduction on the total cost and emissions in shipping, applied to profit-maximizing shipping companies. The authors took into account fuel price and CO_2 emission taxes. Two possible scenarios were considered. The first scenario consisted of assuming ships traveling at lower speeds with less frequent travels. The second scenario assumed speed reduction with the introduction of additional vessels. The authors showed that CO_2 emission taxes can contribute to fuel consumption reduction, and a reduction in emissions of 70% and 45% can be achieved by reducing speed by 50% in scenario 1 and scenario 2 respectively.

Norstad et al. (2011) considered the SOP in tramp shipment context within a framework of ship routing and scheduling. The authors first modeled the problem as a mixed-integer mathematical program where the authors used speed discretization to overcome the non-linearity of the model. Two algorithms were developed to solve the problem. The first algorithms consisted in solving a shortest-path problem on a time-space network where arrival times at customers were discretized. The second algorithm applied a recursive process based on customers hard time windows and the convexity of the objective function. The second was proven better than the first one but is applicable only if the consumption function does not depend on the load. Hvattum et al. (2013) later proved the optimality of this recursive algorithm.

Wang and Meng (2012) considered the SOP in the case of liner shipping. The authors modeled the problem as a mixed integer non-linear program with an objective function that accounts for ship operating cost, bunker cost and container handling cost. Since the objective function is proven to be convex, the author proposed an ϵ-optimal outer-approximation method algorithm. The algorithm solved efficiently a case study with with 46 ports with an approximation of 0.1%.

Fagerholt et al. (2015) studied the problem of routing and speed optimization for ships in the presence of EAs. The authors developed a mixed integer linear program, discretizing speed, to investigated how emission-controlled areas impact ship routes and speeds, as well as the fuel consumption. The authors studied several realistic case studies, and

showed that ship routes may change with the introduction of EAs, resulting in ships sailing longer roads to avoid these areas. It was also showed, that ships may travel at slower speeds within EAs, and at higher speed outside. The authors also affirm that the introduction of EAs is likely to reduce general greenhouse gases emissions.

Andersson et al. (2015) the SOP in a roll-on roll-off shipping context. The authors developed a mixed integer program that integrates speed optimization in the planning of shipping routes. The authors also proposed a rolling horizon heuristic (RHH) to solve the problem with a large planning horizon. The RHH obtained good solutions within reasonable times and performed better than solving the problem using a commercial mixed-integer programming solver for all instances of realistic size. Instances with up to 53 ships over a 10-month planning horizon were solved.

Fagerholt (2001) studied the problem of ship scheduling with soft time windows. The author first generated a set of ship routes, from which the most promising solutions with respect to defined heuristic rules were retained. For each promising route, the optimal speeds were computed using a time-space network, and the best schedule was computed by solving a set partitioning problem. This solution method was designed to solve real-word ship scheduling problems. In our study, we generalize this speed optimization algorithm to make it applicable to a wider range of transportation modes.

Fukasawa et al. (2016a) considered a joint route and speed optimization problem with the objective of minimizing the total cost, including fuel costs. The authors developed a generic branch-cut-and-price (BCP) algorithm that can be applied to any problem with a convex fuel consumption function. The authors proposed a novel set partitioning formulation that takes advantage of the structure of the speed optimization problem. The method was applied to both maritime and road transportation. On maritime instances, the BCP algorithm was able to solve more instances to optimality than a commercial solver, and was three times faster on average. On road transportation instances, the authors solved instances from the PRP-lib and compared the result with those obtained by the branch-and-cut of Fukasawa et al. (2016b). The BCP algorithm outperformed the branch-and-cut and solved some instances to optimality for the first time.

He et al. (2017) considered the SOP with heterogeneous arc costs. The authors argued that fuel consumption also depends on vessel load and weather, and that fuel costs change between ports which can be captured by using a fuel consumption function for each arc. The authors developed an efficient algorithm capable of solving instances with up to 1,000 ports within one second, that is 20 to 100 times faster than a standard non-linear solver. The SOP has also been studied in road transportation, particularly since the introduction of the PRP by Bektaş and Laporte (2011). Demir et al. (2012) solved the SOP as a subproblem of the PRP. The authors adapted the recursive algorithm of Norstad et al. (2011) as a subroutine of an adaptive large neighborhood search (ALNS) metaheuristic.

Our aim in this chapter, is to build on the existing body of literature and study the SOP with Soft Time Windows (SOPSTW) where penalties are imposed for early or late arrivals. In some road and maritime applications, customer time window constraints are allowed to be violated at the expense of a penalty. Similar to (Demir et al., 2012; Norstad et al., 2011; He et al., 2017) we propose an algorithm capable of solving the problem efficiently. In addition, we develop methods that do not rely on the infeasibility in case of time window violation. We show that these outperform the approach presented in (Fagerholt, 2001) to solve the SOPSTW. We solve instances of the problem in both road and maritime transportation as in (He et al., 2017), and we propose an acceleration procedure that allows solving the problem quicker. We first model the problem and then introduce transformations to overcome its non-linearities, and we develop algorithms that solve the problem without resorting to a black-box optimization software. An extensive computational experiment is carried out to derive managerial insight and study the effect of considering soft time windows.

The remainder of the paper is organized as follows. Section 4.2 formally defines the problem, and models it by means of a non-linear formulation. Section 4.3 describes several discretization schemes and algorithms. Section 4.4 presents the computational results. Concluding remarks follow in Section 4.5.

4.2 Problem description and non-linear model

We first formally define the SOPSTW and present a non-linear mathematical model. The SOPSTW is defined on a path composed of n arcs representing legs of a route on $n+1$ nodes defining the set $N = \{0, 1, \ldots, i, \ldots, n\}$. Each arc $(i, i+1)$ is referred to as arc i. Node 0 represents the depot, and $N_0 = \{1, \ldots, n\}$ is the set of customers. All customers are served by a single vehicle. Each customer $i \in N_0$ requires service of duration t_i and can be serviced at no cost within a time window $[a_i, b_i]$. The length of an arc and the speed at which the vehicle travels on it are denoted by d_i and v_i, respectively. The speed used on any arc must lie within the interval $[v^l, v^u]$, as dictated by speed enforcement laws. The vehicle arrives at node i at time y_i in an interval $[a_i - e_i,\ b_i + l_i]$, where e_i and l_i are non-negative decision variables. Unit time penalties p_e and p_l are applied for early and late arrivals, respectively. The objective function consists of minimizing the sum of time window penalties, driver wages with f_d as the cost of driver per unit of time, and a non-linear convex emission function $E_f(v, d)$ that computes the cost of fuel consumption and GHG emissions for a vehicle traveling at speed v on an arc of length d. An explicit description of $E_f(v, d)$ is given in Sections 4.4.1 and 4.4.2 for road and maritime transportation, respectively. A non-linear model (NLM) of the problem

is described as follows:

$$\text{Minimize} \sum_{i \in N \setminus \{n\}} E_f(v_i, d_i) \tag{4.1}$$

$$+ f_d y_n \tag{4.2}$$

$$+ p_l \sum_{i \in N_0} l_i \tag{4.3}$$

$$+ p_e \sum_{i \in N_0} e_i \tag{4.4}$$

subject to

$$y_{i+1} = y_i + t_i + d_i / v_i \qquad i \in N \setminus \{n\} \tag{4.5}$$

$$a_i - e_i \leq y_i \leq b_i + l_i \qquad i \in N_0 \tag{4.6}$$

$$v_i \leq v^u \qquad i \in N \setminus \{n\} \tag{4.7}$$

$$v_i \geq v^l \qquad i \in N \setminus \{n\} \tag{4.8}$$

$$y_0 = 0 \tag{4.9}$$

$$e_i \geq 0 \qquad i \in N_0 \tag{4.10}$$

$$l_i \geq 0 \qquad i \in N_0. \tag{4.11}$$

The first term of the objective function computes the cost of fuel consumption and emissions. Term (4.2) represents the driver wage, and terms (4.3) and (4.4) compute the penalties for arriving late and early at customer locations, respectively. Constraints (4.5) compute the arrival time at each customer. Constraints (4.6) model the soft time windows. Constraints (4.7) and (4.8) ensure that the vehicle speeds lie within the legal limits. Constraint (4.9) sets the departure time of the vehicle from the depot at time 0. Constraints (4.10) and (4.11) indicate the non-negativity of e_i and l_i variables.

Due to (4.1) and (4.5), the above formulation is non-linear, and solving it with a standard optimizer can be time consuming, in particular for large-scale instances. In Section 4.3 we describe alternative models and procedures to overcome the difficulty yielded by this non-linear model.

4.3 Discretization schemes and algorithms

This section presents discretization schemes and algorithms to overcome the non-linearities (4.1) and (4.5) in the NLM.

4.3.1 Discretized speed model (DSM)

One possible discretization scheme is to represent the travel speed on each arc as a discrete and finite set of speed levels. This technique is similar to that introduced by Bektaş and Laporte (2011) for the PRP. The travel speed on an arc belongs to a set $R = \{v^1, \ldots, v^r, \ldots\}$. We define a binary variable z_i^r equal to 1 if and only if the vehicle travels on arc $(i, i+1)$ at speed v^r. The mathematical model is described as follows:

$$\text{Minimize} \quad \sum_{i \in N \setminus \{n\}} \sum_{r \in R} E_f(z_i^r v^r, d_i) \tag{4.12}$$

$$+ f_d y_n \tag{4.13}$$

$$+ p_l \sum_{i \in N_0} l_i \tag{4.14}$$

$$+ p_e \sum_{i \in N_0} e_i \tag{4.15}$$

subject to

$$y_{i+1} = y_i + t_i + d_i \sum_{r \in R} (z_i^r / v^r) \qquad i \in N \setminus \{n\} \tag{4.16}$$

$$a_i - e_i \leq y_i \leq b_i + l_i \qquad i \in N_0 \tag{4.17}$$

$$\sum_{r \in R} z_i^r = 1 \qquad i \in N \setminus \{n\} \tag{4.18}$$

$$y_0 = 0 \tag{4.19}$$

$$e_i \geq 0 \qquad i \in N_0 \tag{4.20}$$

$$l_i \geq 0 \qquad i \in N_0 \tag{4.21}$$

$$z_i^r \in \{0, 1\} \qquad i \in N \setminus \{n\}, r \in R. \tag{4.22}$$

Expressions (4.12)–(4.17) are equivalent to (4.1)–(4.6). Constraints (4.18) ensure that only one speed level is chosen per arc. Constraints (4.19)–(4.21) are the same as (4.9)–(4.11). Constraints (4.22) indicate the binary nature of variables z_i^r.

While the DSM allows overcoming the non-linearity of the model, it may result in solving a considerably larger model (Norstad et al., 2011). In fact, if the distances between customers are large, then a small change in speed would induce a significant change in arrival times at customers. For this reason, one should consider very fine speed discretizations to achieve sufficient accuracy. One way of avoiding this problem is to discretize the possible arrival times at customers. This approach is conceptually equivalent to discretizing speed, but requires fewer discretization steps, and hence results in smaller models to solve. This will be discussed in the following section.

4.3.2 Discretized arrival times: time-space networks

In the presence of soft time windows, one cannot readily apply the methodology used by Norstad et al. (2011) designed for SOP with hard time windows, which consists of discretizing the possible arrival times at customers. In this section, we present different ways to discretize arrival times at customers using time-space networks.

To resolve this issue, we first create artificial hard time windows. The lower end of a hard time window a_i^h is computed by assuming that the vehicle travels on all the arcs at the maximal possible speed $a_i^h = a_{i-1}^h + d_i/v^u$. This will yield the earliest possible arrival time for each customer. We then compute the upper end of the time window b_i^h by using this time the slowest possible speed, which will yield the latest possible arrival time $b_i^h = b_{i-1}^h + d_i/v^l$. We then define a graph $G = (V, E)$, where V and E are the node and arc sets, respectively. Let T_i be the set of possible arrival times at customer i. Each customer i is represented by a set of nodes $\{i_1, i_2, \ldots, i_{|T_i|}\}$, representing the discrete arrival times at that customer, and node i_t corresponds to the vehicle arriving at customer i at time $t \in T_i$. An arc between two nodes i_t and $(i+1)_k$ exists if and only if the vehicle can depart after servicing customer i at time t to arrive at customer $i+1$ at time k. The cost of this arc includes the driver wage, the fuel consumption cost and a penalty for violating the soft time window at customer $i+1$. We define a dummy node $n+1$ connected with arcs of cost 0 to all the nodes in the graph corresponding to the last customer in the route. An optimal solution to the SOPSTW corresponds to a shortest path from node 0 to node $n+1$. Two examples of such graph are displayed in Figure 4.1.

The graph G does not contain cycles, and hence computing shortest path from node 0 to node $n+1$ can be achieved efficiently by means of topological sorting of the graph (Cherkassky et al., 1996). The nodes from 0 to $n+1$ are sorted in such a way that if a node x is ranked before node y, there is no arc going from y to x. The shortest path is represented as a sequence of nodes $SP = (0, 1_{t_1}, 2_{t_2}, \ldots, n+1)$ where $t_i, i \in N_0$ is the index corresponding to the vehicle arriving at customer i at time t_i. The shortest path from node 0 to node $n+1$ is computed as shown in Algorithm 3.

4.3.2.1 Fixed time step (FTS)

Discretizing the arrival times at customers can also be achieved in different ways. The first technique that we present is that of Fagerholt (2001), which consists of considering as an input parameter a fixed time step $\delta > 0$ that separates each two consecutive arrival times, i.e., $\delta = i_t - i_{t-1}$. However, for the last possible arrival time $i_{|T_i|}$ at customer i, the inequality $i_{|T_i|} - i_{|T_i|-1} \leq \delta$ holds. Algorithm 4 shows the computation technique of the possible arrival times at customer i with a hard time window $[a_i, b_i]$.

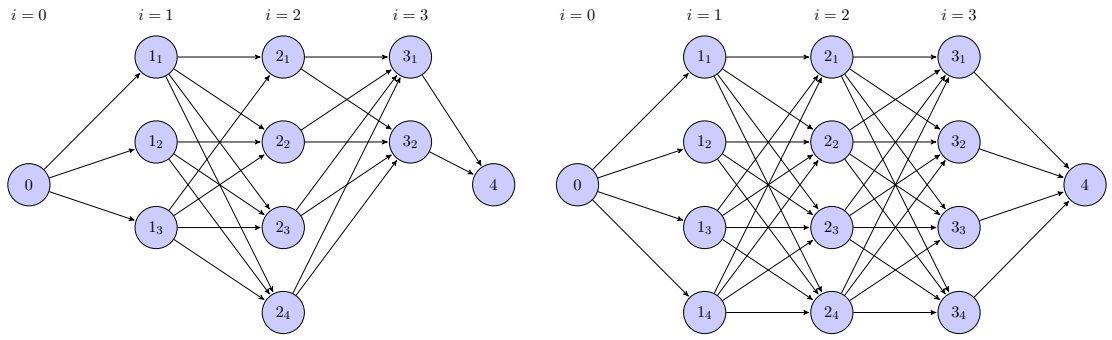
Algorithm 3 Discretized arrival time algorithm

Input: distances, customer sequence, service times, soft time windows
Input: cost function cost(x, y), defined for every arc (x, y) in graph G
Create artificial hard time windows
Create graph G
Sort topologically graph G (x comes before y, if and only if $\nexists$ arc $(x, y) \in G$)
Vector label
label[0] = 0
for node $x \in G$, $x \neq 0$ **do** label[x]=∞
end for
for arcs $(x, y) \in G$ **do**
    if label $[y]$ > label $[x]$ + cost(x, y) **then**
        label $[y] \leftarrow$ label $[y]$ + cost(x, y)
    end if
end for
Compute a shortest path SP
return SP

Algorithm 4 Possible arrival times computation

Input : time step δ
Input : customer index i
$p \leftarrow a_i$
$t \leftarrow 1$
while $p < b_i$ **do**
    $i_t \leftarrow p$
    $p \leftarrow p + \delta$
    $t \leftarrow t + 1$
end while
$i_t \leftarrow b_i$

This technique considers more possible arrival times at a customer having a large time window than at a customer with a narrower time window. The time step δ can thus be seen as the desired accuracy of the discretization. Figure 4.1a depicts an instance where customer 2 has the largest time window, followed by customer 1 and customer 3. A node label i_t means that the vehicle arrives at customer i at time t.



(a) Fixed time step graph

(b) Fixed size graph

Figure 4.1: Time-space networks example with different discretization techniques

4.3.2.2 Fixed size graph (FSG)

An alternative technique to discretize the possible arrival times consists of considering a discretization set T_i of a fixed and equal cardinality $|T_i| = m$ for all customers $i \in N_0$. The number of nodes in the graph, taking into account the depot and node $n + 1$, is then $nm + 2$. Figure 4.1b depicts a graph with $m = 4$.

4.3.2.3 Fixed size graph with zooming (FSG-Z)

Artificial hard time windows built using the procedure described in Section 4.3.2 are usually large, which yields large-scale graphs. We propose an iterative procedure to overcome this problem and accelerate the FSG technique. It consists of solving a series of small problems rather than a single large one. We iteratively solve a series of FSGs with a fixed and small number of discretization steps until reaching a stopping criterion. At every iteration, the customer time windows are narrowed down in a zooming fashion. This approach is described in Algorithm 5.

In Algorithm 5, we first create the artificial hard time windows for each customer. We then create the graph and find the shortest path SP from 0 to $n + 1$ using the FSG method presented in Section 4.3.2.2. We then iterate on the elements of the shortest path $i_{t_i} \in SP, i \in N_0$ that represent the optimal arrival time at each customer. For all customers $i \in N_0$, we recompute the artificial time window taking into account the arrival time selected in the shortest path in the previous iteration. We repeat this process until meeting the stopping criterion.

Algorithm 5 FSG-Z Procedure

```
Input : distances, customer sequence, service times, soft time windows
Converged ← False
Create artificial hard time windows
while not Converged do
    Create graph G
    Solve FSG on graph G using Algorithm 3 and compute the shortest path SP
    if SP satisfies stopping criterion then
        Converged ← true
    else
        for node i_{t_i} ∈ SP do
            if t_i = 0 then
                a_i ← i_{t_i}
            else
                a_i ← i_{t_i − 1}
            end if
            if t_i = m then
                b_i ← i_{t_i}
            else
                b_i ← i_{t_i + 1}
            end if
        end for
    end if
end while
return SP
```

4.3.2.4 Theoretical complexity of time-space algorithms

All three algorithms FTS, FSG and FSG-Z operate on a time-space network, in which shortest path problems are solved using Algorithm 3 in an acyclic, topologically sorted graph $G = (V, E)$. We denote by $K = |V|$ the number of nodes in the graph, and we denote the number of its arcs by $M = |E|$. The algorithms use labels for every node of the graph, and does so by visiting all the arcs of the graph following the topological order. Therefore, the algorithm does $O(K + M)$ iterations to find a shortest path.

The complexity of Algorithm 3 depends on the number of nodes and arcs in the graph. Therefore, to compare between the approaches FTS, FSG and FSG-Z, we evaluate the sizes of the graphs produced by each approach, and how many graph each approach solves.

The FTS approach creates arrival times such that there is a fixed time step δ between each two discretizations. The maximum number of discretization per customer is achieved at the customer i with the largest time window of width w_i. The number of nodes of the graph generated with FTS is $O(nw_i/\delta)$, where n is the number of customers. Since the nodes representing the same customer are not connected, and only the nodes of consecutive customers are connected, the number of arcs in the graph is $O(n(w_i/\delta)^2)$.

The FSG approach creates a fixed number discretizations m per customer, and therefore the number of nodes is $O(nm)$. The number of arcs of the graph is $O(nm^2)$. We can conclude that if m is smaller than (w_i/δ), then FSG is easier to solve, otherwise FTS will be easier.

The FSG-Z approach creates the same number of nodes and arcs as FSG, but it operates differently to attain the desired accuracy. In particular, FSG-Z solves a series of problems using a smaller number m of discretizations. Since the number of arcs is polynomial in terms of m, a much smaller number of arcs are generated, which makes FSG-Z faster than both FSG and FTS.

4.3.2.5 Illustrative example

To show how the algorithms mentioned above operate, we provide an illustrative example consisting of a route with four ports, starting at Antwerp (Belgium), calling at Milford Haven (United Kingdom), Boston (United States) and finishing at Charleston (United States). This example is a sub-route of the example presented by Fagerholt et al. (2010). Figure 4.2 shows the distances between ports (miles), and time windows (hours) above each arc and node, respectively. The objective function considered in this illustrative example is explained in Section 4.4.2.

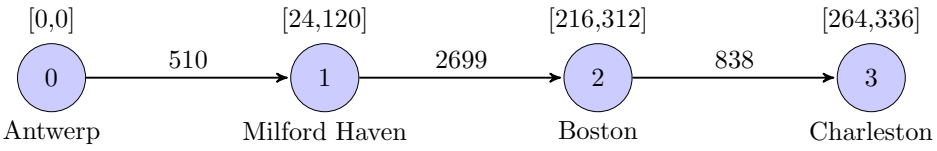


Figure 4.2: Distance and time window data

We solve this instance with FTS, FSG and FSG-Z. We present In Figures 4.3 and 4.4 two graphs created by using FTS in Figure 4.3, with a discretization step of 24 hours, and by using FSG in Figure 4.4, with 100 discretization steps. We also solve the instance with the FSG-Z algorithm, we consider 20 discretization steps and stop when the difference between each two consecutive discretization is 10 hours. Since the graphs created by FSG-Z are similar to those created by FSG, showing a graph created by using FSG is sufficient.

In Table 4.1 we give detailed results of solving the instances with FTS, FSG and FSG-Z. For every algorithm, we report the number of discretizations *Nb disc* for all four nodes denoted from n1 to n4. Line *Opt speed* shows the optimal speed in knots found by the algorithm on departure from the corresponding node. The cost, number of nodes *Nb nodes* and number of arcs *Nb arcs* in the resulting graph, total number of iterations performed by the algorithm *Nb iterations*, the solution time in milliseconds *Sol time* and

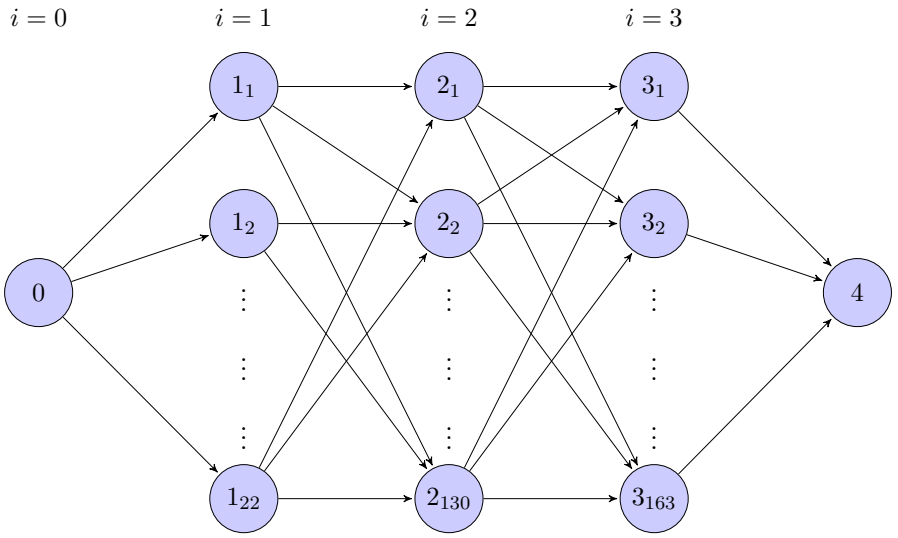


Figure 4.3: Time-space network created using FTS

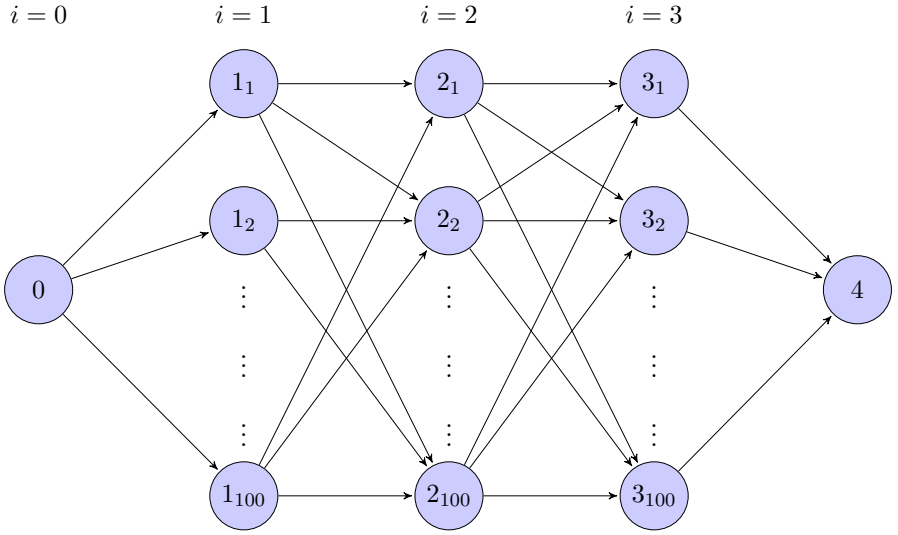


Figure 4.4: Time-space network created using FSG

the number of graphs generated *Nb graphs* are also reported. We can see from Table 4.1 that the number of discretization in the FTS method depends on the node, while for the FSG and FSG-Z is fixed. The number of arcs and nodes created by the FTS and FSG are almost 10 times larger than those created by FSG-Z. We can also see that FSG-Z yields the lowest cost after solving three graphs. Table 4.2 provides information about the three graphs solved by FSG-Z represented by columns FSG-Z1, FSG-Z2 and FSG-Z3. Additionally to what has been reported in Table 4.1, we provide the earliest discretization *Earliest disc* and the latest discretization *Latest disc* for each customer to show how the zooming procedure works.

	FTS				FSG				FSG-Z			
Nodes	n1	n2	n3	n4	n1	n2	n3	n4	n1	n2	n3	n4
Nb disc	1	22	130	163	1	100	100	100	1	20	20	20
Opt speed	11.5	13.2	14.7	-	14.5	14.5	14.5	-	14.1	14.1	14.1	-
Cost	332				323				322			
Sol time	10				10				0			
Nb nodes	317				302				186			
Nb arcs	24072				20100				2460			
Nb iterations	24389				20402				2646			
Nb graphs	1				1				3			

Table 4.1: Algorithms run statistics

We see from Table 4.2 that the number of nodes and arcs is small, which results in a small number of iterations. We see also that the difference between the earliest discretization of customers decreases significantly after two iterations, from 490 hours to 5 hours in the case of the first customer. We see also that the cost is reduced after each iteration and speed is adjusted accordingly. By solving small instances, the FSG-Z method was able to provide better solutions faster on this instance.

	FSG-Z1				FSG-Z2				FSG-Z3			
Nodes	n1	n2	n3	n4	n1	n2	n3	n4	n1	n2	n3	n4
Earliest disc	0	20.4	128	161	0	20.4	128	161	0	34	213	269
Latest disc	0	510	3209	4047	0	72	452.63	570	0	39	247	312
Opt speed	0	11	11	11	0	13.9	13.9	13.9	0	14.1	14.1	14.1
Cost	393				322.4				322.1			
Sol time	0				0				0			
Nb nodes	62				62				62			
Nb Arcs	820				820				820			
Nb Iterations	882				882				882			

Table 4.2: FSG-Z iterations analysis

4.4 Computational experiments

This section aims to evaluate the performances of the linearization methods described in Section 4.3 compared to the non-linear formulation (NLM), and to derive managerial insights for maritime and road transportation. We first carry out intensive computational experiment solving 120 instances in the road transportation context, which allows us to conclude which method is better and to derive managerial insights for road transportation. We then solve 30 maritime transportation instances with the aim of deriving insights related to the consideration of soft time windows.

All tests were performed on a computer equipped with an Intel Core i7-3770 processor, 3.4 GHz and a RAM of 8GB. The non-linear model described in Section 4.2 was solved using the open-source solver Interior Point Optimizer (Ipopt) 3.12.4 using the

AMPL modeling language. The mixed integer linear program described in Section 4.3.1 were solved using CPLEX 12.6.0.1 with default options within the C++ programming language.

4.4.1 Road transportation

There exist several emission models depending on the vehicle type. Demir et al. (2014b) reviewed and compared the available models for road freight transportation. Here, as in Chapter 2, we use the emission model of Barth and Boriboonsomsin (2008) which is defined in Section 2.3. The values of the fuel consumption model parameters are the same as those used in Table 2.4.

4.4.1.1 Instance generation

All experiments were conducted on randomly generated instances with a number of nodes varying between 20 and 200, where, for a given number of nodes, we generate and solve 20 different instances. The distances in meters between successive customers were generated from a continuous uniform distribution $\mathcal{U}(3000, 15000)$. Using these distances, the minimum speed v^l, and the maximum speeds v^u, we generated the largest possible time window $[a_i^{min}, b_i^{max}]$ for each customer using an approach similar to that described in Section 4.3.2. The upper and lower limits of the soft time window were drawn from the continuous uniform distribution $\mathcal{U}(a_i^{min}, b_i^{max})$. The service times (in seconds) were drawn from the continuous uniform distribution $\mathcal{U}(300, 1500)$.

4.4.1.2 Parameters and results

We solved all instances using the NLM and the algorithms described in Section 4.3. The results are presented in Table 4.3, where the name and the parameters of each algorithm appear in the columns "Algorithm" and "Parameter". The remaining columns show the number of the nodes in the instances solved.

For every algorithm and parameter, we provide the average solution time in milliseconds (ms), and the normalized cost compared with the optimal value obtained with the non-linear model, expressed as the ratio (rt) to the value of optimal solution. For example, a value of 1.02 means that over the 20 instances solved, the gap between the solution found and the optimum is 2%.

The non-linear model does not contain any parameter. The parameter of the DSM method represents the speed step between every two speed levels. We considered three different speed step values which are one km/h (the possible travel speeds are $20, 21, \ldots, 96$), five km/h and 10 km/h. For the FSG algorithm, the parameter is the

number of possible arrival times at each customer. We considered four different values 50, 100, 150 and 200 of possible arrival times per customer. The parameter of the FST is the length of the time step δ between every two possible consecutive arrival times. We considered four time step values of 60, 180, 300 and 900 seconds.

The FSG-Z method uses two parameters which are displayed in Table 4.3. The first parameter is the number of possible arrival times at each customer. We considered 10, 20, 50 and 100 possible arrival times per customer. The second parameter is the stopping criterion. Since at every iteration the customers time windows are narrowed, we stop the method once the largest time window of any customer is smaller than a given accuracy threshold. We considered four different threshold values of 60, 300, 600 and 900 seconds.

4.4.1.3 Methods performance: trade-off between accuracy and time

We can see from Table 4.3 that the NLM found an optimal solution on each instance. The average solution time increases significantly with the size of the instance. This algorithm needed almost one second to solve instances with 200 customers. In attempt to reduce solution time for the NLM, we allowed Ipopt to stop if the optimality gap becomes less than 2%. Due to the convexity of the problem, this could only save one iteration and the Ipopt returned the same results and did not change the average solution time of the method.

The DSM algorithm found solutions within less than 0.2% of the optimum for all values of the single parameter. The solution times are drastically affected by the value of the speed step. The computation times obtained with one km/h speed step is larger than those of the NLM on all instances, and this algorithm is therefore not suitable for the problem. However, the computation times obtained with a speed step of five km/h are between one and three times smaller than those the NLM, and using a 10 km/h speed step is slightly faster.

The performance of the FSG algorithm and solution times depend on the number of nodes of the graph. Using 150 or more discretizations per customer always yields longer computation times than those of NLM. Using 50 and 100 discretizations yields similar results in terms of gap with respect to the optimum. However, setting the parameter at 50 discretizations, as opposed to 100, is almost three times faster on all instances.

The FTS with a 900 second time step provides poor solution values lying between 36% and 53% of the optimum. A time step of 60 or 180 seconds results in slower computing times than those of the NLM on all instances with at least 50 nodes. Using a 300 second time step quickly provided solution values quickly within less than 3% of the optimum for instances with 20 and 50 nodes; however, the algorithm was slower than solving the NLM by Ipopt on larger instances.

The FGS-Z algorithm contains two parameters, both of which affected solution time and accuracy. On the 20-node instances all settings yielded faster times and produced solution values within less than 1% of the optimum. However, using 10 or 20 discretizations can be up to 20 times faster than the NLM.

On instances with 50 nodes and more, FGS-Z with 100 discretizations is slower than NLM; FGS-Z with 50 discretizations yields solutions with an optimality gap between 1% and 2%, and a solution time similar to or slightly smaller than that of NLM. Applying FGS-Z with 10 or 20 discretizations yields solutions with an optimality gap of less than 2% and 3%, respectively, and a solution time between 10 to 20 times smaller than that of solving the NLM by Ipopt. Using 10 nodes provides in most cases the same results as for 20 nodes, and reduces the solution time by up 50%.

All algorithms tested possess different strengths, and none is strictly faster than the others or provides the best results on all instances. Whilst the SOP is a stand-alone problem, it generally arises within a more general problem of finding optimal routes and schedules where it may need to be repeatedly solved to optimality within very short time scales. We can obtain solutions within less than 0.2% of the optimum by using the DSM with five or 10 km/h speed steps, which is faster than solving NLM. If optimality is not required, and the SOP is solved as subproblem within heuristic methods, we can obtain solutions up to 10 times faster than with DSM and 20 times than with NLM and an average optimality gap of 2% by FSG-Z with 10 or 20 discretizations.

4.4.1.4 Soft time window, penalty and fuel consumption

We now investigate the effect of the penalty function on fuel consumption. In Table 4.4 we solve the problem, using the DSM model, with three different values of late and early arrival penalties: 0.0014, 0.0028 and 0.0056 (£/second). We compare the fuel cost (FC), the driver cost (DC) and the average time window violation per customer (ATVC) in minutes. The solution values obtained for the tests with penalties 0.0028 and 0.0056 are expressed in terms of percentage increase with respect to the solution values obtained for the 0.0014 penalty; negative percentages correspond to a decrease in the solution value. Every line reports average values across all solutions obtained for different instances with the same number of nodes.

We can see from Table 4.4 that using a larger penalty increases both the fuel consumption and the driver cost. This is due to the fact that the vehicle has to slow down to avoid arriving early at customer locations, which generally increases the fuel consumption and the driver cost. We can also see that the fuel cost increase is more important than that of driver cost; on the other hand, it decreases the ATVC. It is worth mentioning that for large instances, the ATVC decreases slightly and is almost steady, while the fuel and driver costs continue to increase.

Algorithm	Parameters	20 nodes		50 nodes		75 nodes		100 nodes		150 nodes		200 nodes	
		Cost (rt)	Time (ms)	Cost (rt)	Time (ms)	Cost (rt)	Time (ms)	Cost (rt)	Time (ms)	Cost (rt)	Time (ms)	Cost (rt)	Time (ms)
NLM	-	1.00	144	1.00	157	1.00	209	1.00	310	1.00	554	1.00	945
DSM	1	1.00	642	1.00	241	1.00	430	1.00	512	1.00	1021	1.00	1309
	5	1.00	49	1.00	127	1.00	173	1.00	193	1.00	290	1.00	378
	10	1.00	44	1.00	106	1.00	130	1.00	169	1.00	293	1.00	332
FSG	200	1.00	197	1.01	500	1.01	763	1.01	1007	1.02	1526	1.03	2016
	150	1.00	118	1.01	302	1.02	459	1.02	608	1.03	922	1.03	1216
	100	1.00	61	1.02	154	1.03	236	1.03	308	1.04	472	1.04	620
	50	1.01	19	1.04	46	1.05	69	1.05	93	1.05	141	1.04	183
FTS	60	1.00	143	1.00	2138	1.00	7119	1.00	16577	1.00	56336	1.00	130471
	180	1.01	22	1.01	275	1.01	884	1.01	1998	1.01	6592	1.01	15115
	300	1.03	11	1.02	114	1.02	351	1.02	783	1.02	2511	1.01	5695
	900	1.36	3	1.48	20	1.48	58	1.50	115	1.53	349	1.52	752
FSG-Z	100, 60	1.00	131	1.00	334	1.01	504	1.01	661	1.02	992	1.02	1453
	100, 300	1.00	127	1.00	336	1.01	507	1.01	665	1.02	993	1.02	1299
	100, 600	1.00	57	1.00	339	1.01	514	1.01	670	1.02	1002	1.02	1305
	100, 900	1.00	57	1.01	281	1.01	510	1.01	665	1.02	1000	1.02	1300
	50, 60	1.00	41	1.01	128	1.01	197	1.02	318	1.02	481.5	1.02	632
	50, 300	1.00	42	1.01	104	1.01	156	1.02	203	1.02	304	1.02	460
	50, 600	1.00	40	1.01	102	1.01	151.5	1.02	198.5	1.02	298.5	1.02	387
	50, 900	1.01	19.5	1.01	102	1.01	152.5	1.02	200.5	1.02	298.5	1.02	389
	20, 60	1.01	16	1.02	41	1.01	74	1.01	86	1.02	150	1.02	211
	20, 300	1.01	11	1.02	34	1.01	62	1.01	80	1.02	121	1.02	156
	20, 600	1.01	11	1.02	27	1.01	55	1.02	63	1.02	121	1.02	158
	20, 900	1.01	12	1.02	26	1.02	39	1.02	62	1.02	105	1.02	156
	10, 60	1.01	9	1.02	24	1.02	40	1.02	49	1.03	85	1.02	110
	10, 300	1.01	7	1.02	20	1.02	34	1.02	44	1.03	67	1.02	86
	10, 600	1.01	6	1.02	17	1.02	30	1.03	34	1.03	66	1.02	86
	10, 900	1.02	5	1.02	17	1.02	26	1.03	35	1.03	56	1.02	88

Table 4.3: Comparison of all algorithms in terms of cost and CPU time

To further investigate the effect of time windows, we consider the following three different configurations. In the first one we allow the vehicle to arrive at customers before, during or after the time window. In the second configuration we allow the vehicle to arrive only during or after the time window. In the third configuration, we allow the vehicle to arrive only before and during the time window. In all three configurations, the penalty for late and early arrivals, if applicable, is the same at 0.0028 £/second. The column labels are the same as in Table 4.4, and in addition, we show the total cost (TC) in £. The results are displayed in Table 4.5.

We can see from Table 4.5 that both the TC and ATVC increase significantly when we only allow late or early arrivals. Allowing early arrivals only yields a reduction in the FC and the DC, but this is done at the expense of the ATVC which increases drastically. The TC of the latter is larger than the TC obtained by allowing only late arrivals. We see therefore that allowing both early and late arrivals, at the same time, is the best option. If this is not possible, allowing only late arrivals is the second best option.

Instance	Penalty 0.0014			Penalty 0.0028			Penalty 0.0056		
	FC (£)	DC (£)	ATVC (mins)	FC (%)	DC (%)	ATVC (%)	FC (%)	DC (%)	ATVC (%)
20	23	83	14	5	3	−12	12	7	−19
50	63	226	37	4	2	−3	11	5	−6
75	98	348	58	4	2	−2	9	4	−3
100	129	461	83	4	2	−1	9	3	−2
150	199	704	119	3	1	−1	7	2	−1
200	266	951	173	4	1	0	8	2	−1
Average	130	462	81	4	2	−3	9	4	−5

Table 4.4: Comparison of costs under three penalty values

Instance	Early and late arrivals				Late arrivals only				Early arrivals only			
	FC (£)	DC (£)	ATVC (mins)	TC (£)	FC (%)	DC (%)	ATVC (%)	TC (%)	FC (%)	DC (%)	ATVC (%)	TC (%)
20	25	86	13	153	17	12	51	23	−3	−8	88	19
50	65	231	36	599	20	14	89	53	0	−12	132	62
75	102	355	56	1166	17	12	89	59	−2	−16	169	98
100	135	468	82	1975	21	17	96	72	−1	−15	165	111
150	205	709	119	3902	20	16	107	86	−3	−18	197	148
200	275	957	172	7012	20	17	105	90	−5	−21	203	164
Average	135	468	80	2468	19	15	90	64	−2	−15	159	100

Table 4.5: Comparison of costs for three time window types

4.4.1.5 Waiting times at customers

We now assess the effect on fuel consumption of allowing idle waiting times at customers, before or after service. We ran the experiments on all instances, and we provide the results, obtained by the DSM model, in Table 4.6, where the column labels are similar to those of Table 4.5. The results under column "No waiting times" displays the values of the column labels, whereas the results under column "With waiting times" are the percentage increase with respect to the values of "No waiting times" column.

We can see from Table 4.6 that allowing idle waiting times at customers reduces fuel consumption, with an average saving of 23% across all instances. The average time window violation per customer is also significantly decreased, by 18% on average. A slight average increase of 4% in driver cost can be observed, which means that the time to complete the route is longer. However, the total cost decreases by 10% on average.

Instance	No waiting times				With waiting times			
	FC (£)	DC (£)	ATVC (mins)	TC (£)	FC (%)	DC (%)	ATVC (%)	TC (%)
20	25	86	13	153	−24	3	−36	−12
50	65	231	36	599	−23	5	−20	−11
75	102	355	56	1166	−24	3	−18	−12
100	135	468	82	1975	−22	5	−13	−9
150	205	709	119	3902	−21	3	−11	−9
200	275	957	172	7012	−21	3	−8	−7
Average	135	468	80	2468	−23	4	−18	−10

Table 4.6: Comparison of costs with disallowed or allowed waiting time at customers

4.4.2 Maritime transportation

For maritime transportation, we use the fuel consumption model introduced by Fagerholt et al. (2010) which was developed using real-word data, and has been then frequently used in the literature. Equation (4.23) expresses fuel consumption in metric tons (mt) as function of speed and distance.

$$E_f(v, d) = d(0.0036v^2 + 0.1015v + 0.8848) \tag{4.23}$$

We test the solving methods on instances generated by He et al. (2017), available at http://www.menet.umn.edu/ qhe/. We solve 30 instances that have between 10, 100 and 1,000 ports, 10 instances are available for each size. Each instance was generated with distances uniformly drawn between 100 and 1,000 nautical miles, and the time windows for each port were set to 240 hours (10 days). A referential bunker price of 470$ is used according to https://shipandbunker.com. To calculate the penalty cost

incurred in case of time window violation, we use a linear penalty function in terms of duration of violation, as was done by Fagerholt (2001). For early arrivals a penalty cost equivalent to the operation cost of the vessel for the violation duration is incurred. For late arrivals, a penalty equivalent to three times the penalty for early arrivals is applied. Daily operating costs for a vessel is taken from (Grenier, 2013; Faury and Cariou, 2016) is considered to be 9,000$ per day. To the best of the authors' knowledge, there is no widely accepted penalty value since it depends heavily on ports and ship types. We will therefore run a cost analysis with different penalty values in Table 4.8.

4.4.2.1 Methods performance on maritime instances

In this section, we compare on maritime instances the methods that performed best on the road transportation instances. In Table 4.7 each line represents the instance solved, while the cost of solutions is reported in thousands of dollars (k$), and the solution time is in milliseconds (ms). The gap in cost between the DSM and FSG-Z solution values and those of the NLM is also reported in percentage of the optimal solution value. The NLM model found the optimum on all instances, and as in road transportation instances, it is the most time consuming. The FSG-Z model with 20 discretizations and an accuracy of 10 hours found the optimal solutions on 10 ports instances, while the DSM model had a gap of 0.2%. In instances with 100 ports, FSG-Z found solutions with less than 0.1%, whereas the DSM, which was slightly faster, had a gap of 0.6% on average, and a worst gap of 1% on instance 100_8. FSG-Z was the fastest on 1,000 ports instances, and found solutions with a gap not exceeding 0.1% in gap, while the DSM model was slower and found solutions with a gap of 0.9%. We also see from Table 4.7 that when the number of ports becomes larger, the solution time of the NLM increases drastically, followed by that of the DSM model, while the FSG-Z solution time increases in a linear fashion. Table 4.7 also shows that the NLM solution time is heavily dependent on the instance solved, whereas the DSM solution time is less instance-dependent, and the FSG-Z solution time is not instance-dependent.

4.4.2.2 Soft time windows vs hard time windows

In this section we study how the total cost (TC), average time window violation per customer (ATVC) in hours (h) and average speed (AS) in knots (kt) change with change in time window type and penalty values. The tests were run on five instances with 100 customers, and we report the average results in Table 4.8. We solve the same instances with three different time window lengths of three, five and 10 days. The first line shows the results obtained with the standard penalty costs. The second line shows the results obtained with consideration of hard time windows. In the third and fourth lines the

Instance	NLM		DSM			FSG-Z (20,10)		
	Cost (k$)	Time (ms)	Cost (k$)	Gap (%)	Time (ms)	Cost (k$)	Gap (%)	Time (ms)
10_1	449	513	450.277	0.2	0	449	0.0	0
10_2	399	37	399	0.2	10	399	0.0	0
10_3	374	34	375	0.2	10	374	0.0	10
10_4	403	47	404	0.2	10	403	0.0	10
10_5	260	38	261	0.2	0	260	0.0	10
10_6	388	38	389	0.2	0	388	0.0	10
10_7	345	55	345	0.2	20	345	0.0	10
10_8	382	58	383	0.2	10	382	0.0	10
10_9	384	50	384	0.2	0	384	0.0	0
10_10	317	42	318	0.2	10	317	0.0	10
Average	370.2	91.0	370.9	0.2	7.0	370.2	0.0	7.0
100_1	6117	186	6152	0.6	90	6119	0.0	110
100_2	6074	196	6108	0.6	90	6076	0.0	100
100_3	6074	229	6107	0.6	90	6076	0.0	100
100_4	6074	190	6107	0.5	120	6076	0.0	110
100_5	6296	186	6324	0.4	80	6298	0.0	110
100_6	6564	193	6586	0.3	100	6567	0.0	110
100_7	6081	349	6117	0.6	80	6085	0.1	110
100_8	5422	288	5479	1.0	100	5424	0.0	110
100_9	6164	187	6200	0.6	90	6166	0.0	100
100_10	5662	212	5707	0.8	80	5665	0.1	110
Average	6052.7	222.0	6088.8	0.6	92.0	6055.1	0.0	107.0
1000_1	74547	1829	75185	0.9	2110	74554	0.0	1320
1000_2	74114	17851	74748	0.9	2090	74125	0.0	1320
1000_3	72854	2527	73485	0.9	2010	72873	0.0	1320
1000_4	74416	15639	75052	0.9	1950	74427	0.0	1310
1000_5	72024	2563	72647	0.9	2090	72039	0.0	1320
1000_6	74675	8262	75319	0.9	2060	74685	0.0	1320
1000_7	74618	2392	75260	0.9	2050	74630	0.0	1320
1000_8	74860	2618	75499	0.9	2100	74870	0.0	1320
1000_9	75004	2524	75643	0.9	2040	75012	0.0	1310
1000_10	73710	14754	74343	0.9	2070	73715	0.0	1330
Average	74082.3	7095.9	74718.0	0.9	2057.0	74092.9	0.0	1319.0

Table 4.7: Methods performance on maritime instances

value of the penalty is divided by four and two, respectively. In the fifth line, the value of the penalty is multiplied by two.

We see from Table 4.8 that the ATVC depends more on the value of the penalty than on the length of the time window. Narrowing the time window from 10 days to three days increases the ATVC from 0.31 h to 0.54 h, while reducing the value of penalty cost by two increases the ATVC to 1.93 h. We conclude that the ATVC can be controlled to achieve a certain service quality level by a fine tuning of the penalty costs. We can also see that carefully tuning the value of penalty costs with soft time windows can reduce the cost without increasing the ATVC significantly. the penalty cost and time window type also have an impact on the average speed of the vessels: with smaller penalty values, it is optimal to travel at lower speeds, which reduces the amount of fuel consumed. Table 4.8 also shows how the change in time window length affects the total cost. In fact, considering hard time windows with a length of five days results in lower cost than three-day soft time windows. This shows that for logistics companies, negotiating larger hard time windows may yield more savings in the total costs than narrower soft time windows.

	10 days Tw			5 days Tw			3 days Tw		
	Cost (k$)	ATVC (h)	AS (kt)	Cost (k$)	ATVC (h)	AS (kt)	Cost (k$)	ATVC (h)	AS (kt)
Standard	6160	0.31	18.27	6887	0.57	18.94	7211	0.54	19.20
HTW	6195	0.00	18.47	6953	0.00	19.24	7277	0.00	19.55
1/4 Penalty	5986	6.16	17.88	6627	8.41	18.55	6937	9.32	18.86
1/2 Penalty	6079	1.93	18.17	6743	2.24	18.86	7065	2.42	19.17
2 Penalty	6160	0.16	18.34	6839	0.19	19.17	7171	0.19	19.42

Table 4.8: Cost and speed variability with time window length, type and penalty values

4.5 Conclusions

We have introduced, modeled and solved the Speed Optimization Problem with Soft Time Windows. We first provided a non-linear model, which was then linearized by discretizing the speed variables. We also proposed a way to model the problem as a time-space network by discretizing the possible arrival times at the customers, and solving the resulting problem as a shortest path problem. We considered two different techniques to discretize the possible arrival times at customers. An iterative method was developed to accelerate the time-space network algorithm.

We tested all algorithms on 120 generated instances with a number of customers varying between 20 and 200. Our results show that although the non-linear model found an optimal solution on all instances, very similar solutions were found up to three times

faster with a discretized speed model. The time-space network methods, including that of Fagerholt (2001), found good solutions on small instances, but both the discretized speed model and the non-linear model provided better solutions faster. The use of the iterative acceleration method provided solutions with optimality gaps varying between 1% and 3%, but could be found 10 and 30 times faster than with the direct solution of the non-linear model. We also showed that the penalty value has a more significant impact on fuel consumption than on driver cost, and can reduce the average time window violations on small-size instances. We showed that to be efficient, soft time windows should allow for both early and late arrivals at customers. Finally, we assessed the effect of allowing idle waiting times at the customers, and showed that it can decrease fuel consumption by almost 25%, and the average time window violations by almost 20%. The time to complete the routes slightly increases by 4% on average, but an average saving of 10% can be achieved on the total cost. We showed that the methods' performance remains similar in maritime and road transportation contexts. The FSG-Z solved instances with 1,000 ports almost six times faster than the NLM. We also showed how total cost, time window violations and average speed change with the penalty values and consideration of hard time windows. We pointed out that larger hard time windows may result in lower total costs than narrower soft time windows.

Chapter 5

Conclusions

5.1 Short summary

Throughout the realization of this thesis, minimizing the impact of freight transportation on the planet was at the heart of the thesis, and its author. Numerous studies have suggested that an optimal control of vehicle routes and speeds can yield minimized GHG emissions. In the second chapter of this thesis, we aimed at generalizing these results, and considered uncertainty in traffic conditions, which was a limitation in the applicability of previous studies. The third chapter provides decomposition algorithms to solve this problem for larger instances. In the fourth chapter, we contributed to the speed optimization problem literature by considering soft time windows and assessing their value for maritime and road transportation.

In this chapter we highlight the main contributions of the work carried out in this thesis. We first present the findings of the three main chapters in response to the research questions proposed in Chapter 1. We then list the research outputs of this study, and discuss its limitations. We finally conclude by proposing several directions for future research.

5.2 Main findings

In Chapter 2 we have proposed several two-stage stochastic programs to model and solve the Pollution-Routing Problem under stochastic traffic conditions. Innovative modeling techniques were used to linearize the original model. The models proposed were able to represent traffic speed uncertainty as discrete random variables. Two different recourse strategies were proposed, one discrete and optimal and one continuous but not always optimal. The main findings of Chapter 2 can be summarized as:

- Considering uncertainty permits reducing the total cost (including the cost of GHG emissions) by 7.48% on average for 10-node instances.
- Perfect knowledge of traffic conditions can further save only 1.27% of the total cost compared to the stochastic approach.
- Despite not being optimal theoretically, fixed continuous recourse provided better results than discretized recourse,

In Chapter 3, we adapted the Integer L-shaped method to the PRP under uncertain traffic conditions. We also proposed two variations of the algorithm that embed new cuts, lower-bounds and local branching. When tested on 20, 25 and 50-node instances all the methods provided better solution than the commercial solver CPLEX. The main findings of Chapter 3 are:

- Integer L-shaped based algorithms outperformed commercial solver CPLEX on all instances tested.
- The variations of the algorithm yielded similar results on average, but outperformed each other on individual instances.
- The cuts proposed for Lshaped_A significantly improved the lower bound compared to Lshaped_O.

In Chapter 4 we have considered the Speed Optimization Problem with Soft Time Windows. We provided a non-linear model that we linearized and proposed two algorithms based on time-space networks, which differs in the way possible arrival times at customers are discretized. We also proposed an acceleration method for the time-space network algorithms. We solved instances of the problem in road and maritime transportation context. The answers proposed by Chapter 4 are:

- The discretized speed model finds optimal solutions up to three times faster than the non-linear model.
- The accuracy and computational time of time-space network depend on the discretization technique.
- The acceleration method proposed proposed solutions up to 30 times faster than the non-linear model, with an optimality gap less than 3%.
- Allowing vehicle to wait idly at customers can reduce fuel consumption by up to 25% and yields a total cost saving of 10% on average.
- Wider hard time windows may yield better solution values than narrower soft time windows.

5.3 Research outputs

The present thesis has generated scientific papers and conference presentations. One paper has been published:

- Nasri, M I. Bektaş, T., Laporte, G., 2018. "Route and Speed Optimization for Autonomous Trucks", *Computers & Operations Research.*

One paper has received a "revise and resubmit" verdict and has been resubmitted :

- Nasri, M I. Bektaş, T., Laporte, G., 2018. "Speed Optimization under Soft Time Window Constraints", *Transportation Research Part E: Logistics and Transportation Review.*

Five conferences and presentations.

- Nasri, M I. Bektaş, T., Laporte, G. " The Pollution-Routing problem with stochastic travel times", Southampton Business School Student Conference, 2015, Southampton.
- Nasri, M I. Bektaş, T., Laporte, G. 2018. " The Pollution-Routing problem with stochastic travel times", Optimization Days, 2015, Montreal.
- Nasri, M I. Bektaş, T., Laporte, G. " The Pollution-Routing problem with stochastic travel times", Vehicle Routing and Logistics Conference (VeRoLog), 2015, Nantes.
- Nasri, M I. Bektaş, T., Laporte, G. " Integer L-shaped Based Algorithms for the Pollution-Routing Problem under Traffic Uncertainty", Vehicle Routing and Logistics Conference (VeRoLog), 2016, Amsterdam.
- Nasri, M I. Bektaş, T., Laporte, G. " Route and Speed Optimization for Autonomous Trucks", Southampton Business School Student Conference, 2018, Southampton.

5.4 Research limitations and future research directions

Although the thesis responded to the research questions, it has done so with unavoidable limitations. These limitations are due to some necessary assumptions, and some simplifications made necessary by the time limit to accomplish this thesis. Below are some limitations of the study with future research directions for each of the three main chapters.

Chapter 2

- We represented the traffic uncertainty by discrete random variables. This is a sound technique, that has previously been used successfully in the literature, however only three scenarios were considered due to the complexity of the problem. Considering more traffic condition scenarios can prove more realistic.
- We represented the problem on a network where customers are connected by arcs. This approach can be used to assess the advantages of taking uncertainty into account, however it can underestimate the fuel consumption costs due to speed variability. Therefore, considering a more detailed road network that represents an actual road network would provide more accuracy.
- Traffic conditions is a dynamic variable that changes with time. Considering the problem with time-dependent traffic speed can therefore be more realistic.

- Due to the complexity of the problem, developing a metaheuristic to solve large instances can prove to be efficient.

Chapter 3

- We have developed in this chapter new cuts and lower-bounds for the problem which help the algorithm to provide good solutions on some instances, however stronger cuts able to eliminate a larger number of solutions from the set of feasible solutions would be better.
- All the algorithms provided outperformed CPLEX on all the instances tested, however they were not able to provide proof of optimality.

Chapter 4

- We have explored in this chapter the speed optimization problem and included soft time windows, however in practice choosing the optimal speed may not always be possible, therefore traffic conditions uncertainty should be taken into account.
- The computational results fulfilled the aim of assessing which algorithm performed best, but including a practical problem instance could have been beneficial.
- The potential of integrating the algorithms developed in this chapter within a metaheuristic or a matheuristic to solve a larger route and speed optimization problem would be within investigation.

Bibliography

Agostinho Agra, Marielle Christiansen, Rosa Figueiredo, Lars Magnus Hvattum, Michael Poss, and Cristina Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

Henrik Andersson, Kjetil Fagerholt, and Kirsti Hobbesland. Integrated maritime fleet deployment and speed optimization: Case study from roro shipping. *Computers & Operations Research*, 55:233–240, 2015.

Gustavo Angulo, Shabbir Ahmed, and Santanu S Dey. Improving the integer L-shaped method. *INFORMS Journal on Computing*, 28(3):483–499, 2016.

Matthew Barth and Kanok Boriboonsomsin. Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board*, (2058):163–171, 2008.

Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

Tolga Bektaş, Emrah Demir, and Gilbert Laporte. Green vehicle routing. In Harilaos N. Psaraftis, editor, *Green Transportation Logistics: The Quest for Win-win Solutions*, pages 243–265. Springer, New York, 2016.

John R Birge. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Mathematical Programming*, 24(1):314–325, 1982.

John R Birge and François V Louveaux. *Introduction to Stochastic Programming*. Springer Science & Business Media, New York, 2011.

Abraham Charnes and William W Cooper. Programming with linear fractional functionals. *Naval Research logistics quarterly*, 9(3-4):181–186, 1962.

Boris V Cherkassky, Andrew V Goldberg, and Tomasz Radzik. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73(2):129–174, 1996.

James J Corbett, Haifeng Wang, and James J Winebrake. The effectiveness and costs of speed reductions on emissions from international shipping. *Transportation Research Part D: Transport and Environment*, 14(8):593–598, 2009.

Bob Costello and Rod Suarez. Truck driver shortage analysis 2015. Technical report, The American Trucking Associations, Arlington, VA, October 2015.

Said Dabia, Emrah Demir, and Tom Van Woensel. An exact approach for a variant of the pollution-routing problem. *Transportation Science*, 51(2):607–628, 2016.

George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 223(2):346–359, 2012.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478, 2014a.

Emrah Demir, Tolga Bektaş, and Gilbert Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014b.

Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.

Moshe Dror, Gilbert Laporte, and François V Louveaux. Vehicle routing with stochastic demands and restricted failures. *Zeitschrift für Operations Research*, 37(3):273–283, 1993.

Moshe Dror, Gilbert Laporte, and Pierre Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, 23(3):166–176, 1989.

Richard W Eglese and Tolga Bektaş. Green vehicle routing. In Paolo Toth and Daniele Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, volume 18, pages 437–458. MOS-SIAM Series on Optimization, Philadelphia, 2014.

Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W Thomas. Data-driven approaches for emissions-minimized paths in urban areas. *Computers & Operations Research*, 67:34–47, 2016a.

Jan Fabian Ehmke, Ann Melissa Campbell, and Barrett W Thomas. Vehicle routing to minimize time-dependent emissions in urban areas. *European Journal of Operational Research*, 251(2):478–494, 2016b.

Jan Fabian Ehmke, Ann Melissa Campbell, and Timothy L Urban. Ensuring service levels in routing problems with time windows and stochastic travel times. *European Journal of Operational Research*, 240(2):539–550, 2015.

Reza Eshtehadi, Mohammad Fathian, and Emrah Demir. Robust solutions to the pollution-routing problem with demand and travel time uncertainty. *Transportation Research Part D: Transport and Environment*, 51:351–363, 2017.

Kjetil Fagerholt. Ship scheduling with soft time windows: An optimisation based approach. *European Journal of Operational Research*, 131(3):559–571, 2001.

Kjetil Fagerholt, Nora T Gausel, Jørgen G Rakke, and Harilaos N Psaraftis. Maritime routing and speed optimization with emission control areas. *Transportation Research Part C: Emerging Technologies*, 52:57–73, 2015.

Kjetil Fagerholt, Gilbert Laporte, and Inge Norstad. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society*, 61(3): 523–529, 2010.

Yunfei Fang, Feng Chu, Saïd Mammar, and Ada Che. An optimal algorithm for automated truck freight transportation via lane reservation strategy. *Transportation Research Part C: Emerging Technologies*, 26:170–183, 2013.

Olivier Faury and Pierre Cariou. The northern sea route competitiveness for oil tankers. *Transportation Research Part A: Policy and Practice*, 94:461–469, 2016.

Matteo Fischetti and Andrea Lodi. Local branching. *Mathematical programming*, 98 (1-3):23–47, 2003.

Anna Franceschetti, Dorothée Honhon, Tom Van Woensel, Tolga Bektaş, and Gilbert Laporte. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.

Ricardo Fukasawa, Qie He, Fernando Santos, and Yongjia Song. A joint routing and speed optimization problem. *arXiv preprint arXiv:1602.08508*, 2016a.

Ricardo Fukasawa, Qie He, and Yongjia Song. A disjunctive convex programming approach to the pollution-routing problem. *Transportation Research Part B: Methodological*, 94:61–79, 2016b.

Michel Gendreau, Ola Jabali, and Walter Rei. Stochastic vehicle-routing problems. In Paolo Toth and Daniele Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, pages 213–239. MOS-SIAM Series on Optimization, Philadelphia, 2014.

Michel Gendreau, Ola Jabali, and Walter Rei. Future research directions in stochastic vehicle routing. *Transportation Science*, 50(4):1163–1173, 2016.

Michel Gendreau, Gilbert Laporte, and René Séguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29 (2):143–155, 1995.

Emilia Grass and Kathrin Fischer. Two-stage stochastic programming in disaster management: A literature survey. *Surveys in Operations Research and Management Science*, 21(2):85–100, 2016.

Richard Grenier. Ship operating costs current and future trends moore stephens. http://www.propellerclub.gr/files/OpCostPresentation_MrRichardGreiner.pdf, 2013. Accessed on 09.07.2018.

Ge Guo, Gabriel Hackebeil, Sarah M Ryan, Jean-Paul Watson, and David L Woodruff. Integration of progressive hedging and dual decomposition in stochastic integer programs. *Operations Research Letters*, 43(3):311–316, 2015.

Qie He, Xiaochen Zhang, and Kameng Nip. Speed optimization over a path with heterogeneous arc costs. *Transportation Research Part B: Methodological*, 104:198–214, 2017.

Juan C Herrera, Daniel B Work, Ryan Herring, Xuegang Jeff Ban, Quinn Jacobson, and Alexandre M Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, 2010.

Aude Hofleitner, Ryan Herring, and Alexandre Bayen. Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach. In *91st Transportation Research Board Annual Meeting*, number 12-0798, 2012.

Yixiao Huang, Lei Zhao, Tom Van Woensel, and Jean-Philippe Gross. Time-dependent vehicle routing problem with path flexibility. *Transportation Research Part B: Methodological*, 95:169–195, 2017.

Lars Magnus Hvattum, Inge Norstad, Kjetil Fagerholt, and Gilbert Laporte. Analysis of an exact algorithm for the vessel speed optimization problem. *Networks*, 62(2): 132–135, 2013.

Taesung Hwang and Yanfeng Ouyang. Urban freight truck routing under stochastic congestion and emission considerations. *Sustainability*, 7(6):6610–6625, 2015.

Kibaek Kim and Sanjay Mehrotra. A two-stage stochastic integer programming approach to integrated staffing and scheduling with application to nurse management. *Operations Research*, 63(6):1431–1451, 2015.

Çağrı Koç, Tolga Bektaş, Ola Jabali, and Gilbert Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

Raphael Kramer, Anand Subramanian, Thibaut Vidal, and F Cabral Lucídio dos Anjos. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015.

Gilbert Laporte and François V Louveaux. Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies. In Jean J Gabszewicz, Jean-François Richard, and Laurence A Wolsey, editors, *Economic Decision-Making: Games, Econometrics, and Optimisation : Contributions in Honour of Jacques H. Drèze*, pages 443–455. North-Holland, Amsterdam, 1990.

Gilbert Laporte and François V Louveaux. The integer l-shaped method for stochastic integer programs with complete recourse. *Operations research letters*, 13(3):133–142, 1993.

Gilbert Laporte, François V Louveaux, and Hélène Mercure. The vehicle routing problem with stochastic travel times. *Transportation Science*, 26(3):161–170, 1992.

Chungmok Lee, Kyungsik Lee, and Sungsoo Park. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9):1294–1306, 2012.

Jos Lelieveld, JS Evans, M Fnais, Despina Giannadaki, and A Pozzer. The contribution of outdoor air pollution sources to premature mortality on a global scale. *Nature*, 525 (7569):367–371, 2015.

Canhong Lin, King Lun Choy, George TS Ho, SH Chung, and HY Lam. Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications*, 41(4):1118–1138, 2014.

Will Maden, Richard W Eglese, and Dan Black. Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society*, 61(3): 515–522, 2010.

National Careers Service. Large goods vehicle driver. https://nationalcareersservice.direct.gov.uk/job-profiles/large-goods-vehicle-driver, 2017. Accessed on 30.11.2017.

Inge Norstad, Kjetil Fagerholt, and Gilbert Laporte. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies*, 19 (5):853–865, 2011.

International Maritime Organization. Prevention of air pollution from ships. http://www.imo.org/en/OurWork/Environment/PollutionPrevention/AirPollution/Pages/Default.aspx/, 2018. Accessed on 04.06.2018.

Éric Prescott-Gagnon, Guy Desaulniers, and Louis-Martin Rousseau. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks*, 54(4):190–204, 2009.

Harilaos N. Psaraftis. *Green Transportation Logistics: The Quest for Win-Win Solutions*. Springer, New York, 2016.

Jiani Qian and Richard W Eglese. Finding least fuel emission paths in a network with time-varying speeds. *Networks*, 63(1):96–106, 2014.

Jiani Qian and Richard W Eglese. Fuel emissions optimization in vehicle routing problems with time-varying speeds. *European Journal of Operational Research*, 248(3): 840–848, 2016.

Hesham Ahmed Rakha, Ihab El-Shawarby, Mazen Arafeh, and Francois Dion. Estimating path travel-time reliability. In *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pages 236–241. IEEE, 2006.

Majid Salavati-Khoshghalb, Michel Gendreau, Ola Jabali, and Rei Walter. An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. Technical report, Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT), Montreal, 2017.

George Scora and Matthew Barth. Comprehensive modal emissions model (CMEM), version 3.01. User's guide. Technical report, Centre for Environmental Research and Technology. University of California, Riverside, 2006.

Nicola Secomandi. A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research*, 49(5):796–802, 2001.

Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. Technical Report DOT HS 812 115, National Highway Traffic Safety Administration, Washington, DC, February 2015.

Michael Spielmann and Roland Scholz. Life cycle inventories of transport services: Background data for freight transport. *The International Journal of Life Cycle Assessment*, 10(1):85–94, 2005.

Anand Subramanian, Lúcia Maria de Assumpção Drummond, Cristiana Bentes, Luiz Satoru Ochi, and Ricardo Farias. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11):1899–1911, 2010.

Nazanin Tajik, Reza Tavakkoli-Moghaddam, Behnam Vahdani, and Seyed Meysam Mousavi. A robust optimization approach for pollution routing problem with pickup and delivery under uncertainty. *Journal of Manufacturing Systems*, 33(2):277–286, 2014.

Duygu Taş, Nico Dellaert, Tom Van Woensel, and Ton De Kok. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1):214–224, 2013.

Duygu Taş, Michel Gendreau, Nico Dellaert, Tom Van Woensel, and Ton De Kok. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, 236(3):789–799, 2014.

The Guardian. Convoy of self-driving trucks completes first european cross-border trip. https://www.theguardian.com/technology/2016/apr/07/convoy-self-driving-trucks-completes-first-european-cross-border-trip, 2016. Accessed on 30.11.2017.

Paolo Toth and Daniele Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, Philadelphia, 2014.

S Tsugawa. Energy its: What we learned and what we should learn. onlinepubs. trb. org. http://onlinepubs.trb.org/onlinepubs/conferences/2012/Automation/presentations/Tsugawa.pdf, 2012. Accessed on 30.11.2017.

U.S. Environmental Protection Agency. Inventory of U.S. greenhouse gas emissions and sinks : 1990-2014. Technical report, April, 2016. Available from: https://www3.epa.gov/climatechange/Downloads/ghgemissions/US-GHG-Inventory-2016-Main-Text.pdf.

Richard M Van Slyke and Roger Wets. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4):638–663, 1969.

Benoit Vanholme, Dominique Gruyer, Benoit Lusetti, Sébastien Glaser, and Saïd Mammar. Highly automated driving on highways based on legal safety. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):333–347, 2013.

Shuaian Wang and Qiang Meng. Sailing speed optimization for container ships in a liner shipping network. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):701–714, 2012.

Wired.com. Uber's self-driving truck makes its first delivery: 50.000 beers. https://www.wired.com/2016/10/ubers-self-driving-truck-makes-first-delivery-50000-beers/, 2016. Accessed on 30.11.2017.

Peng Wu, Feng Chu, Ada Che, and Yunfei Fang. An efficient two-phase exact algorithm for the automated truck freight transportation problem. *Computers & Industrial Engineering*, 110:59–66, 2017.

Wen-Huei Yang, Kamlesh Mathur, and Ronald H Ballou. Stochastic vehicle routing problem with restocking. *Transportation Science*, 34(1):99–112, 2000.