Department of Mathematics and Systems Analysis

Demand-Responsive Transport: Models and Algorithms

Lauri Häme

Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 80/2013

Demand-Responsive Transport: Models and Algorithms

Lauri Häme

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Science, at a public examination held at lecture hall K of the school on 31st May 2013 at 12.

Aalto University
School of Science
Department of Mathematics and Systems Analysis

Supervising professors
Prof. Esko Valkeila
Prof. Olavi Nevanlinna

Thesis advisor
Dr. Harri Hakula

Preliminary examiners
Prof. Nelly Litvak, University of Twente, The Netherlands
Dr. Silvio Nocera, Università Iuav di Venezia, Italy

Opponent
Prof. Lasse Leskelä, University of Jyväskylä

Aalto University publication series
DOCTORAL DISSERTATIONS 80/2013

ISBN 978-952-60-5162-8 (printed)
ISBN 978-952-60-5163-5 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-5163-5

Unigrafia Oy
Helsinki 2013

Finland

NORDIC ECOLABEL
441 697
Printed matter

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
Lauri Häme

Name of the doctoral dissertation
Demand-Responsive Transport: Models and Algorithms

Publisher School of Science

Unit Department of Mathematics and Systems Analysis

Series Aalto University publication series DOCTORAL DISSERTATIONS 80/2013

Field of research Mathematics

Manuscript submitted 18 January 2013 **Date of the defence** 31 May 2013

Permission to publish granted (date) 16 April 2013 **Language** English

☐ **Monograph** ☒ **Article dissertation (summary + original articles)**

Abstract

Demand-responsive transport is a form of public transport between bus and taxi services, involving flexible routing of small or medium sized vehicles. This dissertation presents mathematical models for demand-responsive transport and methods that can be used to solve combinatorial problems related to vehicle routing and journey planning in a transport network.

Public transport can be viewed as a market where demand affects supply and vice versa. In the first part of the dissertation related to vehicle routing, we show how a given demand for transportation can be satisfied by using a fleet of vehicles, assuming that the demand is known at the individual level. In the second part, by considering the journey planning problem faced by commuters, we study how the demand adapts to the supply of transport services, assuming that the supply remains unchanged for a short period of time. We also present a stochastic network model for determining the economic equilibrium, that is, the point at which the demand meets the supply, by assuming that commuters attempt to minimize travel time and transport operators aim to maximize profit.

The mathematical models proposed in this work can be used to simulate the operations of public transport services in a wide range of scenarios, from paratransit services for the elderly and disabled to large-scale demand-responsive transport services designed to compete with private car traffic. Such calculations can provide valuable information to public authorities and planners of transportation services, regarding, for example, regulation and investments. In addition to public transport, potential applications of the proposed methods for solving vehicle routing and journey planning problems include freight transportation, courier and food delivery services, military logistics and air traffic.

Keywords demand-responsive transport, public transport, graph problems, algorithms, networks, discrete optimization, stochastic optimization, Markov decision process, economics of transportation

ISBN (printed) 978-952-60-5162-8 **ISBN (pdf)** 978-952-60-5163-5

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Espoo **Location of printing** Helsinki **Year** 2013

Pages 168 **urn** http://urn.fi/URN:ISBN:978-952-60-5163-5

Aalto-yliopisto

Tiivistelmä

Aalto-yliopisto, PL 11000, 00076 Aalto www.aalto.fi

Tekijä
Lauri Häme

Väitöskirjan nimi
Demand-Responsive Transport: Models and Algorithms

Julkaisija Perustieteiden korkeakoulu

Yksikkö Matematiikan ja systeemianalyysin laitos

Sarja Aalto University publication series DOCTORAL DISSERTATIONS 80/2013

Tutkimusala Matematiikka

Käsikirjoituksen pvm 18.01.2013 **Väitöspäivä** 31.05.2013

Julkaisuluvan myöntämispäivä 16.04.2013 **Kieli** Englanti

☐ **Monografia** ☒ **Yhdistelmäväitöskirja (yhteenveto-osa + erillisartikkelit)**

Tiivistelmä

Kysyntäohjautuvalla joukkoliikenteellä tarkoitetaan bussi- ja taksipalvelujen välimuotoa, joka perustuu pienten tai keskisuurten ajoneuvojen joustavaan reititykseen. Tässä väitöskirjassa esitetään matemaattisia malleja kysyntäohjautuvalle joukkoliikenteelle, ja menetelmiä, joilla voidaan ratkaista ajoneuvojen reitinlaskentaan ja matkansuunnitteluun liittyviä kombinatorisia ongelmia liikenneverkossa.

Joukkoliikennettä voidaan tarkastella markkinana, jossa kysyntä vaikuttaa tarjontaan ja päinvastoin. Väitoskirjan ensimmäisessä osassa, joka käsittelee ajoneuvojen reitinlaskentaa, näytetään miten tunnettuun kysyntään voidaan vastata käyttämällä tiettyä ajoneuvokantaa, kun oletetaan kysyntä tunnetuksi yhden matkustajan tarkkuudella. Toisessa osassa tarkastellaan matkustajien matkansuunnittelua joukkoliikenneverkossa, eli sitä miten kysyntä mukautuu liikennepalvelujen tarjonnan mukaan, kun oletetaan tarjonta muuttumattomaksi lyhyellä aikavälillä. Lopuksi esitetään menetelmä taloudellisen tasapainopisteen, eli kysynnän ja tarjonnan kohtaamispisteen, määrittämiseksi, kun oletetaan että matkustajat pyrkivät minimoimaan matka-aikaa ja liikennepalvelujen tarjoajat pyrkivät maksimoimaan taloudellista voittoa.

Tässä työssä esiteltyjen mallien avulla voidaan simuloida useita erityyppisiä liikennepalveluja vanhuksille ja liikuntarajoitteisille suunnatuista kutsulinjoista henkilöautoliikenteen kanssa kilpaileviin laajamittaisiin kysyntäohjautuviin joukkoliikennejärjestelmiin. Mallien avulla tehdyt laskelmat voivat tuottaa arvokasta tietoa viranomaisille ja liikennepalvelujen suunnittelijoille liikenteen säännöstelyyn ja investointeihin liittyen. Joukkoliikenteen lisäksi esiteltyjä reitinlaskenta- ja matkansuunnittelumenetelmiä voidaan soveltaa muun muassa rahti- ja lentoliikenteessä, lähetti- ja ruoankuljetuspalveluissa sekä sotilaslogistiikassa.

Avainsanat kysyntäohjautuva joukkoliikenne, kutsujoukkoliikenne, älykäs joukkoliikenne, algoritmit, diskreetti optimointi, stokastinen optimointi, Markov-päätösprosessi, liikennetalous

ISBN (painettu) 978-952-60-5162-8 **ISBN (pdf)** 978-952-60-5163-5

ISSN-L 1799-4934 **ISSN (painettu)** 1799-4934 **ISSN (pdf)** 1799-4942

Julkaisupaikka Espoo **Painopaikka** Helsinki **Vuosi** 2013

Sivumäärä 168 **urn** http://urn.fi/URN:ISBN:978-952-60-5163-5

Preface

In the beginning of my doctoral studies I worked in a research project related to demand-responsive public transport, based in the Department of Computer Science in Helsinki University of Technology (currently a part of Aalto University), and launched by professor Reijo Sulonen. One of the main tasks in the project was to design "from scratch" the routing strategy for a fleet of mini-buses without fixed routes, for combining passengers' trips in an on-line fashion. After working in the project I continued to research on the same topic in the Department of Mathematics and Systems Analysis under the guidance of Dr. Harri Hakula, which led to the publication of this dissertation. In the beginning of 2013, as an extension to the research project, a test version of a fully automatized demand-responsive transport service was deployed in Helsinki.

First, I would like to thank Dr. Harri Hakula for being the instructor of this work and for providing professional guidance throughout my doctoral studies. I would also like to express my gratitude to Professors Esko Valkeila and Olavi Nevanlinna for supervising the work.

Furthermore, I would like to thank colleagues and superiors Dr. Esa Hyytiä, Dr. Aleksi Penttinen, Teemu Sihvola, Jani-Pekka Jokinen, Jeremias Kangas, Tuukka Sarvi, Prof. Samuli Aalto, Prof. Reijo Sulonen and Prof. Jorma Virtamo for scientific support, and Prof. Nelly Litvak, Dr. Silvio Nocera, Prof. Venkat Anantharam and Prof. Juuso Töyli for examining this dissertation.

Finally, I would like to thank my family and friends for their insightful comments to improve the quality of this work.

This dissertation is dedicated to the memory of Esko Valkeila (1951–2012), late professor in stochastics at Aalto University, and the supervisor of my master's thesis and this work until November 2012.

Helsinki, April 17, 2013,

Lauri Häme

Contents

Preface 1

Contents 3

List of Publications 5

Author's Contribution 7

1. Introduction 9

1.1 Demand-responsive transport today. 9
1.2 . . .and tomorrow 10
1.2.1 Opportunities 11
1.2.2 Possible issues 11
1.3 Problem statement 12

2. Vehicle Routing 15

2.1 Single-vehicle dial-a-ride problem 15
2.1.1 Problem formulation 16
2.1.2 Adaptive insertion algorithm [Publication I] 17
2.1.3 Routing by ranking [Publication V] 21
2.1.4 Numerical experiments 25
2.2 Multi-vehicle dial-a-ride problem 27
2.2.1 The maximum cluster algorithm [Publication VI] 28
2.2.2 A priori screening 28
2.2.3 Numerical experiments 30

3. Journey planning 33

3.1 Stochastic model of a scheduled network [Publication III] 35
3.1.1 Service legs and walking legs 35
3.1.2 Transfers 35

3.1.3 Paths 36
3.2 Problem solution 37
3.2.1 Markov Decision Process 37
3.2.2 Optimal policy 38
3.2.3 Backward induction algorithm 39
3.2.4 Expected number of paths 40
3.3 Numerical experiments 41
3.3.1 Description of instances 41
3.3.2 Results 41

4. Economic equilibrium **43**
4.1 Demand 44
4.2 Supply 44
4.3 Competitive market 45
4.3.1 Network equilibrium 46
4.3.2 A three node example 46
4.3.3 Long-run example 47

5. Conclusions **51**

Errata **55**

Bibliography **57**

Publications **63**

List of Publications

This thesis consists of an overview and of the following publications which are referred to in the text by their Roman numerals.

I Lauri Häme. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows. *European Journal of Operational Research*, 209, p. 11–22, February 2011.

II Lauri Häme, Harri Hakula. Dynamic journeying under uncertainty. *European Journal of Operational Research*, 225, p. 455-471, March 2013.

III Lauri Häme, Harri Hakula. Dynamic Journeying in Scheduled Networks. *IEEE Transactions on Intelligent Transportation Systems*, 14, p. 360-369, March 2013.

IV Lauri Häme, Jani-Pekka Jokinen, Reijo Sulonen. Modeling a competitive demand-responsive transport market. In *Kuhmo Nectar Conference on Transport Economics*, Stockholm, Sweden. 20 pages, June-July 2011.

V Lauri Häme, Harri Hakula. Routing by Ranking: A Link Analysis Method for the Constrained Dial-A-Ride Problem. *Operations Research Letters*, Under minor revision, 6 pages, 16.3.2012 .

VI Lauri Häme, Harri Hakula. A Maximum Cluster Algorithm for Checking the Feasibility of Dial-A-Ride Instances. *Transportation Science*, Under minor revision, 16 pages, 16.3.2012 .

Author's Contribution

Publication I: "An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows"

This article was written by the author.

Publication II: "Dynamic journeying under uncertainty"

The author was the main author of this article.

Publication III: "Dynamic Journeying in Scheduled Networks"

The author was the main author of this article.

Publication IV: "Modeling a competitive demand-responsive transport market"

The author was the main author of this article, which was nominated for the best student paper award in Kuhmo Nectar 2011. The article is under revision for publication in Transportation Research Part B.

Publication V: "Routing by Ranking: A Link Analysis Method for the Constrained Dial-A-Ride Problem"

The author was the main author of this article.

Publication VI: "A Maximum Cluster Algorithm for Checking the Feasibility of Dial-A-Ride Instances"

The author was the main author of this article.

1. Introduction

1.1 Demand-responsive transport today...

Demand-Responsive Transport (DRT) is often referred to as a form of public transport between bus and taxi services involving flexible routing and scheduling of small or medium sized vehicles. This means that the vehicle routes are updated daily or in real time by incorporating information on the demand for transportation. Usually, the customers of a DRT service are required to request and book their trips in advance by placing trip requests including information on the origin and destination of the trip as well as the desired pick-up or drop-off time. The vehicle operator uses this information to provide service that satisfies the passenger needs.

DRT services are often fully or partially funded by local authorities as providers of socially necessary transport. They are typically used to provide transportation in areas with low transportation demand where a regular bus service might not be as efficient. Another common application of DRT arises in door-to-door transportation of elderly or handicapped people (paratransit). Most services provided by private companies for commercial reasons are related to transporting passengers between airports and urban areas.

The implementation of demand-responsive transport is strongly dependent on the target group or the business concept of the service. Some DRT systems make use of terminals, at one or both ends of a route, such as an urban center or airport. In these applications (one-to-many or many-to-one), customers may specify either the origin or destination of the desired trip. In many-to-many services, vehicle routes are built freely according to customer requests. Such systems provide either a door-to-door service within a specific area or a transport service between a set of specified stops. For example, a DRT service operating in Nurmijärvi (Finland) aims to improve the level and accessibility of services in a sparsely populated area and to reduce the costs of public transport. The service operates on a "many-to-many" basis, that is, there are no predefined routes. The stop points are located at a maximum of 900 m from origins and destinations. In the case of special users, the stop points are non-predefined (door-to-door service).

The popularity of demand-responsive transport has recently grown mainly due to the shortcomings of conventional bus and taxi services, and new technical developments. In addition, flexible public transport services provided by local authorities and bus operators in partnerships with employers, stores, and leisure centres are thought to help to break down social exclusion [62]. However, current DRT services have often been criticised because of their relatively high cost of provision, their lack of flexibility in route planning, and their inability to manage high demand [54].

At the present moment, a large number of demand-responsive transport services are in operation. Most of such services operate within relatively small neighbourhoods and during low-use daytime hours, when there is not enough demand for traditional public transport. Thus, while the current services meet their requirements, demand-responsive transport remains a relatively small business compared to traditional transportation services, not to speak of private cars.

What if a DRT system was implemented in large scale, in a way that service could be provided for an entire metropolitan area?

1.2 . . .and tomorrow

Several new ideas and concepts related to demand-responsive transport services operating in urban areas have been recently presented, see for example [24, 45]. These ideas are often motivated by problems arising from the congestion of urban areas caused by the increasing number of private cars. Thus, one of the main present goals of planning demand-responsive transport is seen to be the development of *functional public transport services able to compete with private car and taxicab traffic*.

The popularity of the private car as a means of transport is partly based on a direct connection between the origin and destination of a trip and a short total travel time. Another major advantage of the private car is seen to be the availability of the car at any time, even without planning beforehand. The study of large scale demand-responsive transport has therefore been directed towards highly dynamic services, which allow customers to request service not long before they are willing to depart. In addition, a demand-responsive public transport system should be able to offer an alternative for transportation without the inconvenience related to conventional public transport.

The total travel time in public transport consists of *walking time* from origin to pick-up point, *waiting time* at the pick-up point, *ride time*, that is, the time spent in the vehicle, possible *transfer time* and walking time from drop-off point to destination. In order to attract people with private cars, it is necessary that the waiting and riding times are within acceptable bounds. In addition, it can be suggested that the service should be as close to a door-to-door service as possible and the amount of transfers between vehicles should be minimal.

Intuitively, the idea of a large scale DRT system seems promising. With state-of-the-art engineering, there should be no severe technical obstacles in implementing such a service.

1.2.1 Opportunities

The fact that demand-responsive transport is "there for you when you want and where you want" is thought to be a major advantage compared to conventional public transport. While it may not be feasible to think that DRT could provide a level of service substantially better than that offered by taxicabs, a system that could combine customers' trips efficiently could be more cost-efficient than a conventional taxi organization. This would make it possible to provide more inexpensive service without compromising too much on the level of service experienced by customers.

Compared to private cars, demand-responsive transport is thought to have several advantages in urban areas. For a model of a hypothetical large-scale demand-responsive public transport system for the Helsinki metropolitan area, simulation results published in 2005 demonstrated that "in an urban area with one million inhabitants, trip aggregation could reduce the health, environmental, and other detrimental impacts of car traffic typically by 50 - 70%, and could attract about half of the car passengers, and within a broad operational range would require no public subsidies" [79]. In addition to providing affordable transportation without the additional expenses related to maintenance, taxes and parking fees, demand-responsive transport could eliminate many other, possibly concealed, concerns related to private cars, including the difficulty of finding parking space and the stress related to driving in hazardous conditions or traffic jams.

At this point, one might ask: If the large scale demand-responsive transport system is superior compared to the alternatives, why has it not been implemented in practice?

1.2.2 Possible issues

While it is clear that implementing a large-scale demand-responsive transport system would require significant investments, it is not clear whether there would be enough demand for such a service were it implemented.

For example, it might not be realistic nor beneficial from the social point of view to think that a conventional heavy rail system was replaced by demand-responsive minibuses, due to the high efficiency of heavy rail. Moreover, traditional public transport in general has many significant advantages compared to demand-responsive transport: Taking into account the current experience from DRT services, a major issue can be seen to be the reliability of the service. So far, estimating ride times accurately in a service with no fixed routes has proven to be a major challenge, not least because of the human drivers, who are required to follow routes that are constantly changing, and the differences in their driving styles. Another disadvantage of DRT arises when customers are required to book their trips in advance and thus commit themselves to the service or payment at the time the trip is booked. In traditional bus services, this problem does not exist since customers may adjust their personal schedules dynamically according to known timetables, without pre-commitments. A commitment to a trip can be even more binding than in a taxi service: A normal taxi can wait for some time for the customer, for example, if the customer is at home when the taxi arrives, but it might not be reasonable that a demand-responsive minibus with customers on board would wait many minutes for one customer with the expense of other customers.

While demand-responsive transport has many advantages when compared to the private car as argued above, the car has many characteristics that are hard to compensate with public services. Firstly, a person who has already invested in a car and thus settles with yearly taxes and maintenance costs, is often not willing to use other transportation services since it would cost more than the marginal cost of using the car. Secondly, the car is unbeatable in terms of flexibility: It is available at any time of the day without planning beforehand. Even if a demand-responsive transport service accepted immediate requests without a minimum pre-order time, the customer would still be committed to wait for the designated vehicle to arrive. Thirdly, the private car is thought to be the most convenient way of carrying large amounts of luggage and goods. The car is also often used for storing equipment, which is not likely to be possible in a public service.

Despite the above issues related to large scale demand-responsive transport, the concept should be studied carefully. Even if the private car has its advantages in the current state of the world, it may become practically useless in congested urban areas.

1.3 Problem statement

This work is focused on the discrete and combinatorial problems arising in the planning of public transport in general and demand-responsive transport in particular. The main goals are (i) to develop models for a priori studying different forms of demand-

	Deterministic model	Stochastic model
Single vehicle / single customer	Publications I, V	Publications III, II
Multiple vehicles and customers	Publication VI	Publication IV

Table 1.1. Classification of publications.

responsive transport without having to implement them in practice and (ii) to develop algorithms for solving combinatorial problems related to public transport.

Generally, public transport can be viewed as a market where demand affects supply and vice versa. For a given demand, we are interested in the optimal actions for the transport operator and for a given supply, we study the optimal actions for customers with specific travel needs. By relaxing both demand and supply, we seek to determine the economic equilibrium, where the demand meets the supply. More precisely, the three main research questions considered in this work are stated as follows:

In Chapter 2: *Vehicle routing*, the demand for transportation is assumed to be known at the passenger level. The research question is: Using a given fleet of vehicles, what is the best way to satisfy the demand?

In Chapter 3: *Journey planning*, the routes of transport services are assumed to be known during a specific time horizon, and the travel times are assumed to be stochastic. Taking into account the uncertainty in transport services, what is the best way for a commuter to travel from a given origin to a given destination?

In Chapter 4: *Economic equilibrium*, we define the demand model by assuming that customers seek trips with small travel times and the supply model by assuming that transport service providers aim to maximize profit. Given these models, at which point does the demand for transportation meet the supply and how do regulation policies affect the economic equilibrium?

The remainder of this work is organized as follows. Section 2.1 in Chapter 2 presents deterministic algorithms for constructing a route for a single vehicle serving a set of customers (Publication I and Publication V). An extension to the multi-vehicle case is discussed in Section 2.2 (Publication VI). Chapter 3 studies a stochastic journey planning problem involving a single customer and a set of transport services (Publication III and Publication II). Finally, Chapter 4 introduces a stochastic network model, with multiple vehicles and stochastic demand, for determining the economic equilibrium in a transport network (Publication IV). The classification of the publications is summarized in Table 1.1.

2. Vehicle Routing

A vast majority of theoretical studies related to demand-responsive transport are formalized as combinatorial optimization problems involving the construction of vehicle routes with respect to a set of customers, whose pick-up and drop-off points are known a priori [78]. This problem is often referred to as the *dial-a-ride problem*. A demand-responsive transport service operating in real time induces the following major challenge: In order to be able to compete with private cars, service should be available within a short period of time from the trip request. This calls for routing algorithms capable of adapting to high-demand situations, since the modifications in vehicle routes have to be executed in real-time, possibly in a distributed fashion. In order to ensure a sufficient level of service, the customers' waiting and ride times should be relatively limited. Thus, the vehicle dispatching algorithms should be designed in a way that the constrained nature of the problem is taken into account.

In Section 2.2, we study the *multi-vehicle dial-a-ride problem* which involves the construction of a set of vehicle routes serving a set of customers. The multi-vehicle problem can be decomposed as a set of *single-vehicle dial-a-ride problems* (Section 2.1). That is, solutions to the single-vehicle problem can be used as subroutines in environments with multiple vehicles [67, 68].

2.1 Single-vehicle dial-a-ride problem

Different types of dial-a-ride services give rise to different types of mechanisms for controlling vehicle operations. For example, if a dial-a-ride service requires customers to request service during the previous day, the nature of vehicle dispatching will certainly differ from a service in which customers may request immediate service. This Section focuses on the *single-vehicle DARP with time windows*, in which the goal is to determine the optimal route for a single vehicle serving a certain set of customers.

The consideration of time windows means that the vehicle route is restricted by time limits for pick-up and delivery of each customer. Narrow time windows emerge in transport services, in which each customer is given an estimate or guarantee regarding the pick-up and delivery times in the form of time windows. These time windows are examined as hard limits to be met by the vehicle. Time windows have been incorporated in many early and recent studies of the DARP, see for example [68, 44, 53, 77, 21, 26, 81, 20, 11]. In these studies it is noted that in dynamic settings, time windows eliminate the possibility of indefinite deferment of customers and strict time limits help provide reliable service.

In [67], the objective function is defined as a generalization of the objective function of the Traveling Salesman Problem (TSP), in which a weighted combination of the time needed to serve all customers and of the total degree of dissatisfaction is minimized. In [68], the approach is extended to handle time windows on departure and arrival times, but only the route duration is minimized.

In Publication I, both aspects of the problem (general objective function and time windows) are considered. The main contribution is a solution method designed in a way that (i) it is capable of handling practically any objective function suitable for dynamic routing, and (ii) the computational effort of the algorithm can be controlled smoothly: If the problem size is reasonable, the algorithm produces optimal solutions efficiently and as the problem size increases, the search space may be narrowed down in order to achieve locally optimal solutions with a small computational effort.

Publication V discusses a single-vehicle algorithm based on hyperlink-induced topic search [48] for maximizing the number of customers in a single vehicle route. The algorithm is designed to find feasible solutions to highly constrained instances without defining a specific objective function. This approach is motivated by the fact that in such instances, the number of feasible solutions becomes so limited that often any feasible solution will be close to the optimal solution [68]. The method is seen to be useful in determining the feasibility of multi-vehicle instances (Section 2.2).

2.1.1 Problem formulation

The single-vehicle dial-a-ride problem is defined as follows [11]. Let $G = (V, A)$ be a complete and directed graph with node set $V = \{0\} \cup P$, where node 0 represents the depot, and P represents the set of pick-up and drop-off nodes, where ($|P| = 2n$). The set P is partitioned into sets P^+ (pick-up nodes) and P^- (drop-off nodes). Each arc $(i, j) \in A$ has a non-negative travel time T_{ij}. With each node $i \in V$ associate a time window $[E_i, L_i]$, a service duration D_i and a load q_i, where $D_0 = 0$ and $q_0 = 0$. Let $H = \{1, \ldots, n\}$ be the set of customers and let $T^{\max}$ be the maximum ride time for any customer. With each customer i is associated a pickup node $i^+ \in P^+$, a delivery node $i^- \in P^-$ and a load $q_{i^+} = -q_{i^-}$. The time parameters are illustrated in Figure 2.1.

Let Q be the capacity of the vehicle. A *route* is a directed circuit over a set of nodes in P, starting and finishing at node 0. The goal is to construct a vehicle route such that: (i) for every customer i, the pick-up node is visited before the drop-off node; (ii) the load of the vehicle does not exceed the capacity Q at any time; (iii) the ride time of each customer is at most $T^{\max}$; (iv) the service at node i begins within the interval $[E_i, L_i]$; (v) a specific cost function is minimized.

Reference [67] defines the cost function as a linear combination of route duration and the total dissatisfaction of customers. The adaptive insertion method described in Publication I is designed in a way that several cost functions, that are thought to be suitable for dynamic settings [43], may be incorporated with minimal work.

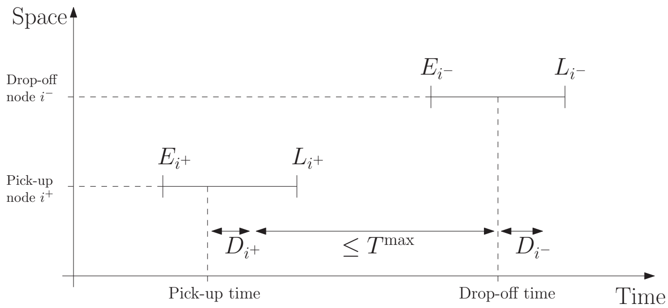


Figure 2.1. Pick-up and drop-off time windows. The pick-up point of customer i is denoted by i^+ and the drop-off point is denoted by i^-. The customer should be picked up at i^+ within the time window $[E_{i^+}, L_{i^+}]$ and the customer should be dropped off at i^- within the time window $[E_{i^-}, L_{i^-}]$. The service times needed for the customer to get on the vehicle and get off the vehicle are denoted by D_{i^+} and D_{i^-}. The time between the drop-off and the pick-up (excluding D_{i^+}) should not exceed the maximum ride time $T^{\max}$.

2.1.2 Adaptive insertion algorithm [Publication I]

In general, exact procedures for solving routing problems are computationally very demanding, since the complexity is always more or less equal to the classical traveling salesman problem. In addition, exact optimization can be seen to be needless at run-time if routes are modified often. Despite these facts, exact algorithms are useful in the sense that the performance of different heuristics may be compared to the optimal solution.

The main idea in the adaptive algorithm presented in Publication I is that customers are added to the vehicle route one by one by using an *exhaustive insertion* method, which leads to a globally optimal solution, that is, a vehicle route which is feasible with respect to all customers and minimizes a given cost function.

Many studies related to the dial-a-ride problem, see for example [44, 53, 26, 81], make use of what is called the insertion procedure, in the classical version of which the pick-up and delivery node of a new customer are inserted into the *current optimal sequence* of pick-up and delivery nodes of existing customers. That is, the classical insertion algorithm does not take into account the fact that the insertion of a new customer may render the optimal sequencing of existing customers no longer optimal.

The main idea in the *adaptive insertion algorithm* is to construct the optimal route iteratively by implementing an insertion algorithm for each customer, one by one *for all feasible sequences of pick-up and delivery nodes of existing customers*. Namely, the procedure involves two steps for each customer:

1. Perform insertion of the new customer to all feasible service sequences with respect to existing customers.
2. Determine the set of feasible service sequences with respect to the new customer and existing customers.

It can be readily shown that the insertion of a new customer to all feasible service sequences with respect to existing customers produces all feasible service sequences with respect to the union of existing customers and the new customer and leads to a globally optimal solution but is computationally expensive if the number of feasible service sequences grows large. However, if the route is constructed under relatively narrow time window constraints, the number of feasible routes with respect to all customers will be small compared to the number of all legitimate routes. Furthermore, the algorithm is easily extended to an adjustable heuristic algorithm capable of handling any types of time windows.

The idea of the advanced insertion method is clarified by the following example, where no capacity or time constraints are taken into account.

Example. Let i^+ denote the *pick-up node* of customer i and let i^- denote the *delivery node* of customer i. A *service sequence* is defined as an ordered list consisting of pick-up and delivery nodes. For instance, the service sequence (i^+, j^+, j^-, i^-) indicates the order in which customers i and j are picked up and dropped off.

Let us start the advanced insertion process with customer 1. Since the pick-up 1^+ of customer 1 has to be before the delivery, 1^-, the only possible service sequence at this point is $(1^+, 1^-)$. Thus, the set of potential service sequences with respect to customer 1 consists of this single service sequence. By insertion of customer 2 into the service sequence $(1^+, 1^-)$ we get the six service sequences presented in Table 2.1.

By inserting the pick-up and delivery node of customer 3 into all of these service sequences we get a total of $6(5 + 4 + 3 + 2 + 1) = 90$ new potential service sequences. However, if the time and capacity constraints are taken into account, not all service sequences described above are necessarily feasible.

Table 2.1. Potential service sequences with respect to customers 1 and 2. No capacity or time constraints are taken into account. i^+ denotes the pick-up node and i^- denotes the delivery node of customer i.

A:	1^+	1^-	2^+	2^-	B:	1^+	2^+	1^-	2^-	C:	1^+	2^+	2^-	1^-
D:	2^+	1^+	1^-	2^-	E:	2^+	1^+	2^-	1^-	F:	2^+	2^-	1^+	1^-

In this way the algorithm produces the set S_N of all feasible routes with respect to customers $1, \ldots, N$. Then the solution to the problem is obtained by choosing the sequence $s \in S_N$ with minimal cost $C(s)$, where $C(\cdot)$ denotes the cost function, for example, route duration.

In the worst case, the number of feasible solutions is of order $O(\sqrt{N}(N^2/2)^N)$. However, an average case study of the exact algorithm discussed in Section 2.1.4 suggests that as time limits reduce the number of feasible solutions, instances with up to 20 customers can be solved up to optimality with reasonable computational effort. In addition, the algorithm has a special property of being extendable to an adjustable heuristic, as described in the following section.

A heuristic extension

Even if the capacity and time constraints were not highly restrictive, the algorithm can be modified easily by bounding the size of the set S_i of service sequences, in which new customers are inserted, by including only a maximum of L service sequences for each customer i. More precisely, if after inserting customer i, the number of feasible service sequences with respect to customers $1, \ldots, i$ is larger than L, the set of feasible service sequences S_i with respect to customers $1, \ldots, i$ is narrowed by including only L service sequences that seem to allow the insertion of remaining customers. After the last customer has been inserted, the feasible service sequences are evaluated by means of the objective function.

This modification leads to a heuristic algorithm, in which the computational effort can be controlled by the parameter L, referred to as the *degree* of the heuristic. The resulting algorithm is somewhat sophisticated in a way that it produces globally optimal solutions for small sets of customers. When the number of customers is increased, the algorithm still produces locally optimal solutions with reasonable computational effort. In the special case where $L = 1$, the algorithm reduces to the classical insertion algorithm. If $L \geq \frac{(2N)!}{2^N}$, the heuristic coincides with the exact version of the algorithm as no routes are discarded.

Objective functions

In order to be able to efficiently make use of the above heuristic extension idea, the set of service sequences is narrowed by means of a certain heuristic objective function after the insertion of each customer. Since the main purpose of heuristics at the operational level is to always produce some implementable solutions very quickly, even if they were only locally optimal, such an objective function should be defined in a way that the algorithm is capable of producing *feasible* solutions even if the complexity of the problem was high.

Looking only at the cost may eliminate from consideration sequences that are marginally costlier but would easily allow the insertion of remaining customers in the route. Thus, more sophisticated criteria should be considered to help ensure that the heuristic will find a feasible solution when one exists.

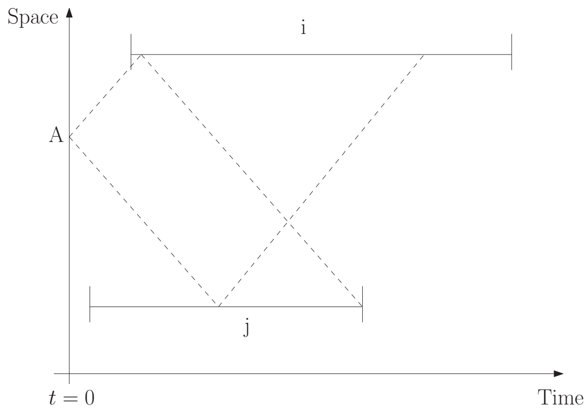


Figure 2.2. Route flexibility. A vehicle is located at A at $t = 0$, and two customers are due to be picked up within the presented time windows at i and j. The dashed lines represent two possible routes for the vehicle. If the route duration were minimized, i should be visited before j. However, since there would be no "slack time" at j, this decision would a priori exclude the possibility that new customers could be inserted between i and j. On the other hand, if j were visited before i, there would be more possibilities for inserting new customers on the route before i.

In other words, the function should favor service sequences with enough *time slack* for those customers, that have not been inserted into the sequences (see Figure 2.2). Publication I suggests the following heuristic objectives.

Route duration favors service sequences in which the time to serve all customers is as small as possible. This objective can be justified by the fact that it is likely that new customers may be inserted at the rear of a route that is executed quickly.

Total time slack stores sequences in which the sum of excess times (or the average excess time) at the nodes is maximized, that is, sequences which are likely to allow the insertion of a new customer before the last node.

Max-min time slack seeks sequences in which the minimum excess time at the nodes of the route is maximized. In other words, the sequences in which there is at least some time slack at each node are considered potential.

2.1.3 Routing by ranking [Publication V]

Several web information retrieval (IR) methods have been developed for finding the most appropriate web pages corresponding to queries given to search engines. The most sophisticated methods, such as HITS [48], PageRank [13] and SALSA [52] in use today make use of the hyperlinked structure of the web, since the goodness of a web page and the position of the page with respect to other web pages seem to have a certain connection. For example, a web page may be considered good if there are many other web pages linking to that page. In other words, web pages are ranked by search engines not only by means of the content of the page, but also by exploiting information regarding the hyperlink-induced relationships between pages.

The bringing of hyperlinks to bear on the ordering of web pages has given rise to a mathematical analysis related to hyperlink-induced web IR methods, such as in [28, 51, 61, 1], in which the behaviour of several IR methods is studied from the computational point of view.

The HITS (Hyperlink-Induced Topic Search) algorithm defines *authorities* (web pages with several inlinks) and *hubs* (several outlinks). The HITS thesis is that *good hubs point to good authorities and good authorities are pointed to by good hubs*. Based on this thesis, both a hub score and an authority score is assigned to each web page [51].

Publication V presents an application of HITS on the dial-a-ride problem (DARP) [9, 10, 11]. Here the DARP is examined as a *constraint satisfaction problem*, in which the goal is to find a feasible vehicle route that serves all customers, or to maximize the number of served customers.

In the context of the dial-a-ride problem, the links between nodes are defined as feasible transitions with respect to the constraints of the problem: If node j can be visited after i, a link from i to j is formed. Thus, a good hub score of a pick-up or drop-off node i means that many nodes can be reached in time from i. Thus, in order to efficiently find feasible solutions to the dial-a-ride problem, we suggest that nodes with large hub scores should be visited first since there are many nodes that can be visited after such nodes.

The HITS algorithm [48]

Given a web graph $G = (V, A)$ consisting of pages V and links A between pages, the authority and hub scores a_i and h_i are computed for each page i as follows. Letting (i, j) represent a link from page i to page j, given that each page has been assigned an initial authority score $a_i(0)$ and hub score $h_i(0)$, HITS successively refines these scores by computing

$$a_i(k) = \sum_{(j,i)\in A} h_j(k-1), \qquad h_i(k) = \sum_{(i,j)\in A} a_j(k-1)$$

for $k \in \{1, 2, \ldots\}$. By using matrix notation, these equations can be written the form $a(k) = L^T h(k-1)$ and $h(k) = La(k-1)$, where $a(k)$ is the authority vector containing the authority scores of each of the pages at step k, $h(k)$ is the corresponding hub vector and L is the adjacency matrix of the graph with elements $L_{ij} = 1$ if $(i, j) \in A$ and $L = 0$ otherwise [51].

It has been shown in [28] that the authority and hub vectors describing the authority and hub scores of nodes of a given graph in the limit are given by the dominant eigenvectors of the matrices $L^T L$ and LL^T (or equivalently, dominant singular vectors of L), where L is the adjacency matrix of the graph.

Modified HITS

For constrained routing problems, a modified version of the HITS algorithm is used, in which only the hub scores are considered. More precisely, the thesis is that *good hubs point to good hubs*. This formulation is motivated by Theorem 1 below, which states that for a specific class of graphs, the hub score of a node i corresponds to the number of self-avoiding paths from i to a given destination node. When attempting to construct a path that visits all nodes, or to maximize the number of visited nodes, the modified HITS idea induces the following intuitive policy: Good hubs are visited first, since many nodes can be reached from good hubs.

The hub scores are calculated as follows. Let L denote the adjacency matrix of a directed graph G. Similarly as in the HITS algorithm, the hub vector containing the *hub scores* of nodes is first initialized, $h(0) = (1, 1, \ldots, 1)$ and the hub vector is successively updated by means of the power method

$$h'(k) = Lh(k-1). \tag{2.1}$$

Similarly as in the original HITS algorithm, the hub vector converges to a dominant eigenvector of L.

Theorem 1 characterizes the hub scores produced by the modified HITS algorithm for *sink graphs* defined as follows.

Definition. *Let $G = (V, A)$ be a directed acyclic graph and let $s \in V$ be a node such that $(s, i) \notin A$ for all $i \in V$. The graph $G_s = (V, A \cup (s, s))$ is called a* sink graph.

In other words, a sink graph is a directed acyclic graph (V, A) with the exception that one node $s \in V$ with zero outdegree is associated with a loop (s, s).

Theorem 1. *Let L denote the adjacency matrix of a sink graph $G_s = (V, A)$, where $V = \{1, \ldots, |V|\}$, let h_i denote the number of self-avoiding paths from i to s for $i \in V \setminus \{s\}$ and let $h_s = 1$. Then, $h = (h_1, \ldots, h_{|V|})^T$ is a unique dominant eigenvector of L.*

Note that since $h_i \geq h_j$ for all $i, j \in V$ for which $L_{ij} = 1$, the vector h defines a *topological ordering* [23] of the nodes for which $h_i > 0$. The result gives us an idea of the behaviour of the modified HITS method: There are many paths beginning from nodes with high hub scores.

In the following section, we show how the hub scores are used to guide a backtracking algorithm for the dial-a-ride problem.

The ranking method

Publication V presents an exact constraint programming method for the single-vehicle DARP that produces a feasible solution whenever one exists or proves that the problem is infeasible. In the latter case, the algorithm returns a route that maximizes the number of served customers.

Letting node $0 \in V$ be the location of the vehicle at $t = 0$, we aim to find a feasible path $(0, p_1, \ldots, p_{2n}, 0)$ consisting of the pick-up and drop-off nodes of n customers. The number of permutations is $(2n)!$ and the number of feasible permutations with respect to precedence constraints is $(2n)!/2^n$.

Briefly, the approach to the problem is a *depth-first* search, in which the remaining nodes are ranked by means of hub scores at each step and the order of the depth-first search is determined by the ranking. The search begins from node 0 by ranking the pick-up and drop-off nodes P of all customers. Then, the node p^* with the highest ranking is added to the sequence and the ranking procedure is repeated for the remaining nodes $P \setminus \{p^*\}$.

More formally, at each recursion step, we have the current sequence S and the set of remaining nodes R_S (nodes that can possibly be added to the sequence). Then, the remaining nodes R_S are ranked by hub scores. The recursion continues for all sequences (S, i), where $i \in R_S$ in ranked order. In the following, we describe the ranking of remaining nodes in more detail.

Hub scores

The ranking of the remaining nodes R_S is determined in two phases: First, the adjacency matrix of the remaining nodes is determined by studying *feasible transitions* between the nodes. Then, the hub vector is calculated by means of Equation (2.1).

The definition of feasible transitions is similar to the one studied in [27], in which the set of *admissible arcs* between the pick-up and delivery nodes is constructed as a preprocessing step in the pickup and delivery problem. The set of admissible arcs is made up of arcs which a priori satisfy the precedence, capacity and time constraints of the problem. The difference in Publication V is that the feasibility of transitions is defined as a function of the state of the vehicle (see Figure 2.3).

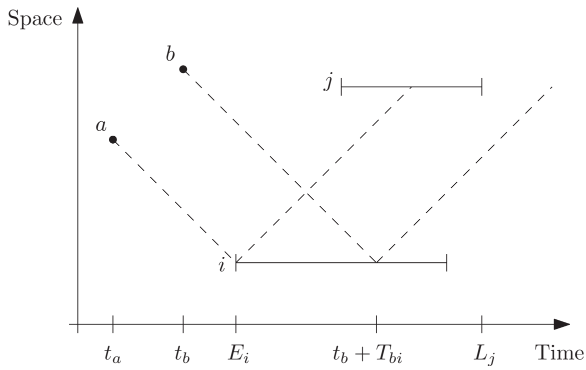


Figure 2.3. Feasible transitions. The figure shows the locations and time windows of two nodes i and j. If the vehicle left from a at the instant t_a, visited node i and moved directly to node j, the time constraint L_j would be satisfied. However, assuming that at the instant t_b the vehicle is located at b, the transition $i \to j$ is seen to be S-infeasible.

The elements of the adjacency matrix are defined for all $i, j \in R_S$ by $L_{ij} = 1$, if $i \neq j$ and $i \to j$ is S-feasible, $L_{ij} = 0$ otherwise.

After the adjacency matrix L has been determined, the hub vector $h = (h_1, \ldots, h_{|R_S|})$ of L is determined by Equation (2.1) (here the remaining nodes $i \in R_S$ are numbered from 1 to $|R_S|$ for clarity). A large hub score of node i means that many nodes can be visited after i. The hub ranking method is defined as follows.

Definition (Hub ranking)**.** *The remaining nodes $i \in R_S$ are sorted in descending order of the hub scores h_i: The branches are evaluated in the order $i_1, \ldots, i_{|R_S|}$, where $h_{i_1} \geq h_{i_2} \geq \ldots \geq h_{i_{|R_S|}}$.*

The hub scores are used to give guidance to the algorithm regarding the order in which the depth-first search visits the nodes. In the best case, no infeasible states are encountered and thus the complexity is of order $O(n^3)$.

A heuristic extension

The effort of the algorithm can be controlled by limiting the number of branches that are evaluated by means of a positive parameter J. For example, if $J = 1$, the algorithm constructs a single sequence and stops when the set of remaining nodes is empty. By increasing J the search space is expanded and if $J = (2n)!/2^n$ (the number of permutations that satisfy precedence constraints), the heuristic coincides with the exact algorithm.

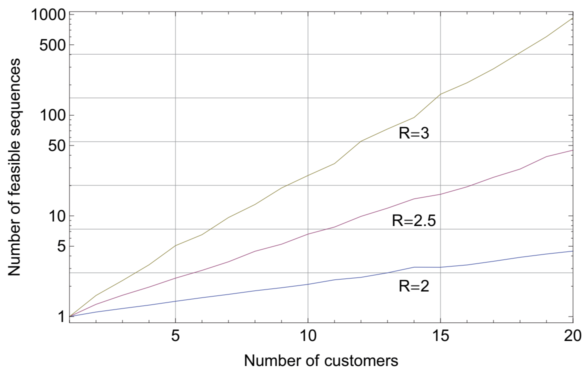


Figure 2.4. Experiment 1. The complexity of the exact algorithm with respect to the number of customers on a logarithmic scale. The three curves represent, as a function of the number of customers, the average number of feasible sequences for $R = 2, 2.5, 3$ and $\mu = 1800$s.

2.1.4 Numerical experiments

The following paragraphs present a summary of the computational results reported in Publication I. The exact and heuristic versions of the single-vehicle advanced insertion algorithm were tested on a set of problems involving different numbers of customers and different time window widths determined by a travel time ratio R describing the maximum allowed ratio of travel time to direct ride time. The pick-up and drop-off points of customers were chosen randomly from a square-shaped service area and the ride times between the points were modeled by euclidean distances.

At first, the complexity of the problem was studied with respect to three parameters, namely i) the number of customers N, ii) travel time ratio R and iii) the average time interval μ between customer requests.

Then, the performance of the heuristic with different objective functions was evaluated. The complexity was measured in terms of the number of sequences evaluated by the heuristic.

The experiments were performed on a standard laptop computer with a 2.2 GHz processor. The CPU times and the number of evaluated sequences appeared to have a roughly linear relationship. A typical problem instance involving 20 customers could be solved up to optimality within less than a second.

Experiment 1: Number of customers

Figure 2.4 shows, as a function of the number of customers, the average number of feasible sequences in feasible problem instances on a logarithmic scale.

Referring to the figure, it can be seen that the complexity of the problem increases exponentially with respect to the number of customers with all studied values of the travel time ratio. In addition, the complexity is increased with the travel time ratio.

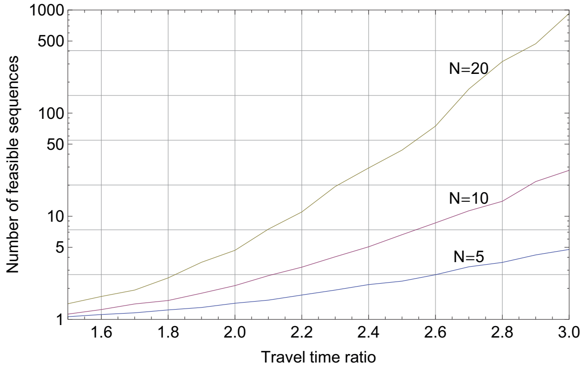


Figure 2.5. Experiment 2. The complexity of the exact algorithm with respect to travel time ratio on a logarithmic scale. The three curves represent, as a function of the travel time ratio, the average number of feasible sequences for $N = 5, 10, 20$ and $\mu = 1800$s.

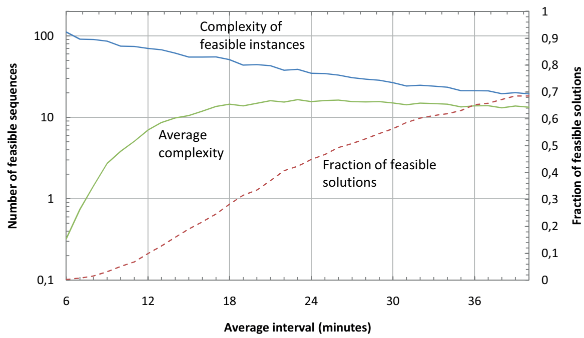


Figure 2.6. Experiment 3. The complexity of the exact algorithm with respect to the average time interval between customers. The solid lines represent the average number of feasible sequences in feasible problem instances and all problem instances, on a logarithmic scale for $N = 10$ and $R = 3$. The dashed line corresponds to the fraction of problem instances, for which at least one feasible solution was found.

Experiment 2: Travel time ratio

Figure 2.5 shows, as a function of travel time ratio, the average number of feasible sequences in feasible problem instances on a logarithmic scale.

The figure shows that the effect of the travel time ratio on the complexity of the problem is significant. The fact that the slopes of the curves increase with R on the logarithmic scale indicates that the relation between complexity and R is super-exponential.

Experiment 3: Time interval

The solid lines in Figure 2.6 represent, as a function of average time interval between requests, the average number of feasible sequences in (i) feasible problem instances and (ii) all problem instances, on a logarithmic scale. The dashed line corresponds to the fraction of problem instances, for which at least one feasible solution was found.

The figure indicates that the complexity of feasible problem instances decreases

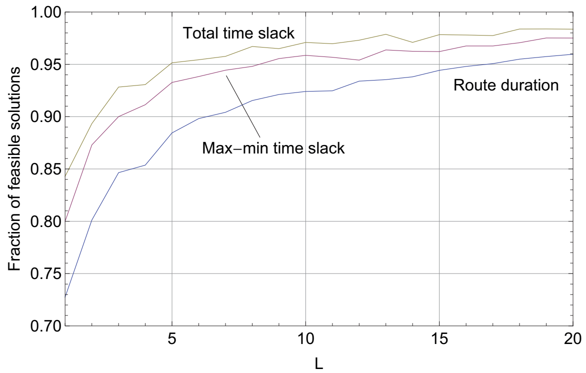


Figure 2.7. Experiment 4. The performance of three different heuristic objective functions as functions of degree L. The curves represent the fractions of instances for which a feasible solution was found by the objective functions (compared to the exact algorithm), for $N = 20$, $R = 3$ and $\mu = 1800$s. The total slack time objective function outperforms the other two in all studied cases.

exponentially with respect to the average time interval μ. On the other hand, the probability of finding at least one feasible solution is increased with μ. By looking at the curve corresponding to the average complexity of all problem instances (including infeasible cases), it can be seen that the complexity is maximized at a certain time interval ($\mu = 24$ minutes in this case), in which both the probability of finding a feasible solution and the number of feasible sequences in feasible cases are relatively large.

Experiment 4: Objective functions

Let us study the performance of three different heuristic cost functions as a function of the degree L of the heuristic. Figure 2.7 shows the fraction of problem instances for which a feasible solution was found by the heuristic (compared to the exact algorithm).

Referring to the figure, it can be seen that the total time slack cost function is capable of finding a feasible solution to randomized problems most often, while the performance of the route duration cost function is worst of the three algorithms. Note that as the degree L is increased, the fraction of feasible solutions converges to 1 for any heuristic cost function, since whenever $L \geq \frac{(2N)!}{2^N}$, the heuristic coincides with the exact algorithm regardless of the studied problem.

2.2 Multi-vehicle dial-a-ride problem

Most recent studies related to the dial-a-ride problem are related to the multi-vehicle case, in which a set of vehicle routes is designed for a predefined set of customers, see for example [21, 20, 8, 70, 84, 56, 63, 33, 9, 11, 12]. Reference [11] examines the multi-vehicle dial-a-ride problem as a *constraint satisfaction problem*, in which the goal is to find a set of m feasible vehicle routes that serve all customers, where m is the number of vehicles, or to prove that such a set of routes does not exist. In this reference, it is noted that an algorithm for checking the feasibility of a multi-vehicle DARP instance has two main applications: 1) Determining the feasibility can be the first phase in an optimization algorithm in a *static* setting, where all trip requests are known, for example, one day in advance. 2) In *dynamic* services, a constraint satisfaction algorithm can be used for deciding whether to accept or reject incoming user requests.

2.2.1 The maximum cluster algorithm [Publication VI]

Publication VI presents the following exact approach to the multi-vehicle DARP as a constraint satisfaction problem. By using the ranking algorithm described in Section 2.1.3 as a subroutine, the method produces a feasible solution to any instance or proves that the instance is infeasible. The main idea is that the vehicle routes are constructed one by one, each maximizing the number of served customers in the set of remaining customers. The customers are denoted by numbers $1, \ldots, n$ and the vehicles are denoted by numbers $1, \ldots, m$.

First, a route is constructed for vehicle 1, serving as many customers as possible. Then, the process is repeated with vehicle 2 for the set of customers that were not served by vehicle 1 and so forth (see Figure 2.8). If a feasible solution is not found directly, the process is repeated by attempting to add the remaining customers to the existing vehicle routes.

The above approach produces a set of customer-vehicle assignments such that for each vehicle there exists a feasible route serving all customers assigned to the vehicle or proves infeasibility by going through all possible sets of customer-vehicle assignments.

2.2.2 A priori screening

Since the number of possible partitions of n customers into m sets is equal to the Stirling number of the second kind, that is, $|\mathcal{P}| = S_2(n, m) = \frac{1}{m!} \sum_{i=0}^{m} (-1)^i \binom{m}{i} (m - i)^n$, going through all possible partitions is computationally taxing. However, some instances are rendered infeasible by studying routes consisting two customers: If all possible routes consisting of customers $\{i, j\}$ are infeasible, an arc between i and j is formed (customers i and j can not be assigned to the same vehicle). Then, we find the *maximum clique* C within the set of customers (the largest set for which there is

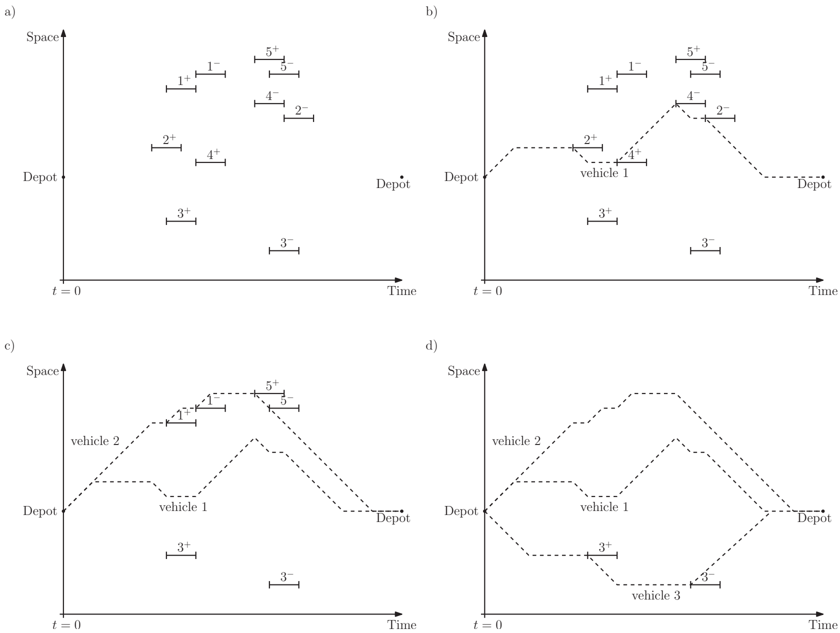


Figure 2.8. A one-dimensional example of the approach to the multi-vehicle problem involving five customers and three vehicles. The first route is constructed by maximizing the number of customers that can be served by a single vehicle (Figure b). Then, the customers 2 and 4 served by the first vehicle are removed from the set of remaining customers and the process is repeated for the second vehicle (Figure c). Finally, a route is constructed for the third vehicle, serving the last remaining customer 3 (Figure d).

an arc between all $i, j \in C$). If the size $|C|$ of the maximum clique is greater than the number m of vehicles, the instance is infeasible.

Otherwise, since the customers in the maximum clique $C = \{c_1, \ldots, c_{|C|}\}$ all have to be assigned to different vehicles, with no loss of generality we may initially assign customer c_j to vehicle j for all $j \in \{1, \ldots, |C|\}$. Then, the set of *feasible vehicles* V_i for each customer i is determined by noting that if there is an arc between i and $c_j \in C$, customer i may not be assigned to the same vehicle as c_j.

Note that assigning each customer i a vehicle number $v_i \in V_i$ defines a unique partition. Since the vehicle number of customer i can be chosen from the set V_i of feasible vehicles, the number of a priori feasible partitions is given by $|\mathcal{P}'| = \prod_{i=1}^{n} |V_i|$, which yields the following result.

Theorem 2. *The number of a priori feasible partitions satisfies* $|\mathcal{P}'| \leq m^{n-|C|}$.

In summary, the worst case complexity is high especially when the set of a priori feasible partitions $\mathcal{P}'$ is large. However, as will be seen in the next section, solutions to problems with loose constraints are usually found with little effort. On the other hand, tight constraints reduce the set of a priori feasible partitions $\mathcal{P}'$ and thus also the complexity of the algorithm.

2.2.3 Numerical experiments

In the following, an algorithm using the ranking idea is compared with two existing solution methods, namely, a tabu search algorithm [21] and a constraint programming (CP) algorithm [11].

The ranking algorithm was implemented in Matlab and the tests were performed on a 2.2 GHz Dual Core Intel PC. The tabu and CP algorithms were tested on a 2.5 GHz Dual Core AMD Opteron computer [9].

In the studied instances the pick-up and drop-off points are located in a 20×20 square and the ride times between points (in minutes) are equal to Euclidean distances. The time windows have 15 minutes of length. The instances are described in more detail in [20, 69, 11].

The results of the tests are shown in Table 2.2. The first column shows the instance labels of the form *am-n* or *bm-n*, where *m* indicates the number of vehicles and *n* corresponds to the number of customers. The other columns show the average time (in seconds) needed to solve the instances and the corresponding modifications by using the different solution methods, calculated over ten runs. A number in parentheses indicates that the instance was proven to be infeasible, a dash indicates that a solution was not found in three minutes computing time and a star indicates that results for the instance have not been reported.

By looking at the results obtained by the ranking algorithm we see that most instances were solved within a fraction of a second. Except for a single modified instance (b7-84, 75% of vehicles), the ranking algorithm produced a feasible solution or proved infeasibility in all instances. Note that the ranking algorithm produced a feasible solution or proved infeasibility in the modified instances with six vehicles and 48 customers (b6-48), for which results have not been previously reported.

The CPU times obtained by the ranking algorithm are typically of order ten times smaller compared to the results of the tabu and CP algorithms. The best improvement factor is $109.5/0.04 \approx 2700$ compared to CP (instance b5-60) and $78.5/0.06 \approx 1300$ compared to tabu (instance a4-48, 75% of vehicles). Although the algorithms were tested on different platforms, the results seem to justify the efficiency of the ranking algorithm on the test instances. We acknowledge that there are instances in which tabu and CP produced a feasible solution or proved infeasibility faster than the ranking algorithm. However, the results suggest that the ranking algorithm may have practical importance since it is capable of handling large problems in short computation times.

Instance	Original			RT=30			RT=22			75 % of vehicles		
	Tabu	CP	Ranking	Tabu	CP	Ranking	Tabu	CP	Ranking	Tabu	CP	Ranking
a4-40	0.8	0.5	0.02	0.8	0.5	0.02	0.5	0.3	0.05	1.7	0.3	0.64
a4-48	1.0	0.5	0.05	1.0	0.5	0.05	1.3	0.4	0.05	78.5	0.6	0.08
a5-40	0.3	0.3	0.02	0.3	0.3	0.02	0.3	0.3	0.02	1.3	0.3	0.02
a5-50	0.7	0.5	0.04	0.7	0.5	0.04	0.6	0.6	0.03	3.9	1.3	2.0
a5-60	1.4	1.0	0.05	1.4	1.0	0.05	1.5	0.9	0.05	-	24.5	64.7
a6-48	0.4	0.6	0.03	0.4	0.6	0.03	0.4	0.7	0.03	1.1	0.5	0.07
a6-60	1.0	5.6	0.04	1.0	5.6	0.04	-	(1.1)	(0.63)	11.6	6.2	10.7
a6-72	1.9	5.0	0.06	1.9	5.0	0.06	-	(1.7)	(0.77)	5.7	2.0	0.90
a7-56	0.5	1.8	0.04	0.5	1.8	0.04	-	(0.6)	(0.40)	0.9	1.5	0.06
a7-70	1.7	41.5	0.06	1.7	41.5	0.06	1.4	7.6	0.06	3.2	5.1	0.06
a7-84	2.7	3.4	0.08	2.7	3.4	0.08	3.0	4.1	0.08	7.5	3.5	0.07
a8-64	0.8	6.2	0.05	0.8	6.2	0.05	-	(1.9)	(0.84)	1.1	2.5	0.04
a8-80	1.5	8.5	0.07	1.5	8.5	0.07	1.8	6.0	0.07	3.0	3.6	0.07
a8-96	3.5	37.7	0.11	3.5	38.7	0.11	-	(3.7)	(1.72)	8.1	5.3	0.15
b4-40	0.6	0.4	0.02	0.4	0.3	0.05	0.4	0.3	0.05	-	(0.1)	(0.40)
b4-48	1.3	0.4	0.03	1.6	0.4	0.08	-	(0.1)	(0.32)	-	(0.1)	(0.59)
b5-40	0.4	0.3	0.03	0.4	0.4	0.03	-	(0.1)	(0.27)	-	(0.1)	(0.37)
b5-50	1.2	6.8	0.06	0.9	0.8	0.05	1.1	0.9	0.18	-	(0.1)	(0.58)
b5-60	1.5	109.5	0.04	1.8	3.1	0.04	1.6	1.2	0.04	-	(0.1)	(0.78)
b6-48	0.3	*	0.02	*	*	0.02	*	*	0.02	*	*	(0.54)
b6-60	0.9	5.1	0.04	1.5	1.5	0.04	1.0	1.4	0.04	5.3	3.8	3.2
b6-72	2.1	27.3	0.05	2.3	2.4	0.06	2.4	2.4	0.05	9.7	25.6	5.1
b7-56	0.5	13.3	0.04	0.6	1.5	0.03	0.5	1.5	0.04	2.1	3.9	0.19
b7-70	1.4	5.6	0.08	1.6	18.4	0.08	1.3	2.7	0.05	12.6	78.7	4.3
b7-84	3.0	25.8	1.60	2.9	6.3	0.08	-	(0.1)	(0.64)	-	-	-
b8-64	0.7	12.7	0.09	0.8	1.9	0.09	0.8	1.9	0.04	1.8	4.8	0.98
b8-80	1.9	23.9	0.07	1.8	19.2	0.07	-	(0.5)	(1.28)	4.0	13.9	0.27
b8-96	4.1	149.1	0.12	3.9	34.8	0.12	3.9	56.1	0.15	8.2	-	0.74

Table 2.2. Comparison between a tabu search algorithm [21], a constraint programming algorithm [9] and the ranking algorithm. The results obtained by the first two algorithms are given in [9, 11]. The upper table shows the average time (in seconds) needed to solve the instance of the dial-a-ride problem in the first column by using the different solution methods, calculated over ten runs. The times without parentheses indicate that a feasible solution was found and the times in parentheses indicate that the instance was proven to be infeasible. A star (*) indicates that the computing time has not been reported.

3. Journey planning

The urban journey planning problem involves determining a path, possibly involving transfers between different transport modes, from a specified origin to a similarly specified destination in a transport network. Common criteria used for evaluating journeys include the total duration, number of transfers and cost [2, 89, 64, 58, 41, 42, 40, 75, 76, 88, 14, 19, 15, 17, 49, 36, 5, 73].

Reference [89] classifies these journey planning models into the following types of formulations: 1) the headway-based model, in which a constant headway for each transit line is assumed [83] and 2) the schedule-based model, which assumes a fixed route and timetable for each transit line.

Publication II extends approach 2) into a dynamic model taking into account the uncertainty of transport services (buses, trams, trains, ferries, ...). In contrast to existing itinerary planning algorithms designed for scheduled public transport networks, where the path is a priori optimized with respect to an objective, for example, [89, 2], the realized journey may differ from the original plan.

Passenger information systems provide real-time information on the status of transport services (buses, trams, trains, ferries, ...) via mobile devices and displays at public transport stops. This makes it possible for a commuter to dynamically modify the planned journey in case of a delay or cancellation. For example, if a transfer from a transport service to another is unsuccessful due to a delay, the commuter may reconsider the remaining path to the destination. Clearly, the importance of being able to modify the planned journey dynamically is emphasized when the number of transfers between different transport services is increased.

Taking into account the uncertainty in transport services is particularly important in difficult weather conditions when delays are common. In addition to traditional public transport with fixed schedules, uncertainty should be given special attention in flexible transport services without fixed routes [60]. If the vehicle routes are modified in real time, the estimation of travel times between subsequent stops is more difficult than in the case of fixed routes.

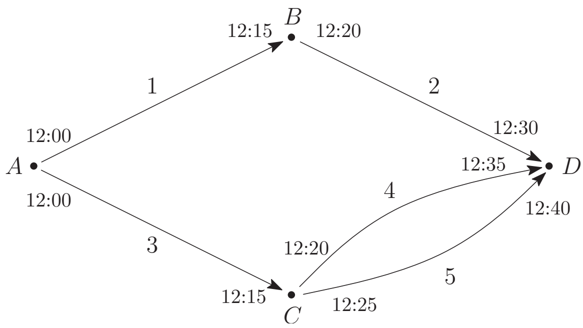


Figure 3.1. The difference between stochastic and deterministic journey planning for a commuter traveling from A to D. The four points represent public transport stops (A, B, C, D) and the arrows between them represent public transport services $(1, \ldots, 5)$. Initially, there are three possible journeys from A to D: $(1,2), (3,4)$ and $(3,5)$. If the commuter initially chooses service 1, the success of the journey is dependent of the success of the transfer from 1 to 2 at stop B. If the commuter chooses service 3 first, the destination is reached if one of the transfers $3 \rightarrow 4$ or $3 \rightarrow 5$ is successful at stop C.

A simplified example clarifying the main difference between dynamic and a priori journey planning is shown in Figure 3.1.

Generally, a commuter wishes to travel from an origin node v_o to a destination node v_d within a time horizon $[0, T]$ using different transport services. Each transport service is represented as a sequence of *legs*. Each leg is associated with a *start node* and *end node*, as well as a random *start time* and *end time*. Adjacent nodes in the network are connected with similarly defined walking legs.

A path from the origin to the destination is represented as a sequence of legs, in which the start node of each leg is equal to the end node of the previous leg. We assume that during the execution of a leg, the commuter receives information on which services have already visited the end node and which are yet to arrive. In other words, the customer "sees" the available successor legs of the current leg and may choose to (i) stay in the vehicle, (ii) transfer to another vehicle or (iii) get off the vehicle and start walking towards a nearby stop (or the destination). The goal the above context is to determine an optimal policy specifying the actions that are executed in different situations in order to optimize reliability, ride time, waiting time, walking time, the number of transfers of a combination of these objectives.

Dynamic path finding problems (see [35, 6, 32, 57, 25, 47, 46, 4, 30, 74, 80]) are often modeled as *Markov decision processes* [66, 65], in which the *actions* of a decision maker at a given *state* are independent of all previous actions and states. Publication II presents a conditional Markov model, in which the path history is included in each state by defining states as sequences of legs in the transport network. That is, the current state is determined by the path taken so far. This model is further approximated in Publication III, where the current state is defined as the current leg.

In addition to public transport, a journey planning problem with a similar objective arises in freight transportation by for-hire carriers.

3.1 Stochastic model of a scheduled network [Publication III]

Let $\mathcal{V}$ denote a set of *nodes* representing public transport stops in a specific area and let $\mathcal{K} \subset \mathbb{N}$ denote a set of public transport *services* operating in this area, indexed by natural numbers.

The route of each service $k \in \mathcal{K}$ is represented as a sequence of nodes $(v_1^k, \ldots, v_m^k)$ in $\mathcal{V}$. Service k departs at node v_1^k at a specific time and proceeds to nodes $v_2^k, \ldots, v_m^k$ in the order determined by the route. The *expected passing time* of service k at node v_j^k is denoted by t_j^k. Thus, a service k can be represented as a sequence of nodes and expected passing times $((v_1^k, t_1^k), \ldots, (v_m^k, t_m^k))$, see Figure 3.2.

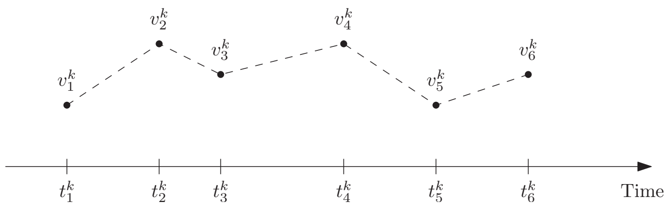


Figure 3.2. The route and schedule of a transport service. The points represent the nodes v_j^k that define the route of service k and the real numbers t_j^k on the timeline represent the expected passing times of the service at the nodes.

3.1.1 Service legs and walking legs

Each service $k \in \mathcal{K}$ is decomposed as a set of scheduled *legs* between subsequent stops. That is, each leg has a start node, end node, expected start time and expected end time. By this decomposition, any path in the transport network can be represented as a sequence of legs.

After each service leg, the commuter may choose to continue the journey by foot. Thus, with each service leg is associated leg a set of walking legs beginning at the end of the service leg, see Figure 3.3a.

3.1.2 Transfers

A transfer from leg i to leg j is possible only if leg j begins at the node from which i ends. Letting $\mathcal{L}$ denote the set of legs and S_i denote the *successor set* of leg i, the

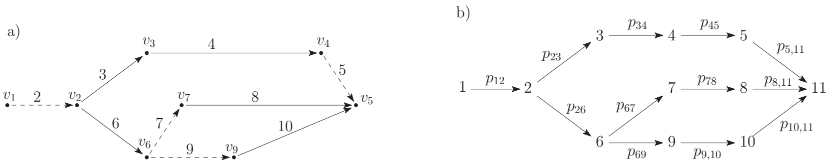


Figure 3.3. a) A sample transport network with eight stops and five transport services consisting of a single leg (solid arrows $3, 4, 6, 8, 10$). Adjacent stops are connected with walking legs (dashed arrows $2, 5, 7, 9$). b) A directed graph representing the relations of legs in Figure 3.3a. The origin and destination are represented by legs 1 and 11. The transfer probability from leg i to leg j is denoted by p_{ij}. Note that $p_{12} = p_{45} = p_{67} = p_{69} = 1$, since $2, 5, 7$ and 9 are walking legs.

transfer probability from leg i to leg j is defined as a real number p_{ij} satisfying

$$p_{ij} \begin{cases} = 1, & \text{if } j \in S_i \text{ and } k_j < 0 \quad (j \text{ is a walking leg}), \\ \in [0, 1], & \text{if } j \in S_i \text{ and } k_j > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Clearly, the legs and transfer probabilities form a directed acyclic graph, as in [66], see Figure 3.3b. For analysis, we assume that the transfers between legs are independent events.

Note that transfers to walking legs are always possible. However, the success probability of a transfer between successive legs of the same service may be less than one due to vehicle breakdowns. Moreover, in the determination of transfer probabilities, one should take into the account the fact that the breakdown of a vehicle running on rail tracks blocks the way for other vehicles using the same tracks.

3.1.3 Paths

A path can be represented as a sequence of legs $(i_1, \ldots, i_m)$ satisfying $i_{h+1} \in S_{i_h}$ for all $h \in \{1, \ldots, m-1\}$, see Figure 3.4. The path is *successful* with probability $\prod_{h=1}^{m-1} p_{i_h i_{h+1}}$ and the *subjective price* $C((i_1, i_2, \ldots, i_m))$ of the path is defined as a linear combination of expected waiting time, number of transfers, expected ride time, expected walking time and reliability.

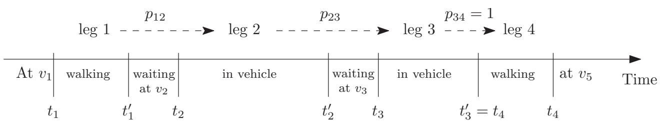


Figure 3.4. Representation of a path. The ticks on the time axis denote the schedule of a path from v_1 to v_5. The path is represented by four legs: 1) A commuter starts walking from the origin v_1, and arrives at stop v_2. 2) A transport service departs at v_2 and travels to stop v_3. 3) A transport service departs at v_3 and arrives at stop v_4. 4) The commuter continues by foot to the destination v_5. Note that the path is successful with probability $p_{12}p_{23}$.

3.2 Problem solution

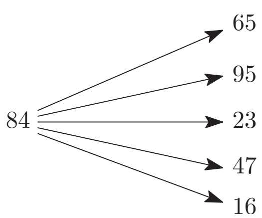
In Section 3.2.1, we model the journeying problem as a Markov Decision Process and propose an algorithm for maximizing the general objective function. In Section 2.1.3, we present a simplified algorithm for maximizing the reliability of a journey.

3.2.1 Markov Decision Process

In order to be able to handle a general objective function, we propose a finite-state Markov decision process, where the goal is to determine an optimal *policy* with respect to the objective function.

A policy defines for each leg $i \in \{1, \ldots, n-1\}$ a *ranking* of successor legs $i' \in S_i$. The policy is used to choose the best transfer options during the journey as follows: At the end of each leg $i \in \{1, \ldots, n-1\}$, the transfer options to successor legs $i' \in S_i$ are revealed. After the transfer options are revealed, letting $X \subset S_i$ denote the set of successor legs to which a transfer is possible, the commuter transfers to the leg $x \in X$ with the highest ranking, see Figure 3.5.

The above approach produces relatively easy to follow travel information, since the commuter knows the ranking of successor legs before each transfer. Note that the policy can be re-optimized during the journey if the transition probabilities between legs are updated by using real-time traffic information. That is, a new optimal policy for the remaining part of the journey can be calculated at $t \in [0, T]$ by using the updated transition probabilities available at t. This type of dynamically adaptive policy could be useful in situations in which there is substantial uncertainty in travel times.



Ranking of successors of leg 84	End of leg 84: transfer options are revealed
1 : 23	~~1 : 23~~
2 : 16	2 : 16 ←
3 : 65	~~3 : 65~~
4 : 47	~~4 : 47~~
5 : 95	5 : 95

Figure 3.5. A ranking example. The left column of the table shows the a priori ranking of successors of leg 84. When the commuter is at the end of leg 84, the available transfer options (16 and 95) are revealed. The commuter then transfers to leg 16, which has the highest ranking among the possible transfer options.

Formally, the parameters of the Markov decision process $(S, A, P_{\cdot}(\cdot, \cdot), R_{\cdot}(\cdot, \cdot))$ are defined as follows.

States = Legs

The set of *states* S is equal to the set of legs[1] numbered from 1 to n, that is, $S = \{1, \ldots, n\}$. State 1 is referred to as the *origin state* and state n is referred to as the *destination state*.

Actions

The set of *actions* A consists of sets A_s of actions available at states $s \in S$. An action $a \in A_s$ is defined as a *preference order* of the successor states $s' \in S_s$, that is, a bijection $a : S_s \to \{1, \ldots, |S_s|\}$, where $a(s')$ denotes the ranking of the successor state $s' \in S_s$ in the preference order. The successor states of s ranked by the preference order a are denoted by $s^{a,a(s')}$. Given the the sorted successor states $s^{a,1}, \ldots, s^{a,|S_s|} \in S_s$ of s, the commuter transfers to state $s^{a,k}$ if (i) the transfer to $s^{a,g}$ is unsuccessful for $1 \leq g < k$ and (ii) the transfer to $s^{a,k}$ is successful.

Transition probabilities

$P_a(s, s')$ is the probability that action a in state s at step t will lead to state $s' \in S_s$ at step $t + 1$. Given the preference order $s^{a,1}, \ldots, s^{a,|S_s|} \in S_s$ defined by action $a \in A_s$, since successful transfers are assumed to be independent[2] events, we have

$$P_a(s, s^{a,k}) = \sum_{k=1}^{h} p_{ss^{a,k}} \prod_{g=1}^{k-1} (1 - p_{ss^{a,g}}), \tag{3.2}$$

where $p_{ss^{a,k}}$ denotes the transfer probability from s to $s^{a,k}$, as defined in Equation (3.1).

Rewards

$R(s, s')$ is the expected immediate reward received after transition from state $s \in S$ to state $s' \in S$ with transition probability $P_a(s, s')$. Since the objective is to minimize the subjective price of the trip, $R(s, s')$ is defined as the difference in subjective price due to the transition from s to s'.

3.2.2 Optimal policy

The solutions to Markov decision processes are characterized as policies, that is, functions π that specify the action $a(s)$ that the commuter chooses when in state s. The goal is to find an optimal policy, that is, a policy that maximizes the expected reward. Generally, the calculation of an optimal policy requires two arrays indexed by state: value V, which contains real values, and policy π which contains actions. The value $V(s)$ of s corresponds to the expected rewards to be earned by following a policy that maximizes the expected rewards from s onwards.

[1] A more detailed model described in Publication II defines states as sequences of legs.

[2] See Publication II for a model in which the transfers between services are not independent.

Reference [7] defines the value $V(s)$ by

$$V(s) := \max_{a \in A_s} \left\{ \sum_{s' \in S_s} P_a(s, s') \left(R(s, s') + V(s') \right) \right\} \tag{3.3}$$

for all $s \in S$. Note that $V(n) = 0$ since $S_n = \emptyset$.

An optimal policy is characterized as follows: When at state s, the available transfer options $R \subset S_s$ to successor states are revealed. The commuter transfers to a state $s' \in R$ for which $R(s, s') + V(s')$ is maximized. Thus, an optimal action a at state s is defined by

$$R(s, s^{a,1}) + V(s^{a,1}) \geq \ldots \geq R(s, s^{a,|S_s|}) + V(s^{a,|S_s|}), \tag{3.4}$$

where the successor states ranked by action a are denoted by $s^{a,a(s')}$ for all $s' \in S_s$. Equation (3.4) gives an optimal action for a state s, given that the values $V(s')$ of its successor states are known. In the following, we present an algorithm for calculating the values for all states that are reachable from the origin state.

3.2.3 Backward induction algorithm

The values $V(s)$ of states can be determined by means of backward induction. The recursive function Rec(s) executes the following procedures at each recursion step:

1. Check if the value of the current state s has been determined. If yes, return the value.
2. For all successors s' of the current state: Calculate the value of s' by calling Rec(s').
3. Determine the optimal action for the current state s by ranking the successors.
4. Calculate and return the value of the current state.

By executing Rec(1), the program recursively calculates the values and optimal actions for all states s that are reachable from the origin state 1. Initially, we only know the value of the destination state, that is, $V(n) = 0$. Thus, the first states for which the value can be calculated are the ones that precede the destination state. The algorithm then proceeds backwards until the value $V(1)$ of the origin state is calculated.

Since the algorithm involves sorting, the complexity is bounded above by $O(nm \log m + nm) = O(nm(1 + \log m))$, where $m = \max_{s \in \{1,\ldots,n-1\}} |S_s|$ is the size of the largest successor set. Note that although the total number of possible actions for a successor set S_s equals $|S_s|!$, the complexity of the algorithm is smaller due to the fact that the successors of each state are ordered only once. That is, the algorithm goes through all states reachable from the origin state but not through all possible actions.

3.2.4 Expected number of paths

In the following we present a straightforward method for maximizing the reliability of a journey by redefining the value of a state. Namely, the value of state $s \in \{1, \ldots, n-1\}$ is defined as the *expected number of successful paths* from s to the destination state n. The approach is motivated by the idea that paths that allow several detours are considered more reliable than paths with no alternatives (see Figure 3.1).

Definition. *Let $\mathcal{P}(s)$ denote the set of paths from state $s \in \{1, \ldots, n-1\}$ to state n. For each path $r \in \mathcal{P}(s)$, let $P(r)$ denote the probability of success of r. The value of state s is defined by $h_s = \sum_{r \in \mathcal{P}(s)} P(r)$.*

In this formulation, when the commuter is at state s and sees the available transfer options F to successor states, the commuter transfers to the state $s' \in F$ for which $R(s, s') + h_{s'}$ is maximized.

Theorem 3 establishes a relation between eigenvectors and the expected number of successful paths h_s. For this purpose, we define a (weighted) *sink graph* as follows.

Definition. *Let $G = (X, A)$ be a weighted directed acyclic graph, where a weight $p_{ss'} \in [0, 1]$ is assigned to each arc $(s, s') \in A$ and let $k \in X$ be a node such that $(k, s) \notin A$ for all $s \in X$. The graph $G_k = (X, A \cup (k, k))$, where $p_{kk} = 1$, is called a* sink graph.

Similarly as in Definition 2.1.3, a sink graph is a directed acyclic graph (X, A) with the exception that one node $k \in X$ with zero outdegree is associated with a loop (k, k).

Theorem 3. *Let P denote the adjacency matrix of a sink graph $G_k = (X, A)$, where $X = \{1, \ldots, |X|\}$, let h_s denote the expected number of successful paths from s to n for $s \in X \setminus \{k\}$ and let $h_k = 1$. Then, $h = (h_1, \ldots, h_{|X|})^T$ is a unique dominant eigenvector of P.*

Theorem 3 states that by constructing a sink graph, the expected number of successful paths h_s from state s to the destination state n is given by the dominant eigenvector of the adjacency matrix P consisting of the transfer probabilities $p_{ss'}$ between states (and $p_{nn} = 1$).

The expected number of successful paths can also be calculated for all states that are reachable from the origin state by means of a procedure similar to the one described in Section 3.2.3. Since calculating the expected number of successful paths does not involve sorting, the complexity of the algorithm is of order $O(n + |A|)$, where A is the set of transfers for which $p_{ss'} > 0$. Letting $m = \max_{i \in \{1, \ldots, n-1\}} |S_i|$ denote size of the largest successor set, the complexity is bounded above by $O(n(m + 1))$.

3.3 Numerical experiments

In the following, computational results obtained by backward induction are presented. The transfer probabilities between legs are determined by assuming gamma distributed ride times similarly as in [71, 18]. More precisely, given the expected ride time t_i of leg i, we define the ride time τ_i as a random variable $\tau_i \sim \text{Gamma}(\alpha t_i, \beta, \delta t_i)$, where the $\text{Gamma}(\alpha, \beta, \delta)$ distribution is defined by the probability density function

$$f(x) = \frac{(x-\delta)^{\alpha-1} e^{-(x-\delta)/\beta}}{\beta^{\alpha}\Gamma(\alpha)} \quad \text{for } x > \delta \geq 0. \tag{3.5}$$

The numerical examples are restricted to calculating the the expected number of feasible paths and the probability of reaching the destination during a pre-defined time horizon. However, a general objective function defined as a combination of expected waiting time, number of transfers, expected ride time, expected walking time and reliability is incorporated with minimal effort (see Equation (6) in Publication III).

3.3.1 Description of instances

The Helsinki tram network consists of ten tram lines, operated by 50 trams, and 154 stops. The tram schedules of Helsinki Region Transport are available at [39] and the instances are available at `http://math.tkk.fi/%7elehame/dju`.

In each instance, the origin and destination nodes v_o and v_d are chosen randomly from the set of 154 tram stops, see Figure 3.6. The departure time is set equal to 9:00 and the length of the time horizon is defined by $T = 1.6d(v_o, v_d)$, where $d(v_o, v_d)$ is the expected duration of the shortest path from v_o to v_d in the tram network. Each (tram) service k operating within the time horizon is determined by a *scheduled route* extracted from the timetables, that is, a sequence $((v_0^k, t_0^k), \ldots, (v_p^k, t_p^k))$ of nodes, where t_i^k is the expected passing time at node v_i^k for $i = 0, \ldots, p$. Each service is decomposed as a set of legs.

3.3.2 Results

The computational results of ten instances are shown in Table 3.1. For each instance the name, the probability $V(1)$ of reaching the destination during the time horizon, the expected number of successful paths h_1, and their respective computation times are given. By looking at the computation times, we see that all instances were solved within a fraction of a second. In addition, calculating the expected number of successful paths is slightly faster than calculating probability, which involves sorting.

Since the expected number of successful paths represents the number of alternatives, even a large value of h_1 does not guarantee that the destination will be reached in time (b10). On the other hand, small values of h_1 seem to correspond to small prob-

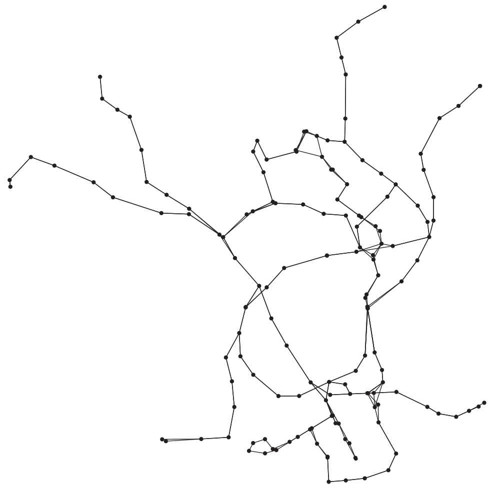


Figure 3.6. The tram network of Helsinki consisting of ten tram lines and 154 stops.

Table 3.1. Computational results. The table shows the probability $V(1)$ of reaching the destination leg n from the origin leg 1, the expected number of successful paths h_1 from 1 to n and the corresponding computation times in seconds.

Instance	Probability		Exp. number of paths	
	CPU time (s)	$V(1)$	CPU time (s)	h_1
b1	0.23	0.97	0.15	106.88
b2	0.06	1.	0.04	42.31
b3	0.06	1.	0.04	20.72
b4	0.06	0.46	0.04	1.25
b5	0.31	0.97	0.22	7.05
b6	0.1	0.91	0.07	2.67
b7	0.09	0.85	0.06	12.72
b8	0.05	0.7	0.03	0.7
b9	0.12	0.99	0.08	25.67
b10	0.08	0.79	0.06	83.34

abilities (b4 and b8). For instances b2 and b3 with $V(1) = 1$, the expected number of successful paths gives additional information: Instance b2 provides more alternative routes for the commuter than b3.

4. Economic equilibrium

Publication IV introduces a stochastic network model to characterize the behavior of customers and transport operators in a demand-responsive transport market. Customers seek rides to minimize travel time and transport operators aim to maximize profit. Demand-responsive transport is considered as a complementary service to conventional public transport services and customers are assumed to choose the utility-maximizing alternative from the available transport modes. An *economic equilibrium* is defined as a state in which the demand meets the supply of trips: the choices of transport operators do not change if the demand remains constant and the choices of customers do not change if the supply of trips remains constant.

This approach is somewhat similar to the taxi model proposed in [85]. The main difference is that in a taxi service, customers are delivered to their destinations directly, whereas in a demand-responsive transport service, a vehicle can serve several customers simultaneously and therefore a customer's trip from an origin to a destination is not necessarily a direct one. In other words, the route of a taxi is determined by two points (origin and destination), but demand-responsive transport routes may include several stops, similarly as bus routes.

The stochastic model for demand-responsive transport is governed by the following preliminary assumptions.

1. There are N vehicles that produce trips between origins and destinations in a specific operating zone. Different trips may have different prices and travel times. A single vehicle can simultaneously serve several customers.

2. The subjective price of a trip is defined as a combination of ticket price and travel time. Customers choose between trips provided by DRT and a virtual mode[1] by comparing the subjective prices of different trips.

[1] The virtual mode represents alternatives for the DRT service.

4.1 Demand

Let us consider a set of *nodes* I representing the origins and destinations of customers in a specific operating zone. Each pair of nodes $(i, j) \in I \times I$ is associated with a specific *direct ride time* t_{ij}.

A *trip* from an origin $i_0 \in I$ to a destination $i_d \in I$ is defined as an acyclic sequence of nodes $(i_0, i_1, \ldots, i_{d-1}, i_d)$ in I. When a customer takes a trip $(i_0, i_1, \ldots, i_{d-1}, i_d)$, the customer enters a vehicle at i_0, which visits the nodes $i_1, \ldots, i_{d-1}$ before the drop-off of the customer at i_d. That is, the trip denotes the path of the vehicle that transports the customer from i_0 to i_d. For example, a *direct* trip from $i_0 \in I$ to $i_d \in I$, denoted by (i_0, i_d), describes a trip in which a customer enters a vehicle at i_0 and the vehicle drives directly to i_d without stopping between i_0 and i_d.

Each trip r has a specific ticket price p_r. Note that the prices of trips with the same origin and destination may be different. Each customer seeks a trip to minimize the *subjective price* g_r^{DRT}, defined as a combination of ticket price p_r and *travel time* t_r^{DRT}, that is,

$$g_r^{\mathrm{DRT}} = p_r + \beta t_r^{\mathrm{DRT}}, \tag{4.1}$$

where p_r is the ticket price for the trip r and β is the customers' monetary value of unit travel time.

The choices of customers are determined by a logit model as follows. The subjective trip price of a trip from i to j provided by the virtual mode is denoted by $\bar{g}_{ij}$. The probability that a customer traveling from i to j chooses a DRT trip $r \in \mathcal{R}_{ij}$, where $\mathcal{R}_{ij}$ denotes the set of trips from i to j, is defined by

$$P_r^{\mathrm{DRT}} = \frac{\exp(-\theta g_r^{\mathrm{DRT}})}{\exp(-\theta \bar{g}_{ij}) + \sum_{r' \in \mathcal{R}_{ij}} \exp(-\theta g_{r'}^{\mathrm{DRT}})}, \tag{4.2}$$

where θ is a nonnegative parameter describing the uncertainty in transport services and demand from the perspective of customers. Clearly, the above logit model has the property of independence from irrelevant alternatives, that is, the ratio $P_r/P_{r'}$ depends on the subjective prices of trips r and r' but not on the subjective prices of other trips [72].

The expected demand for a trip $r \in \mathcal{R}_{ij}$ is given by $Q_r^{\mathrm{DRT}} = Q_{ij} P_r^{\mathrm{DRT}}$, where Q_{ij} denotes the total demand from i to j, including the demand Q_{ij}^{DRT} for DRT and the demand for the virtual mode.

4.2 Supply

We assume that there are N vehicles available for transporting customers. At any point in time, each vehicle follows a specific route $(i_0, i_1, \ldots, i_m)$ determined by a sequence of nodes in the transportation network. The vehicle starts at the first node i_0 and proceeds by visiting the other nodes i_k for $k = 1, \ldots, m$ in the order determined by the route. Each node corresponds to a stop during which customers may enter or exit the vehicle.

During the execution of the route $(i_0, i_1, \ldots, i_m)$, the vehicle produces trips that are subsequences of $(i_0, \ldots, i_m)$. This idea of producing trips is in fact similar as in traditional public transport: a bus with a given route produces trips that are subsequences of the route.

The *state* of a vehicle describes which part of the route the vehicle is currently executing. Each time a vehicle following a route $(i_0, \ldots, i_m)$ arrives at stop i_k, we say that the vehicle *transfers* to a new state. Similarly as routes, the states are defined as sequences of nodes. The difference between states and routes is that the state of a vehicle corresponds to the *remaining part* of its current route. That is, the vehicle transfers to a new state each time it arrives at a stop, even if the route remains unchanged. During the execution of a route $(i_0, \ldots, i_m)$, the vehicle successively transfers to states $(i_0, \ldots, i_m), (i_1, \ldots, i_m), (i_2, \ldots, i_m), \ldots, (i_{m-1}, i_m)$.

4.3 Competitive market

Most existing demand-responsive transport services provide door-to-door transportation for elderly or handicapped people and require customers to book trips at least one hour in advance [22, 38]. Conventionally, the trips are organized centrally via *travel dispatch centers*, which have the capability of assigning customers to vehicles and optimizing the routes [54]. In contrast to such centralized services, Publication IV considers a competitive form of demand-responsive transport in which each driver providing service attempts to maximize his/her profit. That is, the movement of vehicles is governed by the decisions of individual drivers, instead of a travel dispatch center controlled by a single transport operator. This market structure is in fact similar to conventional taxi-markets, which have been extensively studied, see for example [34, 3, 16, 31, 50, 82, 55, 29, 59, 86, 87, 85].

In a competitive market, the drivers attempt to transfer to states at which the *expected profit rate* is maximized.

By assuming a logit model similar to the one in Equation (4.2), the probability that a vehicle at state s transfers to state $s' \in \mathcal{S}$ is given by

$$P_{s,s'} = \begin{cases} \frac{\exp(\theta^d U(s'))}{\sum_{s'' \in \mathcal{S}_s} \exp(\theta^d U_{s''})}, & \text{if } s' \in \mathcal{S}_s \quad (\mathcal{S}_s = \text{ successor set of } s), \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

where θ^d is a nonnegative parameter reflecting the uncertainty on demand and DRT services from the perspective of drivers, $U(s)$ is the expected profit rate at state s and $\mathcal{S}_s$ is the set of states to which a transfer from state s is possible.

4.3.1 Network equilibrium

The drivers attempt to maximize profit rate by transporting as many customers per unit time as possible. Thus, we expect that the drivers prefer detours instead of direct routes in order to serve more customers. In some cases, however, producing only direct trips may be more profitable.

We also expect that the customers prefer direct trips instead of detours. However, if many vehicles produce non-direct trips, the travel times in non-direct trips may be smaller than in direct trips due to small waiting times.

The movement of vehicles is described by means of the *arrival rates* of vehicles at different states and the movement of customers is described by means of the demands for different DRT trips. A *network equilibrium* denotes a situation in which the arrival rates at states and demands for different trips meet. By using Brouwer's fixed point theorem, the following result is obtained.

Theorem 4. *For any finite transportation network, there exists a network equilibrium.*

The idea for finding a network equilibrium is to solve both the customer choice subproblem and the vehicle movement subproblem iteratively until a convergence criterion is met, similarly as in [85].

4.3.2 A three node example

Let us demonstrate the calculation of the competitive network equilibrium in a simple case in which the network consists of three nodes denoted by $1, 2, 3$, as shown in Figure 4.1. The set of states consists of the 12 possible sequences of two and three nodes. In large networks, one might want to limit the number of states by including only a part of all possible sequences, since the number of possible sequences of n nodes equals $n!$.

The solid lines in Figure 4.2 show the arrival rates T_s of vehicles at different states $s \in \mathcal{S}$ after each step of the equilibration algorithm. The black and grey dashed lines show on a logarithmic scale the convergence of the arrival rate vector T and the demand vector Q^{DRT}, respectively.

By looking at the solid lines, we see the oscillatory nature of the arrival rates: When the arrival rate of vehicles in a specific state s increases during the equilibration, the number of customers available for a single vehicle in that state decreases. This causes the drivers to choose other states instead of s. When the arrival rate at state

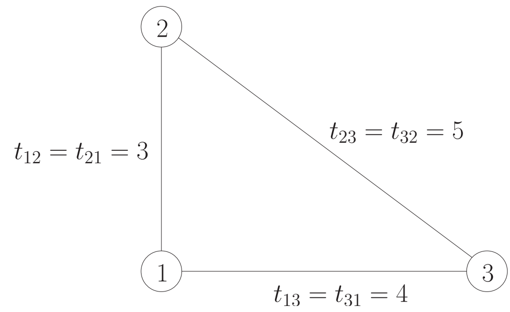


Figure 4.1. A three node example network. The distances between the nodes (in kilometers) are equal to the direct ride times t_{ij}.

s decreases, it becomes more profitable for individual vehicles and results in drivers choosing state s more often. In the network equilibrium, the arrival rate is highest at state $(2, 1, 3)$ and lowest at state $(2, 1)$.

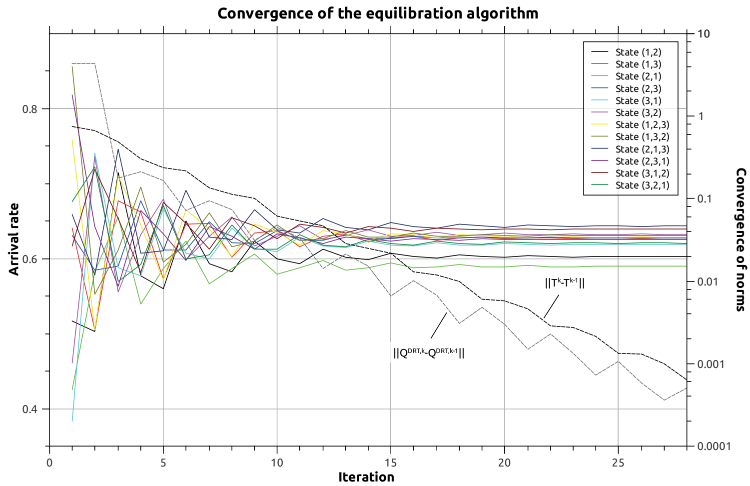
Referring to the dashed lines, which are roughly straight lines on a logarithmic scale, we see that the norms of the arrival rate and demand vectors converge exponentially with respect to the number of iterations.

4.3.3 Long-run example

In an unregulated demand-responsive transport service operated by a single operator, we expect that the number of vehicles and ticket price are chosen in a way that the total profit rate is maximized in the long run. In a competitive demand-responsive market with no entry limits, we expect that the number of vehicles increases as long as the profit rate of vehicles is positive. Regulating the number of vehicles and ticket price could improve the service from the customers' point of view as well as from the perspective of transport operators. In the following, we study different characteristics of demand-responsive transport as a function of the number of vehicles N and average price per kilometer p. The network equilibrium was calculated for different numbers of vehicles N and prices per kilometer p. For each combination of N and p, the total profit rate, profit rate per vehicle and customer surplus[2] were calculated. The results are shown in Figure 4.3.

The solid curves represent contour lines in which the total profit rate $U(N, p)$ of vehicles is equal $(= 0, 10, 20)$. In particular, the outermost contour line corresponding to $U(N, p) = 0$ encloses the *feasible region*, that is, the area in which the service is

[2] *Customer surplus* is defined as the difference between the subjective price of the virtual mode and the subjective price of DRT ,namely, by $\sum_{(i,j)\in I\times I} \sum_{r\in\mathcal{R}_{ij}} Q_r^{DRT}(\bar{g}_{ij} - g_r)$, where Q_{ij}^{DRT} is the total demand for DRT trips from i to j, $Q_{ij} - Q_{ij}^{DRT}$ is the demand for the virtual mode from i to j, $\mathcal{R}_{ij}$ is the set of DRT trips from i to j, g_r is the subjective price of trip r and $\bar{g}_{ij}$ is the subjective price of the virtual mode from i to j.



Arrival rates at states in the network equilibrium T^*												
State s	(1,2)	(1,3)	(2,1)	(2,3)	(3,1)	(3,2)	(1,2,3)	(1,3,2)	(2,1,3)	(2,3,1)	(3,1,2)	(3,2,1)
Arrival rate T_s	0.603	0.627	0.590	0.628	0.619	0.632	0.631	0.631	0.644	0.626	0.639	0.621

Figure 4.2. Convergence of the equilibration algorithm in the three node example. The solid lines in show the arrival rates T_s of vehicles at different states $s \in \mathcal{S}$ after each step of the algorithm. The black and grey dashed lines show on a logarithmic scale the convergence of the arrival rate vector T and the demand vector Q^{DRT}, respectively. The network equilibrium is defined by the arrival rates at different states shown in the table below the figure.

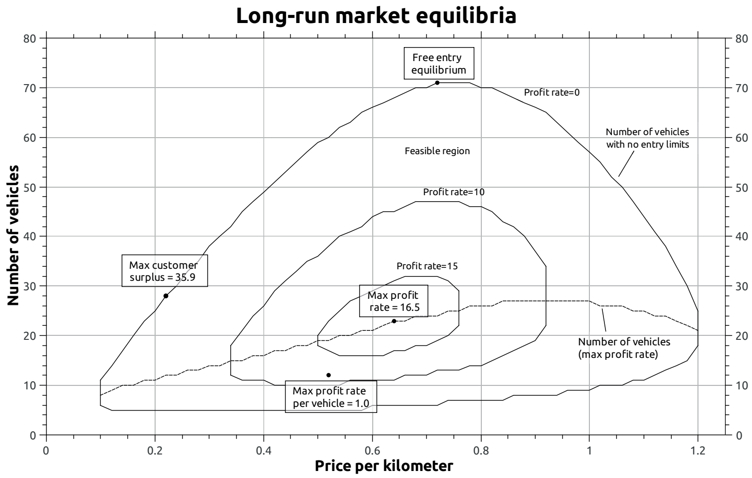
profitable for drivers. The dashed curve shows the number of vehicles for different prices for which the total profit rate is maximized. The points represent the long-run market equilibria defined in the previous section.

The table in Figure 4.3 shows the demand for DRT, total profit rate, profit rate per vehicle, customer surplus, the average ratio of travel time to direct travel time and the average number of customers in a single vehicle in the four equilibrium points.

By looking at the figure, we note that there is a significant difference between the equilibria. In the free entry equilibrium, the number of vehicles is significantly higher than in the other points. The average travel time ratio and the average occupancy are extremely low. This indicates that with no regulation, the DRT service would approach a taxi-type service, in which all customers are transported privately and all trips are direct trips.

The difference in price between the free entry equilibrium and the point in which the total profit rate is maximized is small. In addition, the number of served customers (demand for DRT) is only slightly smaller in the maximum profit rate case compared to the free entry case. However, maximizing the profit rate of vehicles would decrease the customer surplus and decrease the average level of service, as can be seen by looking at the average travel time ratio and average occupancy.

The profit rate per vehicle is maximized with an extremely small number of vehi-



Equilibrium	Number of vehicles	price per kilometer	Demand for DRT (customers/min)	Tot. profit rate / per vehicle (EUR/min)	Surplus (EUR/min)	Travel time ratio	Average occupancy
Max. customer surplus	28	0.22	16.1	0.13 / 0.005	35.9	1.24	2.4
Maximum total profit rate	23	0.64	10.8	16.5 / 0.7	3.9	1.28	1.9
Maximum profit rate per vehicle	12	0.52	8.4	12.6 / 1.0	0.2	1.50	3.0
Free entry equilibrium	71	0.72	12.4	0.1 / 0.001	8.7	1.11	0.7

Figure 4.3. Long-run market equilibria for the three node example. The first column shows the four studied equilibrium points. The second and third columns show the number of vehicles and price per kilometer with which the equilibrium is achieved. The remainder of the columns show the total demand for DRT, total profit rate and profit rate per vehicle, customer surplus, the average ratio of travel time to direct travel time and the average number of customers in a single vehicle.

cles. In this case, only a small number of customers could be served and the level of service would be poor.

The customer surplus is maximized by using a significantly lower ticket price than would be optimal from the perspective of drivers. This is mainly due to the fact that the low price results in a high demand for DRT. Moreover, we note that the customer surplus equilibrium is achieved by using price regulation exclusively. That is, the results suggest that the optimal solution from the customers' point of view would be to regulate price and allow free entry.

Reference [37] repeats the above example for a demand-responsive transport monopoly, where a single transport operator controls the transition probabilities of vehicles between states, in contrast to the competitive market in which the routing decisions are made by individual drivers. In this case, the matrix P containing the transition probabilities $P_{s,s'}$ between states is referred to as a *routing strategy* and the goal is to find a routing strategy that maximizes a given objective.

The results suggest that centralized routing strategies can be used to improve the efficiency of demand-responsive transport: By maximizing profit by means of a centralized strategy, the total profit rate, demand and social welfare are increased from the competitive equilibrium. The difference between the mechanisms is explained by the fact that in the competitive market, the routing decisions are based on the drivers' incomplete market information whereas in the case of a single transport operator, the routing strategy is optimized by using information on the state of the entire network.

5. Conclusions

This work presents mathematical models for demand-responsive transport and methods that can be used to solve combinatorial problems related to vehicle routing and journey planning in a transport network.

First, we show how the demand for transportation can be satisfied by constructing routes for a fleet of vehicles, assuming that the origins, destinations, and time limits of customers' trips are known. Then, by considering a stochastic journey planning problem in a public transport network we determine the optimal actions of commuters, assuming that the vehicle routes are fixed during a specific time horizon. Finally, we present a stochastic network model for determining the economic equilibrium in a transport network, that is, the point at which the demand meets the supply, by assuming that commuters attempt to minimize travel time and transport operators aim to maximize profit.

The proposed models can be used to simulate the operations of public transport services ranging from paratransit services for the elderly and disabled to large-scale demand-responsive transport services. These calculations can provide valuable information to public authorities and planners of transportation services, regarding, for example, regulation and investments. The new methods for solving vehicle routing and journey planning problems can be used to improve the performance of different types of intelligent transportation systems and to provide real-time travel information via mobile devices and electronic displays. In addition to public transport, potential applications of the proposed algorithms include freight transportation, courier and food delivery services, military logistics, and air traffic.

The main contributions to scientific methodology are summarized a follows.

The *adaptive insertion algorithm* introduced in Publication I generalizes the insertion algorithm which is used to solve routing problems. The computational complexity of the adaptive algorithm can be controlled smoothly, closing the gap between a greedy heuristic and complete enumeration.

Publication V presents the *routing by ranking* method, which connects recommendationtype link analysis to combinatorial path-finding problems. The idea is based on HITS, an eigenvector algorithm originally developed for web information retrieval.

The *maximum cluster algorithm* (Publication VI) is used to solve constrained routing problems with multiple vehicles by finding maximal sets of customers that can be assigned to a single vehicle. The article also introduces the *clique detection method*, which finds a maximal set of customers, such that no pair of customers in the set can be served by a single vehicle due to the constraints of the problem. This method is used to narrow down the search space and to detect infeasible problem instances.

Publication III and Publication II model the journey planning problem in a scheduled transport network as a Markov Decision Process (MDP). The *actions* of the MDP are defined as *preference orders* of possible alternatives in each part of the journey. The articles also show how the calculation of an optimal policy can be accelerated by assuming history independence and how the reliability of a journey is maximized by considering the *expected number of successful paths* to the destination. In this context, a *path* is defined as a sequence of scheduled legs in the transport network and for each pair (i, j) of successive legs there is a specific transfer probability from i to j.

Publication IV introduces a stochastic network model for determining the economic equilibrium for demand-responsive transport. The existence of such an equilibrium is proved by using Brouwer's fixed point theorem. The model is used for optimizing fleet size and pricing as well as studying the effects of different regulation policies.

The following directions for future work are suggested: One could attempt to extend the routing by ranking algorithm to handle different types of cost functions. For example, the reliability of solutions could be optimized by defining travel times as random variables. The dynamic journey planning models could be enhanced by partitioning public transport stops into clusters, which would reduce the complexity of dynamic journey planning in large transport networks. Finally, the economic equilibrium model could be used to study the feasibility of demand-responsive transport services in different types of real-life scenarios.

As a conclusion, it can be stated that there are many computational results that support the technical viability of demand-responsive transport. State-of-the-art algortihms are capable of efficiently solving complex routing problems with multiple vehicles (see for example Table 2.2. in Chapter 2). However, as suggested in [24], one of the key issues in a large scale demand responsive service ever becoming a reality, is the institutional inertia against change in transit paradigms. No models exist that are directly applicable in finding to what extent a completely new transportation system is possible in real life. How to accurately estimate the demand for a hypothetical transportation service remains a relatively open question. In order to be able to access such practical problems, future work calls for real-life pilot services, which would probably give valuable information on the demand for and performance of demand-responsive transport.

Errata

In Publication III, Theorem 1, L should be replaced with P.

Bibliography

[1] M. Agosti and L. Pretto. A theoretical study of a generalized version of kleinberg's hits algorithm. *Information Retrieval*, 8:219–243, 2005.

[2] K. N. Androutsopoulos and K. G. Zografos. Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. *European Journal of Operational Research*, 192:18–28, 2009.

[3] R. Arnott. Taxi travel should be subsidized. *Journal of Urban Economics*, 40(3):316–333, 1996.

[4] A. Azaron and F. Kianfar. Dynamic shortest path in stochastic dynamic networks: Ship routing problem. *European Journal of Operational Research*, 144:138–156, 2003.

[5] J. Bander and C. C. White. A new route optimization algorithm for rapid decision support. *Proc. Vehicle Navigat. Inf. Syst. Conf. Part 2*, P-253:709–728, 1991.

[6] J. L. Bander and C. C. White III. A heuristic search approach for a nonstationary shortest path problem with terminal costs. *Transportation Science*, 36:218–230, 2002.

[7] R. Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, (6), 1957.

[8] R. Bent and P. Van Hentenryck. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research*, 33(4):875–893, 2006.

[9] G. Berbeglia. *Complexity Analyses and Algorithms for Pickup and Delivery Problems*. Ph.D. Thesis, HEC Montreal, 2009.

[10] G. Berbeglia, J.-F. Cordeau, and G. Laporte. A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS: Journal on Computing*, 24:343–355, 2012.

[11] G. Berbeglia, G. Pesant, and L.-M. Rousseau. Checking feasibility of the dial-a-ride problem using constraint programming. *Transportation Science*, 45:399–412, 2011.

[12] G. Berbeglia, G. Pesant, and L.-M. Rousseau. The pickup and delivery problem with fixed partial routes: A complexity analysis. *Transportation Science*, 46(3):359–373, 2012.

[13] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International Conference on World Wide Web*, volume 7, pages 107–117. 1998.

[14] J. Bérubé, J. Potvin, and J. Vaucher. Time-dependent shortest paths through fixed sequence of nodes: Application to a travel planning problem. *Comput. Oper. Res.*, 33(6):1838–1856, 2006.

[15] X. Cai, T. Kloks, and C. K. Wong. Time-varying shortest path problems with constraints. *Networks*, 29(3):141–149, 1997.

[16] R.D. Cairns and C. Liston-Heyes. Competition and regulation in the taxi industry. *Journal of Public Economics*, 59(1):1–15, 1996.

[17] I. Chabini. Discrete dynamic shortest path problems in transportation applications. *Transp. Res. Rec.*, 1645:170–175, 1998.

[18] Y. S. Chiang, , and P. O. Roberts. A note on transit time and reliability for regular-route trucking. *Transport Research Part B*, 14s(1-2):59–65, 1980.

[19] K. L. Cooke and E. Halsey. The shortest route through a network with time-dependent intermodal transit times. *J. Math. Anal. Appl.*, 14:493–498, 1966.

[20] J.-F. Cordeau. A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54:573–586, 2006.

[21] J.-F. Cordeau and G. Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research B*, 37:579–594, 2003a.

[22] J.-F. Cordeau, G. Laporte, J.-Y. Potvin, and M.W.P. Savelsbergh. Transportation on demand. In *Transportation*, pages 429–466. Amsterdam: North-Holland, 2007a.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (2nd ed.)*. MIT Press and McGraw-Hill, 2001.

[24] Cristian E. Cortes and R. Jayakrishnan. Design and operational concepts of a high coverage point-to-point transit system. *Center for Traffic Simulation Studies*, 2001.

[25] C. Davies and P. Lingras. Genetic algorithms for rerouting shortest paths in dynamic and stochastic networks. *European Journal of Operational Research*, 144:27–38, 2003.

[26] M. Diana and M. Dessouky. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B*, 38:539–557, 2004.

[27] Y. Dumas, J. Desrosiers, and F. Soumis. The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54:7–22, 1991.

[28] A. Farahat, T. LoFaro, J. C. Miller, G. Rae, and L.A. Ward. Authority rankings from hits, pagerank, and salsa: existence, uniqueness, and effect of initialization. *SIAM Journal on Scientific Computing*, 27:1181–1201, 2006.

[29] J.E. Fernandez, J. De Cea, and J. Briones. A diagrammatic analysis of the market for cruising taxis. *Transportation Research Part E*, 42(6):498–526, 2006.

[30] M. C. Ferris and A. Ruszczynski. Robust path choice in networks with failures. *Networks*, 35:181–194, 2000.

[31] D. Flores-Guri. An economic analysis of regulated taxicab markets. *Review of Industrial Organization*, 23(3-4):255–266, 2003.

[32] L. Fu and L. Rillet. Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research*, 32(7):499–516, 1998.

[33] T. Garaix, C. Artigues, D. Feillet, and D. Josselin. Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operational Research*, 204:62–75, 2010.

[34] J. Hackner and S. Nyberg. Deregulating taxi services: a word of caution. *Journal of Transport Economics and Policy*, 29(2):195–207, 1995.

[35] R. W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation science*, 20(3):182–188, 1986.

[36] H. Hamacher, S. Ruzika, and S. Tjandra. Algorithms for time-dependent bicriteria shortest path problems. *Discrete Optim.*, 3(3):238–254, 2006.

[37] L. Häme and J.-P. Jokinen. A stochastic network model for demand-responsive transport. In *Kuhmo Nectar Conference on Transport Economics, Berlin*, 2012.

[38] Helsinki Region Traffic. Jouko neighbourhood routes and service routes, October 2010.

[39] Helsinki Region Traffic. Kalkati.net XML representation of helsinki regional transport authority (HSL) timetable data, Oct 2010.

[40] M. Horn. An extended model and procedural framework for planning multi-modal passenger journeys. *Transp. Res. Part B*, 37(7):641–660, 2003.

[41] R. Huang and Z. R. Peng. An integration of network data model and routing algorithms for online transit trip planning. In *Proc. 80th Annu. Meeting Transp. Res. Board*. Washington, DC, 2001.

[42] R. Huang and Z. R. Peng. Schedule-based path finding algorithms for online transit trip planning. In *Proc. 81st Annu. Meeting Transp. Res. Board*. Washington, DC, 2002.

[43] E. Hyytiä, L. Häme, A. Penttinen, and R. Sulonen. Simulation of a large scale dynamic pickup and delivery problem. In *3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools 2010)*. 2010.

[44] J.J. Jaw, A.R. Odoni, H.N. Psaraftis, and N.H.M. Wilson. A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows. *Transportation Research Part B*, 20:243–257, 1986.

[45] Jani-Pekka Jokinen, Teemu Sihvola, Esa Hyytiä, and Reijo Sulonen. Why urban mass demand responsive transport? In *IEEE Forum on Integrated and Sustainable Transportation Systems (FISTS)*, Vienna, Austria, June-July 2011.

[46] S. Kim, M.E. Lewis, and C. C. White III. State space reduction for non-stationary stochastic shortest path problems with real-time traffic congestion information. *IEEE Transactions on Intelligent Transportation Systems*, 6(3):273–284, 2005.

[47] S. Kim, M.E. Lewis, and C.C. White III. Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):178–188, 2005.

[48] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the Ninth Annual ACM-SIAM Symposium of Discrete Algorithms*, pages 668–677. ACM Press, New York, 1998.

[49] M. Kostreva and M. Wiecek. Time dependency in multiple objective dynamic programming. *J. Math. Anal. Appl.*, 173(1):289–307, 1993.

[50] R. Lagos. An analysis of the market for taxicab rides in new york city. *International Economic Review*, 44(2):423–434, 2003.

[51] A.N. Langville and C.D. Meyer. A survey of eigenvector methods of web information retrieval. *SIAM Review*, 47:135–161, 2005.

[52] Ronny Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *In The Ninth International WWW Conference, May 2000*.

[53] O. B. G. Madsen, H. F. Ravn, and J. M. Rygaard. A heuristic algorithm for the a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, 60:193–208, 1995.

[54] J. Mageean and J. D. Nelson. The evaluation of demand responsive transport services in europe. *Journal of Transport Geography*, 11(4):255–270, 2003.

[55] K. Matsushima and K. Kobayashi. Endogenous market formation with matching externality: an implication for taxi spot markets. In *Structural Change in Transportation and Communications in the Knowledge Economy*, pages 313–336. Edward Elgar, 2006.

[56] E. Melachrinoudis, A. B. Ilhan, and H. Min. A dial-a-ride problem for client transportation in a healthcare organization. *Computers & Operations Research*, 34:742–759, 2007.

[57] E. D. Miller-Hooks and H. S. Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34:198–215, 2000.

[58] P. Modesti and A. Siomachen. A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *Eur. J. Oper. Res.*, 111(3):495–508, 1998.

[59] A.T. Moore and T. Balaker. Do economists reach a conclusion on taxi deregulation? *Econ Journal Watch*, 3(1):109–132, 2006.

[60] C. Mulley and J. D. Nelson. *Research in Transportation Economics*, 25(1):39 – 45, 2009.

[61] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Stable algorithms for link analysis. *SIGIR01, September 912, 2001, New Orleans, Louisiana, USA*.

[62] Department of the Environment Transport and the Regions. *Social exclusion and the provision and availability of public transport*. DETR, London, 2000.

[63] S. N. Parragh, K. F. Doerner, and R. F. Hartl. Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 37:1129–1138, 2010.

[64] Z. R. Peng and R. Huang. Design and development of interactive trip planning for web-based transit information systems. *Transp. Res. Part C*, 8(1):409–425, 2000.

[65] G. H. Polychronopoulos and J. N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27:133–143, 1996.

[66] H. E. Psaraftis and J. N. Tsitsiklis. Dynamic shortest paths in acyclic networks with markovian arc costs. *Operations Research*, 41(1):91–101, 1993.

[67] H.N. Psaraftis. A dynamic programming approach to the single-vehicle, many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14:130–154, 1980.

[68] H.N. Psaraftis. An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science*, 17:351–357, 1983.

[69] S. Ropke, J.-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithm for pickup and delivery problems with time windows. *Networks*, 49:258–272, 2007.

[70] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

[71] R. A. Russell and T. L. Urban. Vehicle routing with soft time windows and erlang travel times. *Journal of the Operational Research Society*, 59:1220–1228, 2007.

[72] K. A. Small and E. T. Verhoef. *The Economics of Urban Transportation*. Routledge, 2 Park Square, Milton Park, Abingdon, OX14 4RN, 2007.

[73] M. C. Tan, C. O. Tong, S. C. Wong, and J. Xu. An algorithm for finding reasonable paths in transit networks. *J. Adv. Transp.*, 41(3):285–305, 2007.

[74] B. Thomas and C. White. The dynamic shortest path problem with anticipation. *European Journal of Operational Research*, 176(2):836–854, 2007.

[75] C. O. Tong and A. J. Richardson. A computer model for finding the timedependent minimum path in transit systems with fixed schedules. *J. Adv. Transp.*, 18(2):145–161, 1984.

[76] C. O. Tong and S. C. Wong. A stochastic transit assignment model using a dynamic schedule-based network. *Transp. Res. Part B*, 33(2):107–121, 1999.

[77] P. Toth and D. Vigo. Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31:60–71, 1997.

[78] P. Toth and D. Vigo. *The vehicle routing problem*. Philadelphia, PA, 2002.

[79] Jouni T Tuomisto and Marko Tainio. An economic way of reducing health, environmental, and other pressures of urban traffic: a decision analysis on trip aggregation. *BioMed Central, November 25*, 2005.

[80] S. Waller and A. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.

[81] K.I. Wong and M.G.H. Bell. Solution of the dial-a-ride problem with multi-dimensional capacity constraints. *International Transactions in Operational Research*, 13:195–208, 2006.

[82] K.I. Wong, S.C. Wong, M.G.H. Bell, and H. Yang. Modeling the bilateral micro-searching behavior for urban taxi services using the absorbing markov chain approach. *Journal of Advanced Transportation*, 39(1):81–104, 2005.

[83] S. C. Wong and C. O. Tong. Estimation of time-dependent origin-destination matrices for transit networks. *Transp. Res. Part B*, 32(1):35–48, 1998.

[84] Z. Xiang, C. Chu, and H. Chen. A fast heuristic for solving a large-scale static dial-a-ride problem under complex constraints. *European Journal of Operational Research*, 174:1117–1139, 2006.

[85] H. Yang, C. W. Y Leung, S. C. Wong, and M. G. H. Bell. Equilibria of bilateral taxi-customer searching and meeting on networks. *Transportation Research Part B*, 44:1067–1083, 2010.

[86] H. Yang, S.C. Wong, and K.I. Wong. Demand-supply equilibrium of taxi services in a network under competition and regulation. *Transportation Research Part B*, 36:799–819, 2002.

[87] H. Yang, M. Ye, W. H.-C. Tang, and S. C. Wong. A multiperiod dynamic model of taxi services with endogenous service intensity. *Operations Research*, 53(3):501–515, 2005.

[88] A. Ziliaskopoulos and W. Wardell. An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. *Eur. J. Oper. Res.*, 125(3):486–502, 2000.

[89] K. G. Zografos and K. N. Androutsopoulos. Algorithms for itinerary planning in multimodal transportation networks. *IEEE Transactions on Intelligent Transportation Systems*, 9(1), 2008.

Aalto-DD 80/2013

Demand-responsive transport is a form of public transport between bus and taxi services, involving flexible routing of small or medium sized vehicles. This dissertation presents mathematical models for demand-responsive transport and methods that can be used to solve combinatorial problems related to vehicle routing and journey planning in a transport network.

The mathematical models proposed in this work can be used to simulate the operations of public transport services in a wide range of scenarios, from paratransit services for the elderly and disabled to large-scale demand-responsive transport services designed to compete with private car traffic. In addition to public transport, potential applications of the proposed methods for solving vehicle routing and journey planning problems include freight transportation, courier and food delivery services, military logistics and air traffic.

9 789526 051628

ISBN 978-952-60-5162-8
ISBN 978-952-60-5163-5 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934
ISSN 1799-4942 (pdf)

Aalto University
School of Science
Department of Mathematics and Systems Analysis
www.aalto.fi

BUSINESS +
ECONOMY

ART +
DESIGN +
ARCHITECTURE

SCIENCE +
TECHNOLOGY

CROSSOVER

DOCTORAL
DISSERTATIONS