MASTERARBEIT

Titel der Masterarbeit

Finding the Trade-off between Gas Emissions and Disturbance in an Urban Context

verfasst von

Jasmin Grabenschweiger, BSc

angestrebter akademischer Grad

Master of Science (MSc)

Wien, 2016

Studienkennzahl lt. Studienblatt:	A 066 920
Studienrichtung lt. Studienblatt:	Masterstudium Quantitative Economics, Management and Finance UG2002
Betreut von:	Univ.-Prof. Mag. Dr. Karl Franz Dörner
Mitbetreut von:	Dr. Fabien Tricoire

Abstract

We present three different modelling approaches for investigating the trade-off between gas emissions and disturbance in the context of delivering customers in urban areas. The amount of emitted polluting gases is assumed to be dependent on travelled distance and carried load weight. Disturbance refers in the course of this work to how much residents are affected by traffic resulting from freight transportation. We will relate this rather intangible value to population density when we develop a concept for generating disturbance data.

In the first model only shortest distance paths between locations are considered and the task is to solve a bi-objective Travelling Salesman Problem on a simple graph. Model 2 includes in addition to the mandatory nodes so called optional nodes that must not, but can be visited, if beneficial in terms of disturbance. The alternative paths in the third approach are modelled though a graph with multiple arcs, what classifies it as a bi-objective Travelling Salesman Problem on a multi-graph.

We will show that Model 2 yields the same Pareto front as Model 3, when we transform the optional nodes of Model 2 into multiple arcs for Model 3 by solving an all-pairs bi-objective Shortest Path Problem with Martins' algorithm.

Computational experiments on randomly generated instances are conducted in order to demonstrate solution characteristics of the three approaches and to compare their performance with respect to solution quality and CPU effort. The ε-constraint method will be implemented to solve the bi-objective optimisation problems to optimality. Moreover, the computation study involves a real world case study that aims at comparing the solutions of our problem to tours carried out in reality.

Acknowledgement

We will consider social issues in the following primarily as a scientific part of the work. However, here I want to address another important social aspect of this work, namely the people who contributed in different ways to its successful completion.

First of all I would like to express my gratitude to my supervisor Univ.-Prof. Dr. Karl F. Dörner, who always guided me in the right direction and introduced me to the topic of this thesis. I appreciate he showed great interest in the development of my work and always took the time for meeting and discussing, what established a motivating environment and a pleasant working relationship.

My special thanks also go to Dr. Fabien Tricoire, who was not only my co-supervisor but rather became kind of a mentor for me during the last few months. I could benefit a lot from his broad knowledge, particularly because his constructive ideas added a great value to my work. Besides, I'm very glad that the numerous hours of fruitful collaboration were not only devoted to hard work but also to funny moments and encouraging talks.

Furthermore I want to thank o. Univ.-Prof. Dipl.-Ing. Dr. Richard Hartl and Dipl.-Wirtsch.-Ing. Prof. Dr. Heinrich Kuhn who laid the foundation for my enthusiasm for logistic problems.

Last but not least I'm very grateful for the support I got from my family and friends throughout my studies - especially when working on this thesis. They were always willing to listen, believed in my success and offered me sometimes the needed off-topic diversion. However, they also gave me some useful external feedback and rather practical views on my research topic.

Contents

List of abbreviations iii

List of figures iv

List of tables vi

1 Introduction 1
- 1.1 About the subject of this thesis 1
- 1.2 About the Green City Hubs project 2
- 1.3 Motivation 3

2 Literature review 5

3 Bi-objective optimisation 8
- 3.1 General case: Multi-objective optimisation 8
- 3.2 Exact methods for solving multi-objective optimisation problems 10
  - 3.2.1 Weighted-sum method 11
  - 3.2.2 ε-constraint method 12

4 Mathematical problem formulation 16
- 4.1 Features valid for all three approaches 16
  - 4.1.1 Data 16
  - 4.1.2 Graphical representation 17
  - 4.1.3 Decision variables 18
  - 4.1.4 Objective functions 19
  - 4.1.5 Constraints 20
  - 4.1.6 Mixed Integer Programming 21
- 4.2 Model 1 21
  - 4.2.1 Classification: Bi-objective Travelling Salesman Problem 21
  - 4.2.2 Bi-objective MIP formulation 22
- 4.3 Model 2 24
  - 4.3.1 Why optional nodes? 24
  - 4.3.2 Classification: Bi-objective Shortest Path Problem with mandatory nodes 25
  - 4.3.3 Bi-objective MIP formulation - first attempt 27
  - 4.3.4 Model 2A: Allow at most one visit for optional nodes 30

4.3.5 Model 2B: Duplication of optional nodes 31
4.3.6 Model 2C: Introduction of a position index 34
4.4 Model 3 37
4.4.1 Why multiple arcs? 37
4.4.2 Classification: Bi-objective Travelling Salesman Problem on a multigraph 37
4.4.3 Bi-objective MIP formulation 38
4.5 Relation between the three models 40
4.5.1 Model 1 as a special case of Model 3 41
4.5.2 Transformation of Model 2 into Model 3 41

5 Computational study **51**
5.1 Experimental results on randomly generated instances 51
5.1.1 Test instances 52
5.1.2 Compared performance of Model 2B and Model 2C 55
5.1.3 Solution characteristics of the three models illustrated by means of an example instance 57
5.1.4 Compared performance of the three models 65
5.1.5 Limits on the optimality of the ε-constraint method 68
5.1.6 Compared performance of the ε-constraint method once with total emission as f_1 and once with total disturbance as f_1 70
5.2 Real world case study 73
5.2.1 Test instances 73
5.2.2 Compared real tours with solutions of Model 1 76
5.2.3 Taking 'unused customers' as optional nodes 79

6 Conclusion and relevance of the work **82**

7 Perspectives **84**

Appendix **86**

References **89**

Supplements **94**
Deutsche Zusammenfassung 94
Curriculum Vitae 94

Abbreviation index

MIP	Mixed Integer Programme
GCH	Green City Hubs
PF	Pareto front
SPP	Shortest Path Problem
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem

List of Figures

1.1 The Green City Hubs project . . . 3
1.2 The problem of this work . . . 3

3.1 The concept of Pareto optimality . . . 10
3.2 ε-constraint method . . . 14

4.1 Example graphs for the three modelling approaches . . . 18
4.2 Going over optional node reduces disturbance . . . 25
4.3 Multiple visits in optional node cause load variable problems . . . 30
4.4 Including an optional node between each two different mandatory nodes . . 32
4.5 Multiple arcs are assumed to be Pareto optimal . . . 38
4.6 Relation between Model 2 and bi-objective SPP . . . 42
4.7 Illustrative example for Martins' algorithm . . . 47
4.8 Martins' algorithm yields three Pareto optimal paths . . . 49

5.1 Densitygrid . . . 53
5.2 Calculating disturbance between two points . . . 54
5.3 Triangle inequality does not hold for disturbance data . . . 55
5.4 Relevant locations for Model 1 and Model 3 . . . 57
5.5 Relevant locations for Model 2 . . . 58
5.6 Pareto front of Model 1 . . . 59
5.7 Distance minimising solution . . . 60
5.8 Graphical representation of Pareto optimal solutions of Model 1 . . . 60
5.9 Pareto front of Model 1 and 2 . . . 62
5.10 Graphical representation of Pareto optimal solutions of Model 2 . . . 63
5.11 Development of CPU time along Pareto front . . . 72
5.12 Location of hub and customers . . . 74
5.13 Population density of Vienna's subdistricts . . . 75
5.14 Real tour . . . 77
5.15 Pareto front of Model 1 and objective value point of real tour . . . 78
5.16 Pareto optimal solutions with lower emission but higher disturbance than real route . . . 79
5.17 Pareto optimal solutions that dominate the real route . . . 79
5.18 Pareto optimal solutions with lower disturbance but higher emission than real route . . . 80

5.19 Pareto front of Model 3 in comparison to the Pareto front of Model 1 . . . 81

List of Tables

5.1 Compared performance of Model 2B and Model 2C 56
5.2 Output Model 1 58
5.3 Output Model 2 61
5.4 Output Model 3 64
5.5 Compared CPU time 66
5.6 Compared Pareto front 67
5.7 Limits on the optimality of the ε-constraint method 69
5.8 Compared performance of two versions of the ε-constraint method 71

1 Introduction

1.1 About the subject of this thesis

This thesis deals with the following problem: A set of fixed customers located in an urban area has to be delivered from a fixed hub. Only one period is considered in the present framework. Furthermore a single vehicle is assigned to the hub, whose capacity restriction is assumed to be not effective. This implies that all given customers can be served within one, single tour. The question of interest is then in which order to visit the customers such that gas emissions and disturbance to urban neighbourhood are minimised. The amount of emitted gases is measured over travelled distance and transported load. Disturbance is related to how much residents are affected by noise, risk, vibration, etc., resulting from carrying out the delivery tours. We identify population density to be a proper indicator for disturbance. In general less disturbing ways through sparsely populated areas tend to be longer in distance and will thus yield higher emissions. So the two objectives of emission and disturbance minimisation are conflicting, what calls for a bi-objective formulation of the problem.

Three different approaches for modelling the underlying problem scenario will be presented. Each of them belongs to a different problem class of combinatorial optimisation. The first one is a bi-objective Travelling Salesman Problem (TSP) on a simple graph, which means only the minimal distance path between two locations is given and all given locations must be visited in a closed, single tour. In the second approach there are in addition to the hub and customers (=the mandatory nodes) so called optional nodes available. Including one or more optional nodes between two mandatory nodes may lead to an disturbance-reducing alternative path. Since not all nodes but only the mandatory one must be visited this approach can be classified as a bi-objective Shortest Path Problem (SPP) with mandatory nodes. Two different, valid formulations for the optional-node approach will be presented, where the first is based on the concept of duplicating nodes (only the optional nodes) and the latter makes use of an arc position index to guarantee that load flow is in accordance with node sequence. Model 3 also incorporates alternative paths, but now in form of pre-computed multiple arcs between each two mandatory nodes. Thus the approach falls into the class of bi-objective TSP on a multigraph.

Although Mode 2 and Model 3 differ heavily in their structure and mathematical formulation, we will show that they yield the same result when we transform the optional nodes of Model 2 into multiple arcs for Model 3. This is done by solving an all-pairs bi-objective SPP with Martins' algorithm.

While Model 1 constitutes a rather simplistic and - in terms of disturbance minimisation - restrictive approach, the other two models provide more room for optimisation. So differences in the solutions can be expected. But more sophisticated approaches tend to be also more complex and may consequently be harder to solve.
Hence, computational experiments are carried out in order to compare the performance of the three modelling approaches in terms of solution quality and CPU effort. The models will be tested on randomly generated instances as well as on real world data related to the 'Green City Hubs' project (which will be shortly introduced in the next section). In the course of generating instances also a method for generating disturbance data will be presented.

The work will be concluded by some thoughts about the practical and theoretical relevance of the studied problem and about possible future research directions.

1.2 About the Green City Hubs project

The subject of this work was derived from a project called 'Green City Hubs' (for more information see [38] and [39]). The purpose of the project is to design a sustainable logistics system for inner-city freight transportation, where sustainability comes from various aspects: One is to establish inner-city distribution centres - the so called city hubs - from where customer deliveries are carried out. Compared to delivering customers from one big hub outside the city this allows for a more efficient planning of transportation and bundling of requests what reduces system-wide delivery distances and consequently also gas emission. A further move towards reducing emissions is made by using alternatively fuelled vehicles, that are loaded and recharged at the city hubs. Beside taking long-term decisions on hub locations and vehicle purchases there is the operational task of delivering customers. Here sustainability is tackled by optimising the routes with respect to gas emission and disturbance to urban neighbourhood. Figure 1.1 depicts the just described problem.

Taking all decisions together leads to a complex multi-objective optimisation problem that comprises different time horizons and that accounts not only for internal monetary costs but also for environmental and social external costs.

To reduce complexity we consider in the course of this work only the sub-problem of

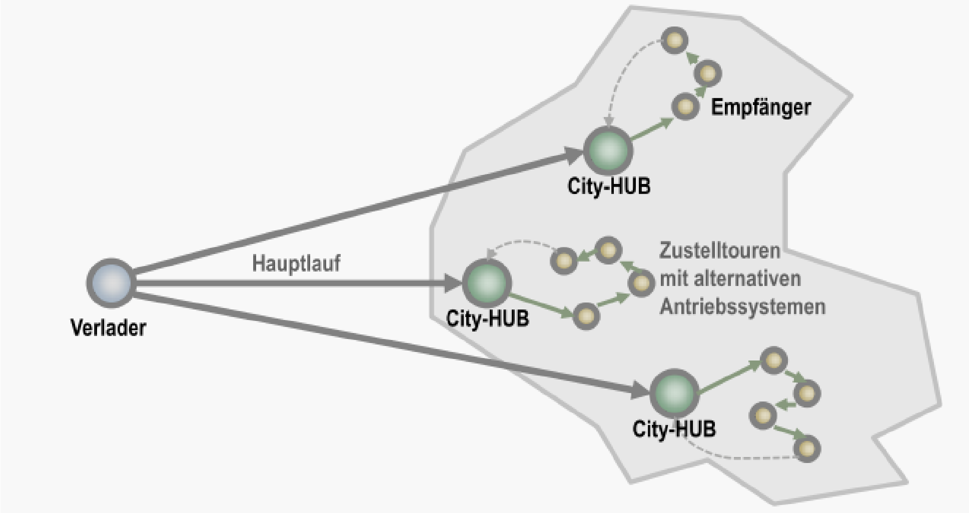


Figure 1.1: The Green City Hubs project (B. Pierkarz, i-LOG)

delivering a fixed set of customers from a fixed hub with a fixed vehicle. So the decisions about hub locations and vehicles are not part of our optimisation problem. In Figure 1.2 the highlighted part illustrates a possible sub-problem of the problem in Figure 1.1. Due to the direct relationship between the problem of this work and the problem of the Green City Hubs project the results we obtain from our study can be interpreted and implemented also in the context of the project framework. However the content of this work is not bounded to it, but serves a much broader field of application.

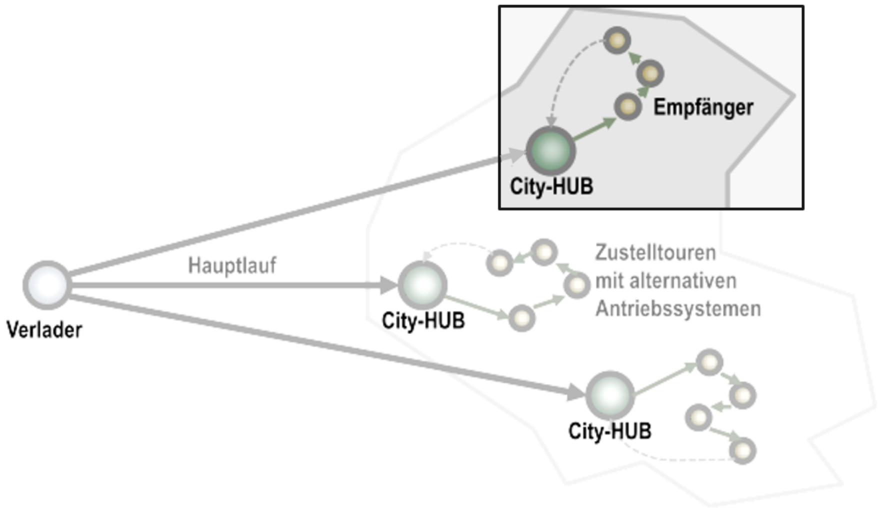


Figure 1.2: The problem of this work

1.3 Motivation

The two types of cost considered in our optimisation problem relate to hot issues in nowadays society and research: environmental pollution and urbanization.

There are many contributors to environmental pollution, but road freight transportation is definitely one of the biggest. While long-haul carriage can be done also by cargo aircraft, ships or freight trains, there is no proper alternative for last-mile deliveries, occurring in particular in urban areas.[1]. Thus air pollution emerging from inner-city freight transportation is extremely critical. In 2011 the European Commission has announced that overall CO_2-pollution coming from transportation should be reduced by 2050 to around 20 % below the 2008 level[7]. For cities they impose an even sharper target by claiming an essentially CO_2-free logistics system by 2030. These goals can not be achieved overnight. Hence it would be wise to implement emission-reducing policies gradually, where optimising customer delivery tours with respect to emissions can contribute much.

Considering an urban context for our transportation problem is not only motivated by environmental concerns but also by the increasing urbanization of world's population. According to the World Urbanization Prospects of 2014 [36] more than 80 % of the European population is expected to live in urban areas by 2050 while this percentage was around 73 % in 2014.
Also the trend towards online-shopping and home delivery shows a greater effect in cities, what supports further the importance of taking into account the particularities of inner-city freight transportation.
One of these particularities is that there are people living next to the streets who are affected by the traffic resulting from inner-city freight transportation. Beside emitted harmful gases there are for example also noise, risk or vibration that disturb the residents. In the present work this problem is addressed by considering external costs in form of urban disturbance when planning customer delivery tours.

So we see, there are many current issues and potential future developments that motivate studying the particularities and difficulties of urban freight transportation.

---

[1] Actually, Amazon recently talked about using drones for delivering their customers[40]. Anyway, heavy and large freight has to be still transported on road.

2 Literature review

The present work contributes essentially to the following up-to-date research directions in Logistics: Green Logistics and City Logistics. Green Logistics means that beside conventional internal economic costs also external costs for harmful environmental, ecological and social effects of transportation are included when planning logistics systems[25]. In the field of City Logistics the difficulties and characteristics of transportation in urban areas are taken into account with the goal of mobility, sustainability and liveability[34]. City Logistics can be thus viewed as a sub-field of Green Logistics since environmental, ecological and social externalities are thereby also of main interest.

Concerning the problem class of this work a bi-objective TSP, once on a simple graph and once on a multigraph, and a SPP with mandatory nodes are considered. They will be introduced and described in the respective sections, where also related literature will be analysed shortly.
The literature review in this chapter focuses thus on Green Logistics and City Logistics. Since the Vehicle Routing Problem (VRP) is a generalisation of the TSP, allowing more than one tour and more than one vehicle, the literature referring to Green VRP comprises also the TSP.

Logistics in general is the science of planning, organizing and managing activities that provide goods or services. This includes amongst others the processes of transportation, warehousing, order picking and packing. However this work addresses only the transportation part - in particular goods transportation.

An extensive literature review on Green Vehicle Routing Problems can be found in Eglese and Bektaş (2014) [12] and Lin et al. (2014) [25]. Green Vehicle Routing splits up in several further areas, but for us primarily the Pollution Routing Problem (PRP) matters, where a reduction greenhouse gas emissions emerging from transportation is targeted. Although the purpose of PRP is of great value for society, business and environment related literature is not that exhaustive, since it is still a rather new research direction.
The first relevant traces in this context can be found in Pronello and André (2000) [31]. They analyse the problem of combing emission and transportation models, since both are usually designed for different purposes that do not necessarily cover the interdependencies between them. As a consequence they suggest that several factors as for example traffic conditions, travel distance, travel time, vehicle speed and load have to be integrated into

a sophisticated and reliable pollution routing problem.

The PhD dissertation of Palmer (2007) [30] explores the relationship between vehicle speed and the amount of emitted gases along a transportation route. However speed is not up to optimisation in his model but taken as fixed input parameter. His findings show that CO_2 emission can be reduced by about 5 % in comparison to time-minimising routes. The effect of vehicle load is not considered explicitly in the approach of Palmer's dissertation.

Kara et al.(2007)[22] bring the load dependant part of emission into play, by introducing an objective function that identifies the cost of traversing an arc as a multiple of arc-length and the corresponding total carried load on this arc. The authors compare the solutions of this load-based approach to those when just distance is to be minimised.

Suzuki (2011)[32] tackles emission minimisation by studying two models that aim at fuel consumption minimisation, where one measures fuel consumption over vehicle speed and the measures it other over truck load. Regarding the load-based model it turned out that fuel consumption can be reduced significantly when heavy goods are delivered rather at the beginning of a tour.

Maden et al. (2010) [26] study a vehicle routing and scheduling problem where total travel time is to be minimised under the assumption of varying travel times - depending on the time of the day and influencing vehicle speed and in turn also gas emissions. They carried out a case study based on this approach and showed that it can lead to around 7 % less CO_2 emissions.

Bektaş and Laporte (2011)[3] present a PRP with a broad objective function that accounts beside travel distance also for the external cost of emission and operational costs of drivers and fuel consumption. This cost function is assumed to depend on load, speed, acceleration and other parameters. They model the framework once with time windows and once without time windows. Computational experiments are conducted to analyse the trade-off between and impact of the various parameters.

Demir et al. (2014) [9] formulate a bi-objective PRP where the two conflicting objectives of fuel consumption minimisation and driving time minimisation are considered. Three exact solution methods (the weighted-sum method, the weighted-sum method with normalisation and the ε-constraint method) and one heuristic approach (a new Hybrid Method) are tested and compared with each other.

Faulin et al. (2011) [14] develop an algorithm for solving a capacitated VRP that considers traditional transportation costs and external environmental costs, but now not only related to pollutant emissions but also to noise, congestion and wear and tear on infrastructure. For these externalities they use cost estimations derived from surveys about road transportation in rural areas (where nature is rather sensitive to traffic). A case study reveals that taking into account the costs for pollution and noise raises economic costs by around 28 %. Thus, the authors suggest that public policies are required in order

to make environmental friendly transportation strategies attractive for companies.
So the work of Faulin et al. (2011) actually addresses the aspect of disturbance by considering the noise impact of road transportation. However, their problem and results apply rather to a rural framework due to the rural evaluation basis of the external cost estimations. Moreover all these externalities are considered to be environmental and not conflicting, whereas the idea of the present work is to treat minimisation of gas emissions and minimisation of noise and congestion impact as two conflicting objectives, where the first one is assumed to be environmental and the second is assumed to be social.

Air pollution and other environmental issues are also critical in urban areas, due to the high traffic volume resulting from providing goods and services to the residents. Basically most of the work on PRPs applies also to the urban framework. Especially the time-dependent models are of interest since urban traffic is more concerned with congestions. However, when designing a sustainable city logistics system in addition to environmental issues also social issues must be taken into account, since people living in the affected areas may suffer from harmful impacts of traffic. Anand et al. (2012) [2], Bozzo et al. (2014) [5] and Taniguchi et al. (2014) [34], for example, provide a review of literature on modelling urban freight transportation. Anand et al. (2012) analyse in particular the stakeholder's involvement, the objectives and the defining criteria of the reviewed urban logistic models. It shows up that efficiency in terms of internal costs is still a major objective but also minimising environmental impact is heavily studied. Social objectives are mostly only just mentioned, but formal and concrete approaches towards minimising social externalities are rare. In case they are tackled it is in form of reducing congestion what reduces in turn the risk of accidents. The approach in Holguín-Veras (2008) [21] addresses this by moving urban freight deliveries to off-hours, where usually traffic is not that dense. However this may increase disturbance to residents since traffic during night, for example, is in general perceived as more disturbing than traffic during day. So this is not of same meaning as the approach in the present work.

So despite the importance of making large cities also liveable, there has not been done much research on VRPs that take amongst others care of the harmful social impacts of freight transportation. In particular we could not find anything similar to the criteria of urban disturbance as we interpret it, namely relating it to population density and indicating thus how much or how many people are affected by traffic. Moreover we oppose the minimisation of this urban disturbance to with the conflicting objective of minimising gas emission, what makes our problem additionally innovative.

3 Bi-objective optimisation

The three modelling approaches developed and evaluated in the course of this work deal all with the same problem, namely optimising customer delivery with respect to the two conflicting objectives of emission and disturbance minimisation. To capture this conflicting situation we need a bi-objective problem formulation.

Bi-objective optimisation is a special case of multi-objective optimisation. In the following a brief overview of the theory of multi-objective optimisation and two possible solution methods is given, where the so called ε-constraint method will be explained in more detail since it is implemented for later computational experiments. Those who are not familiar with multi-objective optimisation and the ε-constraint method are recommended to go through this chapter in order to understand later parts of the work. Those who feel confident about it could see it as a review or skip it and continue with chapter 4.

3.1 General case: Multi-objective optimisation

A multi-objective optimisation problem arises when two or more separated but interconnected objectives have to be optimised simultaneously. This would involve also the case where the objectives are not conflicting at all, which means that optimising one objective yields also the optimal solution for all other objectives. However, we are interested here in non-trivial multi-objective optimisation problems, so the objectives are assumed to be at least partly conflicting. Moreover we assume that all objective functions are to be minimised, since this is the situation we have to deal with in our problem.

A multi-objective optimisation problem can be then mathematically described in the following way (according to [6]):

$$\text{"min" } \mathbf{f}(x) = (f_1(x), f_2(x), ..., f_k(x)) \tag{3.1.1}$$

$$\text{subject to } x \in S. \tag{3.1.2}$$

$S \subset \mathbb{R}^n$ is the set of feasible solutions, also called decision space, that is defined by the constraints of the optimisation problem. The function $\mathbf{f}(x)$ maps a solution vector $x \in S$ onto the objective space $\mathbb{R}^k$. So for a multi-objective optimisation the objective value corresponding to a certain solution is not just a real-valued number but a k-dimensional vector of real-valued numbers. The "min" in front of $\mathbf{f}(x)$ is quoted because it is not possible to minimise $\mathbf{f}(x)$ as a whole due to the conflicting component functions f_i :

$\mathbb{R}^n \to \mathbb{R}$.

When we ask now for solving the problem (3.1.1)-(3.1.2) we have to clarify first what is meant by an optimal solution in the context of multi-objective optimisation.
In a single-objective optimisation problem the optimal solution is the decision vector that minimises the given single objective function subject to the constraints. But is it also possible to find a single solution for the multi-objective optimisation problem that minimises all objectives simultaneously? In the assumed case of at least partly conflicting objective functions the answer is no. A solution that minimises one of the objectives can not minimise at the same time all other objectives in a feasible way. Instead there will be a trade-off between the objective values: Improving one of the objectives can only happen through degrading at least one other objective. This behaviour leads to the concept of Pareto optimality:

- A solution $x^* \in S$ is called *Pareto optimal*, *non-dominated* or *Pareto efficient* if there does not exist any other solution $x \in S$ that is at least as good as x^* in all objectives, i.e. $f_i(x) \leqslant f_i(x^*)$ for all $i \in \{1, \dots, k\}$, and strictly better than x^* in at least one objective, i.e. $f_j(x) < f_j(x^*)$ for at least one $j \in \{1, \dots, k\}$. [1]

- The set of all Pareto optimal solution vectors to a given multi-objective optimisation problem is called *Pareto set*.

- If x^* is a Pareto optimal solution the corresponding objective vector $\mathbf{f}(x^*) = (f_1(x^*), f_2(x^*), \dots, f_k(x^*))$ is referred to as *non-dominated point*.

- All non-dominated points together form the so called *Pareto front*. In the case of a bi-objective problem the Pareto front can be visualised in the two-dimensional objective space what is then referred to as trade-off curve.

Figure 3.1 illustrates the just introduced definitions for an arbitrary bi-objective optimisation problem, where both functions f_1 and f_2 have to be minimised. The green crosses represent the Pareto front or trade-off curve. For a pure minimisation problem the Pareto front will always go from top left to bottom right. The steepness of the curve expresses the trade-off between the two objective functions, i.e. how much an improvement in on objective costs in terms of the other objective. If we take for example the two middle crosses 2 and 3 (assuming we go from left to right) we see that the decline is flatter than

---

[1] The just described concept is actually that of strict Pareto optimality. Beside that there exists also the definition of weak Pareto optimality that says that a solution $x^* \in S$ is weakly Pareto optimal if there does not exist any other solution $x \in S$ that is strictly better in all objectives, i.e. $f_i(x) < f_i(x^*)$ for all $i \in \{1, \dots, k\}$.
For our purposes we stick with strict Pareto optimality, so when talking about Pareto optimality we actually mean strict Pareto optimality.

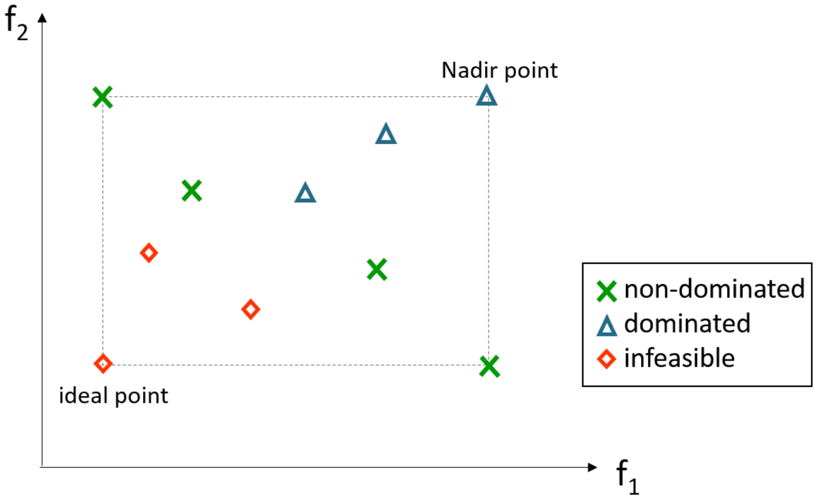


Figure 3.1: The concept of Pareto optimality

from 1 to 2 or 3 to 4. This means that improving f_2 from 2 to 3 causes a larger deterioration in f_1 than between the other points.

To better understand what distinguishes the non-dominated points from all other points in the objective space we have a look at the rest of the given points. In fact all points above and right to the non-dominated points are dominated, what means that there is a solution that is better in at least one objective. The solution corresponding to the first triangle (assuming again we go from left to right) is dominated in the sense of strict Pareto optimality. If we supposed weak Pareto optimality it would be non-dominated because none of the (strict) Pareto optimal solutions is better in both objectives. The third triangle represents a special point called Nadir point, whose components are the worst objective values of the solutions of the entire Pareto set [8].

Solutions corresponding to points below and left to the non-dominated points would be better in at least one objective, but they are ruled out by the constraint set, i.e. they are infeasible. The rhombus at bottom left shows the ideal point whose components are the objective values when minimising each of the two functions individually.

The Nadir point and ideal point give upper and lower bounds for the Pareto front as illustrated by the dotted lines in Figure 3.1. So in general computing the two points may be of interest when solving a multi-objective optimisation problem. However we do not make use of them in the course of this work.

3.2 Exact methods for solving multi-objective optimisation problems

So we know now that solving a multi-objective optimisation problem means basically creating the Pareto front. Therefore several methods exist, that differ in solution quality and performance. Some yield only parts of the Pareto front or approximations to it. However, we are here interested in finding the whole and exact Pareto front, what will bring us in the end to the ε-constraint method. But before we will present another idea for solving multi-objective optimisation problems.

3.2.1 Weighted-sum method

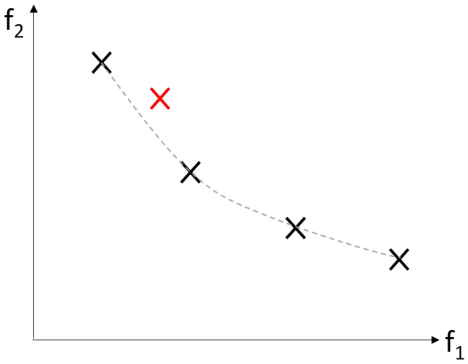
One can argue that there is a way around solving a multi-objective optimisation problem by combining the objective functions linearly to one objective function such that it can be solved as a single objective optimisation problem.Doing so implies putting a weight on each of the objective functions and only the solution corresponding to this specific weight combination is obtained. We would be able to do this if we had extra information on the decision maker's preferences who is the one interested in the solutions of the problem and finally deciding which ones are implemented based on his/her preferences. In the context of the present problem the decision maker could attach more importance to either emission or disturbance or prefer a rather balanced solution. If his/her preferences were known a priori we would be able to attach the corresponding weights to the objective functions and solve it as a single-objective optimisation problem. However, this extra information is rarely available in practice. Mostly decision makers are not willing or able to formulate their preferences in advance. Furthermore preferences can change very quickly, which means a solution that was appropriate last week may not be useful any more today. Hence they usually prefer to see first all potential solutions and the trade-off between the objective functions before evaluating which solution fits best.

One can further argue that the weights of the objective functions could be changed gradually in order to get several different or even all points on the Pareto front. But there remain some drawbacks that prevent us from using it:

- Each time a weight factor is changed a new single objective optimisation problem has to be solved what could result in high computational burden. Moreover it can happen that a change in the weight vector does not even yield a new Pareto optimal solution.

- The relation between the weight factors and the objective values is not consistent. For example in a bi-objective optimisation problem: When we double the weight for the first objective function and half the weight for the second objective function this does not necessarily imply that the objective values change accordingly. Thus it is difficult to anticipate which weight factor leads to which trade-off between the objectives.

- The weighted-sum method cannot find Pareto optimal solutions that lie in non-convex regions of the objective space what makes the approach inadequate for our purposes because we want to find the whole Pareto front. The picture below shows a situation where the weighted-sum method would fail in finding the Pareto optimal solution off the dashed line (that indicates the boundary of the convex region). More on the weighted-sum method can be found in Marler et al. (2010) [28].



3.2.2 ε-constraint method

The so called ε-constraint method [23] is another exact solution method for multi-objective optimisation problems that does not require any weighting and combining of the objective functions and finds even solutions in the non-convex region of the objective space[2]. The basic idea behind it is to choose only one of the objective functions to be optimised and convert the remaining ones into constraints by imposing an upper bound on their objective values. In mathematical terms this writes as follows:

$$\begin{aligned} \min\ & f_i(x) \\ \text{s.t.}\ & f_j(x) \leqslant \Theta_j,\ \forall j \in \{1,\dots,k\}\backslash\{i\} \\ & x \in S \end{aligned}$$

In the case of a bi-objective optimisation problem, as we face it in this work, this formulation reduces to

$$\min\ f_1(x) \tag{3.2.1}$$

$$\text{s.t.}\ f_2(x) \leqslant \Theta \tag{3.2.2}$$

$$x \in S. \tag{3.2.3}$$

---

[2] There are of course many further methods for solving multi-objective optimisation problem, either exactly or heuristically. Marler and Arora (2004) [27] give some insight in this context.

So only one of the the two objectives is chosen to be minimised (3.2.1) under the original constraint set (3.2.3) plus the upper bound constraint on the other objective (3.2.2). When solving this 'new' optimisation problem with an appropriate value for Θ, one Pareto optimal solution to the original bi-objective problem can be identified. To get the whole Pareto front the single-objective optimisation problem as formulated in (3.2.1)-(3.2.3) has to be solved several times for variations of Θ.

The question is now how Θ should be chosen and varied such that in the end the whole Pareto front is obtained. The ε-constraint method described in Algorithm 1 (based on [35]) uses a complying variation scheme.

Algorithm 1 ε-constraint method

```
Require: Increment $\varepsilon \in \mathbb{R}_+$
Ensure: Set of Pareto optimal solutions
 1: $\Lambda \leftarrow \emptyset$
 2: $\varepsilon$-constraint $\leftarrow f_2 < \infty$
 3: add $\varepsilon$-constraint to MIP
 4: repeat
 5:     $x \leftarrow Min(f_1)$
 6:     $localObjectiveBound \leftarrow f_1 = f_1(x)$
 7:     add $localOjbectiveBound$ to MIP
 8:     $x \leftarrow Min(f_2)$
 9:     $\Lambda \leftarrow \Lambda \cup x$
10:     remove $localObjectiveBound$ from MIP
11:     update $\varepsilon$-constraint: $\varepsilon$-constraint $\leftarrow f_2 \leqslant f_2(x) - \varepsilon$
12: until MIP cannot be solved
```

The algorithm in words

The result set Λ is used to store the Pareto optimal solutions. Line 1 says that it is initialized with the empty set. In the beginning the upper bound Θ on f_2 is chosen to be infinity (line 2) [3] what implies that there is in fact no upper bound on f_2 in the first optimisation step. Consequently the first point on the Pareto front we get is the one corresponding to the optimal solution to f_1 when minimising it in complete absence of f_2. Throughout the algorithm the upper bound on f_2 will be gradually decreased by the increment ε what justifies the notation 'ε-constraint'. To allow for these later updates on the upper bound we have to add the ε-constraint to the MIP before the first iteration although it is not yet restrictive (line 3).

In each iteration of the ε-constraint-method the underlying MIP is solved a first time by minimising f_1 subject to the original constraint set plus the ε-constraint (line 5). The resulting objective value $f_1(x)$ serves then as an equality constraint on f_1 when solving the MIP a second time, now with minimising f_2 (line 6-8). Doing so yields a new Pareto

---

[3] Actually the component of the Nadir point belonging to f_2 could be taken as an initial upper bound but doing so requires an extra optimisation step, what we want to avoid.

optimal solution that is added to the solution set Λ (line 9). The reason for naming the objective bound on f_1 'local' is that it is only valid for the current iteration because the objective value of f_1 changes in each iteration (it increases). Thus the local objective bound constraint has to be removed at the end of the current iteration (line 10) and added anew in the subsequent iteration in order make it possible that new solutions can be found.

Beside that, the ε-constraint has to be updated in order to generate a new solution in the next step (line 11). This happens through gradually decreasing the right hand-side of the upper bound constraint by the increment value ε.

The iteration procedure is repeated until no more feasible solution can be found. The last possible Pareto optimal solution is the one that corresponds to minimising f_2 individually.

We can see that the upper-bound variation scheme of the ε-constraint method is designed such that f_1 is gradually deteriorated and f_2 is gradually improved with every iteration.

In Figure 3.2 you can see an illustration of the algorithm by means of an example with three non-dominated points.

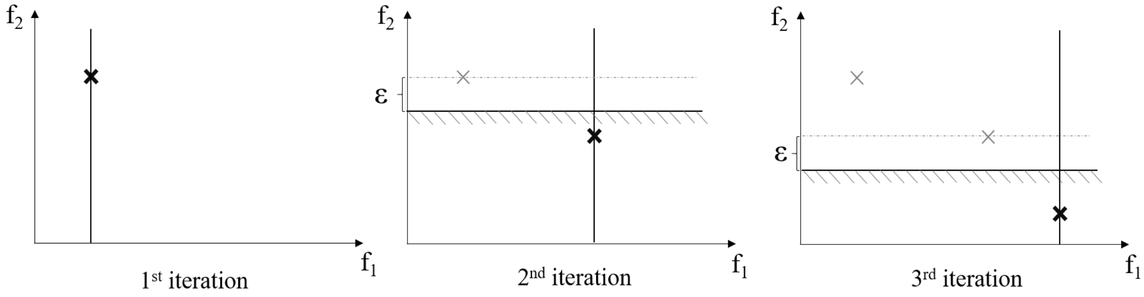


Figure 3.2: ε-constraint method

The horizontal, continuous line with the subjacent diagonal lines represents the ε-constraint and vertical line represents the local objective bound.

The increment parameter ε should be chosen carefully because it has an influence on the performance of the algorithm. If ε was chosen too large, some solutions may be left out. On the other hand, if ε was too small, there may run many iterations without finding any new Pareto optimal solution. A good benchmark for the value of ε can be obtained by evaluating the differences between the objective values of f_1 and f_2 along the Pareto front. So maybe it is necessary to execute several runs of the algorithm before a proper value for ε is found.

Furthermore it has to be decided which of the two objective is taken as f_1 and which one as f_2. Concerning the Pareto front it does not matter because with an appropriate choice of ε both versions yield theoretically the same result. But it could matter with regard to CPU effort. So one version may be faster in finding the Pareto front than the other one.

We will have a closer look on this when we come to the computational study in chapter 5.

4 Mathematical problem formulation

The underlying problem scenario of this work has been already briefly described in chapter 1 and we also know it requires dealing with a bi-objective optimisation problem due to the conflicting objectives. In this chapter things become now more precise in terms of formulation. Three different bi-objective modelling approaches are going to be presented where each of them relies on specific assumptions that totally change the structure of the formulation. However, there are some features that can be generalized across all three models.

4.1 Features valid for all three approaches

4.1.1 Data

Information on locations, vehicle and demand

The task is to deliver a given set of customers C within a single tour. This tour is assumed to start and end at an a-priori fixed hub. So no decisions concerning hubs have to be made in our optimisation problem. The fixed hub acts as depot for the goods as well as recharge station for the alternatively fuelled vehicles. The problem with electric or hybrid vehicles is that recharging stations are not yet very frequent. So running out of energy during a tour should be avoided what could be modelled through imposing an upper bound on the duration of the tour. But we ignore this in our framework.
A further crucial assumption concerning the vehicle is that the capacity restriction is not effective and we do not need to go back to the hub in between for reloading goods.
The absence of restrictions on capacity and tour duration in the optimisation problem requires that the a-priori, fixed set of locations is such that demand pattern and time reachability respect the implicitly given limits. Otherwise it is not guaranteed that the resulting solutions are feasible with respect to capacity and duration.
Concerning demand we assume that each customer $i \in C$ has strictly positive and deterministic demand q_i. Strictly positive demand implies that customers must be visited during the delivery tour. Deterministic demand means that we know already at the time of planning how much must be delivered to each customer.
The time horizon of our optimisation problem is assumed to comprise only one period, e.g. one day, thus demand data refers to this specific day.

Distance and disturbance data

Going from a location i to another location j is assumed to be associated with covering a distance d_{ij} and producing disturbance g_{ij} to urban neighbourhood. All formulations presented in the next sections will be asymmetric, i.e. the distance and disturbance when going from i to j can be different to the distance and disturbance when going from j to i. This should model the situation in real life where asymmetries in the road network occur, for example, through one ways.
Distance data is assumed to fulfil the triangle inequality, so when going from one location to another the way over a third location must be at least as long as the direct way.

A very crucial assumption is that triangle inequality must not hold for disturbance data. So the way over a third location could be less disturbing than the direct one. Actually this assumption makes the idea of including optional nodes in Model 2 meaningful at all.

Quantifying and measuring distance is rather straightforward. Quantifying disturbance, however, is more challenging, since it is a rather intangible value. The lack of related research aggravates the situation. We mentioned already that disturbance should relate in our context to how much people are affected by noise, risk or vibration caused through traffic in the residential areas. When generating disturbance data for the purpose of this work, we measured it in relation to population density. More on this can be found in the computational study part. For formulating the models the most important thing is to have in mind that disturbance data does not fulfil the triangle inequality.

Emission parameters

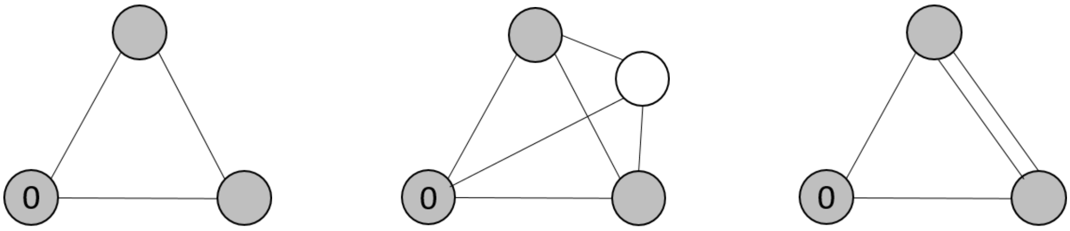
For calculating the total disturbance of a route only disturbance data is needed. For calculating total emission we need in addition to distance data also information on how much pollutant gases the vehicle produces. Therefore we have a parameter e that gives the gas emission when travelling one unit of distance while the vehicle is empty and a second parameter f that gives gas emission per unit of load weight when travelling one unit of distance. Both parameters are constant but depend on the type of vehicle. We do not care about how to come up with these values but just take them as given from the data.

4.1.2 Graphical representation

Each of the modelling approaches induces a different type of graph, but the following features hold true for all three approaches:

The locations are represented by the graph's node set V and ways between them by the corresponding arc set A. The hub will always be node 0. In Model 1 and 3 the customer nodes account for the rest of the node set. For Model 2 the node set V also contains so called optional nodes that must not but can be visited if beneficial in terms of disturbance. We assume that the underlying graph is always complete, which means there is an arc (i, j) for each node $i \in V$ to each other node $j \in V$, with $i \neq j$ since loops are forbidden. The special feature of Model 3 is that the graph can contain more than one arc between each pair of nodes. Such a graph is referred to as multi-graph in graph theory.

Each arc is associated with an arc-weight-vector (d_{ij}, g_{ij}), where the first component refers to the distance from node i to j and the second to its disturbance. As already mentioned, all three modelling approaches are formulated asymmetrically, what implicates considering a directed graph, where the arc-weight-vector for arc (i, j) is not necessarily equal to the arc-weight-vector for the opposite-directed arc (j, i).[1]



(a) Model 1 (b) Model 2 (c) Model 3

Figure 4.1: Example graphs for the three modelling approaches

Figure 4.1 shows the graphs induced from the three different modelling approaches, illustrated for an example instance with two customers. For Model 2 an additional optional node is available and for Model 3 there are two arcs between the two customers.

Arc weights are not displayed in any of the three graphs, but would be available from data.

4.1.3 Decision variables

For each arc there is a binary decision variable x_{ij} that is either 1 when the arc is used in the solution or 0 when not. Using an arc (i, j) means that location j is visited directly after location i. All non-zero x_{ij} together define then the sequence of nodes, i.e. the solution route. Moreover there is for each arc a continuous and non-negative decision variable u_{ij} that gives the total load weight carried in the vehicle when traversing arc

---

[1] Actually a complete digraph would mean that there are two arcs for each pair of nodes, one in each direction. For reasons of simplicity we assume here that the two directions are represented by only one arc. Moreover in graphical illustrations the arcs are displayed without arrows although there is formally a bi-directed arc behind it. If an arc is provided with an arrow it indicates the direction of a route/path as a result of any of the considered optimisation problems.

(i, j). For modelling approach 3 the decision variables are of course available for each of the multiple arcs what requires an additional index but more on this will be given in the respective section.

4.1.4 Objective functions

We know already that the two conflicting objective functions of our optimisation problem are that of total emission and total disturbance, where both are to be minimised. Although they are embedded into three different models the structure of their formulation remains essentially the same across all three models. That is why we set them up in a basic form already here, using the information about data and decision variables we have so far. The model-specific adaptations can be then easily taken later.

$$\min \sum_{i \in V} \sum_{j \in V} (e d_{ij} x_{ij} + f d_{ij} u_{ij}) \tag{4.1.1}$$

$$\min \sum_{i \in V} \sum_{j \in V} g_{ij} x_{ij} \tag{4.1.2}$$

Function (4.1.1) measures total emission and (4.1.2) total disturbance of a tour. We will analyse them now in more detail:

Gas emissions

There are many different factors playing together when total emission for a certain route should be measured. It would be too complex to incorporate all of them into a mathematical, abstract model. Thus we ignore for example vehicle speed and acceleration in our optimisation problem and assume that solely the amount of pollutant gases produced during a delivery-tour depends on the technical features of the used vehicle, the length of the tour and the carried load weight. The type of vehicle is crucial, because when driving a route with an alternatively fuelled vehicle total emission will be lower than driving the same route with, for example, a diesel-powered vehicle. Since our optimisation problem does not contain any decisions on vehicle type this component of emission is fixed and covered by constant input parameters e and f. However tour length and load weight are assumed to be controllable and thus decided from within the optimisation problem. Both factors are interdependent, because load weight values differently depending on over which distance you carry it. So a fully loaded vehicle should preferably go shorter ways than an empty vehicle. However, also an empty vehicle causes pollutant gases. Therefore we make a basic distinction between two different types of controllable emission: the load-independent emission, emerging from just driving the vehicle, and the load-dependent emission, emerging from carrying weight over the travelled distance. The first part of the sum in objective function (4.1.1) refers to the first type and the second part to the latter. Load-dependent emission is measured over driven vehicle-kilometres, that become effective through binary decision variables x_{ij}, times the emission parameter e. Load-dependent emission is measured also over driven vehicle kilometres, but now they are not just counted or not, but are additionally weighted through load decision variables u_{ij}. Emission parameter f is needed to count for emission per unit of distance and per unit of load weight. e and f serve moreover as kind of weighting factors between the two types of emission. So depending on how they are chosen more importance is given to load-independent or load-dependent emission or a balance between them is achieved.[2]

Disturbance

Disturbance data is as well as distance data arc-dependent. So the total disturbance of a route is computed by summing up the used arcs expressed by the binary decision variables x_{ij} and the corresponding disturbance value. We assume that load has no influence on disturbance. So function (4.1.2) is actually just a linear cost function where cost arise is in form of disturbance.

The way the two objective functions are designed links them via the binary decision variable x_{ij}. The consequence of this is that we have to deal with a bi-objective optimisation problem, whose particularities were already described in chapter 3.

4.1.5 Constraints

To complete a mathematical formulation of our optimisation problem also a set of constraints is needed that makes sure the solutions respect all assumptions and requirements, like having a closed, single tour or that load variables are in accordance with the route. However the formulation of the constraint set differs heavily across the three approaches. Hence it does not make sense to state anything general already here. The only thing we can already anticipate is that the set of constraints will always comprise only linear inequalities and equations.

---

[2] Since the parameters e and f are arc-independent and always multiplied linearly to the distance values d_{ij} we could multiply them ex ante, giving then data e_{ij} and f_{ij}. Objective function (1) would read as $\min \sum_{i \in V} \sum_{j \in V} (e_{ij} x_{ij} + f_{ij} u_{ij})$. But doing so would make the formulation less transparent - it would hide the fact that emission is proportional to distance. Moreover it would require extra pre-processing of data, because we get distance data and emission parameters separately.

4.1.6 Mixed Integer Programming

Due to the fact that our optimisation problem contains integer - actually binary - and real valued, continuous decision variables, a linear set of constraints (except for the integer constraint on the binary variable) and linear objective functions we can classify it as a Mixed Integer Program (MIP) [37].
Hence, all model formulations of this work are in fact bi-objective MIPs.

4.2 Model 1

Model 1 can be seen as a kind of basic model because on the one hand it is rather simplistic and on the other hand it serves as a starting point for setting up the other two, more sophisticated models.

4.2.1 Classification: Bi-objective Travelling Salesman Problem

Model 1 falls into the class of bi-objective Travelling Salesman Problem (bi-objective TSP). The bi-objective TSP is an extension of the classical, single-objective TSP. The goal of the classical, single-objective TSP is to find the cost-minimising tour that visits every given location exactly once and returns to the starting point. The TSP is one of the most studied combinatorial optimization problems in operations research due to its relevance in many real-life applications. In Lawler (1985) [24] and Gutin and Punnen (2002) [20] you can find an extensive treatment of this problem class. In transportation logistics the locations are usually cities, customers or any other logistically relevant location (warehouse, airport,...). The term cost can have many different meanings. In transportation logistics mainly geographical distance or time are taken as cost, because they are strongly connected to saving money, what is one of the main interests of decision makers. But increasingly important are nowadays also external costs - like for example risk or - what we consider - gas emission and disturbance to urban neighbourhood.
In a single-objective TSP only one type of cost is considered for decision making. In this work two types of cost appear - namely emission and disturbance - what brings us to a bi-objective TSP. The result for a bi-objective TSP is not only one cost-minimising tour but a set of Pareto-optimal tours.

A bi-objective variation of the classical TSP is the TSP with profits introduced by Feillet et al. (2005) [15], where visiting a location is associated with a creating a certain profit and the two objectives are that of maximising collected profit and minimising travel costs. There can be found many other studies about multi-criteria TSP and attempts to solve them efficiently. Ehrgott and Gandibleux (2000) [13] provide a comprehensive survey on literature about multi-objective combinatorial optimisation, in particular also about multi-objective TSP.

4.2.2 Bi-objective MIP formulation

Data

Data is mainly as described in section 3.1 and can be summarized as follows:

$\{0\}$... fixed, single hub
C... set of customer locations
$V = \{0\} \cup C$... set of all locations
A... set of arcs (i, j) connecting each pair of nodes i and $j \in V$
d_{ij} ... travel distance when going from i to j
e... emission when travelling one unit of distance while vehicle is empty
f... emission per unit of load weight when travelling one unit of distance
q_i... $\begin{cases} > 0 & \text{for } i \in C \\ = 0 & \text{for } i = 0 \end{cases}$
g_{ij} ... disturbance when going from i to j

Decision variables

Also decision variables are as described in section 3.1:

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases}$$

u_{ij}... load when travelling from i to j

Objective functions and constraints

$$\min \sum_{i \in V} \sum_{j \in V} (e d_{ij} x_{ij} + f d_{ij} u_{ij}) \tag{4.2.1}$$

$$\min \sum_{i \in V} \sum_{j \in V} g_{ij} x_{ij} \tag{4.2.2}$$

subject to:

$$\sum_{j \in V \setminus \{i\}} x_{ji} = 1 \qquad \forall i \in V \qquad (4.2.3)$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} = 1 \qquad \forall i \in V \qquad (4.2.4)$$

$$\sum_{j \in V} u_{ji} - \sum_{j \in V} u_{ij} = q_i \qquad \forall i \in C \qquad (4.2.5)$$

$$u_{ij} \leqslant M x_{ij} \qquad \forall i, j \in V \qquad (4.2.6)$$

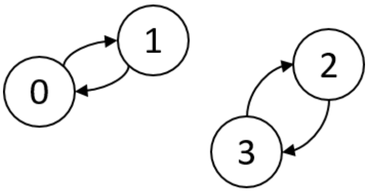
$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \qquad (4.2.7)$$

$$u_{ij} \geqslant 0 \qquad \forall i, j \in V \qquad (4.2.8)$$

The two objective functions measure total emission (4.2.1) and total disturbance (4.2.2) and are formulated as in section 4.1. So data, decision variables and objective functions are very similar to what we introduced in a general form in 4.1. But now also the set of constraints is defined through (4.2.3)-(4.2.8) and described in more detail:

Constraints (4.2.3) say that there must be exactly one ingoing arc for each node and constraints (4.2.4) say that there must be also exactly one outgoing arc for each node. Together they are responsible for route continuity, which means that every entered node must be again left. Constraints (4.2.5) define the load flow. For a certain customer $i \in C$ this means that the load before arriving at this customer, given through $\sum_{j \in V} u_{ji}$, is forced to be reduced by the customer's demand q_i. $\sum_{j \in V} u_{ij}$ is then the load when leaving customer i. For the hub node this can not be claimed, because its zero demand would force the load from leaving the hub until coming back at the end of the tour to remain unchanged. This would be in contradiction to the fact that load is gradually reduced through delivering the strictly positive demand to the customers. For optimality reasons the load variables should be kept as small as possible, so the load will be equal to total customers' demand when leaving the hub and will be zero when coming back to the hub.

Constraints (4.2.5) do not only define the load flow but also prevent subtours in the solutions. (4.2.3) and (4.2.4) alone only guarantee that each entered node is again left - what would be also fulfilled when a solution contains subtours, i.e. circular paths that are disconnected from the hub. To make sure there is only one circular path, namely the one containing all nodes, constraints (4.2.5) are needed. For a better understanding of these constraints we consider a small example with three customers having demand $q_1 = 10$, $q_2 = 50$, $q_3 = 150$. In the figure on the right you can see a solution that respects constraints (4.2.3) and (4.2.4) since there is one ingoing and one outgoing arc for each node.



But it contains a subtour of length 2 containing nodes 2 and 3. The tour containing the hub fulfils (4.2.5) when $u_{01} = 50$ and $u_{10} = 0$ because there is no load reduction requirement on the hub node. But the tour disconnected from the hub does not. For both customers must hold that the difference between the ingoing and the outgoing load is equal to the demand. If we take $u_{23} = 200$ and $u_{32} = 50$ then the difference for node 3 is equal to its demand of 150 but for node 2 this load variable constellation means a difference of -150 between ingoing and outgoing load, so we would actually gain load. If we try to adjust the load variable correctly for node 2 through $u_{32} = 200$ and $u_{23} = 50$, then difference for node 3 is -150. For the example solution also no other load variable combination will lead to a valid result. In fact, only a solution representing a closed, single tour will fulfil the equation of (4.2.5) for each customer.

Constraints (4.2.6) link the two decision variables x_{ij} and u_{ij}. They force u_{ij} to be zero when x_{ij} is zero, so no load can be carried via an arc that is not used. When an arc is used, i.e. when $x_{ij} = 1$, then there should be also positive load (except for the way back to the hub). Therefore big M is needed on the right hand side of (4.2.6) in order to allow the load variable to take also values greater than x_{ij}'s maximal value of 1. When implementing and solving the model, M can be set a priori to the sum of all customers' demand, because u_{ij} will never exceed this value (for optimality reasons, as already mentioned before).

Constraints (4.2.7) define x_{ij} to be binary and (4.2.8) define u_{ij} to be a non-negative, real number.

4.3 Model 2

4.3.1 Why optional nodes?

When developing Model 1 we assumed that a path from one node to an other has been already evaluated and corresponding distance as well as disturbance are given in the data. In Transportation Logistics it is common to consider the path with minimal distance, so let's assume that this was also the case in Model 1. With regard to emission minimisation it is fine to consider minimal-distance paths but we have already mentioned that the shortest, direct way is often one that causes high disturbance, when going for example through a highly populated area. Thus considering only the path with minimal distance may prevent us from finding good solutions in terms of disturbance.

Therefore the idea in Model 2 is to introduce so called optional nodes. Including one

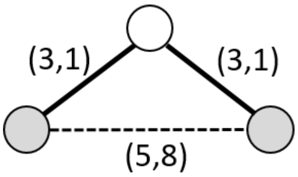


Figure 4.2: Going over optional node reduces disturbance

or more optional nodes between two customer nodes or between a customer and a hub node results in an alternative path to the already existing direct one. In terms of distance the alternative path will constitute as an effect of the triangle inequality. However, the detour-path may have a lower disturbance value than the distance-minimising one, so it may become beneficial to include one or more optional nodes in solutions that aim for an improvement in total disturbance.

We do not impose any must-visit condition or any other restriction on the optional nodes. But they are provided with associated distance and disturbance data. Hence, the decisions on the inclusion of optional nodes are made endogenously from within the optimisation problem.

The idea of considering alternative ways over optional nodes is motivated by the fact that also in real-life there are often less residents disturbed when traffic goes around highly populated areas, using for example bypass streets.

Figure 4.2 illustrates the idea of model 2, where the two shaded circles are assumed to be customer nodes and the non-shaded circle represents an optional node. The dashed line displays the direct, given arc between the two customers. The alternative way over the optional is represented by the fat depicted path, comprising two arcs. Adding up the disturbance values of these two arcs gives a total disturbance of 2 for the alternative path, which is lower than the disturbance 8 of the direct path. Thus, going over the optional node is beneficial in terms of disturbance, but with an increase in length from 5 to 6.

4.3.2 Classification: Bi-objective Shortest Path Problem with mandatory nodes

In Model 1 it was required that all given nodes are visited exactly once during a tour, what classified it as a TSP.[3]

As already mentioned we do not impose any restrictions on the optional nodes - visiting them is just optional - what explains the notation. In contrast we call the customer nodes and the hub from now on mandatory nodes since it is still required that they are visited. Anyway, the absence of a must-visit condition on optional nodes drifts the problem away

---

[3] Actually the hub node is visited twice - once at the beginning and once at the end of a tour. But if we consider the tour as a closed cycle, the hub node is contained only once.

from being a bi-objective TSP. Instead we classify it as a bi-objective Shortest Path Problem with mandatory nodes.

We will first describe the classical single-objective Shortest Path Problem to gain some insight into this problem class and come then to the specific features of the bi-objective Shortest Path Problem with mandatory nodes.

The classical Shortest Path Problem

The Shortest Path Problem (SPP) is as well as the TSP a very well studied problem in Operations Research in particular in Transportation Logistics. The task is to find the shortest path from one specified node - usually called start/source node - to another specified node - usually called end/sink/target node. [4] As well as for the TSP the term shortest may be a bit misleading because it is not necessarily distance or time that is to be minimised, but also other types of cost, like emission or disturbance.

One of the basic works on SPPs was done by Ford (1956) [17], who co-developed the Ford-Fulkerson [16] and Bellman-Ford algorithm [4] for solving a SPP. Ahuja et al. (1988) [1] provide a good overview of SPPs and its applications.

In the case of two decision criteria the bi-objective SPP applies, where each arc is associated with a two-dimensional weight vector and not the shortest path but Pareto optimal paths have to be found.

The main differences between the classical SPP and the TSP are:

- Apart from the source and target node there is no 'must-visit'-condition on the other nodes.
- Arc weights can also be negative. This may implies that some nodes are visited more than once what leads to cycles.
- In general start and target are not the same, so the path will not end where it started.

Since the classical SPP has some characteristics that are not in accordance with our framework, the following adaptations are made - leading us to the bi-objective shortest path problem with mandatory nodes.

Bi-objective Shortest Path Problem with mandatory nodes

- We assume that the customer nodes must be included exactly once in the resulting shortest path. This justifies attaching 'mandatory nodes' in the notation of the

---

[4] We will use the notation source and target node throughout this work.

problem.

- Concerning the optional nodes we stick to the classical SPP, i.e. they can but must not be visited and multiple visits are allowed.
- The hub serves as source and target node, so source and target node are the same. In contrast to the classical problem this assumption makes sense here, because the must-visit condition on customer nodes forces the solutions to not just stay in the hub node.
- Arc weights will not be negative in our setting, since distance and disturbance data is always non-negative. Nevertheless cycles are likely to appear because customers must be visited exactly once and optional nodes can be visited more than once.

4.3.3 Bi-objective MIP formulation - first attempt

The idea for the first formulation attempt is to use just a classical shortest path problem formulation extended and adapted by the assumptions explained just before. But a counterexample will show that this formulation fails in the load flow constraints. So three alternatives are presented, where only the latter two will turn out to be valid approaches for the postulated framework.

Data

Data for Model 2 contains now also the set of optional nodes denoted by O. The demand for these nodes is 0. The enlarged node set V enlarges also the arc set A and distance and disturbance data matrices. The emission parameters e and f as well as everything related only to customer and hub nodes remains the same.

$\{0\}$...	fixed, single hub
C...	set of customer locations
O...	set of optional nodes
$V = \{0\} \cup C \cup O$...	set of all locations
A...	set of arcs (i, j) connecting each pair of nodes i and $j \in V$
d_{ij} ...	travel distance when going from i to j
e...	emission when travelling one unit of distance while vehicle is empty
f...	emission per unit of load weight when travelling one unit of distance
q_i...	$\begin{cases} > 0 & \text{for } i \in C \\ = 0 & \text{for } i \in D \cup \{0\} \end{cases}$
g_{ij} ...	disturbance when going from i to j

Decision variables

Decision variables are needed also for arcs corresponding optional nodes.

$$x_{ij} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases}$$

u_{ij}... load when travelling from i to j

Objective functions and constraints

$$\min \sum_{i \in V} \sum_{j \in V} (e d_{ij} x_{ij} + f d_{ij} u_{ij}) \tag{4.3.3.1}$$

$$\min \sum_{i \in V} \sum_{j \in V} g_{ij} x_{ij} \tag{4.3.3.2}$$

subject to:

$$\sum_{i \in V \setminus \{j\}} x_{ij} = 1 \qquad \forall j \in C \cup \{0\} \tag{4.3.3.3}$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} = 1 \qquad \forall i \in C \cup \{0\} \tag{4.3.3.4}$$

$$\sum_{j \in V \setminus \{i\}} x_{ji} = \sum_{j \in V \setminus \{i\}} x_{ij} \qquad \forall i \in O \tag{4.3.3.5}$$

$$\sum_{j \in V} u_{ji} - \sum_{j \in V} u_{ij} = q_i \qquad \forall i \in C \cup O \tag{4.3.3.6}$$

$$u_{ij} \leqslant M x_{ij} \qquad \forall i, j \in V \tag{4.3.3.7}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{4.3.3.8}$$

$$u_{ij} \geqslant 0 \qquad \forall i, j \in V \tag{4.3.3.9}$$

Constraints (4.3.3.1) and (4.3.3.2) say again that customers and hub must be visited exactly once. Although there is no must-visit condition on the optional nodes, we need constraints (4.3.3.5) to make sure the number of ingoing arcs is equal to the number of outgoing arcs for each optional node.
Constraints (4.3.3.6) define the load flow of the solution tour. For customer nodes their meaning translates one-to-on from model 1. For optional nodes the total ingoing load is defined to be equal to the total outgoing load because their demand is 0. Constraints (4.3.3.7) are equivalent to (4.2.6) of Model 1. And again (4.3.3.8) and (4.3.3.9) define the scope of the decision variables..

Model 2's point of failure

At first glance the formulation above seems to capture everything we want. Unfortunately a glance at the following example reveals that the load flow constraints may fail in case an optional node is visited more than once. The graphical representation of the example can be seen in Figure 4.3, where the the 5 shaded circles represent mandatory nodes and the two not shaded circles represent optional nodes (there may be more, but to keep the graph manageable only the visited ones are illustrated).

The demands of the customers are $q_1 = 20, q_2 = 80, q_3 = 120, q_4 = 30$. The depicted route - showing also the load weight on the corresponding arc - is assumed to be a Pareto optimal solution to Model 2 as formulated right before. So in particular it has to fulfil all conditions expressed through constraints (4.3.3.3)-(4.3.3.9). But when we look at the load variables we see that they do not evolve as desired. The requirement on the load is to be gradually reduced by the customers' demands but remain unchanged when an optional node is passed. The load variables associated with optional node 6 violate this

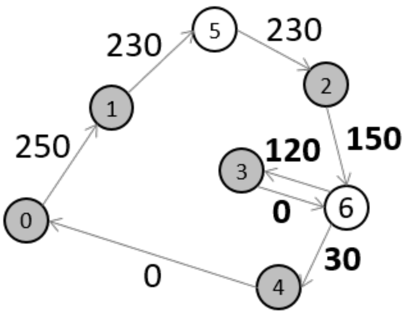


Figure 4.3: Multiple visits in optional node cause load variable problems

condition in the following sense: First we go from customer 2 to 3 over 6. Between 2 and 3 load should not change because there is nothing to deliver to 6. But in the example solution load has decreased by 30 (from 150 to 120) when leaving node 6. So from 6 to 3 the vehicle carries now only the amount that has to be delivered to customer 3, i.e. the demand of 120. When going then from 3 to 4 (the next customer in the tour), passing again optional node 6, the vehicle is first completely unloaded on the arc from 3 to 6 and suddenly carries again a load of 30 when going from 6 to 4, although there is no hub for reloading in between. Such a behaviour is neither realistic nor desired to occur in any of our Pareto optimal solutions.

So, how can it then happen at all that such a solution is obtained? The reason lies in constraints (4.3.3.6). They say that for any customer or optional node the *total* ingoing load must be equal to the *total* outgoing load. This makes sense as long as a node is visited only once and the ingoing as well as the outgoing load weight corresponds thus to only one arc. However, for optional nodes multiple visits are possible, which implies that there can be several times an ingoing and outgoing load. This is where constraints (4.3.3.6) fail, because they do not guarantee that for each of the potentially multiple visits the ingoing load equals the outgoing load, but only equality for the total number of all visits.
This permits then having a solution like shown in the counterexample, where the total ingoing load for optional node 6 is 150 (=$150 + 0$) and also the total outgoing load is 150 (=$120 + 30$). What happens with the load variables directly before and after each visit is thus not defined by constraints (4.3.3.6).

4.3.4 Model 2A: Allow at most one visit for optional nodes

Since the problem in the formulation of Model 2 arises from the eventuality of multiple visits on optional nodes a simplistic idea is to assume that each optional node can be included at most once in a final tour. Formally this is done by adding the following inequalities to the set of constraints:

$$\sum_{i \in V \setminus \{j\}} x_{ij} \leqslant 1 \qquad \forall j \in O \tag{4.3.4.1}$$

Since x_{ij} is assumed to be binary the sum is either 0, which means there is no arc leaving the optional node, or 1 in case of one outgoing arc. But in no case there can be more than one outgoing arc. To ensure consistency with the ingoing arcs constraints (4.3.3.5) are still needed.

Constraints (4.3.3.6) together with constraints (4.3.3.7) induce the load flow to develop as desired.

However this formulation in insufficient in the sense that the Pareto set is restricted to solutions that allow at most one visit in the optional nodes while our purpose is to exploit the availability of optional nodes to full extent.

4.3.5 Model 2B: Duplication of optional nodes

But there is a way to keep the restriction of at most one visit and still allow for multiple visits. What sounds contradictory becomes possible when the optional nodes are duplicated. Duplicating means creating copies that carry exact the same information as the original. In this context we call now O the set of *original* optional nodes. So when we take for example any arbitrary original optional node $i \in O$, then the duplications of i have the same coordinates, demand (which is always 0 for optional nodes) and distance d_{ij} and disturbance g_{ij} to all other nodes $j \in V \setminus \{i\}$. So although there exists from a topographical point of view only the original optional node it is essential to have the copies in the data set of Model 2B. Then constraints (4.3.4.1) allow for multiple visits in optional nodes, because when an original node is visited and exploits therefore the at-most-one-visit limit, there are still its duplications where each of them allows for another single visit.

There arises the question of how many duplications of each optional node are needed. If we knew in advance how often each optional is included in a solution tour, the according number of duplications could be made already before solving the problem. But since nothing is predictable with regard to optional nodes we have to assume every possible case, in particular also the extreme case where an optional node is included between each two different mandatory nodes. An illustration of this case can be seen in Figure 4.4. Shaded nodes are mandatory, where the hub is labelled with 0. The white node is optional. The illustrated example route includes the optional node between each two mandatory nodes.

So in general each original optional node should be available ($|C|+1$) times, where ($|C|+1$)

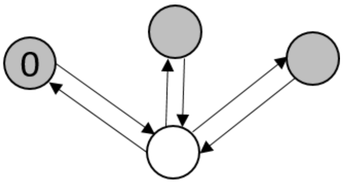


Figure 4.4: Including an optional node between each two different mandatory nodes

is the number of mandatory nodes.[5] Since the original node is already available from the beginning only $|C|$ duplications are needed.

Data and decision variables

We denote by $O_d(p)$ the set that contains an original optional node $p \in O$ and its duplications. O_d refers to the whole set of optional nodes, i.e. $O_d = \bigcup_{p \in O} O_d(p)$. So O_d contains all original optional nodes and their duplications.
The notation for the decision variables is the same as in 4.3.3, but the set of decision variables is now larger due to the larger node set O_d and consequently larger arc set.
Also arc-dependent data d_{ij} and g_{ij} has to be duplicated accordingly.

When we made in section 4.1 general assumptions on the modelling approaches we assumed that graphs induced for our optimisation problems do not contain any loops, which means arcs that connect a node with itself do not exist. However, we have now arcs that connect the duplicated nodes with the original node and also arcs that connect duplicated nodes belonging to the same original node. Such arcs represent actually loops. Their associated weight-vector is always $(0, 0)$, because no distance is travelled and no disturbance is caused. Since we have also defined the corresponding decision variables it can happen that they are included in a solution because it does not contribute anything to the objective values. In order to avoid such a behaviour the following equations are added to the model:

$$x_{ij} = 0 \qquad \forall i, j \in O_d(p), \ \forall p \in O$$

---

[5] The notation $|X|$ refers to the cardinality of a set X, i.e. the number of elements contained in X.

Objective functions and constraints

$$\min \sum_{i \in V} \sum_{j \in V} (ed_{ij} x_{ij} + f d_{ij} u_{ij}) \tag{4.3.5.1}$$

$$\min \sum_{i \in V} \sum_{j \in V} g_{ij} x_{ij} \tag{4.3.5.2}$$

subject to:

$$\sum_{j \in V \setminus \{i\}} x_{ji} = 1 \qquad \forall i \in C \cup \{0\} \tag{4.3.5.3}$$

$$\sum_{j \in V \setminus \{i\}} x_{ij} = 1 \qquad \forall i \in C \cup \{0\} \tag{4.3.5.4}$$

$$\sum_{i \in V \setminus \{j\}} x_{ij} \leqslant 1 \qquad \forall j \in O_d \tag{4.3.5.5}$$

$$\sum_{j \in V \setminus \{i\}} x_{ji} = \sum_{j \in V \setminus \{i\}} x_{ij} \qquad \forall i \in O_d \tag{4.3.5.6}$$

$$\sum_{j \in V} u_{ji} - \sum_{j \in V} u_{ij} = q_i \qquad \forall i \in C \cup O_d \tag{4.3.5.7}$$

$$u_{ij} \leqslant M x_{ij} \qquad \forall i, j \in V \tag{4.3.5.8}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{4.3.5.9}$$

$$u_{ij} \geqslant 0 \qquad \forall i, j \in V \tag{4.3.5.10}$$

Objective functions are as usual those of total emission and total disturbance.

Constraints look basically the same as for Model 2A. So they still include the at-most-one-visit condition (so either no visit at all or only one) expressed through (4.3.5.5) but referring now not only to the optional nodes but also to all their duplications. So when the at-most-one-visit restriction is exploited for the original optional node, there is still the possibility to visit it effectively further times by exploiting the at-most-one-visit restriction of its duplications.

Also constraints (4.3.5.6)-(4.3.5.10) are formulated now for each of the duplicated nodes.

Valid inequalities:

The formulation we established so far for Model 2B is technically complete and will yield a valid result. However, to speed up exploration evaluations of decision variables we add the following valid inequalities:

$$\sum_{i \in V} x_{iu} \geqslant \sum_{j \in V} x_{jv} \forall p \in O, u, v \in O_d(p), v > u$$

So for duplications i, j belonging to the same original node p the one with the higher indices can not be used as long as one with a lower index is not used.

4.3.6 Model 2C: Introduction of a position index

There is yet another way to formulate the problem scenario of Model 2 such that load flow is in accordance with the arc sequence, also in the case of multiple visits, that does not require duplication of optional nodes. Instead the idea is to provide decision variables x_{ij} as well as the load variables u_{ij} with an additional index k that indicates the position of an used arc in the route.

Data and decision variables

The decision variables for this approach are then defined as follows:

$$x_{ij}^k = \begin{cases} 1 & \text{if arc } (i,j) \text{ is in position } k \text{ in the solution tour} \\ 0 & \text{otherwise} \end{cases}$$

u_{ij}^k... load when travelling on arc (i, j) being in k^{th} position in the solution tour

So the two decision variables are now additionally linked via the position index k in the sense that the load variable u_{ij} corresponding to a certain arc (i, j) must carry the same value for index k as the corresponding binary decision variable x_{ij}.
When we consider for example a tour $(0, 2, 1, 3, 0)$ with corresponding load flow $(120, 50, 30, 0)$ the decision variables inducing this result are:

$$\begin{aligned} 0 \rightarrow 2:&\quad x_{02}^1 = 1, \quad u_{02}^1 = 120 \\ 2 \rightarrow 1:&\quad x_{21}^2 = 1, \quad u_{21}^2 = 50 \\ 1 \rightarrow 3:&\quad x_{13}^3 = 1, \quad u_{13}^3 = 30 \\ 3 \rightarrow 0:&\quad x_{30}^4 = 1, \quad u_{30}^4 = 0 \end{aligned}$$

All remaining decision variables are zero.

The data used for this formulation is the same as for the original Model 2, so although the new index k is associated with arcs it has no influence on the arcs themselves or on arc-dependent data d_{ij} and g_{ij}.
Ideally the index k should range from 1 to the number of arcs included in a solution. If we had only mandatory nodes, the maximal tour length and thus number of included arcs would be equal to the number of mandatory nodes. But in the context of Model 2 we have

optional nodes and with each included optional node the tour length increases by one. Since we can not foresee how many optional nodes will be included in any of the Pareto optimal tours we set the upper bound of k's range to the length of the extreme tour that contains all optional nodes between each pair of mandatory nodes. Including $|O|$ many nodes between two mandatory nodes requires $|O| + 1$ arcs. Doing this for all pairs of mandatory nodes results in a total number of $(|O| + 1)(|C| + 1)$ arcs used in the extreme tour. We denote this cardinality by K. Since each decision variable x_{ij} and u_{ij} must be available for each value of index k the complexity of the model is increased considerably.

Objective functions and constraints

$$\min \sum_{k=1}^{K} \sum_{i \in V} \sum_{j \in V} (ed_{ij} x_{ij}^k + fd_{ij} u_{ij}^k) \tag{4.3.6.1}$$

$$\min \sum_{k=1}^{K} \sum_{i \in V} \sum_{j \in V} g_{ij} x_{ij}^k \tag{4.3.6.2}$$

subject to:

$$\sum_{j \in C \cup O} x_{0j}^1 = 1 \tag{4.3.6.3}$$

$$\sum_{k=1}^{K} \sum_{j \in V \setminus \{i\}} x_{ji}^k = 1 \qquad \forall\, i \in C \cup \{0\} \tag{4.3.6.4}$$

$$\sum_{k=1}^{K} \sum_{j \in V \setminus \{i\}} x_{ij}^k = 1 \qquad \forall\, i \in C \cup \{0\} \tag{4.3.6.5}$$

$$\sum_{j \in V} x_{ji}^k = \sum_{j \in V} x_{ij}^{k+1} \qquad \forall\, i \in C \cup O,\ \forall\, k = 1, \cdots, K \tag{4.3.6.6}$$

$$\sum_{i \in V} \sum_{j \in V} x_{ij}^k \leqslant 1 \qquad \forall\, k = 1, \cdots, K \tag{4.3.6.7}$$

$$\sum_{j \in V} u_{ji}^k - \sum_{j \in V} u_{ij}^{k+1} = q_i \sum_{j \in V} x_{ji}^k \qquad \forall\, i \in C \cup O,\ \forall\, k = 1, \cdots, K \tag{4.3.6.8}$$

$$u_{ij}^k \leqslant M x_{ij}^k \qquad \forall\, i, j \in V,\ \forall\, k = 1, \cdots, K \tag{4.3.6.9}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall\, i, j \in V,\ \forall\, k = 1, \cdots, K \tag{4.3.6.10}$$

$$u_{ij}^k \geqslant 0 \qquad \forall i, j \in V,\ \forall\, k = 1, \cdots, K \tag{4.3.6.11}$$

The structure of the objective functions is the same as in all other approaches before, but the formulation accounts now also for the new index k associated with the decision variables x_{ij} and u_{ij}.

A notable change in the constraints is that they have to define the position index k in accordance with the sequence of arcs and load variables.
A tour must start in the hub node. Hence the decision variable $x_{0.}$, that defines the first arc in the solution, must be indexed with $k = 1$. This happens through equation (4.3.6.3).
Constraints (4.3.6.4) and(4.3.6.5) still make sure that each mandatory node is visited exactly once but guaranteeing now also that only one of the, for each node pair formally available, k arcs is used.
Constraints (4.3.6.6) define the correct sequence of decision variable x_{ij}^k with respect to k. If a node i is reached by an arc at position k, it must be left such that its successor in the route is reached by an arc at position $k + 1$. The hub node is excluded here because its outgoing arc is in position 1 (as already covered with (4.3.6.3)) and its ingoing arc is in the last position of the tour. So the values of k corresponding to the arc decision variables of the hub are not consecutive. Although we know that the position index of the hub's ingoing arc should be equal to the tour length we can not define this value through any constraint because, as already mentioned, we can not foresee the tour length.
Constraints (4.3.6.7) ensure that there is only one arc in each position k.
Equations (4.3.6.8) are the load flow constraints. The existence of index k at the load variables u_{ij} makes it now possible to define them in accordance with the arc sequence. So the difference between the load on an arc in position k and the load on the consecutive arc $k + 1$ must be equal to the demand of the node in between.
The inequalities corresponding to (4.3.6.9) force the load variable to be zero on unused arcs.
Constraints (4.3.6.10) and (4.3.6.11) define as usual the domains of decision variables.

A word on notation

Throughout the rest of this work we will name the approach of considering optional nodes always as 'Model 2'. So by 'Model 2' we do not mean the very first formulation attempt, that is not going to be mentioned or used in the following, anyway. When we want to refer to any specific formulation 'Model 2A', 'Model 2B' or 'Model 2C' we will also state it like that.

4.4 Model 3

4.4.1 Why multiple arcs?

The approach in this section presents a further way of how alternative paths between mandatory nodes can be modelled. While in Model 2 the multiple paths emerged from considering optional nodes explicitly in the optimisation problem, the idea for Model 3 is now to evaluate multiple paths ex-ante such that they appear already in the data set.
The additional alternative paths provide more possibilities for taking decisions and consequently expand the scope for optimisation. In modelling approach 1 a solution tour was defined only over the sequence of nodes (resulting from the used arcs) and the corresponding load flow. With Model 3 different solutions emerge now not only from a certain node sequence but also from the arc that is used between two consecutive nodes. So Model 3 offers in comparison to Model 1 more possible solutions. Although only a small fraction will survive the Pareto optimality condition, it can be expected that the final Pareto set is larger for Model 3 than for Model 1.

4.4.2 Classification: Bi-objective Travelling Salesman Problem on a multigraph

With regard to the problem class, Model 3 is as well as Model 1 a bi-objective TSP which means the task is to find Pareto optimal tours that start and end in the hub and visit every customer exactly once. Though, the existence of multiple arcs requires now dealing with a multigraph instead of a simple graph. Tadei et al. (2014) [33] consider a TSP with multiple paths in the context of City Logistics. The multiple paths between two locations are associated with different stochastic travel costs and the objective is to find an expected minimum closed tour. There are also studies on VRPs (that is, as we already know, a generalisation of the TSP) with alternative paths. Garaix et al. (2010) [19], for example, deal with an application to a system of Demand Responsive Transportation (or dial-a-ride problem), where planning upon customer requests is done with the goal of increased flexibility. So in general, formulating a TSP on a multigraph allows being more flexible in optimisation.

The multigraph for Model 3

The node set is the same as for the simple graph, only the arc set is different, namely larger. There is not an explicit condition on the number of arcs for each node pair, in particular the number of arcs may vary from one node pair to another. Actually also a simple graph can be classified as a multigraph because it is just a special case, where the number of arcs between any node pair is one. But this would make the new modelling approach equivalent to Model 1. Hence we assume that for at least one node pair there are at least two different arcs.

Like in the other two models distance and disturbance are the two type of costs associated with each arc. Although we allow any arbitrary number of arcs between any pair of nodes, it makes sense to take only those that are Pareto-optimal with respect to the two criteria distance and disturbance. We explain the reasoning for this by a small example whose illustration can be seen in Figure 4.5. Three different arcs are available between the two nodes, with associated distance and disturbance value represented by the first and second component of the arc-weight vector, respectively. The upper and the middle arc are Pareto optimal, because the upper one is compared to the middle one worse in distance but better in disturbance. The lower arc is dominated by the upper arc in both criteria. So the upper arc will always be chosen over the lower arc, no matter a solution should minimise emission or disturbance. Thus it can be dropped from the input data for Model 3. However, taking only the two Pareto-optimal paths is reasonable because dependent on whether a solution should be low in emission or low in disturbance the upper or middle arc is selected, respectively.

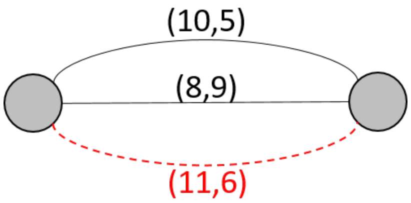


Figure 4.5: Multiple arcs are assumed to be Pareto optimal

A method to come up with Pareto optimal paths between two nodes is to solve a bi-objective SPP. When transforming Model 2 into Model 3 in the subsequent section we will present this in more detail. For the purposes of this section, i.e. for setting up Model 3, it is enough to have in mind that distance and disturbance data is assumed to be such that the multiple arcs between two nodes are Pareto optimal.

4.4.3 Bi-objective MIP formulation

Data

Model 3 needs the type of data introduced in section 4.1, i.e. a set of mandatory nodes V, a corresponding set of arcs A, a demand pattern q_i, emission parameters e, f and distance and disturbance data. But in contrast to Model 1 and Model 2 the set A contains now multiple arcs (i, j, k) from node i to j, where k ranges from 1 to K_{ij}. So $K_{i,j}$ tells us the

number of arcs between node i and j. It has to be indexed with i and j because depending on the node pair i, j a different number of arcs occurs. Since distance and disturbance data is arc-dependent it has to be also indexed with k. Demand pattern and emission parameters are not related to arcs, so they remain unchanged.
In order to ensure dealing with a multi-graph it is assumed that $K_{ij} \geqslant 2$ for at least one node pair i, j.
Data can be summarized as follows:

$\{0\}$...	fixed, single hub
C...	set of customer locations
$V = \{0\} \cup C$...	set of all locations
A...	set of arcs (i, j, k), $i, j \in V$ and $k = 1, \cdots, K_{ij}$
d_{ij}^k ...	travel distance when going from i to j using arc (i, j, k)
e...	emission when travelling one unit of distance while vehicle is empty
f...	emission per unit of load weight when travelling one unit of distance
q_i...	$\begin{cases} > 0 & \text{for } i \in C \\ = 0 & \text{for } i = 0 \end{cases}$
g_{ij}^k ...	disturbance when going from i to j using arc (i, j, k)

Decision variables

The type of decision variables for Model 3 is the same as for the other models, namely a binary decision variable identifying whether an arc is used or not and positive, real valued decision variable giving the load corresponding to an arc. Both are obviously arc-dependent, what entails that they are also indexed with k.

$$x_{ij}^k = \begin{cases} 1 & \text{if arc } (i, j, k) \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases}$$

u_{ij}^k load when travelling from i to j on arc (i, j, k)

Objective functions and constraints

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k=1}^{K_{ji}} (e d_{ij}^k x_{ij}^k + f d_{ij}^k u_{ij}^k) \tag{4.4.1}$$

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{k=1}^{K_{ij}} g_{ij}^k x_{ij}^k \tag{4.4.2}$$

subject to:

$$\sum_{k=1}^{K_{ji}} \sum_{j \in V \setminus \{i\}} x_{ji}^k = 1 \qquad \forall i \in V \tag{4.4.3}$$

$$\sum_{k=1}^{K_{ij}} \sum_{j \in V \setminus \{i\}} x_{ij}^k = 1 \qquad \forall i \in V \tag{4.4.4}$$

$$\sum_{k=1}^{K_{ji}} \sum_{j \in V} u_{ji}^k - \sum_{k=1}^{K_{ij}} \sum_{j \in V} u_{ij}^k = q_i \qquad \forall i \in C \tag{4.4.5}$$

$$u_{ij}^k \leqslant M x_{ij}^k \qquad \forall i, j \in V,\ \forall k = 1, \cdots, K_{ij} \tag{4.4.6}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall i, j \in V,\ \forall k = 1, \cdots, K_{ij} \tag{4.4.7}$$

$$u_{ij}^k \geqslant 0 \qquad \forall i, j \in V,\ \forall k = 1, \cdots, K_{ij} \tag{4.4.8}$$

The existence of multiple arcs entails also some modifications in the objective functions and constraints. So it is necessary to consider the additional index in (4.4.1) and (4.4.2) by summing over all k.

Constraints (4.4.3) and (4.4.4) still say that every time we arrive at a node we have to also leave it. In particular, taking the sum over k guarantees that for each pair of nodes only one of the multiple arcs can be used. Constraints (4.4.5) define again the load flow by reducing the ingoing load of a customer $i \in C$ exactly by its required demand q_i. Here the sum for index k is needed to ensure that the load is transported only via one arc. Constraints (4.4.6) prohibit positive load on unused arcs.

(4.4.7) and (4.4.8) specify as usual the domains of the decision variables.

4.5 Relation between the three models

In the preceding sections three different approaches to model the given problem scenario were presented. Each approach belonged to a different problem class - ranging from a bi-objective TSP over a bi-objective SPP with mandatory nodes to a bi-objective TSP on a multi-graph.

In this section we will show now that all three approaches can be formally related to each other.

4.5.1 Model 1 as a special case of Model 3

When setting up the formulation for Model 3 it has been assumed that for at least one pair of nodes there must be more than one arc to make sure that it does not reduce to Model 1. So the case of having only one arc between each pair of nodes was excluded for Model 3. Now we assume exactly this former 'forbidden' case and show in a rather trivial, short way that Model 1 is a special case of Model 3.

Looking at the MIP formulation of Model 3 the assumption of having only one arc for each pair of nodes means that $K_{ij} = 1$ for all mandatory nodes $i, j \in C \cup \{0\}$. So the upper bound for index k's range is equal to its lower bound, namely 1 what makes the index k to be always just equal to 1. There is actually no need for having this index at all, and it can be cancelled everywhere it appears in the MIP formulation of Model 3. So the summation over index k in (4.4.1)-(4.4.5) is not needed any more and it is not necessary to formulate constraints (4.4.6)-(4.4.8) for each index k.

In the end the MIP formulation of Model 3 proves to be equal to that of Model 1 when $K_{ij} = 1$ for all node pairs. This suggests that setting up Model 1 would not have been necessary at all, when it can be derived as a special case of Model 3 anyway. Nonetheless there are reasons why it was introduced individually. On the one hand it was a good starting point for developing the more complex models. On the other, in practice the available input data may not be that extensive with respect to multiple arcs because evaluating them for real-world problems poses high effort. So one may decide that having only one path between two locations is sufficient and apply therefore Model 1.

4.5.2 Transformation of Model 2 into Model 3

Now we want to transform Model 2 into Model 3. Therefore the alternative paths arising in Model 2 from deciding endogenously on the inclusion of optional nodes are pre-computed such that they can be used in form of multiple arcs for the multigraph of the bi-objective TSP in Model 3.

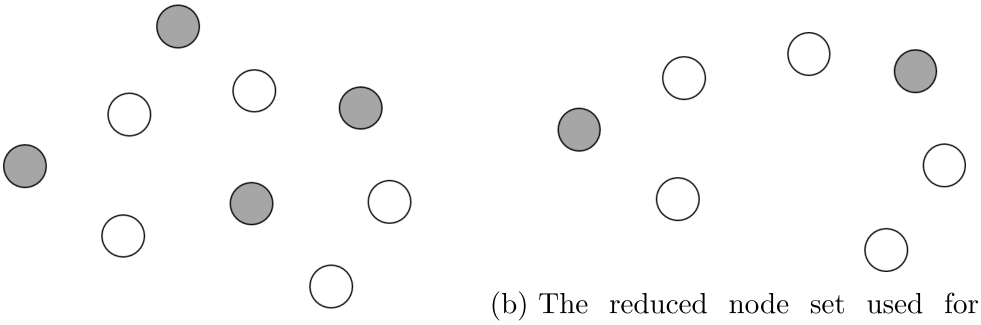
All-pair bi-objective Shortest Path Problem as transformation tool

The bi-objective SPP was already introduced when classifying Model 2 in section 4.3. Now it is applied in yet another context: to pre-compute the multiple arcs for Model 3. Therefore an all-pair bi-objective SPP on the set of mandatory nodes is solved, which actually means here solving a bi-objective SPP for each pair of unequal mandatory nodes where one node serves as source node and the other as target node. In addition to the two mandatory nodes the whole set of Model 2's optional nodes is added to the node set considered for each one-pair bi-objective SPP. Optimisation is again done with respect to the two decision criteria of emission and disturbance (we will see that finding the emission-minimising path between two mandatory nodes reduces to finding the distance-minimising path). The resulting Pareto optimal paths serve then as multiple arcs for Model 3 and the corresponding objective values as distance and disturbance data.

The whole transformation problem, i.e. the all-pair bi-objective SPP, involves solving a bi-objective SPP for each possible pairing of mandatory nodes, i.e. the one-pair bi-objective SPP, that is technically and formally the same for each pairing. Hence we investigate in the following only the one-pair bi-objective SPP.

MIP formulation

We assume that source node s and target node t are two unequal mandatory nodes from Model 2's data set. Together with the optional nodes they constitute the node set V' on which the bi-objective SPP is formulated, so $V' = \{s\} \cup \{t\} \cup O$. To complete the graph the set of arcs $A' = \{(i,j) : i, j \in V'\}$ connecting each pair of nodes i and $j \in V'$ is defined. Each arc (i,j) is associated with a weight-vector containing the two attributes distance d_{ij} and disturbance g_{ij} retrieved from data for Model 2. This new graph can be seen as a reduced version of the Model 2's underlying graph, as illustrated in Figure 4.6. [6]



(a) The original node set of Model 2

(b) The reduced node set used for bi-objective SPP

Figure 4.6: Relation between Model 2 and bi-objective SPP

As usual, binary decision variables x_{ij} are used to indicate whether an arc (i,j) is used, $x_{ij} = 1$, or not, $x_{ij} = 0$. In contrast to the optimisation problems in sections 4.2 to 4.4 there is now no need of emission parameters, demand data and load decision variables, what will be justified when discussing the objective functions.

Equipped with data and decision variables the objective functions and constraints can

---

[6] Special case: When there are only two mandatory nodes, i.e the hub and one customer, then the graph for the bi-objective SPP is the same as for the original problem.

be formulated as follows:

$$\min \sum_{i \in V'} \sum_{j \in V'} d_{ij} x_{ij} \tag{4.5.1}$$

$$\min \sum_{i \in V'} \sum_{j \in V'} g_{ij} x_{ij} \tag{4.5.2}$$

subject to

$$\sum_{j \in V' \setminus \{i\}} x_{ij} - \sum_{j \in V' \setminus \{i\}} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \neq s, t \\ -1 & \text{if } i = t \end{cases} \tag{4.5.3}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V' \tag{4.5.4}$$

Objective function (4.5.2) is essentially the same as (1.1.2), i.e. total disturbance is measured by accumulating the disturbance values of the used arcs. Objective function (4.5.1) relates to that of total emission from the preceding sections, but is now reduced to only measuring total distance. The load-dependent part of emission is not relevant here because only source and target node have strictly positive demand what induces the load to remain unchanged on any path between them. So in fact only the pure distance-part of the total emission function should be incorporated into the bi-objective SPP. However, we can also get rid of the emission parameter e, because it is constant and always multiplied linearly to the distance values d_{ij}. So we come up with objective function (4.5.1), measuring the length of a path.

The objective values of the obtained Pareto optimal paths from node i to j a can thus be directly employed as distance d_{ij} and disturbance g_{ij} for Model 3.[7]

Constraints (4.5.3) are the typical path constraints, where the left sum gives the number of ingoing arcs and the right sum the number of outgoing arcs for a certain node $i \in V'$. Depending on whether node i is the source or target node (i.e. a mandatory node) or as any of the remaining nodes (i.e. optional node) different conditions for the difference between these two sums are formulated:

- If the considered node i is the source node s, the there must be one arc more leaving it than arriving in it. This guarantees that the paths start in this node.
- If i is an optional node, the number of outgoing arcs must be equal to the number of ingoing arcs. So each time we arrive at an optional node we must also exit it.

---

[7] We could also leave the emission parameter e in the objective function and divide then the objective value by e in order to get the distance value used for Model 3. But we want to avoid this extra calculation step.

Important is that it must not be visited at all, so when there is no arc arriving at an optional node then there is also no arc leaving it.

- Finally, if i is the target node t, there must be one more arc arriving in it than leaving it, what forces a solution path to end in this node.

Constraints (4.5.4) define the decision variables x_{ij} to be binary.

Solution method: Martins' algorithm

To solve the bi-objective SPP formulated in the preceding section Martins' algorithm [29] is used. It belongs to the class of labelling algorithms that are very popular when dealing with a SPP. A well known labelling algorithm for single-objective SPPs is Dijkstra's Algorithm[11]. Martins' algorithm is its generalisation for multi-objective SPPs.
A requirement for their applicability is that the arc weights of the underlying graph are non-negative. This is fulfilled in the present framework, because distance and disturbance data is always non-negative.

Before we come to a detailed description of Martins' algorithm we want to state some basic characteristics of labelling algorithms.
In the context of graph theory labelling means that a so called label is assigned to an arc or a node. A label can be basically any number or vector of numbers. In fact, this has been already done throughout the work, for example when bi-criteria weight vectors (d_{ij}, g_{ij}) were assigned to an arc (arc labelling) or when the depot was chosen to be always node zero (node labelling).

For labelling algorithms node-labelling is done in the following way: Labels are assigned to each node i in order to store information about already evaluated sub-paths from source node s to the respective node i. The information refers to the decision criteria, like for example distance, time, cost, emission or disturbance. The outcome of a labelling algorithm is the optimal path (in the case of single-objective SPP) or are the Pareto optimal paths (in the case of multi-objective SPP) from source node s to any other node in the network, in particular to the target node t. This outcome depends on several features of the algorithm. One is the specified rule based on which a label is selected in each iteration. The information contained in this selected label is then propagated to all other labels, what may lead to creating new labels or updating already existing ones. The concept how labels are created or updated also influences the outcome.
For Martins' algorithm the selection rule is to take the lexicographically smallest label, what makes sure that the chosen label always corresponds to a Pareto-optimal path. For creating and updating labels a dominance test based on the concept of Pareto optimality

is applied.
Martins' algorithm uses, as well as some other labelling algorithms, two different sets of labels, namely permanent and temporary ones. Permanent labels are those that were already fixed and serve as final solutions to the problem. Temporary labels are those that are still available for further exploratory and changes. Depending on whether a so called label setting or label correcting method is implemented these two sets are manipulated in different ways. In a label setting algorithm one label is made permanent in each iteration, whereas in a label correcting algorithm all labels become permanent in the final iteration.

Algorithm 2 shows the detailed procedure of Martins' algorithm. Its basic structure is taken from [18].

Algorithm 2 Martins' algorithm

```
Require: $G = (V', A')$ and $(d_{ij}, g_{ij})$ for each $(i,j) \in A'$
Ensure: All Pareto-optimal paths from $s$ to all vertices $i \in V' \setminus \{s\}$
Notation:
    $lt_i \cdots$ entire set of temporary labels of node $i$
    $lp_i \cdots$ entire set of permanent labels of node $i$
    $l_i = (l_i^d, l_i^g, j, h) \cdots$ a label of node $i$, with
        $l_i^d$...distance-performance
        $l_i^d$...disturbance-performance
        $j$....the precedent node from which it was possible to label $i$
        $h$....the position of $j$'s permanent label based on which $l_i$ was calculated
Initialisation:
    $lp_i \longleftarrow \emptyset, \forall i \in V'$
    $lt_i \longleftarrow \emptyset, \forall i \in V' \setminus \{s\}$
    $lt_s \longleftarrow (0, 0, -, -)$
Iteration:
while $(\bigcup_{i \in V'} lt_i \neq \emptyset)$ do
    $l_q \longleftarrow$ min lex $\{\bigcup_{i \in V'} lt_i\}$
    $lt_q \longleftarrow lt_q \setminus \{l_q\};\quad lp_q \longleftarrow lp_q \cup \{l_q\}$
    $h = |lp_q|$
    for $(j \in V'\colon (q,j) \in A')$ do
        $l_j \longleftarrow (l_q^d + d_{qj}, l_q^g + g_{qj}, q, h)$
        if $(\nexists\, l_j^* \in \{lt_j \cup lp_j\}$: $l_j^{*d} < l_j^d$ and $l_j^{*g} \leqslant l_j^g$ or vice versa$)$ then
            $lt_j \longleftarrow lt_j \cup \{l_j\}$
        end if
        $lt_j \longleftarrow lt_j \setminus \{l_j^* \in lt_j\colon l_j^{*d} \geqslant l_j^d$ and $l_j^{*g} \geqslant l_j^g\}$
    end for
end while
```

The algorithm in words

Data and notation: The required input data comprises the two mandatory nodes s, t, the

optional nodes and the according arcs with associated arc-weight-vectors[8].
The set of temporary and permanent labels for each node $i \in V'$ is denoted by lt_i and lp_i, respectively. l_i refers to an arbitrary label of node i - no matter if it is temporary or permanent. The first two components l_i^d and l_i^g give the total distance and disturbance of the label-specific sub-path from s to i. j indicates the precedent node that comes right before node i in the sub-path. One of predecessor j's labels was the basis for propagating information to node i and creating label l_i. The position of this label in j's set of permanent labels is stored in h, that builds the fourth component of label l_i.

Initialisation: The algorithm starts with assigning the label $(0, 0)$ temporarily to the source node s. This does not deliver any new information, but it assures that in the first iteration the source node is used as a starting point. The set of temporary labels of the remaining nodes and the set of permanent labels of all nodes are initialised with the empty set.

Iteration: The while loop is executed as long as there are some temporary labels to explore (line 1). At the beginning of each iteration the minimum lexicographical label l_q of all temporary labels ($= \bigcup_{i \in V'} lt_i$) is selected to become permanent (line 2). This entails dropping it out of the set of node q's temporary labels lt_i and adding it to the set of permanent labels lp_q (line 3). It will thus not be investigated any more.
Since the new permanent label l_q is queued at the end of the set of permanent labels the cardinality of lp_q can be taken as l_q's position h (line 4), that is needed for labelling q's successors.
Node q is the last node of the just evaluated sub-path and serves as a starting point for the following procedure: A current label is computed for all successors of q (line 5). Successors of a certain node q are nodes that can be directly (using only one arc) reached from node q. Formally a successor j of q is identified by the existence of an arc from q to j, i.e. when $(q, j) \in A$ for $j \in V' \setminus \{q\}$. In our case of a complete graph all nodes $i \in V' \setminus \{q\}$ are successors of q.
The current label l_j of a successor j is computed as follows (line 6): the distance between q and j is added to the distance value l_q^d of q's just 'new born' permanent label l_q. The same is done for disturbance. This gives the total distance/disturbance when using the just evaluated subpath, for going from source node s to q, plus the direct way to j. Also q and h are stored into the successor's current label l_i. However this current label is only added to j's set of temporary labels if it is not dominated by any existing permanent or temporary label of node j (line 7 and 8), i.e. if there is no label l_j in lt_j or lp_j that is at least as good in both, distance and disturbance, and strictly better in either distance

---

[8] In case of a not complete graph the distance and disturbance value of the missing arcs could be set to ∞ or any very large number in order to make sure such arcs are not chosen.

or disturbance or both. It can happen that there are 'old' temporary labels that are consequently dominated by the new temporary label. They are deleted, because they would not lead to any Pareto-optimal solution any more (line 10).
So after each iteration there is one new permanent label - the minimal lexicographical one. At the same time the set of temporary labels is updated in case new non-dominated labels result from the new permanent label.
The algorithm stops when there are no more temporary labels left to explore.

Output: The output of Martins' algorithm is then the entire set of permanent labels - containing for each node at least one permanent label. The distance and disturbance performance of these labels represent the objective values of the Pareto optimal paths from the source node s to the respective nodes $i \in V' \setminus \{s\}$. To trace the corresponding Pareto optimal path back to s the third and fourth component of the permanent label is needed.
The procedure of Martins' algorithm makes sure that any sub-path of a Pareto optimal path is again Pareto optimal, what classifies it as a so called dynamic algorithm [10].
For the purpose of transforming Model 2 into Model 3, however, only the final permanent labels of the predefined target node t are needed.

Illustrating Martins' algorithm by means of an example:

To get some feeling for the just presented abstract description of Martins' algorithm, we look at the following example that is rather minimalistic but nevertheless sufficient to point out important steps:

The example consists of four nodes, where node 1 is assumed to be the source node and 4 the target node. Relating the example to the problem scenario of this work these two nodes would correspond to two mandatory nodes and the remaining two nodes would be optional nodes. Moreover the graph is assumed to be complete (in accordance with the general assumption that we made in section 3.1). The induced graph can be seen in Figure 4.7. The first value of the weight-vector shows distance and the second disturbance. It

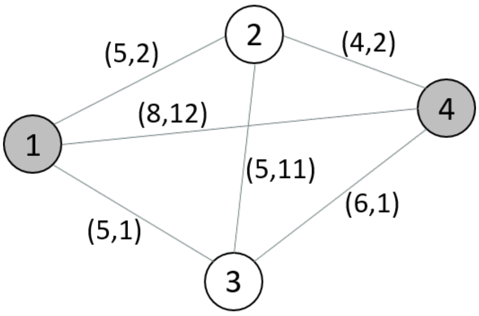


Figure 4.7: Illustrative example for Martins' algorithm

can be observed that disturbance is lower for arcs not going directly through the centre of the depicted network.
The objective is to find all Pareto optimal paths from $s = 1$ to target node 4. The steps according to Algorithm 2 are as follows:

- First the set of temporary labels of source node s is initialized with the label $(0, 0, -, -)$. Since this is the only existing temporary label it is also the minimal lexicographical one and thus transformed into a permanent label. Labelling the successors of s means in the first iteration that $l_i = (d_{1i}, g_{1i}, 1, 1)$ for $i = 2, 3, 4$. Since there are not yet any permanent or temporary labels no Pareto optimality condition must be checked. So the sets of permanent and temporary labels are:
  $lp_1 = \{(0, 0, -, -)\},\ lp_2 = \{\},\ lp_3 = \{\},\ lp_4 = \{\}$
  $lt_1 = \{\},\ lt_2 = \{(5, 2, 1, 1)\},\ lt_3 = \{(5, 1, 1, 1)\},\ lt_4 = \{(8, 12, 1, 1)\}$.

- Finding the minimal lexicographical label in iteration 2 can not be done by just looking for the smallest distance value, because then we would be indifferent between node 2 and 3 since they have both a distance of 5. So their disturbance values have to be compared what gives $l_3 = (5, 1, 1, 1)$ as the new permanent label. Labelling the successors of node 3 yields $l_1 = (10, 2, 3, 1), l_2 = (10, 13, 3, 1),\ l_4 = (11, 2, 3, 1)$. For node 1 the current label l_1 is dominated by the existing permanent label and for node 2 it is dominated by its existing temporary label. So only the current label of 4 passes the optimality test. Its existing temporary label is worse in disturbance but better in distance so it has not to be deleted. Thus, after iteration 2 the label sets are:
  $lp_1 = \{(0, 0, -, -)\},\ lp_2 = \{\},\ lp_3 = \{(5, 2, 1, 1)\},\ lp_4 = \{\}$
  $lt_1 = \{\},\ lt_2 = \{(5, 2, 1, 1)\},\ lt_3 = \{\},\ lt_4 = \{(8, 12, 1, 1), (11, 2, 3, 1)\}$

- In the third iteration the temporary label of node 2, $l_2 = (5, 2, 1, 1)$, is found to be lexicographically smallest. So it is moved from node 2's temporary to permanent set of labels. The current labels of 2's successors are then $l_1 = (10, 4, 2, 1), l_3 = (10, 13, 2, 1),\ l_4 = (9, 4, 2, 1)$. l_1 and l_3 are dominated by their existing permanent labels. l_4 has a worse distance value than the existing temporary label $(8, 12, 1, 1)$ but a better disturbance value. Comparing it to the second existing temporary label $(11, 2, 3, 1)$ we see that it has a worse disturbance value but a better distance value. So l_4 is not dominated and can be added to node 4's set of temporary labels where it dominates neither $(8, 12, 1, 1)$ nor $(11, 2, 3, 1)$. So after iteration 3 the label sets are:
  $lp_1 = \{(0, 0, -, -)\},\ lp_2 = \{(5, 2, 1, 1)\},\ lp_3 = \{(5, 1, 1, 1)\},\ lp_4 = \{\}$
  $lt_1 = \{\},\ lt_2 = \{\},\ lt_3 = \{\},\ lt_4 = \{(8, 12, 1, 1), (11, 2, 3, 1), (9, 4, 2, 1)\}$

- In the subsequent three iterations the three temporary labels of node 4 are made successively permanent. No new temporary labels are created throughout these iterations, so after iteration 6 there are no temporary labels left and as a final result the following sets of permanent labels are obtained:
  $lp_1 = \{(0, 0, -, -)\},\ lp_2 = \{(5, 2, 1, 1)\},\ lp_3 = \{(5, 1, 1, 1)\},$
  $lp_4 = \{((8, 12, 1, 1), (9, 4, 2, 1), (11, 2, 3, 1)\}.$

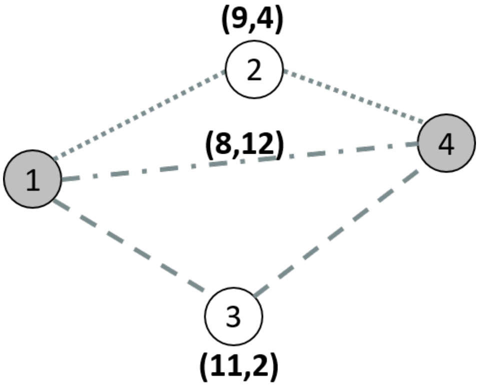


Figure 4.8: Martins' algorithm yields three Pareto optimal paths

Figure 4.8 shows the three obtained Pareto-optimal paths from source node 1 to target node 4.
Due to the symmetry in distance and disturbance data the algorithm yields the same result for the opposite direction, i.e. when taking node 4 as start node and node 1 as target node for the bi-objective SPP.

The solution of the illustrative example shows the idea behind optional nodes: paths with optional nodes are associated with lower disturbance but travelling a longer way. The path with lowest distance goes directly from source to target node without any optional nodes in between, but is the most disturbing one. Depending on whether the decision maker attach more importance to emission minimisation or disturbance minimisation he/she chooses the respective path.
Recalling that the bi-objective shortest path is used to transform Model 2 into Model 3, the three Pareto-optimal paths of the example would correspond to three different arcs connecting mandatory nodes 1 and 4.

The overall transformation procedure

The bi-objective SPP and Martins' algorithm have been demonstrated for only one pair of mandatory nodes. Anyway, the overall purpose of this section is to transform Model 2 to Model 3 and therefore Martins' algorithm has to be applied to each pair of mandatory nodes. Assuming that there are for example 10 mandatory nodes (1 hub and 9 customers) and that data is symmetric, it has to be solved 45 times.

Independent of which two mandatory nodes are considered, always the same entire set of optional nodes is used as intermediate nodes.

The final result of this procedure is then the graph of Model 3 with multiple arcs representing the Pareto-optimal solutions of Martins' algorithm and containing implicitly the optional nodes of Model 2.

5 Computational study

Randomly generated instances as well as instances derived from real-world data available for the Green City Hubs project were used to test the three different modelling approaches for the purpose of drawing conclusion about their characteristics, solution quality and CPU performance.

All three models as well as the ε-constraint method and Martins' algorithm were implemented in Python. Gurobi 6.0.3 was used to solve the MIPs occurring in the ε-constraint method.
If not stated otherwise, total emission was taken as f_1 and total disturbance as f_2 in the ε-constraint method.
The big M appearing in the constraints of the MIP formulations is always set to the vehicle's capacity.
The visualisations of the solution routes were created with Proute [41].

5.1 Experimental results on randomly generated instances

Experimental results on randomly generated data are presented in different ways aiming at different purposes: First the results of an arbitrarily chosen example instance demonstrate how the output of each model typically looks like and which effects the model-specific features have on the solutions.
Afterwards general statements about the performance of all three modelling approaches are made testing them on a variety of instances. In particular we compared how long it takes to compute the whole Pareto front with the different models, or if it is even possible within a predefined time limit of one hour per non-dominated point.
Also results on the limits of instance size for Model 1 and Model 3 will be presented.
And finally we compare the performance of the ε-constraint method for the two variations of taking total emission as f_1 or taking total disturbance as f_1.

5.1.1 Test instances

Information about hub, customers and optional nodes

To represent an urban area in an abstract, simplified way we assume that all relevant points (hub, customers, optional nodes) are located within a square that is integrated into a two dimensional, metric space.
The coordinates of the locations were created randomly while the number of customers and optional nodes is taken as a fixed input parameter. The demand pattern is also generated randomly but with the assumption that total demand amounts to around 90% of vehicle's capacity. Considering not 100% utilization rate is motivated by the situation in real-life applications, where vehicles are rarely fully loaded.

Vehicle information

We assume that the fixed vehicle is characterised by its capacity measured in kg and its emission parameters e and f. We could just take any strictly positive, real valued number for capacity and emission parameters, but to have some meaningful values we will use vehicle information from the Green City Hubs data set (more on it in section 5.2) as a benchmark.

Distance and disturbance data

To obtain distance data the Euclidean Distance for each pair of nodes is computed. In general this does not represent the situation in real life very well, because mostly the way between two logistically relevant locations is not just a direct, straight street. When real data is generated usually a single-objective SPP is solved to identify the minimal-distance path between two locations. However, for our purpose the Euclidean distance is sufficient and seen to be an appropriate choice because it is computationally cheap and because the resulting data will satisfy per definition the triangle inequality. In addition distance data will be symmetric, what would not be necessary because all models are formulated asymmetrically, but it also does not affect the validity of our results. The measurement unit of distance is assumed to be kilometres.

A more complex task is generating disturbance data. We already clarified that the disturbance of a path should be related to how many people are affected by the noise, vibration and risk resulting from using this path. Population density was identified to be a proper indicator for this. So in order to compute disturbance values we need first of all information on the distribution of density values across the city. Therefore a grid with equally sized, square boxes is laid over the city-square. Each grid box is then associated with a certain density value. Figure 5.1 shows an example for a 4×4-density grid, ranging

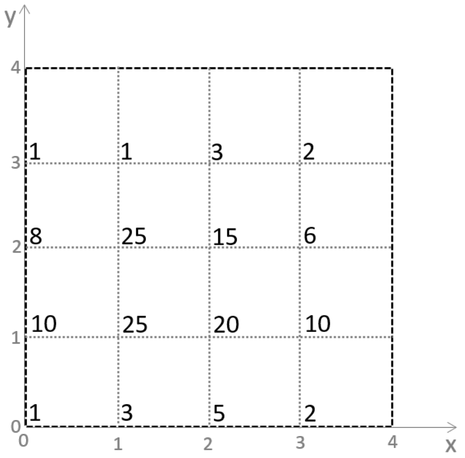


Figure 5.1: Densitygrid

on the x-axis as well as on the y-axis from 0 to 4. The bars of the grid are assumed to lie exactly on the integer x- and y-coordinates. So for example the grid box at the bottom left corresponds to the array $[0, 1] \times [0, 1]$.

It is assumed that the range of location coordinates and density grid coordinates fit together. So in case of a 4×4-density grid the locations of the instances are generated such that the smallest coordinate is not below 0 and the largest coordinate is not above 4.

But density values from the grid can not be taken directly as disturbance values, because it should make a difference if a path crosses a grid box to full extent or only by a small fraction. This makes it necessary to compute the disturbance value of an arc piecewise, in the sense that the arc is split according to the different grid boxes where it goes through and the density value from each passed box is taken only in proportion to the length of the part-arc. The resulting density values of the part-arcs are then summed up to get the density value of the complete arc.

We will explain this procedure in more detail by means of an example, that is based on the example density grid shown in Figure 5.1. The points of interest are assumed to be $A = (0.5, 1.2)$ and $B = (2.4, 3.0)$. Figure 5.2 shows the locations of the two points within the density grid. It can be seen that the arc between A and B lies in four different boxes, associated with density values $10, 25, 25$ and 15.

We start from point A. The first segment of the arc has length 0.68. This is multiplied by the associated density value 10 which results in a part-disturbance of **6.89**. The second part has a length of 0.47, so the density value of 25 contributes **11.75** to total disturbance. For the third part we obtain a disturbance of **22.58**, since the arc crosses the 25-density grid with a length of 0.90. The fourth and last arc segment causes a disturbance of **8.25**, arising from multiplying part-length 0.55 with density 15. Summing up the part-

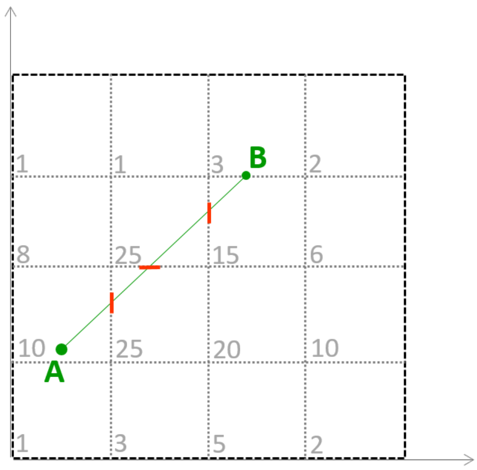


Figure 5.2: Calculating disturbance between two points

disturbances gives a total disturbance of **49.47**. This value is then taken as disturbance when traversing arc (A, B).
To get the entire disturbance input data matrix the calculation above is done for each pair of nodes contained in the instance.

The way how disturbance values are computed entails that disturbance data is on the one hand always non-negative (due to the non-negativity of distance data and density values) and on the other hand also symmetric. The asymmetric formulation of the three modelling approaches covers also the case of symmetric data, so having symmetric disturbance data does not cause a problem. To get asymmetric data the procedure explained above could be modified such that the disturbance for the opposite direction is reduced or increased by a certain number (in case of reduction one hast to be careful that disturbance does not become negative).

Although distance data satisfies the triangle inequality and contributes to disturbance, it will not induce disturbance data to satisfy triangle inequality as well. Let's take the example from before and consider a third point $C = (0.5, 3.2)$. The disturbance for (A, C) is 16.2 and for (C, B) it is 2.71. The disturbance when going from A to B over C accumulates then to 18.91 and is less than the disturbance when going directly from A to B, that is known from before to be 49.47.

This reflects exactly the idea behind optional nodes: If they lie in less disturbing areas, it could get beneficial to include them between two mandatory node, because they lead the route through areas that are less populated and hence less disturbing.
There is no proper measurement unit for disturbance. Thus we can not interpret one disturbance value alone, because we have no idea if a disturbance of, for example, 10 is good or bad. But we do not have to deal with only one disturbance value, neither in the

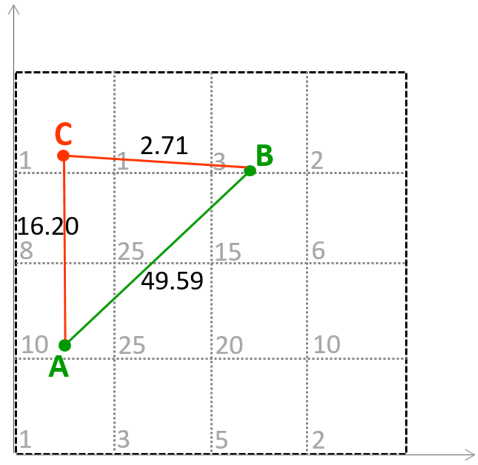


Figure 5.3: Triangle inequality does not hold for disturbance data

data - where in general the disturbance values differ from node pair to node pair - nor in the bi-objective optimisation problem - where several Pareto optimal solutions with different total disturbance arise. So we can always compare disturbance values with each other and say that an arc or a route is less or more disturbing than another one.

5.1.2 Compared performance of Model 2B and Model 2C

In section 4.3 of chapter 4 two different but valid formulations for incorporating optional nodes into the problem have been given by Model 2B and Model 2C, where the first one duplicates optional nodes and the latter one uses a position index to ensure correct load flow. Before we carry any other computational experiments we want to figure out which of the two formulations for Model 2 should be preferred. Since both formulations yield per definition the same Pareto set we make our choice dependant on the performance in terms of CPU time.

The two formulations were tested on instances of different size, ranging from 5 to 9 mandatory nodes and 5 to 18 optional nodes. Table 5.1 reports for each instance the size of the obtained Pareto front and the required CPU time. The abbreviation p.s. stands for 'partly solved' which means that the ε-constraint method stopped before finding the whole Pareto front. The abbreviation n.s. stands for 'no solution' which means not even one solution was obtained within the imposed time limit. CPU time is given in format hh:mm:ss. The notation #M and #O stands in the following always for number of mandatory nodes and number of optional nodes, respectively. The abbreviation PF stands for Pareto front.

#M	#O	size PF		CPU time	
		2B	2C	2B	2C
5	5	8	8	00:01:41	00:01:42
5	10	12	6 (p.s.)	00:45:34	04:36:12
5	15	5	5	00:08:28	02:22:58
6	6	14	14	00:31:15	01:38:27
6	12	20	20	00:12:20	01:31:36
6	18	8 (p.s.)	n.s.	3:18:28	n.s.
7	7	8	8	00:02:22	00:20:02
7	9	7	7	02:34:17	18:04:39
7	14	13 (p.s.)	n.s.	21:29:38	n.s.
8	5	6	6	00:00:43	00:25:52
8	8	1	1	00:00:59	00:18:05
8	9	19	19	05:39:14	34:29:06
8	16	n.s.	n.s.	n.s.	n.s.
9	6	12	12	00:02:38	01:31:27
9	9	9	9	00:34:05	08:08:51
9	13	n.s.	n.s.	n.s.	n.s.
9	18	n.s.	n.s.	n.s.	n.s.

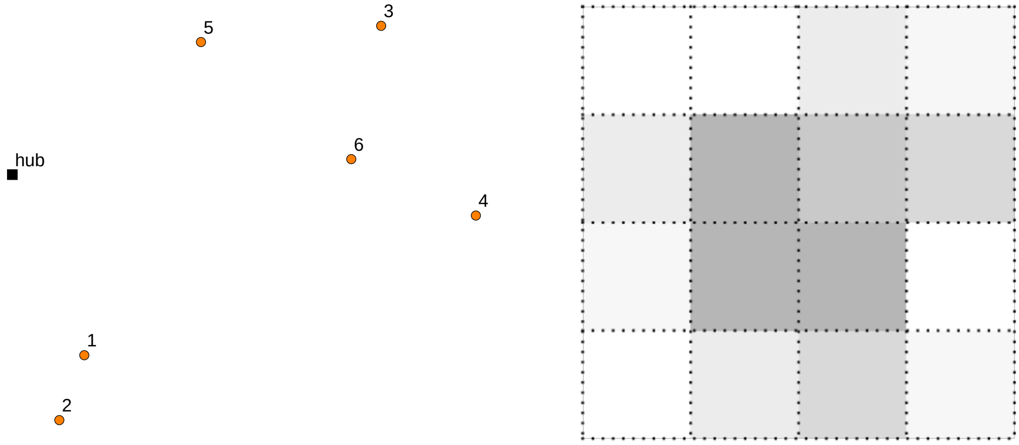
Table 5.1: Compared performance of Model 2B and Model 2C

Analysing the results reveals that Model 2B outperformed Model 2C for all instances. In case it was possible to identify the whole Pareto front with both formulations, Model 2C required considerably more CPU time than Model 2B. If we look for example at instance 9/9, we see that the CPU effort for finding the complete Pareto front with Model 2B was a bit more than half an hour, while for 2C it was more than 8 hours. Only for instance 5/5 the two formulations are competitive. So we conclude that formulation 2B, i.e. duplication of optional nodes, should be used for the subsequent computational experiments.
However also Model 2B showed already problems for the three instances 8/16, 9,/13, 9/18. Thus, when comparing Model 2B to Model 1 and Model 3, we will stick to rather small instances.

5.1.3 Solution characteristics of the three models illustrated by means of an example instance

In this section we want to gain some insight into the characteristics of the solutions obtained with the three different modelling approaches. Therefore the models are tested on an arbitrarily chosen instance that contains the following information:

The considered vehicle has a capacity of 2500 kg. The corresponding emission parameters are $e = 0.2$ and $f = 0.000012$.
The example instance contains six customers and a fixed hub. Their locations are depicted in Figure 5.4. The grey shaded boxes represent the density grid. The darker a box, the higher the density and the higher the disturbance for an arc going through it. We observe that there are two customers at bottom left a bit separated from the other four customers on top right. Moreover the area between them shows a rather high density. So it will be interesting how the different models deal with this situation. The demand pattern is as described in the following table:



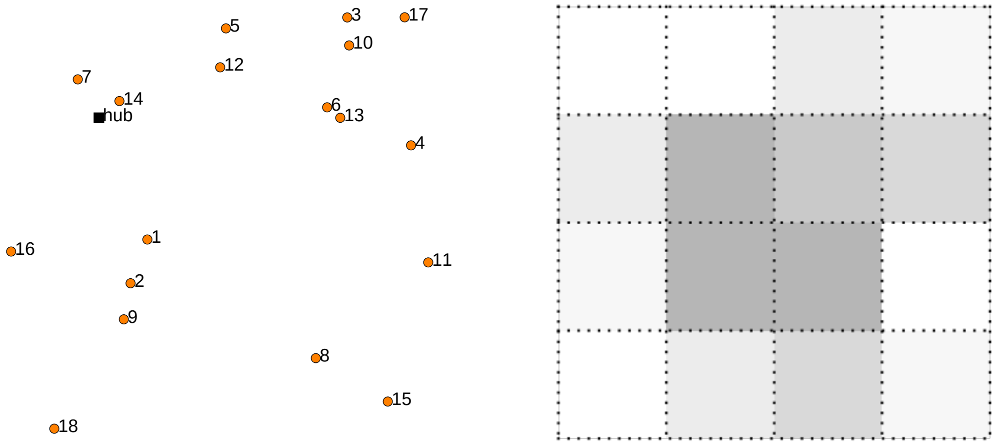
(a) Locations of hub and customers (b) Underlying density grid

Figure 5.4: Relevant locations for Model 1 and Model 3

node	0	1	2	3	4	5	6
demand [kg]	0	434	201	756	768	25	63

As assumed the hub has no demand while all customers have strictly positive demand, ranging in this example from 25 kg to 768 kg. So we see that the demand pattern of this instance is not that uniform what may have an effect when it comes to emission minimisation, because the amount of emitted gases depends on load weight.
Total demand sums up to 2247 kg, what makes around 90% of vehicle's capacity.

The example instance includes also 12 optional nodes that are spread randomly over the virtual city and that have no demand.



(a) All locations (b) Underlying density grid

Figure 5.5: Relevant locations for Model 2

Figure 5.5 shows them in addition with the hub and customers.
Based on the information given on node locations and the underlying density grid the distance and disturbance data is computed as described in section 5.1.1.
Depending on which model is tested the respective required data is extracted from the instance.

Model 1

For Model 1 we do not need any information on optional nodes. Only the mandatory nodes and the given, single arcs connecting them are relevant. Solving the problem with the ε-constraint method as described in algorithm 1 yields three Pareto-optimal solutions. Table 5.2 contains detailed information about them.

	emission	disturbance	route	load flow
1	10.007	80.968	[0, 5, 3, 4, 6, 1, 2, 0]	[2247, 2222, 1466, 698, 635, 201, 0]
2	10.013	77.895	[0, 1, 2, 4, 6, 3, 5, 0]	[2247, 1813, 1612, 844, 781, 25, 0]
3	11.247	76.003	[0, 3, 4, 6, 5, 1, 2, 0]	[2247, 1491, 723, 660, 635, 201, 0]

Table 5.2: Output of Model 1

Each line corresponds to one Pareto optimal solution. The numbers in the first column do not necessarily express any ranking. Primarily they should make later references to specific solutions easier. However the table lists the solutions in increasing order with respect to total emission and in decreasing order with respect to total disturbance (throughout this work the objective values are rounded to the third position after decimal point). Thus the first line shows the emission minimising solution and the last line the disturbance minimising solution. The solution in between balances both objectives. The fourth column shows the solution route identified by the sequence of nodes and in the last column the corresponding load vector is given.

Figure5.6 provides a visualisation of the associated trade-off curve. It reveals that the trade-off between solution 1 and 2 is better (in terms of disturbance) than between solution 2 and 3. Accepting only a bit more emission in solution 1, namely 0.006 in value, results in a significant improvement for total disturbance, namely 3.073 in value. When going from solution 2 to solution 3 we pay more in terms of emission (1.234) but get only a comparably small reduction (1.892) in disturbance. Computing the ratio $\frac{emission2-emission1}{disturbance1-disturbance2}$ gives 0.002 and $\frac{emission3-emission2}{disturbance2-disturbance3}$ gives 0.652. This means that when going from solution 1 to 2 one unit of disturbance costs only 0.002 units of emission, whereas, when going from solution 2 to 3, we have to give up 0.652 units of emission for getting one unit of disturbance.

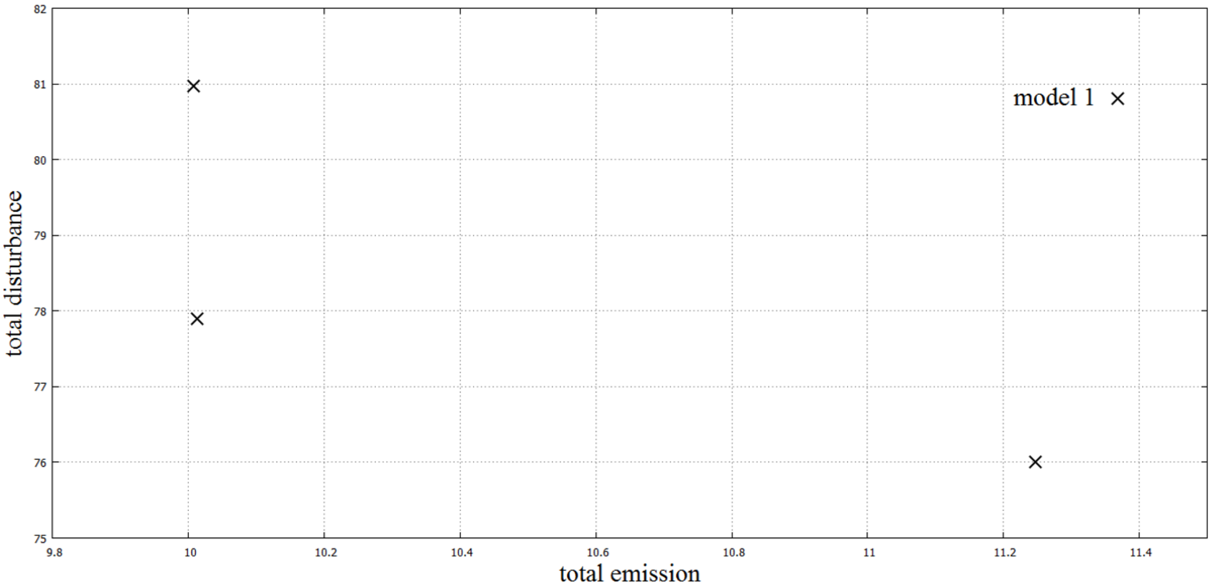


Figure 5.6: Pareto front of Model 1

A visualisation of each of the three obtained Pareto optimal routes is given in Figure 5.8.

The emission minimising solution is depicted on the left. The length of this tour is equal to the length of the distance minimising solution (that is shown in Figure 5.7), namely 46.94 km. This implies going directly through the area with high density values and having thus the worst solution in terms of disturbance. However it is not only distance that accounts for total emission but also load weight. This causes the emission minimising solution to go the opposite direction compared to the distance minimising solution. Customers 3 and 4 have largest demand with over 750 kg each. In the distance minimising solution these two customers are served towards the end, resulting in a total

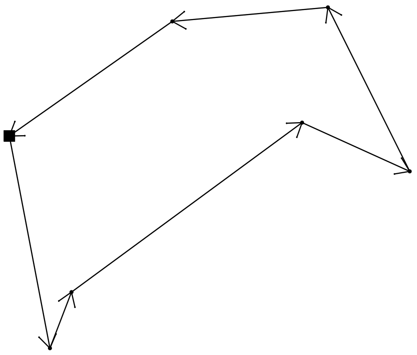
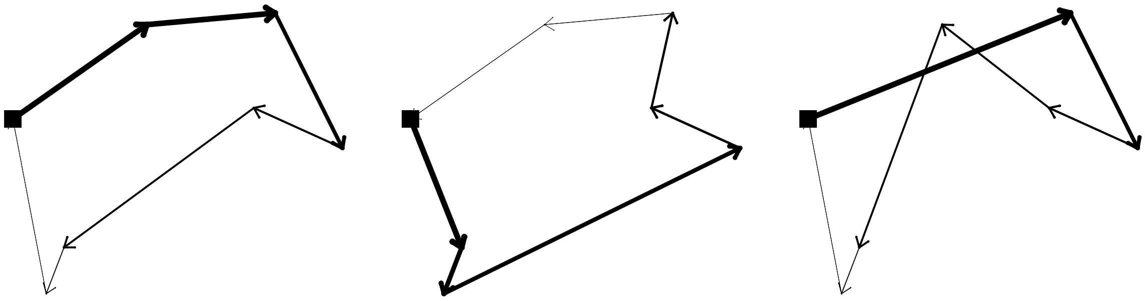


Figure 5.7: Distance minimising solution



(a) solution 1 (b) solution 2 (c) solution 3

Figure 5.8: Graphical representation of Pareto optimal solutions of Model 1

emission of 10.035, while in the emission minimising solution 3 and 4 are served rather at the beginning, resulting in lower total emission of 10.007.

Solution 2 balances the two objectives. So it also takes account of disturbance minimisation what induces the lower part of the emission minimising route to drift a bit away from the highly disturbing region in the middle.

Splitting up total emission of solution 1 and 2 reveals that the load-independent part of emission increased by 0.012 due to increase of 0.06 km in tour length. But at the same time the load-dependent part of total emission decreased by 0.006, the reason why emission increased in total only by 0.006. The better load-dependent emission can be traced back to the new sequence of nodes having a more favourable development of load weight, compared to solution 1.

Solution 3 provides the best possible route in terms of disturbance. The most notable thing is that arc-crossing happens twice during this route, most probably as a result of avoiding highly disturbing area.

Model 2: Adding optional nodes

To solve Model 2 also the optional nodes of the example instance are needed. Details on the resulting Pareto optimal solutions are given in Table 5.3. The load vectors are now not any more displayed, because we will not study them in more detail here. The only

	emission	disturbance	route
1	10.007	80.968	[0, 5, 3, 4, 6, 1, 2, 0]
2	10.013	77.895	[0, 1, 2, 4, 6, 3, 5, 0]
3	10.030	76.873	[0, 1, 2, 4, 6, 10, 3, 5, 0]
4	10.275	75.846	[0, 1, 2, 4, 6, 13, 3, 5, 0]
5	10.459	75.061	[0, 1, 2, 4, 6, 3, 5, 7, 0]
6	10.476	74.039	[0, 1, 2, 4, 6, 10, 3, 5, 7, 0]
7	10.721	73.012	[0, 1, 2, 4, 6, 13, 3, 5, 7, 0]
8	11.313	71.201	[0, 7, 5, 3, 4, 6, 12, 1, 2, 0]
9	11.655	70.762	[0, 5, 3, 4, 6, 12, 1, 2, 7, 0]
10	11.933	70.279	[0, 7, 5, 3, 4, 6, 12, 14, 1, 2, 0]
11	12.096	69.906	[0, 12, 6, 4, 3, 5, 7, 2, 1, 0]
12	12.110	68.652	[0, 1, 2, 7, 5, 3, 4, 6, 12, 14, 0]
13	12.150	68.293	[0, 5, 3, 4, 6, 12, 7, 2, 1, 0]
14	12.161	67.928	[0, 7, 5, 3, 4, 6, 12, 1, 2, 7, 0]
15	12.643	65.459	[0, 1, 2, 7, 5, 3, 4, 6, 7, 0]

Table 5.3: Output of Model 2

interesting thing about load flow in the solutions of Model 2 is that load does not change when passing an optional node. If we take, for example, the load vector corresponding to solution 3, we see that the load of 781 is the same on the arc leading to and leaving from optional node 10: $[2247, 1813, 1612, 844, 781, 781, 25, 0]$.

Very striking in the output table of Model 2 is that we have a lot more Pareto optimal solutions, what confirms the presumption that the availability of alternative paths via optional nodes provides more space for optimisation.
If we compare Table 5.3 to Table 5.2, we can see that the first two solutions of Model 2 are the same as the first two solutions of Model 1. For the emission minimising solution this is clear, because the existence of optional nodes does not provide any possibility to further decrease emission. However, also in the second solution, where disturbance plays already a role, no optional node is included. A possible reason for this is that the second solution provides a very good trade off between the two objectives as already explained when analysing the Pareto curve of Model 1. So obviously no solution containing an optional node dominates solution 2. For the remaining Pareto optimal solutions it is better to have a look at Figure 5.9 that visualises the Pareto front of Model 2 together with that of Model 1. It turns out that solution 3 of Model 1 is dominated by solution 4,5,6 and 7 of Model 2.

So the output table and the Pareto front demonstrate that considering alternative paths via optional nodes provides significant potential for improvement in total disturbance. To get an idea of how such routes look like, a selection of Pareto optimal solutions

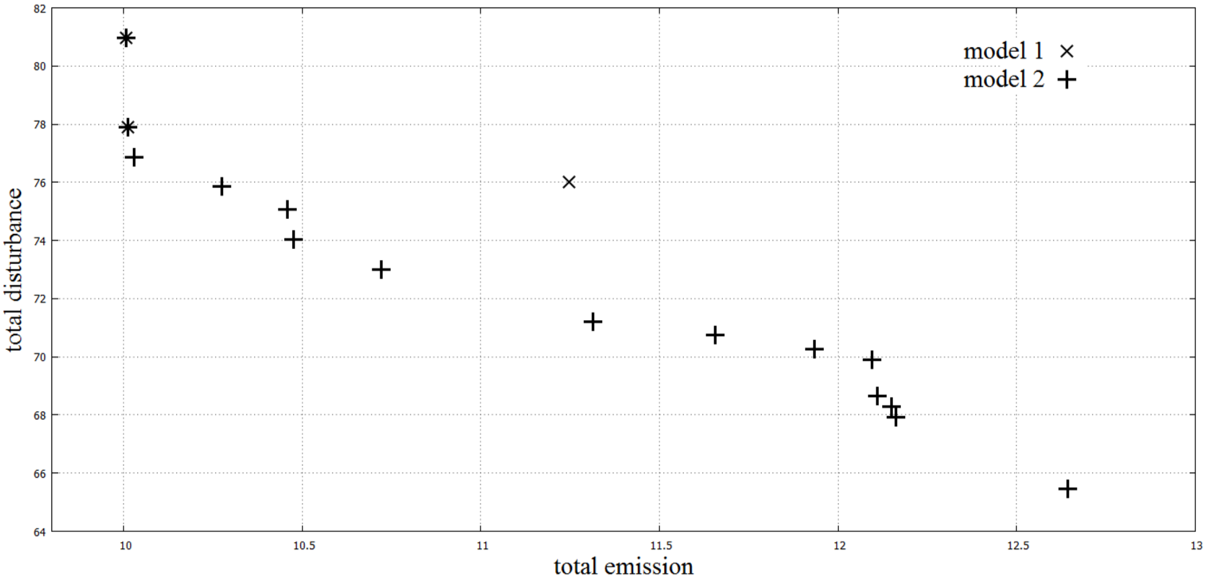
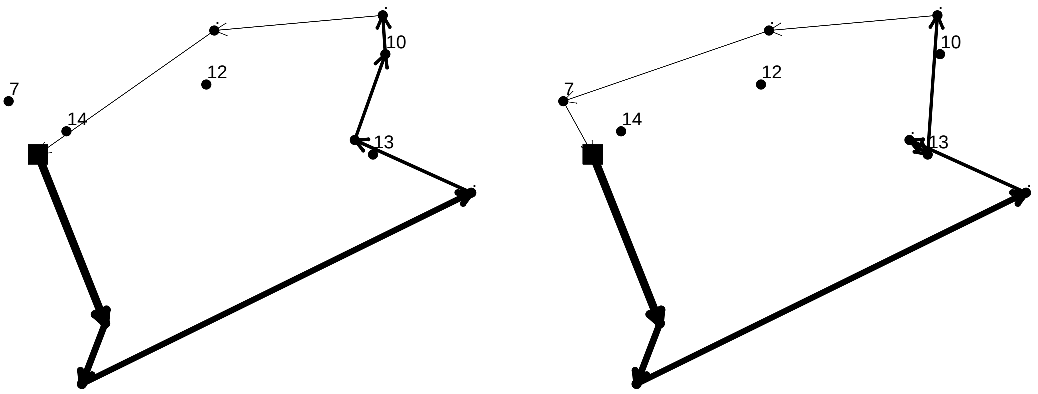


Figure 5.9: Pareto front of Model 2 dominates that of Model 1 partly

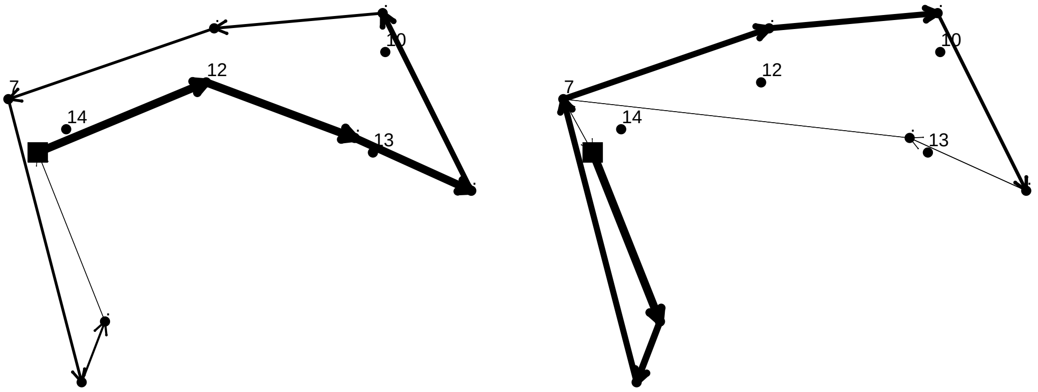
is represented graphically in Figure 5.10. Picture a) shows solution 3 where optional node 10 is included. Its total emission is 10.030 and its total disturbance 76.873. Compared to solution of Model 1 this is only a bit worse in disturbance but much better in emission, what can also be seen in the trade-off curve in Figure 5.9. Picture b) illustrates solution 7, which is one of the solutions that dominate solution 3 of Model 1. Optional nodes 7 and 13 are visited, where the latter leads to a crossing of the fourth and sixth solution arc. Picture c) depicts solution 11 that does not include the high disturbance arc from customer 2 to 4. Instead it leads around the highly disturbing area by going over optional node 12. Solution 15 in picture d) involves a double visit in optional node 7, that is once included rather at the beginning of the route and once just before going back to the hub. This solutions corresponds to the disturbance minimising solution of Model 2. A look at the Pareto front in Figure 5.9 reveals that compared to the second to last solution the disturbance minimising solution provides a significant decrease in total disturbance but at the same time also a significant increase in total emission.

When analysing Model 1 we have seen that an improvement in total disturbance when between solution 2 and 3 entailed crossings of arcs. The Pareto optimal solutions of Model 2 have mostly even lower disturbance than the arc-crossing disturbance minimising solution of Model 1, but only three of them contain arc crossings, namely solutions 4,7 and 13. So for the examined instance we conclude that such a behaviour is reduced when optional nodes are available.



(a) solution 3

(b) solution 7



(c) solution 11

(d) solution 15

Figure 5.10: Graphical representation of Pareto optimal solutions of Model 2

Model 3: Transforming optional nodes into multiple arcs

First we need to transform the optional nodes into multiple arcs, what is done by Martins' algorithm as described in Algorithm 2. The number of Pareto optimal paths obtained for each pair of mandatory nodes is displayed in the following matrix, where the row number indicates the source node and the column number the target node:

$$\begin{array}{c|ccccccc|}
 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\
0 & 1 & 1 & 2 & 4 & 13 & 2 & 6 \\
1 & 1 & 1 & 1 & 6 & 7 & 4 & 5 \\
2 & 2 & 1 & 1 & 6 & 6 & 4 & 6 \\
3 & 4 & 6 & 6 & 1 & 1 & 1 & 3 \\
4 & 13 & 7 & 6 & 1 & 1 & 3 & 1 \\
5 & 2 & 4 & 4 & 1 & 3 & 1 & 1 \\
6 & 6 & 5 & 6 & 3 & 1 & 1 & 1
\end{array}$$

Since distance data is symmetric also the matrix above is symmetric, because the Pareto optimal paths from a node i to a node j are the same as from j to i. For asymmetric data this does not hold true in general.
If the arc set of Model 2 was created through solving a SPP with respect to minimising distance, an entry of 1 in the above matrix means that only the single, minimum-distance arc is available. So no new arcs were created through Martins' algorithm, which means that none of the optional nodes provided a less disturbing way. For each node pair that has an matrix entry greater than 1 new Pareto optimal paths arise from including optional nodes. In total the number of arcs has increased from 49 in Model 1 to 173 for Model 3.

When we solve Model 3 based on this arc set we get the output shown in Table 5.4.

	emission	disturbance	route	used arc number
1	10.007	80.968	[0, 5, 3, 4, 6, 1, 2, 0]	[1, 1, 1, 1, 1, 1, 1]
2	10.013	77.895	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 1, 1, 1]
3	10.030	76.873	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 2, 1, 1]
4	10.275	75.846	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 3, 1, 1]
5	10.459	75.061	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 1, 1, 2]
6	10.476	74.039	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 2, 1, 2]
7	10.721	73.012	[0, 1, 2, 4, 6, 3, 5, 0]	[1, 1, 1, 1, 3, 1, 2]
8	11.313	71.201	[0, 5, 3, 4, 6, 1, 2, 0]	[2, 1, 1, 1, 2, 1, 1]
9	11.655	70.762	[0, 5, 3, 4, 6, 1, 2, 0]	[1, 1, 1, 1, 2, 1, 2]
10	11.933	70.279	[0, 5, 3, 4, 6, 1, 2, 0]	[2, 1, 1, 1, 3, 1, 1]
11	12.096	69.906	[0, 6, 4, 3, 5, 2, 1, 0]	[3, 1, 1, 1, 4, 1, 1]
12	12.110	68.652	[0, 1, 2, 5, 3, 4, 6, 0]	[1, 1, 4, 1, 1, 1, 4]
13	12.150	68.293	[0, 5, 3, 4, 6, 2, 1, 0]	[1, 1, 1, 1, 6, 1, 1]
14	12.161	67.928	[0, 5, 3, 4, 6, 1, 2, 0]	[2, 1, 1, 1, 2, 1, 2]
15	12.643	65.459	[0, 1, 2, 5, 3, 4, 6, 0]	[1, 1, 4, 1, 1, 1, 6]

Table 5.4: Output of Model 3

Not surprisingly it induces the same Pareto front as we got with Model 2, what confirms that Model 2 is equivalent to Model 3 when the optional nodes of Model 2 are transformed into multiple arcs for Model 3.
The optional nodes are not explicitly included in the output of Model 3, so the route vector in column 4 shows only the sequence of mandatory nodes. The vector in the last column contains information on which of the multiple arcs is used between any two consecutive mandatory nodes in the route. When arc number one is used this means the

minimal-distance arc (which is the one already available from data for Model 2) is used. Any other arc number means that one of the newly obtained arcs is used, representing a path that goes via one or more optional nodes. To identify the exact sequence of optional nodes between two mandatory nodes we need the output of Martins' algorithm. We do not report the whole output here, but demonstrate on the basis of solution 15 how information can be extracted from it.

In solution 15 arc number 4 was used to go from customer 2 to 5 and arc number 6 was used to go from customer 6 back to the hub. By Martins' algorithm the following final permanent labels have been evaluated between nodes 2 and 5 and nodes 6 and 0:

$2 \to 5$:	(13.435, 28.805, 2, 1)	$6 \to 0$:	(11.281, 34.934, 6, 1)
	(13.477, 28.703, 15, 1)		(11.567, 24.566, 14, 1)
	(15.379, 27.153, 14, 1)		(12.123, 10.954, 12, 1)
	(18.134, 21.648, 7, 1)		(12.191, 9.700, 14, 1)
			(14.554, 9.403, 7, 1)
			(14.859, 6.507, 7, 1)

When introducing Martins' algorithm we explained the components of the labels, so we know that using arc number 4 between customer 2 and customer 5 implies passing optional node 7 directly before customer 5. The entry of 1 means that the first permanent label at node 7 was used to label node 5, which is the label with minimum distance corresponding to the direct arc-connection to source node 2. Thus there is no further optional node in between. In the same way we get to know that using arc number 6 between customer 6 and the hub implies going via optional node 7. So in fact we have the route $(0, 1, 2, 7, 5, 3, 4, 6, 7, 0)$, which reads now the same as the route vector of solution 15 of Model 2.

5.1.4 Compared performance of the three models

By means of the example instance we presented model-specific characteristics of the solutions. In doing so we observed that the Pareto front is the same for Model 2 and 3. Moreover it turned out that the Pareto front of Model 2 and 3 is significantly larger than that of Model 1.

The question is now, if the observations can be generalized for any instance or if they were just a one-case incident.
So in order to make some general statements about the Pareto sets of the three models, a set of 20 instances was tested. They are also used to analyse the CPU effort for generating the Pareto front with the three modelling approaches.

The number of mandatory nodes varies from 5 to 14, and there is one instance with the same number of optional nodes and a second instance with twice as many optional nodes as mandatory nodes. The vehicle from the example instance in the preceding section is again used, i.e. capacity is 2500 kg and emission parameters are $e = 0.2$ and $f = 0.000012$. Computation of disturbance data is based on the 4×4 density grid used for the example instance.
The instances are processed in the same way as just before in section 5.1.3. This means Model 1 will be tested only on the set of mandatory nodes and corresponding fixed, single arcs. For testing Model 2 also the optional nodes are considered, that are then converted into multiple arcs for Model 3.

Table 5.5 shows the CPU time needed for computing the whole Pareto front, given in the form hh:mm:ss.ms.

#M	#O	CPU time		
		M1	M2	M3
5	5	00:00:00.13	00:00:28.84	00:00:00.23
5	10	00:00:00.14	02:53:27.17	00:00:00.41
6	6	00:00:00.16	02:11:56.41	00:00:03.04
6	12	00:00:00.16	19:13:10.59	00:00:00.34
7	7	00:00:00.29	00:07:30.09	00:00:01.44
7	14	00:00:00.16	21:54:31.89	00:00:00.67
8	8	00:00:00.21	00:02:45.67	00:00:01.06
8	16	00:00:00.16	–	00:00:01.05
9	9	00:00:00.16	00:09:54.07	00:00:00.60
9	18	00:00:01.20	–	00:00:15.92
10	10	00:00:00.20	02:39:56.60	00:00:02.15
10	20	00:00:00.83	–	00:00:34.48
11	11	00:00:01.84	–	00:00:21.18
11	22	00:00:00.61	–	00:00:50.86
12	12	00:00:00.85	11:21:18.07	00:00:12.22
12	24	00:00:01.21	–	00:00:48.62
13	13	00:00:00.92	–	00:00:09.05
13	26	00:00:01.01	–	00:00:36.88
14	14	00:00:03.13	–	00:00:08.41
14	28	00:00:01.22	–	00:00:30.19

Table 5.5: Compared CPU time

The stated CPU time for Model 3 includes also the time needed for pre-computing the multiple arcs with Martins' algorithm. For the larger instances with a mandatory/optional node number of 11/22, 12/24, 13/26 and 14/28 this was between one and three seconds. For all other instances the pre-computing part took less than 0.5 seconds.
A bar means that it was not possible to find all Pareto optimal solutions within the predefined time limit of one hour per Pareto optimal solution. We observe that this happened mainly when solving the instances with Model 2, in particular for those with twice as many optional nodes as mandatory nodes. The smallest instances could be solved exactly with Model 2 but requiring already high CPU effort. With Model 1 and 3 the ε-constraint method yielded for all instances the whole and exact Pareto front in less than one minute, for smaller instances it even took only a few seconds.

#M	#O	Size of PF			First point on PF	Last point on PF		#dom.
		M1	M2	M3	M123	M1	M23	
5	5	1	7	7	(5.765 , 55.294)	(5.765 , 55.294)	(7.166 , 33.526)	–
5	10	2	10	10	(7.677 , 54.342)	(8.055 , 51.280)	(9.470 , 36.135)	1
6	6	2	28	28	(5.108 , 95.852)	(5.969 , 90.198)	(9.151 , 43.108)	1
6	12	1	3	3	(7.612 , 47.648)	(7.612 , 47.648)	(9.388 , 34.422)	–
7	7	4	13	13	(6.194 , 88.246)	(6.834 , 85.278)	(10.362 , 52.600)	2
7	14	2	6	6	(6.143 , 67.04)	(6.367 , 54.897)	(7.889 , 50.37)	–
8	8	2	8	8	(6.897 , 60.019)	(7.064 , 59.054)	(9.158 , 44.011)	1
8	16	2	–	9	(7.921 , 59.746)	(8.550 , 57.737)	(9.001 , 56.705)	–
9	9	1	6	6	(7.720 , 49.236)	(7.720 , 49.236)	(8.354 , 46.613)	–
9	18	9	–	39	(7.716 , 73.583)	(8.673 , 56.643)	(8.562 , 40.037)	4
10	10	2	11	11	(6.067 , 78.676)	(6.203 , 77.92)	(9.205 , 70.911)	–
10	20	4	–	42	(7.365 , 78.816)	(7.481 , 75.658)	(10.705 , 55.033)	3
11	11	6	–	20	(6.981 , 69.246)	(7.476 , 49.864)	(8.649 , 40.741)	5
11	22	3	–	54	(8.199 , 65.246)	(9.310 , 58.023)	(13.506 , 40.522)	2
12	12	3	20	20	(7.926 , 71.258)	(8.247 , 63.987)	(11.351 , 55.155)	–
12	24	5	–	54	(8.809 , 76.38)	(9.291 , 68.014)	(9.907 , 59.728)	4
13	13	3	–	17	(9.297 , 75.194)	(10.022 , 62.133)	(10.149 , 60.271)	–
13	26	3	–	33	(7.819 , 88.264)	(8.091 , 80.569)	(11.181 , 66.57)	–
14	14	7	–	13	(7.955 , 75.637)	(9.221 , 62.922)	(9.938 , 56.476)	5
14	28	3	–	28	(7.559 , 69.057)	(7.674 , 66.674)	(8.944 , 42.919)	1

Table 5.6: Compared Pareto front

Table 5.6 reports for each instance and each model the size of the obtained Pareto

front as well as the first and last point on the Pareto front, where the first component refers to total emission and the second to total disturbance. For instances that could be solved exactly with all three models we observe that the Pareto fronts of Model 2 and 3 have same size. A detailed comparison of the output of the ε-constraint method (that is not given here) reveals that they do not only have same size but are identical, as we expected already after studying the example instance in the preceding section.
In addition we supposed that Model 2 and 3 yield more Pareto optimal solutions than Model 1, what is also confirmed for the larger set of instances. The first point of the Pareto front is the same across all three models, since it refers to the emission minimising solution and the alternative, disturbance-reducing paths in Model 2 and 3 do not have any effect on it. However, a comparison of the last points on the Pareto front shows that that they do have an effect when it comes to disturbance minimisation. In fact, for all instances total disturbance of the last non-dominated point belonging to Model 2/3 could be reduced compared to Model 1. But except for instance 9/18 this happened always at the cost of total emission.
The last column in Table 5.6 gives further information about the relation between the Pareto fronts of Model 1 and Model 2/3. It shows the number of Pareto optimal solutions of Model 1 that are dominated by the Pareto front of Model 2/3. A bar means that none of Model 1's solutions is dominated. We can see that for around half of the instances the Pareto front of Model 2/3 dominates that of Model 1 at least partly. So Model 2/3 does not only provide a more extensive solution set but also a better trade-off curve.

5.1.5 Limits on the optimality of the ε-constraint method

The instances used for simultaneous comparison of all three models were quite small such that some of them could also be solved for Model 2 in reasonable time. Table 5.5 shows that the CPU time needed to solve these small instances exactly with Model 1 and Model 3 was never beyond one hour. So we want to increase the instance size for testing Model 1 and Model 3.

To figure out how far we can indeed go we created 20 instances that comprise mandatory nodes in the range from 35 to 54. For Model 3 we added constantly 35 optional nodes. We did not vary this number because we assume that when 35 optional nodes make computations already difficult than an even larger number of optional nodes would make it even worse. Table 5.7 reports the size of the Pareto front and the corresponding required CPU time, in form hh:mm:ss. The CPU time for Model 3 includes again the time needed for computing the multiple arcs with Martins' algorithm, that was between 40 and 90 seconds. Column 2 shows how many arcs arc included in the optimisation problem of Model 1 and Model 3, respectively, in order to get some idea of the complexity of the problem. A bar means again that it was not possible to find the whole Pareto front within the predefined time limit of one hour per non-dominated point.

#M	#O	Size of PF		#arcs		CPU time	
m1	m3	m1	m3	m1	m3	m1	m3
35	35	1190	9820	13	96	00:03:31	00:33:53
36	35	1260	6764	10	46	00:02:38	00:20:25
37	35	1332	7678	18	54	00:05:35	01:01:02
38	35	1406	8450	44	46	04:28:00	01:51:58
39	35	1482	7107	25	75	00:18:04	00:32:32
40	35	1560	10160	30	50	01:17:29	04:01:43
41	35	1640	12878	14	46	00:20:20	05:16:44
42	35	1722	10670	15	71	00:11:37	01:52:57
43	35	1806	11577	85	171	02:55:03	14:01:52
44	35	1892	11603	14	47	00:11:48	02:26:51
45	35	1980	–	79	–	19:26:22	–
46	35	2070	12182	33	87	01:03:56	17:06:295
47	35	–	–	–	–	–	–
48	35	2256	15479	36	47	05:34:30	18:16:60
49	35	2352	13711	35	106	01:41:35	06:01:23
50	35	–	–	–	–	–	–
51	35	–	–	–	–	–	–
52	35	–	–	–	–	–	–
53	35	–	–	–	–	–	–
54	35	–	–	–	–	–	–

Table 5.7: Limits on the optimality of the ε-constraint method

We see that, except for the instance with 38 mandatory nodes, Model 1 is always less expensive in CPU time than Model 3. This can be traced back to the more complex formulation of Model 3, that includes a larger data and constraint set resulting from the considerably larger number of arcs.

When the instance contained more than 50 mandatory nodes the ε-constraint method failed for both models. But although most of the other instances could be solved exactly it is not desirable to wait more than 1 hour for the whole Pareto front. Thus, in general exact methods can be considered to be inefficient in solving large instances, what calls for heuristic solution methods. This work does not deal with developing and applying heuristics. However it can be kept as a future research topic.

5.1.6 Compared performance of the ε-constraint method once with total emission as f_1 and once with total disturbance as f_1

When solving a bi-objective optimisation problem with the ε-constraint method it has to be decided which of the two objectives is taken as f_1. In the beginning of chapter 5 we assumed that total emission is taken as f_1 if not stated otherwise. But now we want come to the 'otherwise'. Having total emission as f_1 means that with each iteration of the ε-constraint method the obtained solution became better in terms of disturbance but worse in terms of emission. When we set f_1 to the objective function of total disturbance then it is the other way round: the ε-constraint method yields first the disturbance minimising solution, gradually deteriorates total disturbance while improving total emission, until it arrives at the emission minimising solution.

When taking total disturbance as f_1 it may become necessary to adjust the increment parameter ε because the differences between the objective values of total emission are eventually smaller or larger than the differences between the objective values of total disturbance. With the right choice of ε both versions give the same set of Pareto optimal solutions. The question is then, if there are significant differences in terms of CPU time dependent on whether which of the two objectives is taken as f_1. Therefore 25 instances with 30 to 60 mandatory nodes and 40 optional nodes (only relevant for Model 3) were tested, once with total emission as f_1 and once with total disturbance as f_1. Table 5.8 lists for each test instance the number of considered mandatory nodes, the number of optional nodes, and for Model 1 as well as for Model 3 the size of the obtained Pareto front and the required CPU time (in format hh:mm:ss), once with total emission (abbreviated here as e.) as f_1 and once with total disturbance (abbreviated here as d.) as f_1. p.s. and n.s. stands again for 'partly solved' and 'no solution', respectively.

#M	#O	M1				M3			
		Size of PF		CPU time		Size of PF		CPU time	
		f_1=e.	f_1=d.	f_1=e.	f_1=d.	f_1=e.	f_1=d.	f_1=e.	f_1=d.
30	40	16	16	00:02:21	00:07:21	73	p.s.	00:46:13	p.s.
31	40	26	26	00:23:44	00:55:21	83	p.s.	04:11:50	p.s.
32	40	24	24	00:10:53	00:44:48	94	p.s.	01:46:16	p.s.
33	40	21	21	00:04:23	00:17:37	106	p.s.	01:34:06	p.s.
34	40	11	11	00:03:20	00:08:03	54	p.s.	00:26:11	p.s.
35	40	21	21	00:18:04	00:41:52	60	p.s.	07:29:26	p.s.
36	40	21	21	00:38:28	02:20:45	25	p.s.	02:53:45	p.s.
37	40	5	5	00:01:10	00:07:47	21	p.s.	00:14:23	p.s.
38	40	23	23	00:20:57	00:30:36	130	p.s.	03:22:06	p.s.
39	40	53	p.s.	00:13:46	p.s.	154	p.s.	03:01:50	p.s.
40	40	38	38	00:25:56	01:39:42	71	p.s.	01:29:35	p.s.
41	40	42	p.s.	00:16:31	p.s.	182	p.s.	03:53:25	p.s.
42	40	24	p.s.	01:04:23	p.s.	168	p.s.	16:00:58	p.s.
43	40	48	p.s.	06:05:43	p.s.	n.s.	p.s.	n.s.	p.s.
44	40	21	21	00:06:44	00:25:07	38	p.s.	00:46:59	p.s.
45	40	31	31	00:40:20	02:33:34	125	p.s.	11:36:41	p.s.
46	40	36	36	00:18:41	01:18:34	83	p.s.	01:55:08	p.s.
47	40	38	p.s.	02:27:39	p.s.	45	p.s.	16:28:12	p.s.
48	40	30	p.s.	05:15:16	p.s.	n.s.	p.s.	n.s.	p.s.
49	40	43	43	03:03:46	05:25:00	n.s.	p.s.	n.s.	p.s.
50	40	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.
51	40	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.
52	40	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.	n.s.	p.s.
53	40	p.s.	p.s.	p.s.	p.s.	n.s.	n.s.	n.s.	n.s.
54	40	112	p.s.	13:38:33	p.s.	p.s.	p.s.	p.s.	p.s.

Table 5.8: Compared performance of ε-constraint method, taking once emission as f_1 and once disturbance as f_1

We observe that in case both versions yielded the whole Pareto front the version with total emission as f_1 was always faster.

For Model 3 none of the tested instances could be solved exactly when we started the ε-constraint method with minimising disturbance, while when we started with minimising emission the instances up to 47 mandatory nodes did not cause any problem (except for the 43/40-instance).

An interesting fact is that for the instances where none of the two versions succeeded completely, at least parts of the Pareto front could be found when disturbance is taken as f_1, while with emission as f_1 not even a single solution was detected. This suggests that it is more difficult to find solutions which put more importance on emission minimisation. Figure 5.11 supports this conjecture. It shows the accumulated CPU time along the Pareto front when solving the instance with 40 nodes, for both versions of the ε-constraint method, based on modelling approach 1. When taking total emission as f_1 the CPU effort for finding the first solutions on the Pareto front (rather low in emission) is higher than for the last part of the Pareto front (rather low in disturbance). When total disturbance is taken as f_1 it is the other way round, so solutions towards the end of the Pareto front (rather low emission) are obviously harder to find than those at the beginning (rather low disturbance).

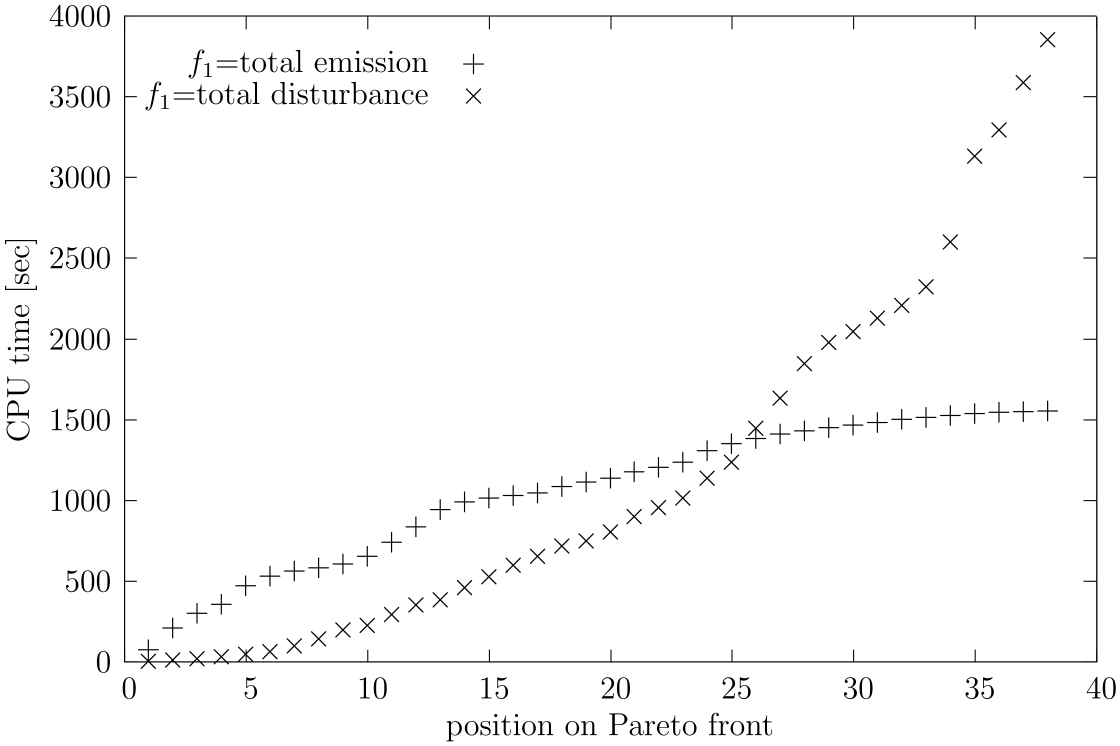


Figure 5.11: Development of CPU time along Pareto front

A possible reasoning for this effect is that the objective function of total emission comprises both decision variables, while through that of total disturbance only the binary decision variable has to be fixed. For Model 3 a similar picture would be obtained.

The observations are relevant for two reasons: First one can apply the version that suits the interests of the decision maker better, i.e. if he/she is more interested in solutions with rather low disturbance the ε-constraint method should be run with total disturbance as f_1. If the whole, exact Pareto front should be found, total emission should be taken as

f_1. Secondly the result of this section can be incorporated when developing heuristics for solving Model 1 and Model 3, because heuristics should be designed in a way such that the harder part of emission minimisation is tackled.

5.2 Real world case study

The randomly generated data created in the preceding section represented a rather abstract framework. To apply our models to a more realistic setting we use the data available from the Green City Hubs project. First the solutions obtained with Model 1 will be compared to a tour that was carried out in reality. Testing Model 2 and Model 3 on Green City Hubs data is not very meaningful because it does not include any information for optional nodes. Anyway, we make an attempt by taking some of the available customers as optional nodes.

5.2.1 Test instances

All instances used for computational experiments in this part are derived from a data set available from the Green City Hubs project. The whole data set can be found in the Appendix and some words on it will be given now:

Hub and customer information

In the context of the Green City Hubs project it has to be decided where to open hubs. In the optimisation problem of this work it is assumed that this decision has already been taken, so only a single, fixed hub is considered. Since the Green City Hubs data set contained at the time these computational experiments were carried out only information for one hub, it is always this hub that is considered in the instances. The information on price and space for the hub is not relevant for our optimisation problem - we only need its location.

The customers are in the context of the Green City Hubs project institutions that have demand for medical products, like for example pharmacies or hospitals. The original data set contains mainly customers located in Vienna but also a few in lower Austria (e.g. one in St. Pölten, a city about 55 km from Vienna). Including them would mean that the tour goes through sparsely populated areas, where disturbance does not play a major role and what makes them rather uninteresting when optimising routes with respect to disturbance. Beside this, disturbance data was generated only for Vienna. So we simply ignore the customers outside of Vienna for the computational experiments, what leaves us with 45 customers. Figure 5.12 shows their distribution across Vienna.

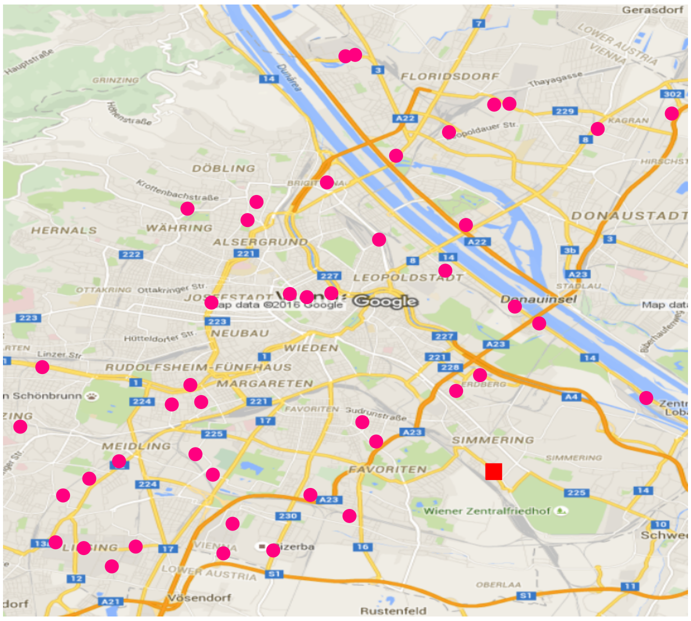


Figure 5.12: Location of hub and customers

As already mentioned, demand refers in the Green City Hubs project to medical products and is measured in kg to be compatible with vehicles' capacities. A demand pattern for a four-day horizon is provided in the data set. Every customer shows at least once within the four days a positive demand, some big customers (e.g. hospitals) even every day. Since our optimisation problem considers only one time period the instances will correspond to only one specific day.

Vehicle information

The emission values for six different vehicles have been evaluated. They range from diesel, hybrid and battery powered. The 'more alternative fuelled' the vehicle is, the smaller it is, what is reflected in lower capacity. Moreover alternative fuelled vehicles are associated with lower emission parameters.

For the overall optimisation problem considered in the context of the Green City Hubs project it has to be decided which vehicle to select/buy, to which hub they are assigned and also which customers to deliver by which vehicle. Since a solution to the optimisation problem on hand comprises only one tour there is also only one vehicle needed that is selected before optimising. Different instances can then be created by altering the type of vehicle.

Distance and disturbance data

The original data set contains information on travel times, distances and disturbances between the locations. In the course of this work only distance and disturbance data is used for optimisation so data on travel times is not needed.

Distance data was generated by solving a SPP between any two relevant locations with the objective of minimising distance.

The concept for generating disturbance data is the same as for randomly generated instances, so the disturbance of a path is identified via a certain population density region and proportional length. However, the more complex nature of a real city and road network complicates the calculations considerably. A density grid for Vienna was established in the following way:
The city of Vienna is administratively divided into 250 subdistricts - what is just a finer partition than the better known 23 districts. The Vienna government provides on-line data sets that contain information about the number of inhabitants for each subdistrict [42] as well as its area in square metres [43]. Out of these two data values a population density value for each subdistrict was calculated, that stands for the population per square metre.
Figure 5.13 shows a graphical representation of these values across Vienna.

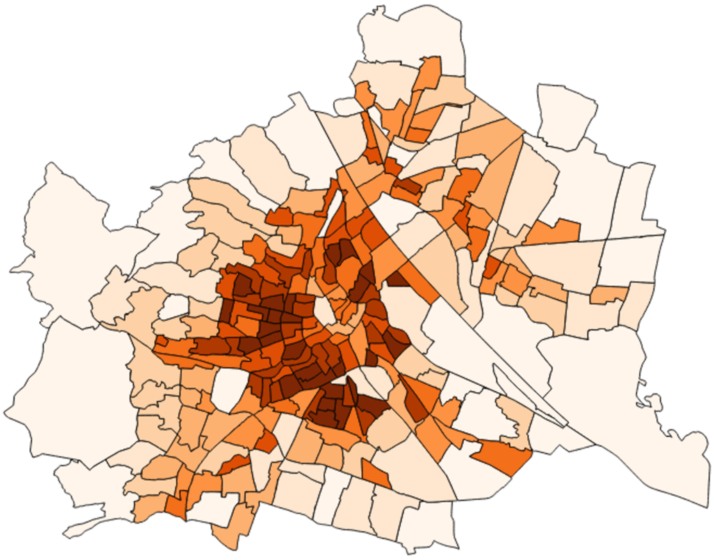


Figure 5.13: Population density of Vienna's subdistricts

It was created with QGIS - an open source Geographic Information System - and guided by the approach given in [44]. The darker the colour of the subdistrict is the denser it is populated. So - not surprisingly - subdistricts in outlying areas have obviously a lower density than those in the center. We can see that also a few subdistricts in the center seem to be rather low populated. This may not be in accordance with the reality, because there may be large areas within those subdistricts, that are not populated at all, like for example rivers or parks. But in general the obtained distribution of density values

can be assumed to be roughly consistent with reality.
Based on this density grid the disturbance values were calculated, taking the density values in proportion of the part-path length that goes through the respective subdistrict.

In contrast to randomly generated data, distance data as well as disturbance data for Vienna is asymmetric, due to asymmetries in the road network. But since all models are formulated asymmetrically this does not pose a problem.
Distance data fulfils the triangle inequality while disturbance data does not, as required for our study of potentially disturbance-reducing alternative ways.

5.2.2 Compared real tours with solutions of Model 1

In the context of the Green City Hubs project also information on real tours is available. These tours were carried out independently of the optimisation goals of the Green City Hubs project. Thus they neither account for emission minimisation nor for disturbance minimisation.

The question of interest is now, if they are comparable to the solutions obtained with Model 1. But there two problems arise. First, the instances corresponding to the real tours contain also customers outside of Vienna. Since disturbance data is only available for locations inside Vienna we can not solve such instances. But if we look at the real tours, we observe that customers outside of Vienna are mostly served at the beginning or at the end. so we assume that they would be also at the beginning or at the end of solution tours to our problem and consequently only the remaining, inner-city customers are relevant for optimisation.
The second problem is that also visits for picking up goods occur during the real tours. These pick-ups change load weight and since our optimisation problem (that is a pure delivery problem) considers load weight as a decision variable it is not very meaningful to compare the real tours to solutions of Model 1. We try to overcome this problem by assuming that the pick-up point is a deliver point.
Both assumptions may entail capacity problems but we do not care about this now.

We have chosen a real tour that corresponds to day 1 and reads as follows $(0, 32, 12, 42, 8, 34, 7, 40, 17, 5, 3, 15, *, *, 0)$. The $*$ represent two outer-city customers that would be visited after inner-city customer 15, but we assume that that the route goes from 15 directly to the hub such that we end up with considering the tour $(0, 32, 12, 42, 8, 34, 7, 40, 17, 5, 3, 15, 0)$. Customer 17 would be actually a pick-up point, but due to our assumptions we consider it as delivery point. Total demand of this tour is 1834 kg, so it could be carried out with any vehicle except the battery powered one. We use vehicle 1 for our tests.

Total emission of the real route sums up to 13.386 and total disturbance to 25.490. Figure 5.14 depicts the route in the background of Vienna's street map[1].

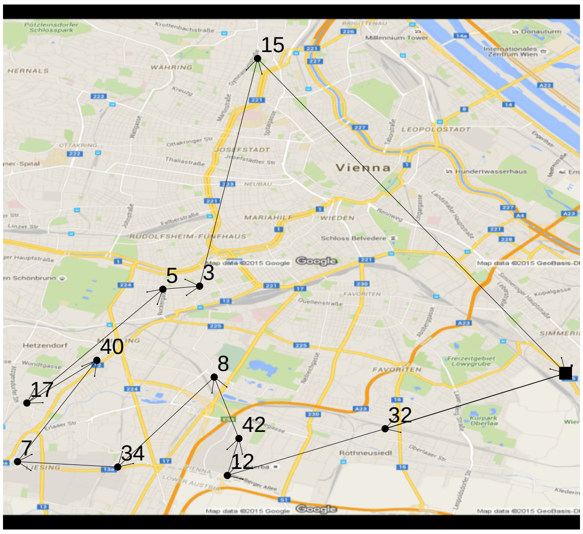


Figure 5.14: Real tour

The instance induced from this route comprises the customer set $C = \{3, 5, 7, 8, 12, 15, 17, 32, 34, 40, 42\}$. Together with the hub the instance contains thus 12 nodes. Solving it with Model 1 results in 143 Pareto optimal solutions, where the real route can not be found amongst them (CPU time: around 2.5 minutes). The visualisation of the associated Pareto front can be seen in Figure 5.15, that also shows the objective value point $(13.386, 25.490)$ of the real tour.

---

[1] The arcs in the visualisations of routes corresponding to Vienna data do actually not represent the real path, that would not be just a straight line. So instead they must be interpreted as a simplified representation of a connection between two location.

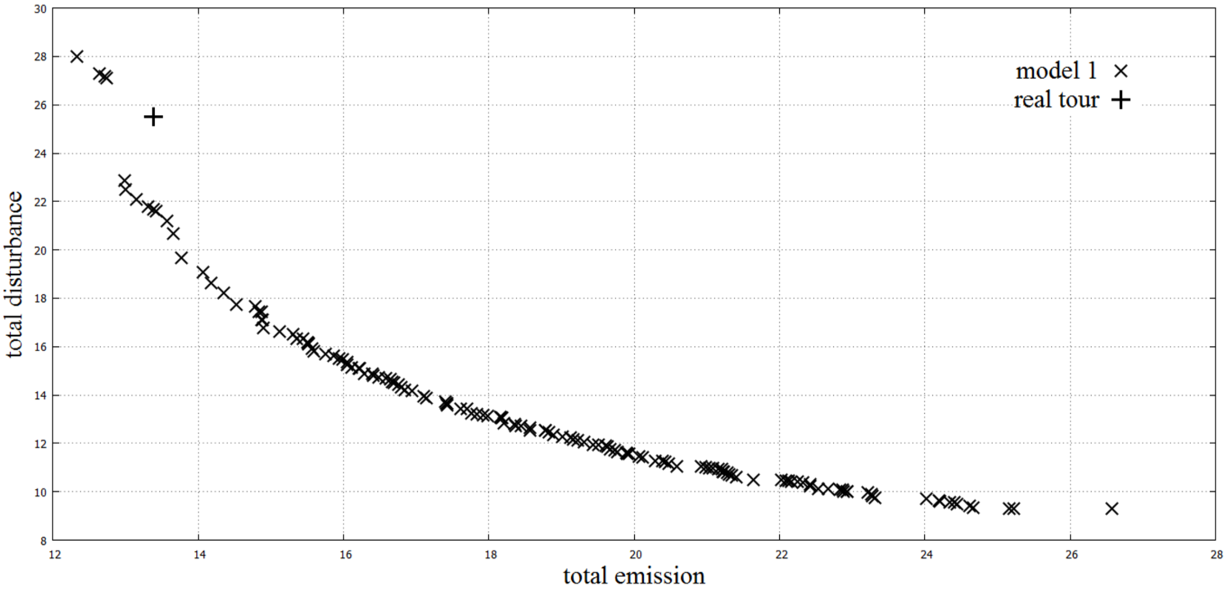


Figure 5.15: Pareto front of Model 1 and objective value point of real tour

The graphical representation makes clear why the solution set of Model 1 does not contain the real route: it is dominated by four solutions that are better in both emission and disturbance. So in the context of our optimisation problem, that aims at emission and disturbance minimisation, the real route is not favourable.

If we look at the Pareto optimal tours of Model 1, we can figure out those that have the smallest deviation (in the sense of node sequence) from the real tour. Figure 5.16 depicts two Pareto optimal solutions that belong to the part of the Pareto front left above the real route objective point in Figure 5.15, which means they have lower emission but higher disturbance than the real route. In the route on the left picture the improvement in emission comes mainly from reduction in distance that comes in turn from going $(\dots - 17 - 40 - \dots)$ instead of $(\dots - 40 - 17 - \dots)$. Also the exchange of customers 42 and 12 contributes to emission reduction. These changes induce using higher disturbing arcs what causes higher total disturbance. In the picture on the right the biggest change compared to the real route is that customer 8 is visited much later. In addition, as in the picture on the left, 17 and 40 are exchanged. The remaining customers are visited in the same sequence. Figure 5.17 illustrates two solutions that dominate the real tour. The only difference between the route on the left picture and the route on the right picture of Figure 5.16 is that 40 and 12 are exchanged in the node sequence. Anyway, this small change reduces disturbance enough to have beside lower emission also lower disturbance. The route in the right picture demonstrates that total disturbance of the route from picture 5.16.a) can be significantly reduced by leading the route around the highly disturbing city centre of Vienna. Figure 5.18 shows two solutions that have lower disturbance but higher emission than the real route. Both routes are regarding the node

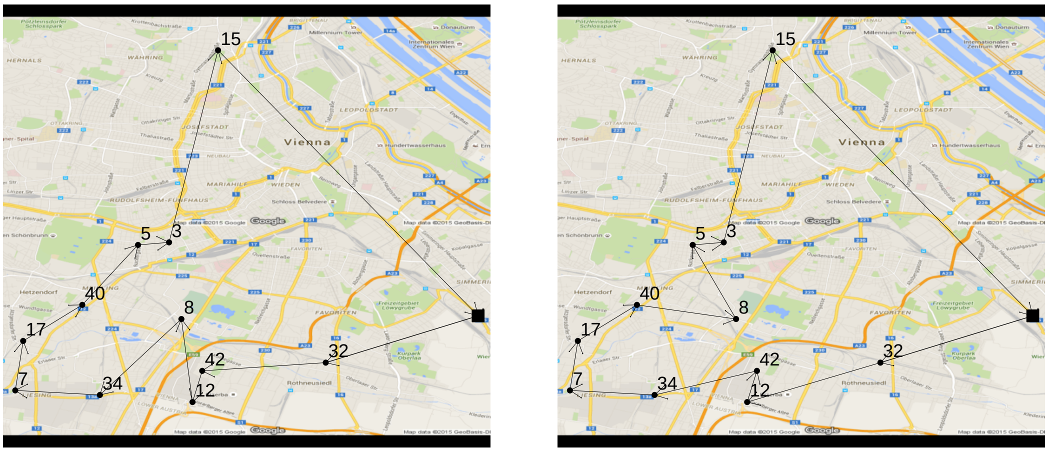


Figure 5.16: Pareto optimal solutions with lower emission but higher disturbance than real route

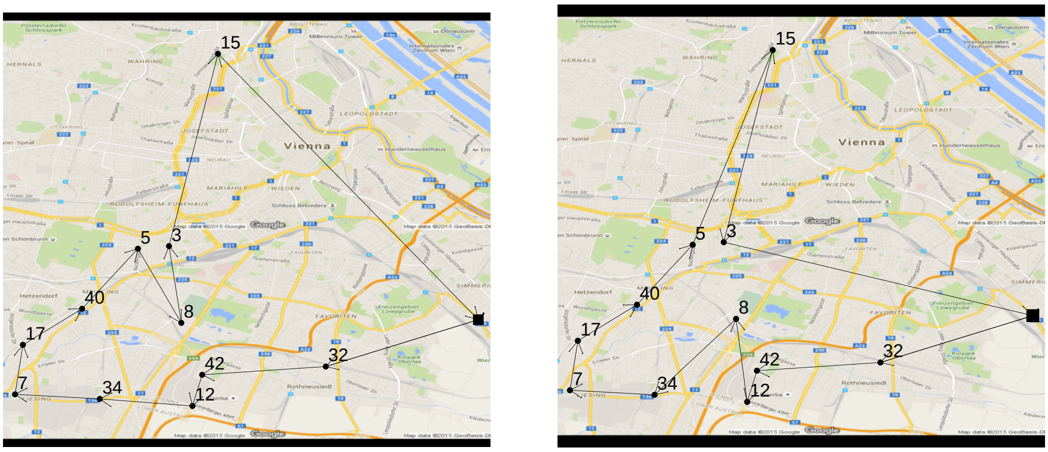


Figure 5.17: Pareto optimal solutions that dominate the real route

sequence to a certain extent comparable to those depicted in Figure 5.17, but contain now arc crossings resulting from the objective of relatively low total disturbance. These arc crossings deteriorate the routes with respect to travelled distance and consequently in total emission.

There is also information on other real tours, in particular for day 2,3 and 4. Testing Model 1 on the instances derived from them leads to similar results as we got for the real tour analysed just before, i.e. the real route is never Pareto optimal in the context of the optimisation problem of Model 1, but there are Pareto optimal solutions that show only small deviations from the real route.

5.2.3 Taking 'unused customers' as optional nodes

For randomly generated data it was rather easy to add optional nodes to the instances used for testing Model 2 and Model 3. For real data it requires considerably more effort

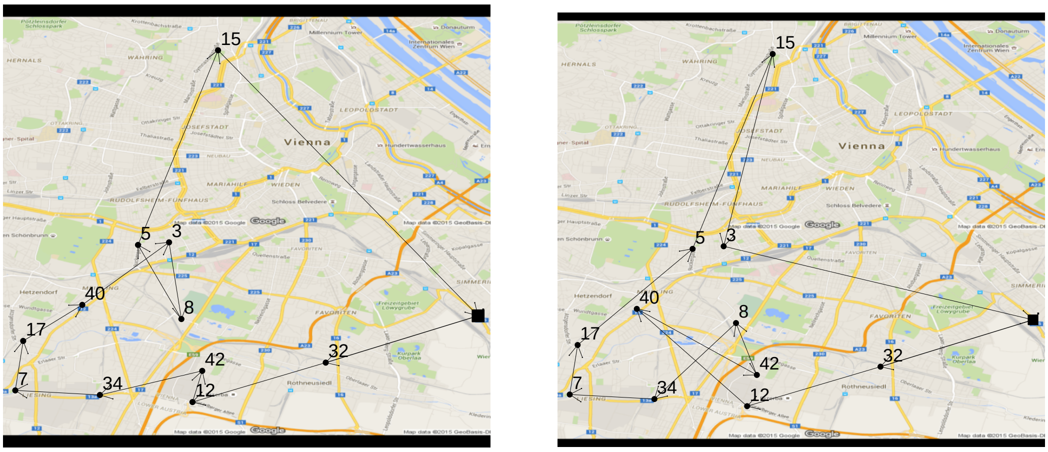


Figure 5.18: Pareto optimal solutions with lower disturbance but higher emission than real route

to evaluate information on optional nodes. First it has to be decided which points in the road network are taken as optional locations. This should not be done just by chance because it does not make sense to consider an optional location that does not provide any potential for reducing disturbance between to mandatory locations. But even if locations for optional nodes are fixed, there is still the need for associated distance and disturbance data.

In the course of the Green City Hubs project nothing related to optional nodes has been done so far. In order to test Model 3 in the context of Vienna anyway, we apply the following idea: We have information for a total of 45 customers. Since the considered test instances relate only to a certain day and a corresponding fixed set of customers that fits the vehicle's capacity. So there remains a set of unused customers that we are going to use as set of optional nodes. Although some of the 'unused' customers have strictly positive demand on the respective day their demand is set to zero in the derived instances because optional nodes are assumed to be 'demandless'.

To analyse the effect of this approach we will take the test instance used in the preceding section, i.e. the customer set $C = \{3, 5, 7, 8, 12, 15, 17, 32, 34, 40, 42\}$. There are then 34 unused customers that are taken as optional nodes and transformed by Martins' algorithm into multiple arcs for Model 3. In total 576 arcs are obtained - compared to 132 arcs in Model 1. So there are obviously disturbance-reducing paths via unused customers. Solving Model 3 with the ε-constraint method yields 193 Pareto optimal solutions (CPU time: around 6.5 minutes). So the Pareto set of Model 3 is larger than that of Model 1, what indicates that the optional nodes are indeed used. To check if the other 143 solutions are the same as in Model 1 or if the Pareto set of Model 3 (partly) dominates that of Model 1 we analyse their Pareto fronts illustrated in Figure 5.19. This reveals that the 143 Pareto optimal solutions of Model 1 are equivalent to the first 142 Pareto optimal solutions of Model 3 represented in the part to the left of the dashed vertical line.

The part to the right shows the 51 new Pareto optimal solutions that further decrease total disturbance but at the cost of increased emission. So for the considered instance including unused customers as optional nodes does not entail that Model 1's Pareto front is (partly) dominated by that of Model 3. Instead it solely extends the Pareto front in direction of higher emission and lower disturbance.

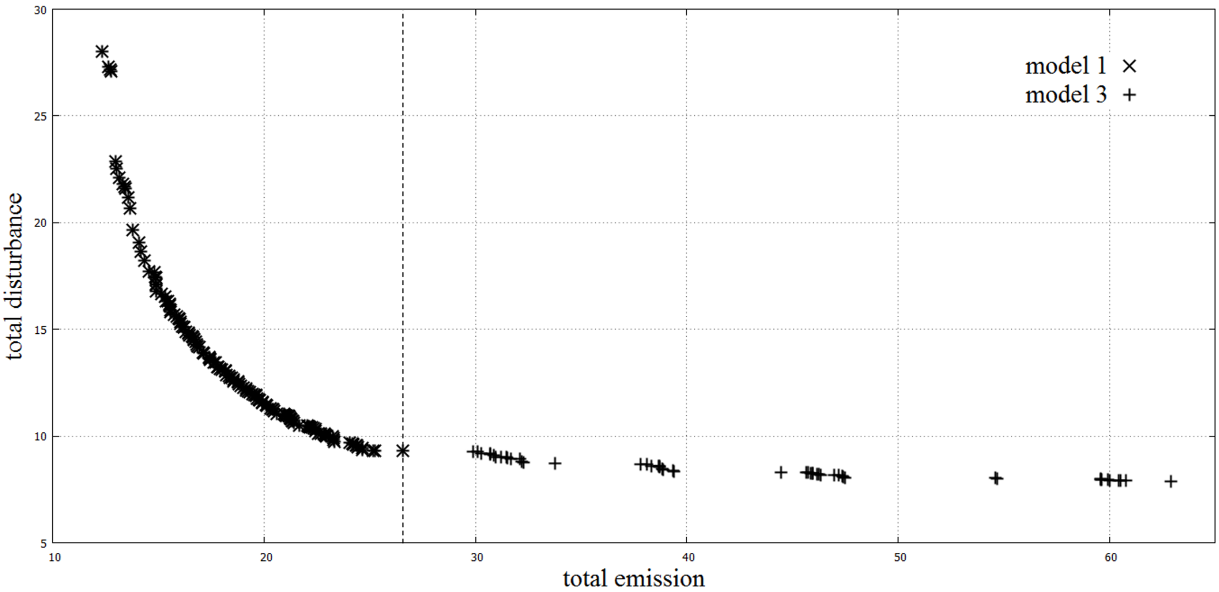


Figure 5.19: Pareto front of Model 3 in comparison to the Pareto front of Model 1

Including an unused customer as an optional node in a Pareto optimal route of Model 3 means that the customer is visited although it does not require any goods delivery, or at least it was assumed to be 'demandless'. However it can be that the visited, optional customer has strictly positive demand on the respective day anyway. And if there was enough capacity left, it would be hypothetically possible to deliver the required demand already within the current tour, although it was originally not planned. Since additional load weight has an effect on total emission it can happen that the resulting extended customer delivery route is not Pareto optimal any more.

However, unused customers are not really a proper choice for optional locations because they were not evaluated for the purpose of leading tours through less disturbing areas. Thus, in order to study the behaviour of Model 3 in a real-world setting we need meaningful data on optional locations, what is kept as a future task.

6 Conclusion and relevance of the work

In Section 1.3 of Chapter 1 we motivated the objective of minimising gas emissions and disturbance when planning customer delivery routes by several aspects. But can we expect that decision makers are indeed interested in the solutions to this problem?

In our framework the amount of emitted gases during a delivery tour is amongst others related to travelled distance. When studying the characteristics of the three modelling approaches by means of an example instance we have seen that solutions putting more importance on emission minimisation are close - or may be even equal - to the distance minimising solution. Since having short tours is still one of decision makers' main interests - because it is strongly related to saving money - we can expect that the part of emission minimisation is attractive for them. But beside emission minimisation also disturbance minimisation was targeted in the problem of this work. The computational experiments in Section 5.1.3 of Chapter 5 revealed the particularities of such solutions. They showed that the objective of disturbance minimisation generates routes containing detours, crossings or loops, what is actually in contradiction with the goal of going short ways.
But decision makers may have nowadays also other incentives than just operating fast and cheap. In the highly competitive business environment companies can benefit from having an ecologically and socially friendly image. Today's society pays increasingly more attention to 'living consciously', what is reflected in particular in the food sector where organic and slow food are on the rise. This change in lifestyle can make it happen that a company that implements a sustainable transportation policy can be preferred over a company that is only interested in saving money.

Moreover regulations for emission and disturbance could make sense. There could be for example a subsidy for companies that keep a very low emission level in their customer delivery tours or - in contrary - some penalties when a certain emission threshold is exceeded. Concerning disturbance, using streets through highly populated areas in the city center may be penalised or associated with paying an inner-city toll. This could be already reflected in disturbance data. We related disturbance only to population density and to some extent to distance, but increasing the disturbance value by the penalty for using the respective arc would also make sense and would even bring some monetary value into disturbance. If such disturbance payments are not considered in the optimisation problem, they still exist from outside the system. All these regulations may influence the interests of the decision maker such that solutions of our problem become attractive.

And above all, it is not about implementing an extreme solution, that takes care either only of disturbance or only of emission. Instead a decision maker can choose solutions that provide a good trade-off between the two objectives. Here the three different modelling approaches come into play. They all deal with the same problem, namely delivering customers in urban areas with respect to emission and disturbance minimisation. But which of the three models should be actually used? The computational study revealed their shortcomings and benefits. First of all the output of Model 2 and 3 showed that including alternative ways via optional nodes expands the scope for optimisation and provides in particular more potential for disturbance minimisation than Model 1. They yield a larger set of Pareto optimal solutions, what means more alternatives for the decision maker and better adjustment to his preferences. So in terms of solution quality one of these two models should be implemented. But also the computational effort is an important decision criteria. So although Model 2 and Model 3 are equivalent in terms of the Pareto front, they are not competitive in terms of CPU effort. Model 3 outperforms Model 2 by far, what makes Model 2 unattractive for practical problems and also for further theoretical considerations. Model 1 is associated with even lower computational burden than Model 3. In addition, evaluating optional nodes and multiple arcs for real-world problems requires quite an effort. So one may decide that having only one path between two locations is sufficient and apply therefore Model 1. However this is at the cost of solution quality that suffers from the absence of alternative paths in Model 1.

Hence, the choice between Model 1 and Model 3 depends on whether low computational effort or a good trade-off curve is desired.

Concerning the application of the presented problem there should be one more thing mentioned: Considering alternative paths via optional nodes does not make much sense when delivery points are too close to each other, as for example in the case of postal delivery where almost every house along a street has to be visited. However, the framework of the Green City Hubs project and also many other real-life transportation problems is such that customers are spread over a spacious region.

7 Perspectives

Some ideas for potential future work have been already mentioned throughout the work, but we want to summarise and deepen them here.

We assumed that the capacity restriction is not effective, what implied that solutions contain only one tour. However, to make the problem more realistic this assumption should be dropped, what requires dealing with a Vehicle Routing Problem in Model 1 and Model 3, where in general not all customers can be delivered within one tour and with one vehicle.

Moreover, we should also take the limit on tour duration into account when using alternatively fuelled vehicles, because otherwise the risk of running out of electricity before reaching a recharge station is high. A limit on duration can become particularly interesting when it comes to disturbance minimisation, because less disturbing ways tend to be longer. So a time/distance constraint could be more restrictive there.

In this work an exact solution method, namely the ε-constraint method, was implemented to solve the presented modelling approaches. It was sufficient when studying the characteristics of the different models or their performance differences. We have seen that also the considered real world instances could be solved in reasonable time with the ε-constraint method. However, our computational experiments demonstrated that the exact method becomes inefficient for larger instances, in particular when testing Model 3, where the existence of multiple arcs increased complexity considerably. For Model 2 the ε-constraint method failed already when testing small instances. Since we have Model 3 that also considers optionals nodes and yields the same solution set as Model 2 but in much shorter time, Model 2 should not be further investigated when developing heuristics. Instead the focus should be put on Model 1 and Model 3. When we evaluated the performance of Model 1 and Model 3 once with total emission as f_1 and once with total disturbance as f_1 in the ε-constraint method we observed that higher computational effort is required for solutions close to emission minimisation. This should be taken into account in an heuristic approach.

We carried out computational experiments on randomly generated data and on real world data. For the latter we had to deal with a lack of information on optional locations. To draw meaningful conclusions about the effect of considering optional locations in a real-world setting appropriate data has to be evaluated. The optional locations should be spread across the city such that direct shortest-distance paths between customers in highly populated areas can be avoided by using an alternative path over an optional location.

There is also much potential for improvement of disturbance data. When generating the density grid for Vienna we noted already that it may not display reality very well. In general, for a more sophisticated approach one should make the disturbance of a street not only dependent on the amount of people living next to it, but also on how big and important this street is. In Vienna, for example, there are heavily frequented streets (e.g. the 'Gürtel') that are close to or in the city center but should not be considered as that disturbing as small streets in residential, where traffic is rather unusual and people are not so used to it. Also social institutions, like schools or retirement homes, should be taken into account in the sense that disturbance is increased on streets going past them.

Appendix

A. Data set available from Green City Hubs project

```
Name: T4_V45_H1_K6.txt
Case:tech_log_vienna
T: 4
V: 45
H: 1
N: 46
K: 6
alpha: 0.2 0.3 0.4 0.1
#Vehicle Info (price per month)
#Q p e f s L P name
2500 250 0.25195 0.000016 22.95 999 3 diesel
2500 270 0.24325 0.000016 22.95 999 3 diesel_(start-stop)
2300 350 0.21035 0.000012 22.95 999 3 parallel-hybrid
2300 380 0.19862 0.000013 22.95 999 3 parallel-plugIn-hybrid_(single_loading)
2300 400 0.17894 0.000013 22.95 999 3 parallel-plugIn-hybrid_(multi_loading)
1800 450 0.12834 0.000013 22.95 150 3 battery-powered
#Hub Info (price per month)
#lat long h R
48.159517 16.42201 6536.98 687.5
#Customer Info
#lat long q
48.168442 16.38718 0 580 108 0
48.283021 16.380951 0 0 0 105
48.180145 16.335358 21 0 0 0
48.172841 16.281779 0 14 0 0
48.17942 16.326651 35 0 0 0
48.17418 16.383087 0 0 125 0
48.138586 16.292243 52 465 0 0
48.158618 16.338816 38 0 1105 0
48.185196 16.332158 0 10 0 0
48.23261 16.413773 20 0 0 0
```

```
48.260121 16.408763 0 0 0 815
48.135311 16.341924 94 142 0 0
48.253128 16.393081 320 0 0 0
48.245223 16.372595 0 55 0 0
48.234055 16.349088 21 0 0 0
48.181337 16.467182 1082 3475 210 2380
48.152464 16.294455 130 0 0 0
48.268529 16.42666 10 0 0 0
48.208431 16.428272 1176 540 608 0
48.203432 16.435476 14 0 0 0
48.21123 16.366682 0 65 0 0
48.190489 16.288299 0 0 32 0
48.164686 16.333666 0 0 162 0
48.239433 16.351752 0 0 0 73
48.136162 16.356695 0 255 150 0
48.13145 16.308887 0 0 1100 0
48.209616 16.338401 0 200 0 0
48.26107 16.452809 0 0 50 0
48.212381 16.373905 0 54 0 0
48.188117 16.417951 0 134 0 0
48.183442 16.410876 50 0 0 0
48.146382 16.379285 4 0 0 0
48.228271 16.388046 0 0 38 0
48.137312 16.315967 30 0 0 0
48.136868 16.300597 0 0 400 0
48.237444 16.331305 0 0 0 22
48.265636 16.474748 180 180 0 0
48.219064 16.407674 0 0 0 55
48.268274 16.422143 164 0 52 5
48.162596 16.311026 1322 169 0 0
48.282602 16.378092 0 2690 0 0
48.144069 16.344656 87 0 0 0
48.152617 16.367711 0 100 0 0
48.21216 16.361613 0 240 0 0
48.15744 16.302206 0 37 0 0
#Travel Time starting with hubs in minutes
0.0 12.8 46.03 28.5 36.5 33.57 16.18 31.52 17.85 32.73 29.03 44.53 ...
...
#Travel Distance starting with hubs in kilometers
```

```
0.0 4.21 16.7 8.01 12.26 8.72 4.32 13.26 8.74 8.83 10.28 14.75 10.47 ...
...
#Disturbance starting with hubs
0.0 1.95 0.51 1.27 0.57 1.16 2.19 0.38 0.72 1.17 0.77 0.48 0.54 0.69 ...
...
EOF
```

Printed Sources

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. Technical report, DTIC Document, 1988.

[2] Nilesh Anand, Hans Quak, Ron van Duin, and Lori Tavasszy. City logistics modeling efforts: Trends and gaps-a review. *Procedia-Social and Behavioral Sciences*, 39:101–115, 2012.

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

[4] Richard Bellman. On a routing problem. 1956.

[5] Riccardo Bozzo, Andrea Conca, and Flavio Marangon. Decision support system for city logistics: Literature review, and guidelines for an ex-ante model. *Transportation Research Procedia*, 3:518–527, 2014.

[6] Massimiliano Caramia and Paolo Dell'Olmo. Multi-objective management in freight logistics: Increasing capacity, service level and safety with optimization algorithms. 2008.

[7] European Commission. White paper roadmap to a single european transport area, towards a competitive and resource efficient transport system. 2011.

[8] Kalyanmoy Deb, Kaisa Miettinen, and Deepak Sharma. A hybrid integrated multi-objective optimization procedure for estimating nadir point. pages 569–583, 2009.

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

[10] Eric V Denardo. *Dynamic programming: models and applications.* Courier Corporation, 2012.

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

[12] Richard Eglese and Tolga Bektaş. Green vehicle routing. *Vehicle Routing: Problems,*

Methods, and Applications, 18:437, 2014.

[13] Matthias Ehrgott and Xavier Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR-Spektrum*, 22(4):425–460, 2000.

[14] Javier Faulin, Angel Juan, Fernando Lera, and Scott Grasman. Solving the capacitated vehicle routing problem with environmental criteria based on real estimations in road transportation: a case study. *Procedia-Social and Behavioral Sciences*, 20:323–334, 2011.

[15] Dominique Feillet, Pierre Dejax, and Michel Gendreau. Traveling salesman problems with profits. *Transportation science*, 39(2):188–205, 2005.

[16] Lester R Ford and Delbert R Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[17] Lester Randolph Ford. Network flow theory. 1956.

[18] Xavier Gandibleux, Frédéric Beugnies, and Sabine Randriamasy. Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function. *4OR*, 4(1):47–59, 2006.

[19] Thierry Garaix, Christian Artigues, Dominique Feillet, and Didier Josselin. Vehicle routing problems with alternative paths: an application to on-demand transportation. *European Journal of Operational Research*, 204(1):62–75, 2010.

[20] Gregory Gutin and Abraham P Punnen. *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media, 2006.

[21] José Holguín-Veras, Michael Silas, John Polimeni, and Brenda Cruz. An investigation on the effectiveness of joint receiver–carrier policies to increase truck traffic in the off-peak hours. *Networks and Spatial Economics*, 8(4):327–354, 2008.

[22] Imdat Kara, Bahar Y Kara, and M Kadri Yetis. Energy minimizing vehicle routing problem. pages 62–71, 2007.

[23] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. An adaptive scheme to generate the pareto front based on the epsilon-constraint method. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2005.

[24] Eugene L Lawler, Jan Karel Lenstra, AHGR Kan, and David B Shmoys. *The traveling*

salesman problem: a guided tour of combinatorial optimization. Vol. 3. Wiley, New York, 1985.

[25] 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.

[26] Will Maden, Richard 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.

[27] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[28] R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. *Structural and multidisciplinary optimization*, 41(6):853–862, 2010.

[29] Ernesto Queiros Vieira Martins. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245, 1984.

[30] Andrew Palmer. The development of an integrated routing and carbon dioxide emissions model for goods vehicles. 2007.

[31] Cristina Pronello and Michel André. Pollutant emissions estimation in road transport models. *INRETS-LTE report*, 2007, 2000.

[32] Yoshinori Suzuki. A new truck-routing approach for reducing fuel consumption and pollutants emission. *Transportation Research Part D: Transport and Environment*, 16(1):73–77, 2011.

[33] Roberto Tadei, Guido Perboli, and Francesca Perfetti. The multi-path traveling salesman problem with stochastic travel costs. *EURO Journal on Transportation and Logistics*, pages 1–21, 2014.

[34] Eiichi Taniguchi, Russell G Thompson, and Tadashi Yamada. Recent trends and innovations in modelling city logistics. *Procedia-Social and Behavioral Sciences*, 125:4–14, 2014.

[35] Fabien Tricoire, Alexandra Graf, and Walter J Gutjahr. The bi-objective stochastic

covering tour problem. *Computers & operations research*, 39(7):1582–1592, 2012.

[36] Department of Economic United Nations and Population Division Social Affairs. World urbanization prospects: The 2014 revision, highlights. *ST/ESA/SER.A/352*, 2014.

[37] Laurence A Wolsey. Mixed integer programming. *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

Internet Sources

[38] Department for production and operations management with interantional focus - research project 'green city hubs'. `http://plis.univie.ac.at/projects/green-city-hubs/`. Accessed: 28.11.2015.

[39] Eine grüne logistik für wien. `http://medienportal.univie.ac.at/uniview/forschung/detailansicht/artikel/eine-gruene-logistik-fuer-wien/`. Accessed: 28.11.2015.

[40] Amazon prime air. `http://www.amazon.com/b?node=8037720011`. Accessed: 10.12.2015.

[41] Fabien Tricoire. Proute, the python software to visualize routing solutions. `https://github.com/fa-bien/proute`. Accessed: 28.01.2016.

[42] Arbeit und Statistik City of Vienna Magistratsabteilung 23 Wirtschaft. Bevölkerungsstand in wien nach geschlecht und staatsangehörigkeit, zählbezirke. `https://www.data.gv.at/katalog/dataset/efbdbab2-a095-422b-acb4-81cc9c380093`. Accessed: 08.09.2015.

[43] City of Vienna Magistratsabteilung 21 Stadtteilplanung und Flächennutzung. Zählbezirksgrenzen wien. `https://www.data.gv.at/katalog/dataset/e4079286-310c-435a-af2d-64604ba9ade5`. Accessed: 08.09.2015.

[44] Anita Graser. Mapping vienna population density. `http://anitagraser.com/2012/11/17/mapping-ogdwien-population-density/`. Accessed: 08.10.2015.

Supplements

Deutsche Zusammenfassung

In der vorliegenden Arbeit werden drei verschiedene Modelle zur Untersuchung des Trade-offs zwischen Abgasemissionen und Ruhestörung bei der Kundenbelieferung in Stadtgebieten präsentiert. Die Menge der umweltverschmutzenden Abgasemissionen hängt dabei von der zurückgelegten Wegstrecke und dem Gewicht der transportierten Ladung ab. Ruhestörung soll ein Maß dafür sein, wie sehr Stadtbewohner durch den Güterverkehr im Sinne von Lärm, Schmutz, Gefahren oder Erschütterungen beeinträchtigt werden. Beide Zielfunktionen sollen minimiert werden. Da aber Routen, die wenig Abgase produzieren, im Allgemeinen eine hohe Ruhestörung mit sich bringen können - wie auch umgekehrt - müssen wir bei der Formulierung der Modellierungsansätze vom Konzept der Pareto-Optimierung Gebrauch machen.

Im ersten Modell stehen nur die Pfade mit minimalem Abstand zwischen den relevanten Orten zur Verfügung. Der Ansatz klassifiziert sich somit als bi-objective Travelling Salesman Problem auf einem einfachen Graphen. Modell 2 bietet zusätzlich zu Kunden und Depot noch so genannte optionale Punkte, die nicht besucht werden müssen, aber können, falls sie die Route durch ein Gebiet mit weniger Ruhestörung führen. Im dritten Ansatz werden die Alternativpfade durch einen Graph mit Mehrfachkanten modelliert, wodurch wir es folglich mit einem bi-objective Travelling Salesman Problem auf einem Multigraphen zu tun haben.

Wir zeigen, dass Modell 2 und Modell 3 dieselben Pareto-optimalen Lösungen liefern, wenn wir die optionalen Punkte von Modell 2 in Mehrfachkanten für Modell 3 umwandeln. Diese Transformierung wird als all-pairs bi-objective Shortest Path Problem formuliert und mithilfe von Martins‘ Algorithmus gelöst.

Die drei Modellierungsansätze werden anhand von zufällig generierten Instanzen getestet, um die Charakteristiken ihrer Lösungen zu untersuchen und Vergleiche hinsichtlich Lösungsqualität und CPU-Performance aufzustellen. Zur exakten Lösung der Probleme wird die sogenannte ε-Constraint Methode implementiert. Im Zuge der Instanzerzeugung wird außerdem eine Methode zur Berechnung von Ruhestörungsdaten entwickelt.
Weiters werden auf Basis von realen Daten Touren aus der Praxis mit den Lösungstouren unserer Modelle verglichen.

CURRICULUM VITAE

Personal data

Name	Jasmin Grabenschweiger
Date of birth	April 12, 1990
Citizenship	Austria
Phone	+43 664 45 04 134
Email	jasmin1204@gmx.at

Education

03/2013 – 02/2016 (presumably)	Master Quantitative Economics, Management and Finance, University of Vienna
	Specialisation: Management Production/Logistics/Transport
since 10/2012	Master Mathematics University of Vienna
10/2009 – 09/2012	Bachelor Mathematics University of Vienna
	Specialisation: Applied Mathematics
09/2004 – 06/2009	College of management and service industries, Perg (Upper Austria)
	Specialisation: Health Care and Social Management