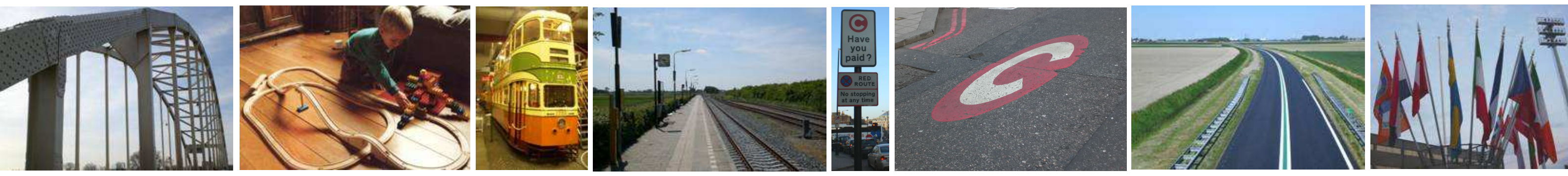
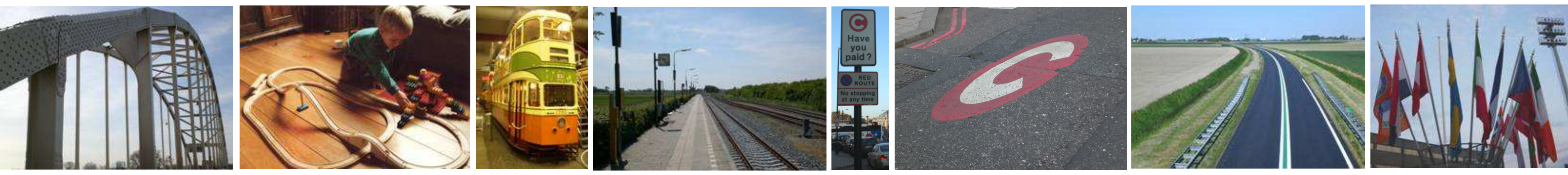


Optimisation of toll levels in networks

An optimal toll design framework, using a pattern search approximation algorithm

Ties Brands



University of Twente
Enschede - The Netherlands

Goudappel Coffeng
Adviseurs verkeer en vervoer

Ties Brands

Optimisation of toll levels in networks
An optimal toll design framework, using a pattern search approximation algorithm

MSc Thesis
Civil Engineering and Management
Applied Mathematics
20-08-2008
Deventer / Enschede

Graduation committee:
- Prof. Dr. Ir. Eric van Berkum
  University of Twente, Centre for Transport Studies
- Dr. Tom Thomas
  University of Twente, Centre for Transport Studies
- Dr. Georg Still
  Universtiy of Twente, Discrete Mathematics & Mathematical Programming
- Dr. Ir. Werner Scheinhardt
  University of Twente, Stochastic Operations Research
- Ir. Dirk van Amelsfort
  Goudappel Coffeng BV

University of Twente
Enschede - The Netherlands

Goudappel Coffeng
Adviseurs verkeer en vervoer

Summary

Within the world of traffic engineering, road pricing is widely accepted to be a measure that can contribute to the solution of several problems in the present traffic system: congestion, environmental damage, use of unsustainable recourses, use of space, etc. Road users, the government, and neighbours of roads are the most important stakeholders of these problems. Although some successful practical applications of road pricing exist (for instance Singapore, London, Stockholm), no large scale implementation of road pricing exists in the world yet. The Dutch government has decided to implement road pricing in the whole country, starting in 2011. In order to be able to influence the traffic system, this road pricing system will be able to handle time and space differentiation of tolls. An important question that rises from this quality is where and when the tolls should have which particular value. This thesis investigates this question.

A stakeholder analysis for the Dutch context confirms the Dutch government's idea: in the Netherlands, a need for a time and space differentiated fare per kilometre exists. The system must have a predefined maximum value and should be clear and comprehensible for the public. The price needs to be differentiated to the vehicle's environmental characteristics. The differentiated road pricing system can be used to influence the traffic system in order to achieve a range of policy objectives, which are assumed to benefit the society as a whole. For example, the minimisation of total congestion, minimisation of average travel time, or minimisation of vehicle emissions.

In order to gain more insight in the problem of optimal toll design, it is formulated as a mathematical program. The policy objectives are made quantitative in the form of objective functions. In this formulation the value of the objective function depends on the space and time differentiated toll levels. Some of the stakeholders' demands are formulated as mathematical constraints. The problem is formulated as a bi-level problem. The upper level consists of the road authority which aims to minimise an objective function, in this case the average travel time in the traffic system. The lower level consists of the behaviour of the road users in the traffic system, who minimise their own perceived generalised traffic costs. In this formulation, the road authority minimises the average travel time, given the reaction of the traffic system. The lower level is operationalised by a dynamic traffic model, which achieves dynamic stochastic user equilibrium.

Solving this mathematical program is not a trivial case, because the traffic model is treated as a black box and one function evaluation takes a lot of computation time.

Based on the problem's previously mentioned characteristics, a local search algorithm has been selected to approximate the solution of the optimisation problem, namely pattern search.

Application of different variants of the pattern search algorithm to the case study showed that it is possible to achieve considerable improvements in the value of the average travel time. From all tested variants, one which monitors improvements performed best in terms of average travel time value, within a reasonable computation time. Another variant which adapts a few different tolls at the same time, performed well in terms of computation time, within a reasonable achieved value of the average travel time. The tests also showed that multiple local minima exist, with many of them having approximately the same average travel time value. So by means of the policy objective, these toll settings are roughly the same. Other political arguments, like the expected revenue of these toll settings, can determine which exact toll setting is to be implemented. It cannot be guaranteed that this applies to other networks, but this research gives an indication that when a local minimum is found, the average travel time is close to the optimal value.

The result of this research is a practical modelling framework for the approximation of optimal toll design. Restrictions to computational power make that at this moment it is not yet possible to approximate highly differentiated tolls in big networks with the proposed modelling framework. The approximation of very roughly differentiated tolls on bigger networks seems possible, which is promising when the space and time differentiation will be designed for the Dutch road pricing system.

Samenvatting

In de verkeerskunde wordt breed gedragen dat het geven van een prijs aan automobiliteit (in Nederland bekend als Anders Betalen voor Mobiliteit, ABvM) kan bijdragen aan de oplossing van verschillende problemen van het huidige verkeerssysteem: files, milieuschade, gebruik van niet hernieuwbare grondstoffen, ruimtegebruik, enz. De belangrijkste belanghebbenden van dit probleem zijn weggebruikers, de overheid en omwonenden van wegen. Er bestaan enkele succesvolle toepassingen van tol in grote steden (Singapore, Londen, Stockholm), maar een systeem voor alle weggebruikers op alle wegen in een land bestaat nog niet. De Nederlandse regering heeft besloten om ABvM in te voeren in het hele land, beginnend in 2011. In dit systeem worden de prijzen gedifferentieerd naar plaats en naar tijd, waardoor het verkeerssysteem beïnvloed kan worden. Hieruit volgt een belangrijke vraag: waar en wanneer moeten de prijzen welke waarde hebben? Dit rapport onderzoekt deze vraag.

Een analyse van de belanghebbenden bevestigt het idee van de Nederlandse regering: in Nederland bestaat er een behoefte aan een tijd en plaats gedifferentieerde kilometerprijs. Verder volgt uit deze analyse dat de prijs een vastgestelde maximumwaarde moet hebben, dat het systeem duidelijk en begrijpbaar moet zijn voor de gebruikers en dat de prijs gedifferentieerd moet zijn naar de milieueigenschappen van het voertuig. Het gedifferentieerde prijssysteem kan gebruikt worden om het verkeerssysteem te beïnvloeden. Daarmee is het mogelijk verschillende beleidsdoelen na te streven, die de samenleving in zijn geheel ten goede komen. Hierbij kan gedacht worden aan minimaliseren van de totale filezwaarte, minimaliseren van de gemiddelde reistijd of minimaliseren van schadelijke uitstoot.

Om meer inzicht in het ontwerpprobleem van een optimaal tolsysteem te krijgen, is het geformuleerd als een mathematisch programma. De beleidsdoelen zijn kwantitatief geformuleerd in doelfuncties. De waarde van deze doelfunctie hangt af van de naar plaats en naar tijd gedifferentieerde prijsniveaus. Sommige eisen van belanghebbenden zijn geformuleerd als wiskundige randvoorwaarden. Het probleem wordt geformuleerd als een bi-level probleem. Het upper level bestaat uit de wegbeheerder, die een doelfunctie wil minimaliseren, in dit geval de gemiddelde reistijd in het verkeerssysteem. Het lower level bestaat uit het gedrag van de weggebruikers van het verkeerssysteem, die hun eigen beleefde gegeneraliseerde kosten minimaliseren. In deze formulering minimaliseert de wegbeheerder de gemiddelde reistijd, gegeven de reactie van het verkeerssysteem. Het lower level wordt geoperationaliseerd door een dynamisch verkeersmodel, dat een stochastisch dynamisch gebruikersevenwicht tot stand brengt.

Het oplossen van dit mathematisch programma is niet triviaal, omdat het verkeersmodel als een 'black box' wordt gezien en het uitrekenen van één functiewaarde veel rekentijd kost. Gebaseerd op deze eigenschappen van het probleem is een lokaal zoekalgoritme geselecteerd om de oplossing van het optimalisatieprobleem te benaderen, namelijk pattern search.

Verschillende varianten van dit pattern search algoritme zijn toegepast op de case studie en laten zien dat het mogelijk is om aanzienlijke verbeteringen in de waarde van de gemiddelde reistijd te bereiken. Een variant die bijhoudt waar verbetering optreedt presteert van de geteste varianten het beste in termen van gemiddelde reistijd, binnen een redelijke rekentijd. Een andere variant die meerdere tarieven tegelijk aanpast heeft minder rekentijd nodig, terwijl er nog steeds een redelijke waarde voor gemiddelde reistijd wordt bereikt. Uit de experimenten is ook naar voren gekomen dat er meerdere lokale minima bestaan, waarvan de meeste ongeveer dezelfde waarde voor gemiddelde reistijd hebben. Dus in termen van het gekozen beleidsdoel zijn deze oplossingen ongeveer hetzelfde. Welke van deze oplossingen geïmplementeerd wordt, kan gekozen worden op basis van andere politieke argumenten, bijvoorbeeld de verwachte tolopbrengsten. Hoewel het niet kan worden gegarandeerd dat dit ook voor andere netwerken geldt, geeft dit onderzoek een indicatie dat de gemiddelde reistijd waarde van een gevonden lokaal minimum waarschijnlijk dichtbij de optimale waarde zit.

Het resultaat van dit onderzoek is een praktisch modelraamwerk dat het ontwerp van een optimaal tolsysteem kan benaderen. Vanwege beperkte rekenkracht is het op dit moment nog niet mogelijk om het raamwerk toe te passen op grote netwerken met gedetailleerde tariefdifferentiatie. Het benaderen van optimale ruw gedifferentieerde tarieven op grotere netwerken lijkt mogelijk. Dit is veelbelovend als de naar plaats en naar tijd gedifferentieerde tarieven voor het Nederlandse ABvM systeem ontworpen moeten worden.

Preface

This thesis is the result of exactly one year of work on the final project of my master's degree. It is the basis of finishing two master degrees at University of Twente: Civil Engineering & Management (Traffic Engineering and Management) and Applied Mathematics (Discrete Mathematics & Mathematical Programming and Stochastic Operations Research). The corresponding research has been conducted at Goudappel Coffeng BV, a large transport consultancy firm.

Already as a child, I was interested in civil engineering, as can be seen on the cover of this thesis. Several fellow pupils in secondary school knew to find me for some extra explanation during maths classes. Viewers of my holiday pictures regularly notice the relatively high number of bridges, train stations, buses, etc. So the combination of mathematics and traffic engineering has been quite natural to me.

In fact, finishing this thesis is just like any other subject I earlier completed during my 6 years at the university in Enschede: send it to the teachers, give a presentation about it, and receive a mark for it. But on the other hand, it feels like much more, because the final project itself is not the only thing to be finished: it completes both my master's studies and it finishes my time in Enschede. It represents the end of the student period in my life and represents the start of my working life as well.

I would like to thank my five supervisors Eric, Tom, Georg, Werner, and Dirk for their easy co-operation and their useful and extensive feedback on my writings. Furthermore, I would like to thank Eric and Dirk for encouraging me to write two papers about my research, and for their feedback on the papers. I experienced the working environment at Goudappel Coffeng BV as inspiring and divers: I would like to thank my colleagues for the interesting and nice conversations on traffic engineering, holidays, good food, Deventer, and other topics. Especially my room mates Dirk, Jantine, and Robert: thank you for the pleasant working environment.

Deventer, August 2008

Ties Brands

1 Introduction

Within the world of traffic engineering, road pricing is widely accepted to be a measure which can contribute to the solution of a range of problems in the present traffic system: congestion, environmental damage, use of unsustainable recourses, use of space, etc. Although some successful practical applications of road pricing exist (Singapore, London, Stockholm), no large scale implementation of road pricing exists yet in the world.

The Dutch government is planning to implement road pricing on all roads in The Netherlands. According to the present status of the plans, in 2011 all trucks start paying per kilometre and in 2018 the system of road pricing should fully implemented for all road users. In order to be able to have influence on the traffic system, the system will be able to handle time- and space differentiation of tolls. Until now, nothing is said about the way to determine the exact way of differentiation. This issue is the subject of this study.

In theoretical economics the most efficient way to apply road pricing is as a continuous function over space, time and vehicle type, because in this way every possible toll can be set and thus every possible desirable behaviour can be achieved. When the social marginal costs are charged as a toll, social welfare optimum is achieved. In the literature this is called the first best pricing principle (Yang and Huang, 2005). However, such a model would be far too complicated for practical implementation. First, it will be technically difficult to charge all vehicles exactly the correct amount of money. But what is more important, the users of the system are not able to understand such a difficult system and they will not know in advance the amount of money they will have to pay. In this way they will not behave optimally, while optimal behaviour is necessary to achieve first best conditions. Furthermore, the first best road pricing measure is not a desirable situation, because the public opinion will not accept the uncertainty in the toll. From the users' point of view, it is desirable to develop a simple system with the smallest possible variation, like a fixed toll over some predefined time interval, usually a peak period (Hau, 2006a).

So several problems in determining the toll levels in a real life situation arise when theory and practise in road pricing meat each other. Three important issues about this are given below (see also section 16.2.5 of Verhoef *et al.*, 2008):

- Different stakeholders have different goals. The road authority aims for social optimum conditions, revenue maximisation, minimisation of the total amount of road kilometres, or any other objective. The road user optimises his or her own situation: minimisation of costs and travel time (or generalised costs). The choice of the objective function of the road authority determines the point at which optimality is reached. So different policy objectives lead to different optimal road pricing measures.

- An infinite number of different road pricing schemes exists: it is possible to charge the toll on many different bases. For example in cordon based tolling, a road user has to pay when he enters a specific zone, usually the centre of a city. Congestion charging in London is an example of cordon based tolling. Another example of a pricing scheme is paying per kilometre, differentiated to time and to space, like the plans in the Netherlands. The question is what pricing scheme is the most effective with respect to the policy objective. Because the number of possible different schemes is infinite, searching for the most effective scheme takes place in a large solution space.
- Economically desirable road pricing systems are politically and/or socially unacceptable. In other words, the problem is to find the optimal road pricing measure given the policy objective while the system is not too difficult for the public. On the other hand, to have the ability to take specific measures, the system should have the ability to differentiate tolls to specific area's and time slots with a certain level of detail. In this research, this level of detail in the differentiation of tolls is incorporated in the definition of pricing scheme (see section 2.2.2).

1.1 Research objective

The former section illustrates the gap between theory and practice when the toll level has to be determined in road pricing. The aim of this research is to bring theory and practice closer to each other.

The overall objective is to design a practical modelling framework to determine the optimal toll level in road pricing, given a price structure, an objective function and constraints.
To this end a mathematical and a computational model are developed, taking into account the demands of the stakeholders. Furthermore, the objective is to apply the method to a case study to gain insight in the effect of different search algorithms.

1.2 Research questions

Given the research objectives, the following research questions result:

1. What pricing scheme corresponds to the demands of the stakeholders of road pricing?
    a. What demands exist from stakeholders?
    b. What pricing schemes can be found in the literature?
    c. Which of these pricing schemes are considered to work well in the literature?

2. In what way can this road pricing problem be formulated in a mathematical model?
    a. What variables and control variables should be defined?
    b. What objective function should be defined?
    c. What constraints should be added to the model?
    d. What underlying traffic model is needed to model the effect of the tolls on the traffic system?
3. What method or algorithm is most suitable to solve the mathematical model?
    a. What methods and algorithms exist in the literature?
    b. Which of these methods and algorithms are suitable for this model?
    c. Which algorithm gives the best results in the case study?
    d. Can the case specific results be generalised?
4. What modelling choices have to be made when a practical situation is analysed by the modelling framework?

1.3 Scope of this research

As follows from the list of research questions, the topic of this research, design of a framework to determine optimal toll levels, involves a range of steps to take. Each of these steps could be an interesting topic for a separate research. This research focuses on the development and testing of an efficient search algorithm (research question 3). So for the stakeholder analysis, for the formulation of a mathematical model, and for the development of a dynamic traffic model, existing literature and methods are used. The used search algorithms are based on existing literature, but are further developed and tested in this research. The focus of this research is on testing the performance of the variants of the different search algorithms. The tests are conducted for a specific case, and generalisations are made where possible.

1.4 Outline of the report

Apart from this introduction and the concluding chapter, the report consists of 5 chapters (see Figure 1.1).

In chapter 2 an analysis is made of the stakeholders of road pricing in the Dutch context. This analysis determines the general requirements for the optimal toll design. The chapter results in a list of demands from the stakeholders. From a wide range of possible road pricing schemes, the demands of stakeholders result in a focus on distance based, time and space differentiated tolls.

In chapter 3 a mathematical programming formulation is given for the toll design problem. The formulations in this chapter are based on the chosen pricing scheme in chapter 2. Furthermore, mathematical constraints are formulated based on the list of stakeholder demands in chapter 2. From a wide range of possible optimisation criteria, average travel time is chosen as an objective function.

When tolls are charged to road users, this will have an effect on their behaviour. So in chapter 4 a model is presented to compute the effects of the tolls on the traffic system. This model is capable of handling the dynamic characteristics of the chosen road pricing scheme and of modelling the behavioural responses to these tolls. The output of the traffic system consists of network performance data which is used to compute the value of the objective function for a given toll setting.

Chapter 5 explores the possibilities of different search algorithms from mathematical optimisation theory. The formulation in chapter 3 and the computational aspects of chapter 4 result in a choice for a pattern search algorithm. A range of case specific properties of the search algorithm is treated.

In chapter 6 the developed framework is applied to a case study. In this case study different variants of the search algorithm are applied. These variants vary in a limited number of properties. The results of this chapter are thus a ranking of the different variants of the pattern search algorithm for this specific case. In a concluding section notes are given on generalisation of the results of this case study.

Finally in chapter 7 a summary is given of the optimal toll design framework. This summary includes the main conclusions of the research on the most appropriate search algorithm to use when a real life toll design problem has to be solved. Furthermore, limitations of the research are discussed, which result in recommendations for improvement of the framework in future research.

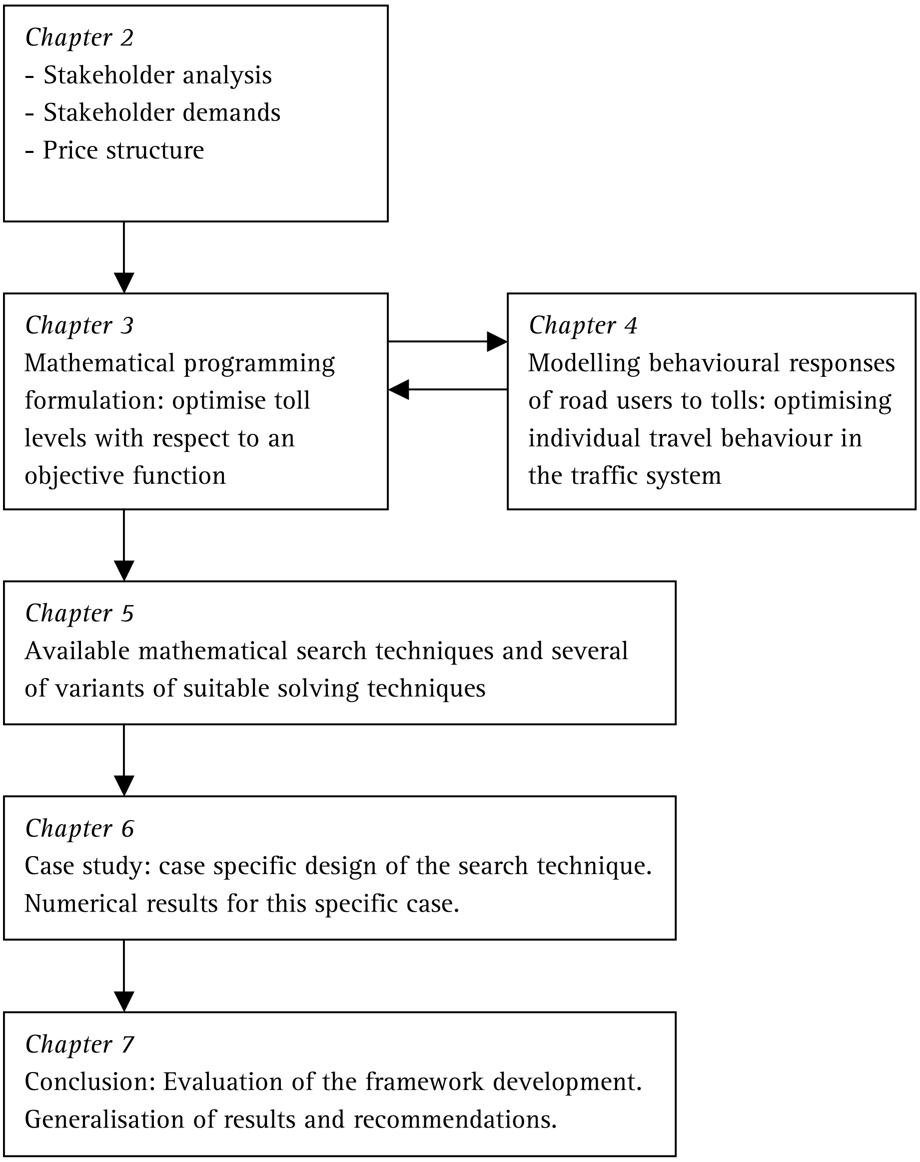


Figure 1.1: Overview of the chapters of this research

2 Stakeholder analysis

In this chapter we investigate the stakeholders that are involved in the implementation of road pricing. The demands and objectives of these different stakeholders together determine the design of an acceptable road pricing system. The demands of stakeholders, described in words, can later be translated later in mathematical constraints or can be incorporated in the objective function. Thus the objective and main result of this chapter is to determine the objectives and constraints of road pricing measures in order to give input to the succeeding chapters on the mathematical model (chapter 3) and on the traffic model (chapter 4). The stakeholders and their objectives and constraints are analysed using a literature review on different road pricing schemes, both planned and realised, literature on stakeholders of road pricing, and reports written by the different stakeholders.

The outline of this chapter is as follows. First a literature review is given of a range of stakeholders in the context of road pricing. After that, a summary of the most important demands from the stakeholder analysis is presented. A second literature review is then given about different possible road pricing schemes. Based on this overview, a motivation is given for the pricing scheme in this research: a link based, time and space differentiated charge.

2.1 Literature review

In this section information is given on the stakeholders of road pricing in the Netherlands in general. We use the word “demands” to refer to all kinds of wishes, needs or requirements from stakeholders, for example the need to set a maximum toll because of the public opinion or the need of the ministry of Finance to respect the current tax laws. For each stakeholder the demands are stated in this chapter. It is also argued what demands are relevant for this research. Finally, a summary is given of all relevant demands, taken into account in this research.

The stakeholders of the problem can be incorporated in two main groups: the road authority on the one hand and the public on the other hand. The road authority will be some governmental institute, which serves the demands of the involved ministries: the Ministry of Transport on account of the traffic effects and the Ministry of Finance on account of the effect on the tax system. Furthermore, this road authority has to respect the national and European legislation. The public is summarised in the categories: interests of the society as a whole and the specific interests of the current road users, represented by associations, because their situation will change most by the system.

2.1.1 Ministry of Transport

The most important policy document in the Netherlands at this time related to road pricing is Nota Mobiliteit (VenW & VROM, 2004). This report represents the Dutch policy on mobility and is published by the Ministry of Transport, Public Works and Water Management and by the Ministry of Housing, Spatial Planning and the Environment. The outline of the policy is that citizens can make mobility choices themselves, but they should also feel and carry the responsibility of the consequences of this choice. They should make a rational and conscious choice. In order to achieve this choice behaviour a variabilisation of car taxes is introduced in the form of a kilometre charge that will be differentiated towards time, space and vehicle characteristics. The total revenue should remain equal and all revenues will be dedicated to infrastructure. When this road pricing measure is implemented, the costs of implementation will be paid out of the general tax revenues, but these costs may not be more than 5% of the generated revenue.

Hau (2006a) gives additional demands for a road authority in general. The relevant demands are shown here.

- *Enhanced efficiency via direct charging.* The system should be able to directly charge (as closely as possible) the external costs of road use. Ubbels (2006) argues it is not possible to measure all external costs of car consumption, so this demand is difficult to achieve. In this research external costs are not used for the optimisation, but an objective function is used.
- *Reliability.* The system should be able to operate in harsh environmental conditions and should charge users correctly. In this research the existence of such a technical system is assumed.
- *Security and enforcement.* The system should be free from theft of revenues by private individuals and operators and it should be free of fraud and abuse to the payment system. In this research the existence of such a technical system is assumed.
- *Provision for occasional visitors.* Non Dutch drivers should also be able to use the system when visit the country. Is assumed to be present.
- *Flexibility (responsiveness to demand).* The tolls should be able to adapt to traffic demand, this implies the tolls should be able to vary to time-of-day and space on a certain level of detail. This demand is relevant for the pricing scheme in this research.

This section results in two demands to be taken into account in this research:

- The average amount of money charged to car users may not become bigger. So the average car user should not pay more after the new pricing measures. The total revenue should remain equal and all revenues will be dedicated to infrastructure. As mentioned earlier the system costs should not exceed 5% of the revenues generated.
- The tolls should be able to adapt to traffic demand, this implies the tolls should be able to vary to time-of-day and space on a certain level of detail.

2.1.2 Ministry of Finance

The Ministry of Finance is responsible for the collection of tax revenues in the Netherlands. For this ministry the reliability of tax revenues is an important issue, because it is responsible for a stable financial system (Ministry of Finance, 2007). Currently the car ownership taxes, which are about 6.5 billion euro annually, are collected by 4 tax collectors. So the current means of collecting tax is very efficient, however it does not incur the desired behaviour of travellers which leads to higher costs elsewhere in government spending. Because the exact response of the car drivers on a new road pricing system is not known in advance, the income of the state becomes less reliable.

A demand from this stakeholder is:

The revenue from taxes on car drivers may not be reduced or become unreliable: a minimum revenue should be determined to ensure a stable state finance. Together with the demand from the Ministry of Transport, the amount of money collected from car users should stay exactly the same after the new road pricing measures. It is not possible to give this guarantee, so the demands may be formulated with some uncertainty bound.

In order to give a better view on the meaning of this constraint, the present road tax system of the Netherlands is described shortly in this section. The present system consists of tax on car ownership (MRB), tax on the purchase of new cars (BPM), and excise on fuels.

For petrol cars the periodical car ownership tax rates are given in Table 2.1. These rates are a periodical tax on vehicle ownership. For cars using other fuel, different rates apply, but the structure of the system is the same.

Weight of the car (rounded to the nearest 100)	Tax per three months	Increased with	For each 100 kg more than
500 or less	€ 12,37		
600	€ 16,76		
700	€ 21,30		
800	€ 27,80		
900 to 3200	€ 36,98	€ 9,95	900 kg
3300 en meer	€ 273,01	€ 6,92	3300 kg

Table 2.1: Monthly tax rates on vehicle ownership (Overheid, 2007b)

The tax on new cars is levied as a percentage of the catalogue price of the car. This amount of money is increased or lowered with some amount depending on the type of fuel the car uses. The percentage is 45,2% of the net catalogue price. Petrol cars pay €1.540.- less and diesel cars pay €328.- more. Furthermore the amount of tax is adapted based on the energy efficiency class of the car. This adaptation varies from a discount of €1,400.- to an extra tax of €1,600.-. Finally, for a car with a CO_2 emission of more than 232 grams per km, an additional tax of €120.- has to be paid for each gram CO_2 emission per km which exceeds this value (Overheid, 2007a).

Finally, the excise duty on fuels is an important tax on car use. Several rates exist, depending on the exact fuel, but all rates are fixed on the volume of the fuel (Overheid, 2007c). So in fact this tax is already a variable tax on car use, which is less expensive for environmental friendly vehicles: vehicles with a small fuel consumption. Problem with rising this tax is the border effect. People who live near the border of the country can very easily drive abroad to buy their fuel.

2.1.3 Road users

In the Netherlands a user association for car mobility exists, ANWB. This association made a policy document in 2004 which stated the vision of the members of the association, who are especially car owners / road users (ANWB, 2004). In general the association is not negative with respect to road pricing. They argue that variable tolls are fairer than the present fixed tax system and that negative externalities of traffic may be internalised in a road pricing fee. They are not pleased that in the present system choosing for the safest, healthiest or cleanest alternative is not always the cheapest. In Hau (2006a) an overview on different demands to a road pricing system from a user's point of view is given. From these two sources, a list of demands from the users' point of view is given here.

- When the system is introduced, the change should be smooth, including temporary measures (ANWB). Not relevant for this research, because the research focuses on the final situation.
- Users of the traffic system should have a guarantee to get the service where they pay for (ANWB). This research does not cover pay lanes, so this demand is outside of the scope of this research.
- Prepayment / post payment options for charging. When both pre- and post payment systems are offered to the road user, it would allow the user to choose the type of technology that suits him or her (Hau). In this research it is assumed that the way of paying does not influence the traffic behaviour.
- The system should be transparent (via ex ante pricing) (Hau). This research focuses on ex ante pricing instead of online pricing, so this demand is satisfied.
- The system should be have enough anonymity (Protection from invasion of privacy) (Hau). In this research the existence of such a technical system is assumed.
- The paying system should be fast, safe, understandable and easy (ANWB). All these issues can be translated to a system which is not too complicated, which is added as a demand. The system should be user-friendly: simple to understand and convenient to use. Also for road safety reasons the system may not be too complex (Hau). To this can be added that the system should not be too complicated, because otherwise road users do not understand the price they have to pay anymore and they will behave less optimal / rational, which makes the system less effective. So the system should be not too much differentiated to space, time or other magnitudes. On the other hand, Ubbels (2006) has investigated that

complexity does not influence the acceptance of the measure. Fast and easy to use are assumed to be present in a well developed charging system.
- Users should pay all costs for the ownership and the use of vehicles, including the costs for avoiding or for cleaning of pollution (ANWB). This is an important argument for distance based charging (demand) and can also be used to justify a congestion charge, because pollution is highest during the peak hour.
- The road user must have a choice between a tolled (and fast) route and a non-tolled (and maybe congested) route (ANWB). The existence of a free route is not in line with the previous point, because on every route car traffic causes a certain level of pollution. So this demand is not taken into account. Note that by the spatial differentiation of tolls still choices may be available between cheap and expensive routes.

In Ubbels (2006) and Jou et al. (2007) a list of acceptability issues concerning public opinion is given: a lot of objections exist against road pricing. Ubbels (2006) contains a list of guidelines for successful implementation to overcome these public issues. These guidelines are added below to the demands from road users.
- For people it is difficult to accept that they have to pay for congestion.
- People think it is not needed, unfair and not effective.
- Local business is afraid of losing customers.
- Unacceptable privacy issues.
- Implementation problems like unreliable technology and boundary issues.
- Revenue use should be clearly stated: use for lowering existing taxes on cars and fuels makes acceptance higher.
- The relation between acceptability and effectiveness is negative for most measures. Some measures are quite good on both criteria and seem the best trade-off between the two. These measures have a time-differentiated, relatively low charge level.
- The objective should be highly valued by the public and the public's behaviour should contribute to it.
- A limited and clear area of application of road pricing, like an inner city is more acceptable. This is not the form which is investigated in this research, so this demand will be omitted.
- Road pricing may deteriorate the socio-economic activities in the road pricing area.
- Road pricing may reduce traffic congestion and save travel time.
- Road pricing is unfair for the poor people.

From this analysis we conclude that the following demands are of interest to include:
- The higher the charges, the less acceptable they are, so the charges should have a maximum value (Ubbels, 2006).
- Fairness issues have to be considered. No system achieves a better situation for everyone, but the new situation should be fairer (charge per km) (Ubbels, 2006).

- The system should not be too complicated: not too much differentiation (ANWB).
- The system should be distance based (ANWB).

2.1.4 Public interests

In Hau (2006a) many demands are given. The demands concerning the society and public interests are shown here.

- Minimum of road work and environmental intrusion. In this research the existence of such a technical system is assumed.
- Provision of mixed traffic. The system should be able to identify different vehicle types, in order to be able to charge more to more polluting and bigger vehicles. In this research the existence of such a technical system is assumed.
- Handling of transitional phase. For public acceptability and softening of the system's full impact, a gradual introduction in the form of an experiment may be necessary. The final road pricing system should be capable of handling the whole vehicle population. In this research the final state will be investigated, so the transition is not taken into account.
- Modularity to add-on options. An ultimate road pricing system should be compatible with parking pricing systems, gasoline purchases, automatic route guidance, information systems, control systems of commercial vehicles of private firms, but this is not the focus of this research.
- Benefit-cost ratio should be high enough (from a social benefits point of view). In this research, the objective function measures the quality of the system. So this objective function should represent social benefits. The results of this research show indeed a reasonable improvement in the objective function value.
- Compatibility with other systems (see also EU demands). Existence of such a technical system is assumed.
- Fairness and availability of alternatives. When road pricing is introduced without additional measures, it is not acceptable to the public, because almost every traffic participant is worse off (even public transport users because of more crowded trains). So for the system to be fair, revenues should be invested in lowering fixed car taxes, in roads and in public transport (see also Ubbels, 2006). This is the case in the plans of the Dutch government and is not considered in this research.

One demand from public interests results:

- The objective function should be a magnitude that measures social benefits.

2.1.5 Legislation from the European Union

From the European Union some legislation can be found. For freight traffic the following two rules are important:

- Tolls shall be levied according to the distance travelled and the type of vehicle; user charges are scaled according to the duration of the use made of the infrastructure and to vehicles emission classes. This demand is in line with this research: a charge per kilometre.

- Both tolls and user charges can only be imposed on users of motorways or multilane roads similar to motorways as well as on users of bridges, tunnels and mountain passes. This demand is not in line with the ideas in this research, because here we would like to charge road users on all roads. Maybe it is not allowed to do so for freight vehicles.

In the White paper on European Transport Policy (European Commission, 2001) is stated that the road tax systems European Union should be more differentiated: the pricing scheme should better reflect the costs to the community. In this document an estimation of the value of external costs of a 100 km trip of a heavy goods vehicle is given, which is shown in Table 2.2. In the table the very high uncertainty of the figures has to be mentioned. But in general, the external costs are not very high, the very from 8 to 36 cents per kilometre for a heavy goods vehicle, so a price from 3,4 cents to 11 cents for cars seems reasonable in terms of external effects.

External and infrastructure cost	*Average range*
Air pollution	2.3 – 15
Climate change	0.2 – 1.54
Infrastructure	2.1 – 3.3
Noise	0.7 – 4
Accidents	0.2 – 2.6
Congestion	2.7 – 9.3
Total	8 – 36

Table 2.2: External and infrastructure costs of a heavy goods vehicle travelling 100 km on a motorway with little traffic (EUR) (European Commission, 2001)

Furthermore, some legislation exists on the interoperability of different European systems (EU guideline 2004/52), because an important aim of the European Union is the free traffic of persons and goods and free trade of services and capital. From 1st of January 2009 a system should be available for lorries to pay road charges in all the member states, having only a singe device in the vehicle. From 1st of January 2011 such a system should also be available for cars. This is also a technical aspect, so it is not considered further.

2.1.6 Conclusion: Demands

In this section a wide variety of demands is treated. Some of these demands are not relevant for this research. In this concluding paragraph the demands that are taken into account in this research are summarised. In the chapter on the mathematical model (chapter 3) these constraints are described quantitatively when possible.

- The system should have a maximum toll value.
- The tolls should be depending on the distance travelled.
- The toll should be differentiated to time and to space, in order to be effective and to leave choices for the road user.
- The toll should be differentiated to environmental properties of the vehicle.

- The fare should not be too much differentiated, because the system should be understandable.
- The revenue of the new system should be reliable and should yield about the same revenue as the present road taxes.
- The social benefit of the system, or the increase in objective value, should be high enough compared to the costs.

2.2 Pricing scheme framework

Designing a road pricing measure from given policy objectives and constraints is a very difficult problem since there is large solution space in terms of price differentiations towards space, time, and users. The objective of this section is to determine a promising pricing scheme given the demands from the previous section. This pricing scheme describes the level of differentiation in space, time and user group, but the price levels within each differentiation are not yet determined. These need to be optimised given policy objectives. In chapter 6 the optimal price levels will be approximated for a specific case study.

In order to illustrate the concept of a pricing scheme, in this section first two examples of possible pricing schemes are given. After that, a schematic overview of possible pricing schemes is given, in order to explore the possibilities concerning pricing schemes. To explore pricing schemes in other literature and practical situations, a short literature review is given. Then, taking into account the demands from the stakeholders from the previous section, a choice is made for a pricing scheme in this study.

2.2.1 Examples of pricing schemes

A well known road pricing scheme is to assign a price to each link in the network in each time slot. In this model each link in the network gets a price at each defined time interval. In this way both space and time differentiation of prices are possible (more details in section 2.2.3).

Another possibility is to charge parking instead of driving. The parking charge will therefore not be based on paying per hour as is currently often implemented, but the price per hour for each destination depends on the arrival time and the departure time or on the arrival time and the distance between the origin and the destination. Because arriving and leaving in the rush hour contributes most to congestion, these times will be the most expensive.

2.2.2 A schematic overview

In this section a schematic overview of possible pricing schemes is given. It is partly based on the literature review in the next section. The overview does not aim to cover all possible pricing schemes, but to give a general overview which gives enough structure to this research and defines the pricing scheme framework of this research. First the used definition of pricing scheme is given. An overview of possible pricing schemes follows.

In road pricing literature different road pricing schemes are known. The used technology, the use of revenue, and the paying procedure are properties that are not considered in this research. So in this research a scheme consists of a fare base, a level of differentiation in possible price levels and a number of attributes to which the fare is differentiated. For each of these attributes a level of differentiation can be chosen (including no differentiation). In this research two specific attributes are chosen to which the fare is differentiated, namely time-of-day and space. In Table 2.3 the possible pricing schemes to be considered in this research are listed. Any combination between a fare base, a number of tolling classes, an amount of differentiation in time-of-day and an amount of differentiation in space is a possible pricing scheme in the sense of this research. It may be possible some particular combinations are not realistic pricing schemes. In this case the scheme will not be considered.

Fare base	**Number of tolling classes**	**Amount of differentiation in time-of-day**	**Amount of differentiation in space**
Distance travelled	Few	None	None
Time spent	Medium	Rough	Rough
Number of passages	A lot	Medium	Medium
Parking time	Continuous	Detailed	Detailed
OD relation			
Other innovative fare base			

Table 2.3: Different possible pricing schemes

Other attributes to which the fare may be differentiated are:

- Income of the car owner / driver
- Trip frequency
- Purpose of the trip
- Vehicle type

In this research these latter ways of differentiation are not considered as a design variable, in order to limit the scope to the relevant levels of differentiation from a practical viewpoint.

2.2.3 Literature review

Hilbers et al. (2007) gives the three pricing schemes which are currently being investigated by the Dutch government. These schemes are listed here. Remarkable is that the toll is fixed at a predetermined level.

- Flat kilometre charge of 3,4 euro cent per kilometre.
- Flat kilometre charge of 3,4 euro cent per kilometre and an additional congestion charge of 11 euro cent per kilometre. This congestion charge will be charged during peak hours (7AM-9AM and 4PM-6PM) on heavily used roads: roads with a

flow / capacity ratio of more than 0,8, which result from the standardised NRM model framework (more information on the NRM model see for example Kiel and Smit, 2004).
- Flat kilometre charge of 3,4 euro cent per kilometre and an additional congestion charge of 11 euro cent per kilometre in combination with investments in extra infrastructure.

From Ubbels (2006) some examples of pricing schemes are listed in Table 2.4. All four examples have different primary goals and some comments are added on the way of implementation. Ubbels also stated that in practice a system is possible in which more than one road authority sets prices and even private parties are possible price setters. For simplicity in this research is assumed that all prices are decision variables: one big road authority without private parties.

Area	*Pricing scheme*	*Goal*	*Comments*
Norway	Cordon	Revenue generation to improve infrastructure	
London	Cordon	Demand management to reduce huge congestion	Revenues invested in public transport
Stockholm	Cordon	Reduce environmental problems	First an experiment, a referendum follows
USA	Priority lanes	Reliability and stimulating carpooling (multi occupancy vehicles are free)	Value pricing is used for marketing purposes and is clearly communicated

Table 2.4: Experience from practice

In Joksimovic (2007) some examples of possible tolling regimes are given and some straightforward mathematical formulations for these regimes are given. Here the tolling regimes are mentioned shortly.
- A fixed fare charged at the entrance / exit of a cordon.
- A variable fare charged at the entrance / exit of a cordon.
- Fare dependent on the length or duration of the trip.
- A space variable fare dependent on the periods of travelling.

In Hau (2006a) an overview is given of different methods of road use charging, including a description of the system and places of implementation or places where plans for this system are made. All these systems are conveniently arranged in a diagram on page 97 of the paper. Furthermore, six particular systems of road pricing are judged based on 20 criteria. In this judgement three possibilities are used: low, medium or high score. No weights are used, because the author argues that different views on the weights exist, so it is better to not use them at all. The six systems are:

- 24 hours manual toll collection in a toll plaza.
- Cordon pricing, combination of manual toll collection and reserved lanes for licence holders.
- Supplementary licensing.
- Electronic toll collection (24 hours) with Automated Vehicle Identification technology.
- Electronic road pricing (differentiated fares) with AVI
- Electronic road pricing (differentiated fares) with smart card technology.

From this analysis the two forms of electronic road pricing are considered to be the best alternatives, with the AVI technology slightly better than the smart card technology. So in the research it is justifiable to investigate time and space differentiated road pricing schemes.

Pieper (2001) proposes a system of road types for the whole country of the Netherlands. In this system every link in the road network is classified as a certain road type. Every road type is associated to a certain price per kilometre. Busy roads get a higher price than quite roads. Furthermore, during the day a certain road can change from one road type to another road type, for example the road becomes busier during the peak period. The incorporation in road types can be communicated to the road users by using colours on maps. In Figure 2.1 an example of this visualisation is given for two different moments in time. All these road categories are predetermined, but can be adapted on the long run, when changes in the traffic performance occur. The system can be characterized as a distance based space and time differentiated toll.

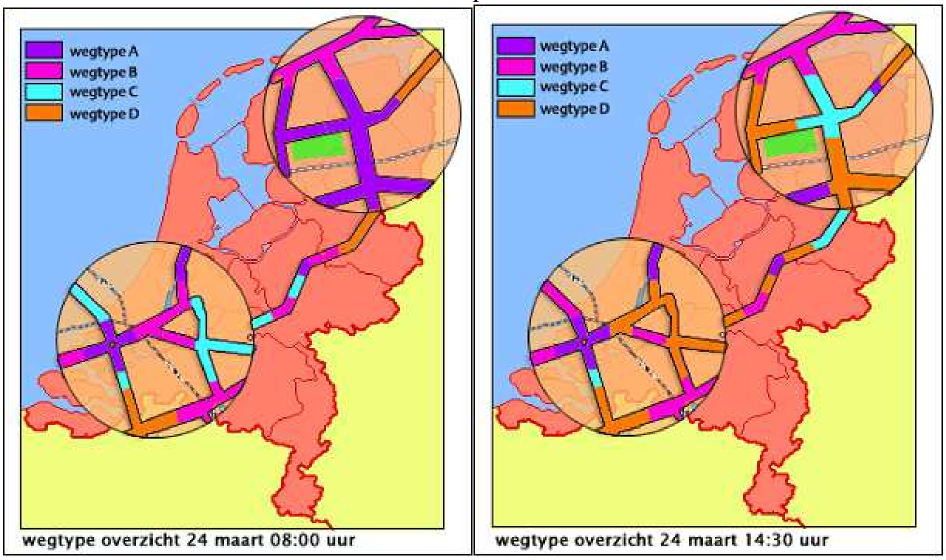


Figure 2.1: Illustration of the concept of road types (wegtype) at two different times. Every road type can be identified by a particular colour, that may change over time (Pieper, 2001)

2.2.4 Selection of a pricing scheme

In this section we justify the choice of a pricing scheme to be used in this research. First the choice is made based on the constraints by the stakeholders. After that, two more justifications are added: one from a computational point of view and one from results from the literature.

From the section on stakeholders a list of constraints to a pricing scheme follows. In this paragraph is explained what scheme is chosen and why. The used fare base will be distance travelled, because this is the base the Dutch government is willing the implement in the Netherlands (Hilbers et al, 2007) and because it seems a fair way of paying: the more you use the roads, the more you pay (as a contribution to construction and maintenance). Furthermore, a scheme with differentiated fares to time and to space is chosen, because this is more effective and this leaves choices for the road user, so it is more acceptable. To keep this system understandable for the public, the fare should not be too much differentiated. It should be clear to people what they pay before they travel, otherwise it is not possible for them to adapt their behaviour to the tolls. So links may be put together in a group with the same fare and the number of time intervals should be not too big. Also the number of price categories should not be too big. In the definition of the price categories a maximum toll value is incorporated.
After this selection a few constraints remain:

- The fares should be differentiated to environmental properties of the vehicle. This constraint is not dependent on the pricing scheme and it is possible to incorporate this constraint in the final implementation. In this research for simplicity only one type of cars is modelled, so no differentiation to environmental vehicle properties was possible.
- The average tax revenue should remain the same. This constraint does not have influence on the chosen pricing scheme, because every scheme is able to collect enough revenue. It is added mathematically to the model in chapter 3.
- The increase in the objective function should be high enough compared to the costs. This value was not known in advance, so it could not be used for the selection of the pricing scheme. In the chapter 6 on results the effect on the objective function is shown.

Finally, Tillema (2007) gives a method was found to judge a pricing scheme easily. Three possible criteria to judge a road pricing system are given. These criteria are relatively simple and investigated fast. These criteria are:

- Effectiveness: the extent to which it reaches a certain goal
- Efficiency: the extent to which it increases social welfare
- Acceptance: the extent to which it is supported by the society

The first and the third criterion compete with each other: highly effective measures are usually not very acceptable for the society (Ubbels, 2006). A time differentiated fare with relatively low fares is the scheme with the best trade-off between the two criteria according to this source. The efficiency could not be taken into account when the pricing scheme was chosen, because the increase in value of the objective function was not known in advance.

2.3 Conclusion

After an investigation of possible pricing schemes and different opinions from stakeholders and from the literature, we decided to use the following pricing scheme in this research:

- A time differentiated,
- space differentiated,
- link based,
- fare per kilometre
- with a nonnegative, not too high fare,
- and not too many different prices.

This corresponds to the idea of Pieper (2001). The exact form of the road pricing model can now be defined in chapter 3 and 4 on the mathematical model and on the traffic model.

3 Mathematical model

In this chapter the optimal toll design problem is formulated as a mathematical optimisation problem. This gives insight into the structure of the problem. The mathematical formulation is adapted to meet the demands we chose from the stakeholder analysis and to fit the chosen pricing scheme (chapter 2). The formulation is used in chapter 4 with the traffic model and in chapter 5 it is operationalised by the solution method.

First an introduction of the bi-level formulation is given. After that, the symbols which are used are presented: an overview of all used sets, indices and variables is given. This is succeeded by a discussion on the different formulations of objective functions and on the mathematical constraints in the model. After this, some possible equivalent alternative formulations are given, which have advantages and disadvantages compared to the formulations in the earlier sections. The chapter ends with a section on the size of the solution space. The formulations in this chapter are based on the formulations of Joksimovic (2007) and Bliemer (2001).

3.1 Bi-level formulation

The problem in this research, the dynamic optimal toll design problem, is formulated as a mathematical problem with equilibrium constraints. This is also called a bi-level network design problem (Joksimovic, 2007; Viti *et al.*, 2003). In such a model an upper level and a lower level exist (see also Figure 3.1) . The upper level contains the objective function, constraints, and the design variables. In this case, the design variables consist of the amount of toll. This amount of toll can be seen as a link characteristic. The lower level is an optimisation problem which reacts to the design variables in the upper level. Here, the lower level consists of the users of the traffic system and their reaction to the tolls: users individually optimise their generalised costs, consisting of their travel time, schedule delay, tolls and other monetary costs. In the lower

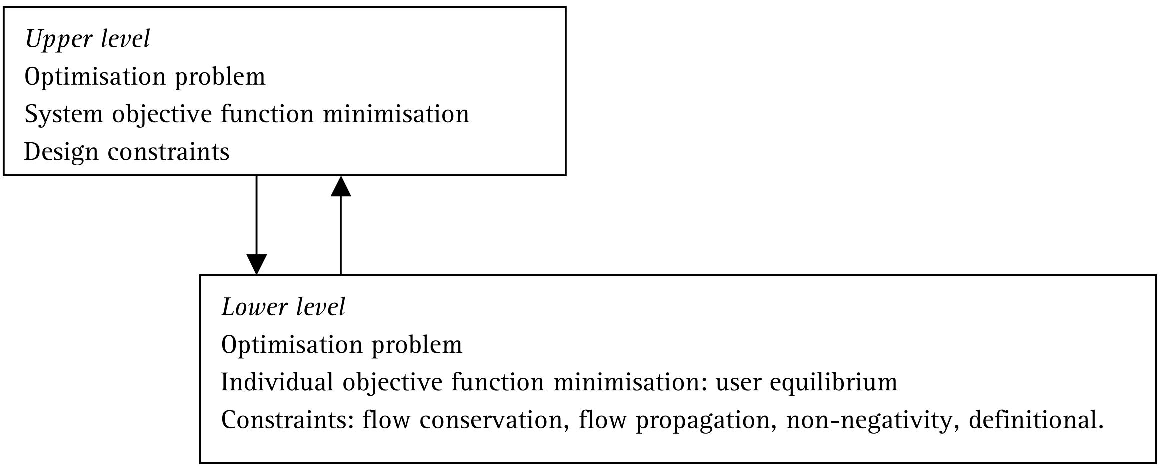


Figure 3.1: An illustration of a bi-level problem: The upper level is subject to the lower level

level it will be assumed that the traffic system will achieve dynamic stochastic user equilibrium (DSUE, see chapter 4 on the traffic model). Due to heterogeneity in the population, not every individual behaves in exact the same way, which is modelled by a stochastic approach. A logit model is used to model this stochastic behaviour: the traffic is distributed along departure times and routes, with most traffic on the most attractive times and routes, and no traffic on very bad times and routes (see section 4.2.1 and 4.2.2). The users' reaction to the tolls is used as a constraint for the optimisation of the upper level problem. From the results of the lower level, the network performance can be measured in the upper level. The constraints that result from the demands of the stakeholders, are added as constraints to the upper level of the problem as well.

3.2 Upper level

The upper level of the mathematical program consists of the optimisation problem from the road authority point of view. This level contains the design variables: the toll levels and direct constraints to the toll levels. The measurement of the value of the objective function also takes place in this level.

3.2.1 Sets / indices

In this section, the used sets are given, including their corresponding indices (see Figure 3.2). Most sets are straightforward and usual in mathematical formulations of traffic problems, like links, nodes, origins, destinations, routes, and user classes. For this particular problem some sets need some extra explanation. The modelling environment is dynamic, but in order to make computations possible, it is necessary to define discrete time windows. These time windows need to be a short period of time, to achieve a sufficient departure time modelling. But in order to improve the comprehensibility of the pricing system, it is not possible for every time window to get a different price. Therefore, groups of successive time windows are defined. The time windows within each group all get the same price. In practice, these groups should have easy beginning and starting times, for example all time windows between 8.00 AM and 8.30 AM are together in a group. Such a time window group is also called a time interval. With the same argument, groups of links are defined. In these groups, links which physically form a unity are put together, for example a corridor of links or all links in some naturally connected area. Finally, a set of toll categories is defined, because again for comprehensibility it is desirable to have only a few different toll levels.

3.2.2 Variables

In this section all variables of the model are defined (see Figure 3.2). A distinction is made between variables that define the network, definition variables, state variables and decision variables. Because a set of toll categories is defined, an indicator

Sets / indices

$a, b \in A$	Links
$n \in N$	Nodes
$i \in I \subseteq N$	Origins
$j \in J \subseteq N$	Destinations
$t, w \in T$	Time windows
$p \in P$	Toll categories
$r \in R_{ij}$	Routes between OD pair *ij*
$m \in M$	User classes
$g \in G$	Groups of links
$h \in H$	Groups of time windows (time intervals)

Network

A_{an}	Network incidence matrix: -1 when link *a* goes into node *n* and 1 when link *a* goes out of node *n*.
d_{ijm}	Total possible demand of user class *m* between origin *i* and destination *j* during total time period $\sum_t k_t$. In case no elastic demand is modelled, this is equal to the actual demand on OD pair *ij*.
PAT_{ijtm}	Fraction of the total demand on OD pair *ij* of user class *m* who prefer to arrive during time window *t*.

Definition

q_p	Toll level category *p* (€)
k_t	Length time window *t* (h)
α_m	Value of travel time (€/h) for user class *m*
β_m	Value of schedule delay early departure (€/h) for user class *m*
γ_m	Value of schedule delay late departure (€/h) for user class *m*
χ_m	Value of schedule delay early arrival (€/h) for user class *m*
ι_m	Value of schedule delay late arrival (€/h) for user class *m*
l_a	Length link *a* (km)
c_a	Capacity link *a* (veh/h)
s_a	Free flow speed link *a* (km/h)
μ_{ag}	Index groups links: equals 1 if link *a* is in link group *g*, and equals 0 otherwise (binary)
θ_{wh}	Index groups time periods: equals 1 if time window *w* is in time window group *h*, and equals 0 otherwise (binary)
δ_{awt}^{rij}	Dynamic route-link index for route $r \in R_{ij}$ and departing during time window *w*: equals one if this traffic flows into link *a* at time window *t*, and equals 0 otherwise (binary)
κ_{rij}	Other constant monetary costs associated to route $r \in R_{ij}$, for example fuel costs

Figure 3.2: Notation in chapter 3 and 4 of this research

State variables

d_{ijwm} Total demand in the situation without tolls of user class *m* between origin *i* and destination *j* departing during time window *w* (veh)

u_{amt} Average inflow on link *a* during time period *t* of user class *m* (veh/h)

v_{amt} Average outflow on link *a* during time period *t* of user class *m* (veh/h)

u_{rijwm} Flow on route $r \in R_{ij}$ departing during time window *w* of user class *m* (veh/h)

υ_{amt} Average speed on link *a* during time period *t* of user class *m* (km/h)

τ_{at} Average travel time on link *a* during time period *t* (h)

τ_{rijwm} Travel time for user class *m* on route $r \in R_{ij}$ when departing during time period *w* (h)

p_{rijwm} Total toll price for user class *m* on route $r \in R_{ij}$ when departing during time window *w* (h)

U_{rijwm} Total utility for user class *m* on route $r \in R_{ij}$ when departing during time window *w* (h)

C_{at} Congestion indicator: equals 1 if $\upsilon_{at} \leq 0,6 s_a$ and equals 0 otherwise

$ASDE_{ijwtm}$ Traveller specific arrival schedule delay early when a traveller who prefers to arrive during time window *t* departs during time window *w* from origin *i* to destination *j* (h)

$ASDL_{ijwtm}$ Traveller specific arrival schedule delay late when a traveller who prefers to arrive during time window *t* departs during time window *w* from origin *i* to destination *j* (h)

$DSDE_{ijwtm}$ Traveller specific departure schedule delay early when a traveller who prefers to arrive during time window *t* departs during time window *w* from origin *i* to destination *j* (h)

$DSDL_{ijwtm}$ Traveller specific departure schedule delay late when a traveller who prefers to arrive during time window *t* departs during time window *w* from origin *i* to destination *j* (h)

ψ_{rijwm} Fraction of the demand of user class *m* on OD pair *ij* during time window *w* which follows route $r \in R_{ij}$ (So $\sum_{r \in R_{ij}} \psi_{rijmw} = 1 \; \forall \; i, j, m, w$)

φ_{ijwm} Fraction of the demand of user class *m* on OD pair *ij* which departs during time window *w* (so $\sum_{w} \varphi_{ijmw} = 1 \; \forall \; i, j, m$)

ϕ_{ijm} Fraction of the possible demand d_{ijm} which decides to travel by car after pricing measures. So $0 \leq \phi_{ijm} \leq 1$. When no elastic demand is modelled $\phi_{ijm} = 1$ holds.

Decision variables

I_{atp} Indicator of the link price: equals 1 if price category *p* is assigned to link *a* during time period *t*, and equals 0 otherwise (binary).

Figure 3.2: Notation in chapter 3 and 4 of this research (continued)

formulation is chosen. The indicator I_{atp} assigns a toll category p to a particular link a in a particular time window t and is the only decision variable. All other variables are given parameters or are dependent on this decision variable.

3.2.3 Objective function

In this section we present four possible objective functions.

Maximisation of toll revenue

In equation (3.1) the objective function for the maximisation of revenues is stated. It is assumed that people have to pay at the time they enter a link, so the inflow is used for toll collection. In this equation a multiplication is made of the indicator, the inflow, and the level of the toll. In order to translate the flow into a number of vehicles, this number is multiplied by the length of the time window and to translate the fare per kilometre into an amount, this number is also multiplied by the length of the link. Now a summation is made over all price categories, user classes, links, and time windows to cover all collected tolls in the network during the complete time period.

$$\max_{I_{atp}} \sum_a \sum_t \sum_p \sum_m u_{atm} I_{atp} q_p k_t l_a \tag{3.1}$$

Minimisation of travel time

In equation (3.2) the objective function is shown for the minimisation of the total travel time. Again, the link inflow is multiplied by the length of the time window and by the quantity to be measured: the actual travel time on each link and during each time window. Note that the objective function does not contain the design variable I_{atp}: this variable determines the outcome of the lower level, which again determines the value of the objective function. In equation (3.3) the objective function is shown for the minimisation of the average travel time. The only difference between the two is the addition of a division by the total number of travellers in the latter case. This total number of travellers should be defined in an additional constraint, which is shown in equation (3.4). The first formulation is in terms of the total flow from all origins during all time windows, while the second is in terms of the total flow towards all destinations during all time windows. This assumes all traffic reaches its destination within the modelling period, so no traffic is on the network at the end of the time period. Furthermore, it assumes that no through traffic exists through origin or destination nodes. This corresponds to the use of feeder links in the network. In terms of the origin destination matrix the total number of travellers can also be defined as shown in equation (3.5). In this case it can be computed before the actual assignment has taken place.

$$\min_{I_{atp}} \sum_a \sum_t \sum_m u_{atm} \tau_{at} k_t \tag{3.2}$$

$$\min_{I_{atp}} \sum_a \sum_t \sum_m (u_{atm} \tau_{at} k_t) / N \tag{3.3}$$

$$N = \sum_i \sum_a \sum_t \sum_m \max(0, A_{ai}) u_{atm} = -\sum_j \sum_a \sum_t \sum_m \min(0, A_{aj}) u_{atm} \tag{3.4}$$

$$N=\sum_{i,j}\sum_{m}d_{ijm}\phi_{ijm} \tag{3.5}$$

Minimisation of delay

The next possible objective function is minimisation of the total delay, which is shown in equation (3.6). This objective is similar to the minimisation of travel time. The delay per link is computed by subtracting free flow travel time from the actual link travel time. The minimisation of the average delay is shown in equation (3.7). This formulation is link based, which can be a problem, as will be illustrated here shortly. When no delays occur in a network, the objective value will be equal to 0, also when drivers make huge detours to reach their destinations. This is not a desirable situation, so when delay is an objective, an OD based formulation would be better. However, on a relatively small network with few possibilities to make such detours, this will only be a minor problem.

$$\min_{I_{atp}}\sum_{a}\sum_{t}\sum_{m}(\tau_{at}-l_a/s_a)u_{atm}k_t \tag{3.6}$$

$$\min_{I_{atp}}\sum_{a}\sum_{t}\sum_{m}((\tau_{at}-l_a/s_a)u_{atm}k_t)/N \tag{3.7}$$

Minimisation of congestion

When congestion is chosen as an objective function, a new variable (C) should be defined which indicates whether a link is considered to be congested or not. The objective is straightforward and is given in equation (3.8). In this equation the congestion indicator is multiplied by the length of the link and by the capacity of the link, in order to give a higher weight to congestion on longer links and on links with multiple lanes. The congestion indicator C is defined in constraint (3.12).

$$\min_{I_{atp}}\sum_{a}\sum_{t}\sum_{m}C_{atm}l_a c_a \tag{3.8}$$

The objective function in this research

The choice of the objective function is dependent on the desired policy objective. In the framework in this research it is possible to choose any objective function. In the case study in this research the average travel time in the network is chosen, because it can be easily interpreted by the reader and because it represents the quality of the system as a whole, from a traffic engineering point of view.

3.2.4 Constraints

Link group constraint

A constraint is needed to force links in the same link group to get the same price.

$$\begin{aligned}(1-\sum_{g}\eta_{ag}\eta_{bg})+I_{apt}\geq I_{bpt} \quad &\forall\, a,b>a,t,p\\ -(1-\sum_{g}\eta_{ag}\eta_{bg})+I_{apt}\leq I_{bpt} \quad &\forall\, a,b>a,t,p\end{aligned} \tag{3.9}$$

In equations (3.9) all links in the same group as link a are forced to get the same price as link a. When a and b are in the same group, $\sum_{g}\eta_{ag}\eta_{bg}=1$ holds. In this case constraints (3.9) will change to $I_{apt}=I_{bpt}$. When a and b are not in the same group,

$\sum_g \eta_{ag}\eta_{bg} = 0$ holds. In this case, constraints (3.9) give no restrictions for the indicator of link *b*, because it is a binary variable. Only links *b* with a higher number than link *a* need to be compared, because otherwise the constraint appears twice for each pair of links.

Time window group constraint

Just like the link group constraint, a constraint is needed to force time windows in the same time window group to get the same price.

$$\begin{aligned} (1-\sum_h \theta_{th}\theta_{wh}) + I_{apt} \geq I_{apw} \quad \forall \quad a,p,t,w>t \\ -(1-\sum_h \theta_{th}\theta_{wh}) + I_{apt} \leq I_{apw} \quad \forall \quad a,p,t,w>t \end{aligned} \tag{3.10}$$

In equations (3.10) the constraints for the groups of time windows are given They are similar to the link group constraints.

Indicator constraint

A link should be assigned to exactly one price category at every time window (a zero toll is also considered to be a price category). So the following constraint has to be taken into account:

$$\sum_p I_{atp} = 1 \quad \forall \quad a,t \tag{3.11}$$

Congestion constraint

When congestion is chosen as an objective function, a constraint should be added to define the congestion indicator *C* properly. When the speed on a link is lower than 60% of the free flow speed, the link is defined as congested. Equation (3.12) forces the congestion indicator *C* to equal 1 when the actual speed is lower than 0,6**s*. In this constraint *M* stands for a sufficiently large number. In this context it is sufficient to define the value of *M* to be 0,6**s*, because the speed cannot become negative. Because the objective is minimized, the value of *C* will be 0 when possible and the value will only be 1 if the speed is really lower than 0,6 times the free flow speed.

$$C_{atm}M + \upsilon_{atm} \geq 0,6 s_{am} \quad \forall \quad a,t,m \tag{3.12}$$

Minimum and maximum toll constraint

A minimum and a maximum should be defined for the amount of toll (€ per km) on all links. In the chosen formulation, this constraint is incorporated in the definition of toll categories, so it is not necessary to add the constraints explicitly.

Speed – flow relationship

In this model a direct relationship between flow and speed is assumed for every link. Flow is a function of the speed, see equation (3.13).

$$u_{atm} = f(\upsilon_{atm}) \quad \forall \quad a,t,m \tag{3.13}$$

Equilibrium constraint

In the formulation in this chapter, the lower level traffic equilibrium model is not formulated explicitly, because it is determined numerically by traffic engineering software. Here, it is summarised by one equilibrium constraint, which can be stated easily by the formulation in equation (3.14).

$$u_{atm} \; satisfies \; SDEU \quad \forall \quad a,t,m \tag{3.14}$$

The modelling framework of the lower level problem is used to approximate DSUE and is treated in chapter 4 on the traffic model. Furthermore, a description of a lower level in a bi-level formulation in the context of road pricing can be read in Joksimovic (2007). Exact mathematical formulations of the lower level constraints can be read in Bliemer (2001).

Min / max revenue
Based on the stakeholder analysis a constraint should be added concerning the minimum and the maximum revenues of the system. This constraint can only be used when revenue maximisation is not used as an objective function and can be easily deduced from this objective function. Let *L* and *H* be the lower and upper bound for the revenue of the system. The constraint in equation (3.15) follows.

$$L \leq \sum_a \sum_t \sum_p \sum_m u_{atm} I_{atp} q_p k_t l_a \leq H \tag{3.15}$$

3.2.5 Alternative formulations

The group constraints are rather complicated and many indicator variables are necessary in this formulation. An alternative formulation could be to give a price to a group instead of giving it to a link. In this formulation, the indicator should be dependent on the groups instead of the links, so a lot fewer variables are needed. In the objective functions and all other formulas the link indicator should be replaced by a group indicator. The relation between those two indicators is shown in the following relation: $I_{atp} = \sum_g \mu_{ag} I_{gtp}$. The new formulation of the max revenue objective is shown in equation (3.16). The sum is now also taken on the index groups. The other objective functions change in a similar way.

$$\sum_a \sum_g \sum_p \sum_t \mu_{ag} I_{gtp} f_{at} q_p k_t l_a \tag{3.16}$$

Another aspect which can be formulated differently is the indicator formulation. It could also be possible to define an integer variable for the toll level (I_{at}), a step size (*d*) and lower and upper bounds $0 \leq I_{at} \leq p_{\max} / d$

3.3 Solution space

This section pays attention to the size of the solution space of the formulated mathematical program. The formulation is based on a limited number of possible price categories |P|. So the solution space is divided in a finite number of discrete points, because the upper level parameter, the toll level, is discretised. Each combination of a time window group *h* and a link group *g* should be assigned to a price category. The number of combinations is equal to |G|*|H|. So the number of possible solutions is equal to $|P|^{|G|*|H|}$. This number grows exponentially with the number of link groups and with the number of time window groups, so care is needed when these are chosen. The growth with the number of price categories is polynomial, though in most cases |G|*|H| will not be a small number, so the growth in |P| will be substantial as well. In Table 3.1, the size of the solution space (the number of possible solutions) is given for some numerical examples. It can be observed that small reductions in |G| or in |H| give huge reductions in the solution space. Reductions in |P| still give big reductions in the solution space, but the effect is less than reducing |G| or |H|. Note that in this example the number of possible solutions stays the same when the numbers of |G| and |H| are switched.

Number of link groups |G|	*Number of time intervals* |H|	*Number of price categories* |P|	*Number of possible solutions*
3	6	5	3.81E+12
3	5	5	3.05E+10
3	4	5	2.44E+08
2	4	5	3.91E+05
3	4	4	1.68E+07
3	4	3	5.31E+05

Table 3.1: the size of the solution space

3.4 Additional constraints

From this analysis can be concluded that the solution space is still too big to be searched exhaustively. One approach to handle this problem is the use of a smart search algorithm. This approach is treated in chapter 5. In this section it is argued that it is possible to add some additional constraints to the solution space, in order to make the number of possible solutions smaller. All constraints are based on assumptions, which can be assumed to be true in certain situations.

1. When a clear peak period is modelled, the prices will also have a peak. So before the peak, the price of certain time windows should always be lower than or equal to the next interval. After the peak, the price of certain time windows should always be higher than or equal to the next interval.
2. When it is clear that the busiest period is in the middle of the modelled period, it can be said that the peak toll should be achieved within some particular time periods.
3. When the network is highly congested, it is likely that in a certain period the maximum toll level is reached. Solutions that do not have a maximum toll level can be omitted.
4. The toll peak can be set symmetric. When there are only four time windows in a peak period, this means that a distinction between the centre of the peak

and the border periods of the peak is made. In fact, this is the same as putting these time windows in the same time window group and will effect the number of time windows. So it is not treated separately here.

5. A ratio between link types, for example between urban roads and highways, can be defined. This can be motivated when certain roads are more expensive to build or when certain roads clearly have more external effects than other roads.
6. The price on a certain link type is always smaller or equal to another link type. This can be motivated in the same way.

To give an indication on the reduction of the solution space by the first three constraints, the effects of these constraints are computed in the case of no space differentiation, 4 time windows and 5 price categories. The results are given in Table 3.2. The application of all three constraints gives a size reduction of the solution space of a factor 5.

Without additional constraints	625
Constraint 1: peak constraint	295
Constraint 1 and 2: peak in period 2 or 3	224
Constraint 1 and 3: peak at maximum toll level	165
Constraint 1, 2 and 3	125

Table 3.2: Reduction of the solution space by additional constraints.

3.5 Bi-level model: a theoretical approach

In this section we shortly discuss a bi-level formulation for traffic control by using tolls from a mathematical viewpoint. We however restrict our study to the simplest case where the individual driver behaves according to a (static) equilibrium network with fixed origin-destination demand.

3.5.1 The wardrop traffic equilibrium

Before introducing the bi-level (BL) model we give a concise introduction into the theory of (Wardrop) traffic equilibria (see Wardrop, 1952).

In the well-known classical Wardrop traffic equilibrium model we have given a traffic network *(N,A)* with a set W of origin-destination pairs (OD-pairs for short) $(i_w, j_w) \in N \times N$ and corresponding demands $d_w \geq 0, w \in W$. Let R_w denote the set of directed routes from i_w to j_w and let $\Re := \bigcup_w R_w$. A (link based) traffic flow for the given demand $d \in \mathbb{R}_+^w$ is a vector x of the form

$$x = \Delta u, \quad u \geq 0, \quad \Lambda u = d \tag{3.17}$$

where the components u_r of $u \in \mathbb{R}^{\Re}$ is the amount of flow on route $r \in R_w$ of OD-pair w and the elements of the matrices $\Delta \in \mathbb{R}^{A \times \Re}, \Lambda \in \mathbb{R}^{W \times \Re}$ are defined by

$$\Delta_{ar} = \begin{cases} 1 & a \text{ is a link of } r \\ 0 & \text{otherwise} \end{cases} \qquad \Lambda_{wr} = \begin{cases} 1 & r \in R_w \\ 0 & \text{otherwise} \end{cases}$$

We let $X_d \subseteq \mathbb{R}_+^A$ denote the set of all feasible traffic flows for demand $d \in \mathbb{R}_+^W$, $X_d = \{x \mid x \text{ satisfies (3.17) with some } u\}$.

Further, let $c_a : \mathbb{R}_+^A \to \mathbb{R}_+$ be nonnegative continuous link cost functions defining costs or travel times $c_a(x), a \in A$, for a given traffic flow $x \in \mathbb{R}_+^A$ (not necessarily separable). We also say that x induces the link costs $c_a(x), a \in A$ and the route costs $c_r(x) = \sum_{a \in r} c_a(x), r \in \Re$.

Now consider a traffic flow $\overline{x} = \Delta \overline{u}$ as in (3.17) with corresponding induced costs $c_a(\overline{x})$. Then $\overline{x}$ is called a *Wardrop equilibrium* flow (W-E flow) if for all $r, s \in \Re_w, w \in W$, the following stability condition is satisfied:

$$\overline{u_r} > 0 \quad \Rightarrow \quad c_r(\overline{x}) \le c_s(\overline{x})$$

In other words, $\overline{x}$ is an equilibrium flow if it routes all the demand d_w along min costs paths (relative to $c(\overline{x}) \in \mathbb{R}_+^A$ in $\Re_w, w \in W$).

The next lemma gives different characterizations of a Wardrop equilibrium, see Bergenhoff *et al.* (1997) for a similar result.

Lemma 3.1 The following is equivalent for $\overline{x} \in X_d$.

(1) The flow $\overline{x}$ is a Wardrop equilibrium

(2) (variational inequality) $c(\overline{x})^T (x - \overline{x}) \ge 0 \; \forall \; x \in X_d$

(3) (Nash equilibrium) $\overline{x}$ is a minimiser of

$$\min_x c(\overline{x})x \quad s.t. \quad x \in X_d \tag{3.18}$$

(4) The (generalized) Karush-Kuhn-Tucker (KKT) relation holds with $u \ge 0$ and multipliers $\lambda \in \mathbb{R}^A, \sigma \in \mathbb{R}^W, \mu \in \mathbb{R}^{\Re}, \mu \ge 0$:

$$\begin{aligned} &\Lambda u = d \qquad c(\overline{x}) = \lambda \qquad u^T \mu = 0 \\ &\Delta u - \overline{x} = 0 \qquad 0 = \Lambda^T \sigma - \Delta^T \lambda + \mu \end{aligned} \tag{3.19}$$

Proof

Note that $x \ge 0$ is automatically implied by the relations $x = \Delta u, u \ge 0$ in (3.17)

$(4) \Leftrightarrow (1)$: Using $\Delta_r^T c(\overline{x}) = c_r(\overline{x})$ (Δ_r^T denotes the rth column of Δ and w denotes the corresponding OD-pair) we find from (4)

$$\sigma_w = \Delta_r c(\overline{x}) - \mu_r \begin{cases} = c_r(\overline{x}) & \text{if } u_r > 0 \\ \le c_r(\overline{x}) & \text{if } u_r = 0 \end{cases}$$

which yields the Wardrop equilibrium conditions and vice versa.

$(3) \Leftrightarrow (4)$: For the linear program in (3) it is well-known that the minimality condition for $\bar{x}$ are equivalent with the KKT conditions in (4).
$(3) \Leftrightarrow (2)$: For $\bar{x} \in X_d$ the condition that $\bar{x}$ solves the LP in (3) is equivalent with $c(\bar{x})x \geq c(\bar{x})\bar{x}$ for all $x \in X_d$, i.e., (2).

Remark 3.1 If in addition the costs are separable, i.e., $c_a(x) = c_a(x_a)$ does only depend on x_a and $c_a(x_a)$ are nondecreasing, the conditions in Lemma 3.1 are equivalent with: $\bar{x}$ is a solution of the following Beckmann minimization problem:

$$\min_{x,u} \sum_{a \in A} \int_0^{x_a} c(\tau)d\tau \quad s.t. \begin{cases} \Lambda u = d \\ \Delta u - x = 0 \\ u \geq 0 \end{cases} \tag{3.20}$$

Note that by the monotonicity assumption the objective of program (3.20) is a convex function and the KKT-condition is equivalent with minimality. If the costs $c_a(x_a)$ are strictly increasing then the objective is strictly convex and the minimizer (KKT-point) is uniquely determined.

3.5.2 The bi-level model

To motivate the model we can argue that the public traffic planner has objectives which are different from the objective of the individual network user. The former wishes *e.g.*, to minimise air pollution and/or congestion. This can possibly be achieved by an appropriate toll pricing system where the requirements of the public planner can be modelled by a BL problem.
Let in the sequel $p \in \mathbb{R}^A$ denote the toll vector with toll price p_a for the links $a \in A$ and suppose the objective (to be minimised) of the planner is given by a function $S(x, p)$. Often the toll vector is subject to bounds

$$p^- \leq p \leq p^+$$

If we assume that for given OD demand *d* the behaviour of the network user is modelled by the W-equilibrium condition, then the system planner tries to choose a toll vector $\bar{p}$ such that together with the corresponding W-E $\bar{x} = x(\bar{p})$ the value $S(\bar{x}, \bar{p})$ becomes minimal.

This is described by the bi-level program:

$$\boxed{P_{BL}: \quad \min_{p,x} S(x,p) \quad s.t. \quad p^- \leq p \leq p^+ \\ \text{and } x \text{ is W-E w.r.t. cost function } c_a(x,p) = c_a(x) + p}$$

The function $S(x, p)$ corresponds to the objective function in section 3.2.3. In the formulation of this section the minimisation of total travel time gets the form:

$$S(x, p) = S_1(x) := \sum_a c_a(x)x \tag{3.21}$$

Note that this function does not depend on the toll vector p .

In general, BL problems are not easy to solve (see Bard, 1998). However, as we shall see in the next section, if the object function $S(x,p)=S(x)$ does not depend on p (as in (3.21)), then the problem P_{BL} can often be solved easier.

3.5.3 Cases where the program P_{BL} can be solved directly

In this section we consider the BL program P_{BL} under the assumption that the objective function $S(x,p)=S(x)$ does not depend on p . We will show that in this case the problem P_{BL} can often be solved simply by solving the reduced minimisation problem of the system planner:

$$\boxed{P_0: \quad \min_x S(x) \quad s.t. \quad x \in X_d}$$

For the special instance of the function $S(x)=\sum_a c_a(x)x_a$ the approach of this subsection is related to the well-known principle of congestion toll pricing as studied in Bergenhoff *et al.* (1997). The idea is as follows.

Problem 3.1 Find a toll vector $p_0, p^- \le p_0 \le p^+$ and a solution x_0 of P_0 such that x_0 is a W-E w.r.t. the costs $c(x_0,p_0)=c(x_0)+p_0$.

To solve Problem 3.1 we simply can compare the KKT optimality conditions for a solution x_0 of P_0 with the conditions for a W equilibrium. For this case the necessary KKT conditions for a solution x_0 of P_0 read:

$$\begin{aligned} &\Lambda u = d \qquad \nabla s(x_0)=\lambda \qquad u^T\mu = 0 \\ &\Delta u - x_0 = 0 \qquad 0 = \Lambda^T\sigma - \Delta^T\lambda + \mu \end{aligned} \tag{3.22}$$

Now, by defining a toll price through

$$c(x_0)+p_0 = \nabla S(x_0) \quad \text{or} \quad p_0 = \nabla S(x_0) - c(x_0) \tag{3.23}$$

according to Lemma 3.1, x_0 is also a W-E w.r.t. the costs $c(x_0)+p_0$. If moreover p_0 satisfies the constraints $p^- \le p_0 \le p^+$ of P_{BL} then (x_0,p_0) is also feasible for P_{BL} . Since the feasible set of P_0 is larger than the (projection onto the x -space of the) feasible set of P_{BL} the minimiser x_0 of P_0 must also yield a minimiser (x_0,p_0) of P_{BL} . We summarise our discussion in corollary 3.1.

Corollary 3.1 *Let x_0 be a solution of P_0 and suppose the toll vector p_0 defined by (3.23) satisfies $p^- \le p_0 \le p^+$. Then (x_0,p_0) is a solution of the bi-level program P_{BL} .*

A toll vector as defined in Problem 3.1 is not uniquely determined in general. This is discussed in Remark 3.2.

Remark 3.2 Clearly the toll vector in Problem 3.1 is not uniquely determined. If p_0 with corresponding solution x_0 of P_0 solves Problem 3.1 then all vectors p which satisfy *e.g.*, the KKT conditions in (3.19) (with $c(\bar{x})$ replaced by $c(x_0)+p$) are also solutions of Problem 3.1. This set of vectors p in general defines a whole polyhedron. A part of this polyhedron is trivially obtained by the following reasoning. Let p_0 be a toll vector as in Corollary 3.1. This means that (x_0, p_0) solves (3.19) with $c(x_0)+p_0=\lambda$. But then for any $\alpha>0$ (3.19) is also solved by $\alpha(c(x_0)+p_0)=\alpha\lambda$ and $0=\Lambda^T\alpha\sigma-\Delta^T\alpha\lambda+\alpha\mu$. This means that with p_0 also any vector $\hat{p}_0$ which satisfies $\alpha(c(x_0)+p_0)=c(x_0)+\hat{p}_0$ is a valid toll vector as well. Combining this with

$$\alpha(c(x_0)+p_0)=c(x_0)+p_0+(\alpha-1)(c(x_0)+p_0) \quad \forall\alpha>0$$

gives

$$\hat{p}_0=p_0+(\alpha-1)(c(x_0)+p_0)=\alpha\nabla S(x_0)-c(x_0) \quad \forall\alpha>0$$

So we also can choose any such vector satisfying $p^- \leq \hat{p}_0 \leq p^+$ as toll.

3.6 Summary

In this chapter optimal toll level design problem is formulated as a bi-level mathematical program. The upper level consists of the design variables (toll level), an objective function, and constraints to the design variables. The objective function in this research is the minimisation of the average travel in the network. For a given set of design variables, the lower level determines the depending variables and thus the value of the objective function. The lower level is treated as a black box in this chapter. Chapter 4 on the traffic model gives more insight in the ideas behind this model and the behaviour of the model. Some mathematical background information is given in the last section of this chapter, by providing some theory on a static bi-level toll model.

4 Traffic model

In order to make the mathematical optimisation problem operational, an implementation of the lower level is needed. In this chapter, the lower level modelling framework that assesses the network effects of a toll setting is presented and is called a traffic model. The framework is adopted from Van Amelsfort (forthcoming). The different parts of this model are presented in this chapter.

An important issue in identifying the key components of the modelling framework is to determine the behavioural responses towards different road pricing measures which are necessary to take into account in the modelling framework. For each of the behavioural responses used in the modelling framework, the issues are to determine what type of choice model to use and the model's attributes and parameter settings. This results in a dynamic stochastic user equilibrium (DSUE).

As we will show this chapter we find that in order to model time-varying road pricing measures correctly, it is of great importance to correctly model the space-time distribution of traffic in detail to produce correct estimates of travel times, changes in travel times and behavioural responses as a consequence. The requirements of the model are treated in section 4.1. The resulting framework therefore uses an analytical multi-user dynamic traffic assignment model for route choice and dynamic network loading (section 4.2.1), together with departure time choice (section 4.2.2). The total demand for car traffic is assumed to be fixed (section 4.2.3). So no mode choice and no elastic demand are taken into account in this study. However, both behavioural responses and network conditions are modelled in such detail that the effects of time-varying road pricing measures can be assessed.

4.1 Requirements for the model

The purpose of developing a modelling framework is to assess the long-term traffic network effects of time-varying, space-differentiated road pricing measures. In this section, we will discuss the requirements for the modelling framework and determine the components that should be included in the framework. The requirements for the modelling framework are determined by two factors: the road pricing measure and its level of differentiation on the one hand and the expected behavioural responses towards the road pricing measure and changing travel conditions on the other hand.

4.1.1 Requirements from road pricing measures

With time-varying road pricing measures, the charge levels change over time. In this case this means that the price levels change within a peak, according to posted charge schedules in time periods of a half an hour. Therefore, the price level does not depend on the actual conditions on the road.

The time differentiation in charge levels will give travellers an incentive to change their departure times towards cheaper time periods. If and how much travellers will actually change their departures time will, among many other factors, depend on the expected travel times of perceived alternative departure time periods. This level of time differentiation therefore poses two requirements on the modelling framework: it needs to be able to handle time-varying road pricing measures (requirement 1) and the travel times for different departure periods need to be calculated accurately (requirement 2).

4.1.2 Expected behavioural responses

Apart from the design of the road pricing measure, the expected behavioural responses pose requirements on the modelling framework as well. Four requirements from the expected behavioural responses are mentioned:

- In this framework, road pricing stimuli invoke behavioural responses of travellers which change the performance of the (transportation) system. This in return causes behavioural responses. In the long run this process is assumed to lead to a new stable equilibrium situation. In this case DSUE is chosen, where users try to minimise their individual perceived generalized costs. The use of perceived costs achieves a more realistic user equilibrium, because not every individual from a heterogeneous population experiences the same disutility for the same route (e.g. comfort, speed, nice views, etc.). Reaching stable equilibrium conditions is therefore requirement 3 of the modelling framework.
- In this research the OD matrix is assumed as fixed, so no changes occur in activities (trip generation), origins and destinations (trip distribution), or modes (model split). Changes in departure time and route choice are taken into account, so the model should be able to handle changes in temporal distributions of demand (requirement 4) and changes in spatial distribution of demand (requirement 5). Note that even with fixed charge levels during the day, people may change departure times as a result of changes in traffic conditions. The necessity of considering departure time choice is therefore not limited to a case with time-varying road pricing measures.
- Due to heterogeneity of the population, different road users react differently to the tolls: some people are willing to pay a lot for their trip in order to arrive on time, while others do not bother about time and want to minimise their monetary costs. In reality this is a continuous variation through the population. This can be modelled by the definition of user classes. In this research, two different user classes exist: one with a high value of time and one with a low value of time. So each user class needs its own choice model and parameter settings for each choice process in the modelling framework (requirement 6). However, as stated in section 3.2.2, every user class pays the same toll.
- The modelling framework focuses only on changes in the performance of the car network. So it is required to be able to measure the changes in car traffic network performance (requirement 7).

4.2 Traffic modelling framework

In the previous section, a list of requirements to the traffic model is made. This list is summarized in Table 4.1. In this section, the resulting traffic model is presented. This model is used to compute the network effects and the effect on the objective function value of a toll setting.

Requirement	
1	Time varying road pricing measures
2	Accurate time-varying travel times
3	User class specific model
4	Equilibrium should be reached
5	Changes in temporal distribution of traffic
6	Changes in spatial distribution of traffic
7	Car traffic system performance can be measured

Table 4.1: The requirements for the traffic model

Requirement 1, 2, 3, 4, 6 and 7 are met by the DTA model INDY (section 4.2.1). The only and important requirement that cannot be satisfied by INDY is number 5: the

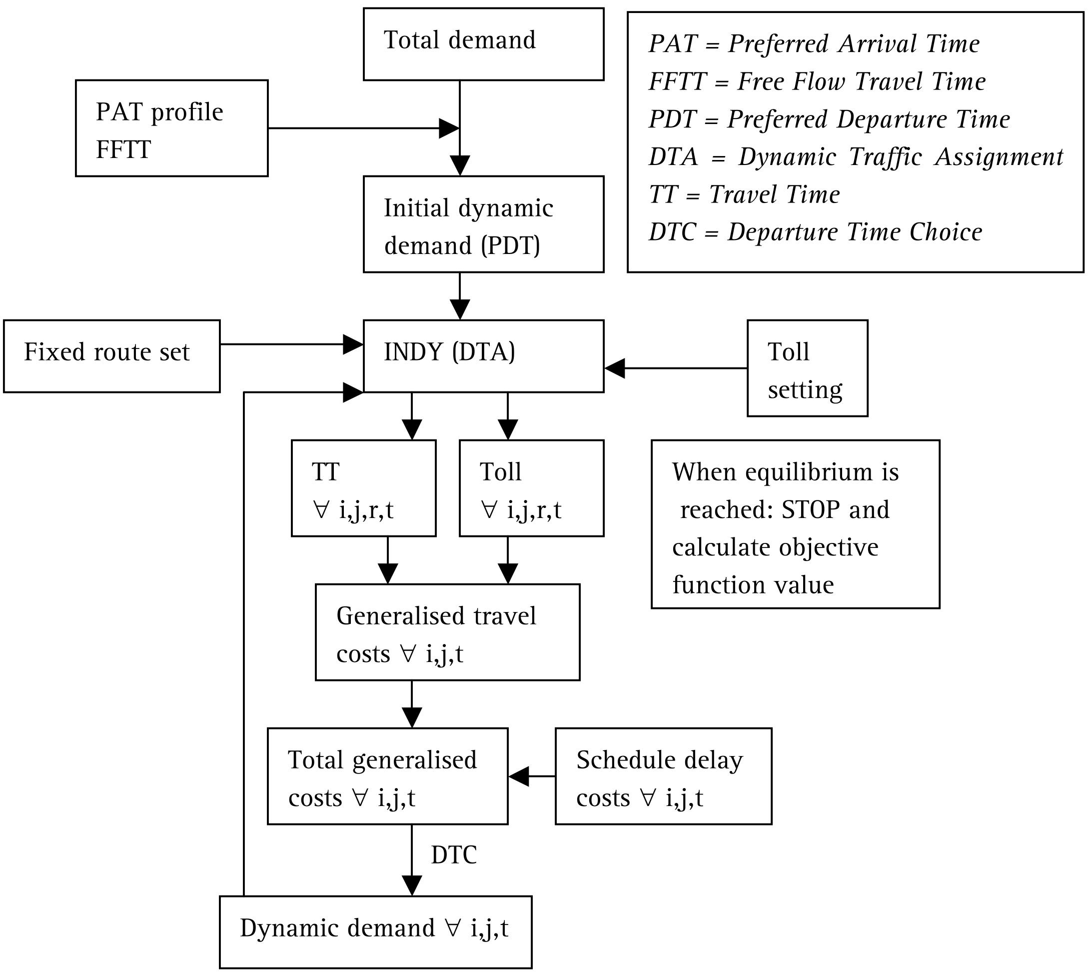


Figure 4.1: schematic overview of the lower level model

ability to change the temporal distribution of the traffic. The departure time choice is added a separate module and is applied iteratively in order to reach equilibrium. The departure time model is treated in more detail in section 4.2.2. In Figure 4.1 the model is shown. It can be seen that INDY uses the dynamic demand data to calculate dynamic travel times and dynamic tolls. These two route based figures are combined in OD based generalised travel costs. Together with schedule delay penalties they form the total generalised costs for departing at a particular time period t. Based on these costs the departure time choice model determines new dynamic demand matrices.

4.2.1 Dynamic traffic assignment model

In this research the DTA model INDY was used, (see Bliemer, 2001; Bliemer et al., 2004; Bliemer, 2004). This DTA model is an analytical macroscopic multi-user class dynamic equilibrium for large scale networks and consists of three model components: a route generation model, a route choice model, and a dynamic network loading model (DNL). Furthermore, it has the possibility to define different user classes, categorized in two groups: different driver classes and different vehicle classes. In this research only two user classes are defined:

- Cars with a maximum speed of 120 km/h and a low value of travel time (€10 per hour)
- Freight with a maximum speed of 85 km/h and a high value of travel time (€32 per hour)

Route generation model

For a given departure time, car drivers are assumed to consider that different route alternatives have certain attributes and choose their subjective optimal route. The route alternatives available to the car drivers are determined by a pre-trip route generation procedure, in which the most likely route alternatives are generated for each origin-destination (OD) pair, in the situation without tolls, see also Bliemer and Taale (2006). Each of these routes is assumed to have a generalised travel cost, which is used in the route choice model, including the tolls. So some routes may become more attractive after tolls are introduced, which may not be in the route set. However, this is considered to be a minor problem, because the toll in this research is distance based, so longer routes will be less attractive, as is also the case when only travel time is considered as a cost. Slower, shorter routes may become more attractive after the introduction of tolls.

Route choice model

The route choice module models the behaviour of the travellers by choosing the best route for themselves from the set of available routes as determined in the route generation model. The route choice model is a logit model based on a route cost function. For each available route between an origin and destination, the route costs are determined. The route cost function in the DTA model can be specified by the modeller, and in this modelling framework the route costs include toll cost and travel time, where a value of time (per user class) is used to determine the generalised costs.

Mathematically formulated, the following generalised route travel cost function is used:

$$U_{rijwm} = \alpha_m \tau_{rijwm} + p_{rijwm} + \varepsilon_{rijwm} \tag{4.1}$$

where U_{rijwm} is the disutility for user class m taking route $r \in R_{ij}$ when departing at time w, τ_{rijwm} is the route travel time for route r departing at time w, and p_{rijwm} is an additional (monetary) cost on that route. The term ε_{rijwm} is a random unobserved cost component, which represents all other cost components. The parameter α_m is the value of time (€/h) for user class m. This is a behavioural parameter to be estimated. In traffic engineering it is common to assume that ε_{rijwm}'s are independently extreme value type I (Gumbel) distributed over all routes. This is useful, because in that case the percentage of car drivers choosing route r during departure time window w can be computed by the following formula (according to the multinomial logit model, see McFadden, 1976):

$$\psi_{rijwm} = \Pr(U_{rijwm} \leq U_{r'ijwm} \ \forall\, r' \in R_{ij}) = \frac{e^{(-\mu_1 U_{rijwm})}}{\sum_{r'} e^{(-\mu_1 U_{r'ijwm})}} \tag{4.2}$$

where μ_1 is a scaling parameter (which is inversely related to the variance of the random unobserved component). Multiplying this fraction by the total number of car drivers from i to j during time window w, d_{ijwm}, yields the route flows u_{rijwm}. See equation (4.3).

$$u_{rijwm} = \psi_{rijwm} d_{ijwm} \ \forall\, r \in R_{ij} \tag{4.3}$$

This dynamic travel demand depends on the departure time choices of the travellers. This relation however assumes independence between different routes, which is not true when routes overlap each other. When the common parts of the routes are not big, this is assumed to be a minor problem.

Dynamic network loading model

The dynamic network loading model is completely incorporated in the INDY software and is not treated in detail here. For a more theoretical formulation of the dynamic traffic assignment problem, see the formulations of Bliemer (2001). In this PhD thesis the problem of dynamic traffic assignment is formulated analytically for different user classes. Just like in INDY, the traffic demand is considered to be fixed, so for every OD pair and for every time window a demand is given. To assign this demand to the network, in this source the following constraints are formulated: flow conservation, flow propagation, non negativity constraints, definition constraints, and First In First Out constraints. These constraints also hold for the model in this research and are incorporated in the INDY software.

4.2.2 Departure time choice

For a correct modelling of departure time choice different attributes need to be implemented in the model framework. Besides travel time and travel costs, which are basic components of transportation models, the utility functions also contain scheduling delay components. There are two important issues with these scheduling delay components: 1) they are based on preferred arrival and departure times which are data that are not vastly available and 2) they may contain departure as well as arrival scheduling delay components. Especially arrival time scheduling delays are harder to model, since the arrival times are only known after the trip has been completed. Both issues are discussed in more detail in the next sections.

Determining preferred arrival times

Determining preferred arrival time profiles is not a trivial issue. Travel diaries and traffic count data are mostly setup to measure the current behaviour, not the desired behaviour. Although it is not part of this research, two modelling approaches seem feasible. The first approach uses departure time choice modelling in the base year estimation of the model. In this case, the preferred arrival times can be calibrated such that the resulting base year model validates well towards independent traffic data (volumes, travel times, etc.). This is a difficult process, but feasible, and has been applied in Van Amelsfort (forthcoming). In this research, this preferred arrival time profile is used (see section 6.1). The second approach uses a reverse engineering approach, as proposed by Teekamp *et al.*, 2002, which involved an equilibrium base year model situation and a departure time choice model, the unknown preferred arrival times are calculated back.

Departure time choice model

The departure time choice or scheduling of trips is a choice process discussed in detail in chapter 6 of Van Amelsfort (forthcoming) and the resulting models are used here to determine the resulting network effects of the tolls. The departure time module incorporated in the modelling framework is a logit model where the parameter values can be set by the modeller.

The total disutility of travelling from i to j when departing during time window w while preferring to arrive during time window t, i.e. U_{ijwtm}, is given by:

$$\begin{aligned} U_{ijwtm} = & -\alpha_m \bar{\tau}_{ijwm} - \beta_m DSDE_{ijwtm} - \gamma_m DSDL_{ijwtm} - \chi_m ASDE_{ijwtm} \\ & - \iota_m ASDL_{ijwtm} - \bar{p}_{ijwm} - \bar{\kappa}_{ijwm} - \varepsilon_{ijwtm} \end{aligned} \tag{4.4}$$

where $\bar{\tau}_{ijwm}$, $\bar{p}_{ijwm}$, and $\bar{\kappa}_{ijwm}$ are the average (flow-weighted) travel time, toll costs and additional costs for user class m travelling from i to j and during time window w (see equation (4.5)).

$$\bar{\tau}_{ijwm} = \sum_r \psi_{rijwm} \tau_{rijwm},\ \bar{p}_{ijwm} = \sum_r \psi_{rijwm} p_{rijwm},\ \bar{\kappa}_{ijwm} = \sum_r \psi_{rijwm} \kappa_{rijwm} \tag{4.5}$$

$DSDE_{ijwtm}$ is the time that the car driver departs in user group m earlier than their preferred departure time, when departing at time w, while preferring to depart at time t when travelling from i to j. $DSDL_{ijwtm}$ is the time that the car driver departs later than preferred, $ASDE_{ijwtm}$ is the time that the car driver arrives earlier than their preferred arrival time and $ASDL_{ijwtm}$ is the time that the traveller arrives later than their preferred arrival time. The term ε_{ijwtm} is a random unobserved component that represents all other cost components. The parameters $\alpha_m, \beta_m, \gamma_m, \chi_m, \iota_m$ are behavioural parameters for different user classes *m*.

The preferred departure time and preferred arrival time of travellers are related by subtracting the free-flow travel time from the preferred arrival time. So, even if the arrival time profiles for all zones are equal, the departure time profiles are different for each zone and destination, because longer trips require earlier departing than shorter trips. Again assuming that the random components ε_{ijwtm} are independently extreme value type I (Gumbel) distributed over all departure times windows, the fraction of car drivers (with preferred departure time *t*) choosing departure time *w* is given by

$$\varphi_{ijwtm} = \Pr(U_{ijwtm} \leq U_{ijw'tm}, \forall\ w') = \frac{e^{-\mu_2 U_{ijwtm}}}{\sum_{w'} e^{-\mu_2 U_{ijw'tm}}} \tag{4.6}$$

where μ_2 is a scaling parameter, which is inversely related to the variance of the random unobserved component. Multiplying this percentage by the fraction of car drivers preferring to depart at time *t*, i.e. $PAT_{ijtm}\varphi_{ijwtm}$, yields the fraction of the total demand who prefer to depart at time *t* and will actually depart at time *w*. Summing over all preferred arrival times, the total fraction of car drivers departing at time *k*, is given by equation (4.7).

$$\varphi_{ijwm} = \sum_t \varphi_{ijwtm} PAT_{ijtm} \tag{4.7}$$

The total demand of user class *m* in time window *w* at OD pair *ij* is now computed in equation (4.8) by multiplying the total demand on this OD pair by the fraction.

$$d_{ijwm} = \varphi_{ijwm} d_{ijm} \tag{4.8}$$

4.2.3 Elastic demand

The framework of Van Amelsfort (forthcoming) has a module to model elastic demand. The idea behind this model is that the demand for traffic on an OD pair is dependent on the generalised costs on this OD pair (A general mathematical formulation is stated in equation (4.9)). In the case of levying tolls, it is assumed that the generalised costs would normally increase, so the demand for traffic would decrease. This phenomenon is the inverse effect of the latent demand effect, which is renowned in traffic engineering. This theory states that new infrastructure between an OD pair generates new traffic, because the generalised costs on this OD pair have gone down. This partly nullifies the positive effects of the new infrastructure on the state of the traffic system. However, for simplicity reasons these effects are not considered in this research: no elastic demand is modeled. The following arguments can be used to strengthen this choice:

- For the short term: it can be argued that a modeling approach without elastic demand models the short term effects of tolls: route choice and departure time choice are easier to adapt than destination choice and trip frequency.
- The average car user will not pay more after the introduction of road pricing, because fixed taxes on cars disappear. Car users may assign the same budget to their cars, resulting in the same demand for traffic.
- For certain user classes, road pricing may make traveling more interesting again because of lower travel times, so it can lead to extra (latent) demand.
- Finally can be argued that the effects of spatial and temporal redistribution are approximately independent of the exact level of the total demand, so it is not necessary to model this total demand very accurately in order to achieve optimal toll settings.

When desired, the elastic demand model could however relatively easily be added to the modeling framework.

$$\phi_{ijm} = f(U_{ijwm}) \tag{4.9}$$

4.2.4 Convergence of the traffic model

The traffic equilibrium is computed by an iterative process (see Figure 4.1). Experiments with the traffic model showed that convergence was slow: one model evaluation lasted about 5 hours. An adaptation to the algorithm is made to make this convergence faster. In the new situation one model evaluation lasts about 48 minutes. In appendix 9.2 a description of the adaptation to the model is given.

4.3 Summary

In this chapter the traffic model has been described. In order to handle dynamic tolls and the behavioural responses to those tolls, a dynamic model is used. The model consists of a departure time choice model, a route choice model, and a dynamic traffic assignment model. The used choice models are simple logit models. Elastic demand is not taken into account for simplicity reasons. Other arguments were used to strengthen this assumption.

5 Solution method

In the previous chapters a model has been presented to describe the problem of optimal toll design. The goal of this chapter is to present a technique to solve this model from a practical point of view. In other words, in this chapter computational aspects are highlighted and the consequences of restrictions in computational power are dealt with. The solution method resulting from this chapter is applied to a test case in the next chapter.

First, a short review of available approximation methods is given. The pattern search method is considered to be useful for this optimisation problem and is highlighted further. Because the pattern search algorithm works with a discrete solution space, some issues about a discrete solution space are added. The chapter ends with a range of possibilities and choices to be made when a pattern search algorithm is implemented in practise.

5.1 Approximation of the optimisation problem

As was stated in section 3.1, the lower level of the mathematical program is considered as a black box: the reaction of the traffic system to the tolls is computed completely numerically. So no analytical optimisation techniques can be used to solve this model. Roughly, two different approaches are possible to handle such a numerical problem. The first approach uses a continuous solution space and when a solution is found, it is made discrete again. This has the advantage that analytical gradient based search techniques can be approximated by a numerical gradient. Also the line search sub problem can be approximated numerically (see Fletcher, 1987). The second approach discretises the solution space and approximates the problem by trying different solutions in an intelligent way, to be as efficient as possible. In this section, a few of them are described shortly. These techniques were considered to be used during this research. The most promising technique turns out to be pattern search. This technique is described in more detail in the next section.

Every search technique in this section starts with an initial solution. From the start solution, the algorithm searches for better solutions and when they are found, the algorithm jumps to this solution. For all treated algorithms it holds that a different initial solution can make the algorithm to end up in a different local minimum. This means that it is not possible to judge the different search algorithms on the quality of the solution in advance. Ideally, each algorithm should be tested with different starting points in order to say something about this. Even when different starting points are used, it is not possible to judge whether the best found solution is also the global minimum, because the solution space has not been search exhaustively.

5.1.1 Criteria for selection of a search algorithm

The search algorithms are judged on the following criteria, in order to select an algorithm to be used.

- The algorithm has to have a reasonable computation time, because the aim of this study is to achieve a solving method that can be applied in practice.
- The algorithm should be able to handle tight constraints to the tolls.
- The extent to which the algorithm is appropriate to handle the fixed step sizes of the tolls (a discrete solution space).
- The ability to overcome a local minimum.

5.1.2 The gradient direction: steepest descent

Here the gradient is computed numerically. In this direction, a line search algorithm is executed. A line search starts with an initial solution, in this case the current solution. In one direction, an optimisation is carried out. In formula: $\min_{\alpha} s + \alpha v$, with s the initial solution, α a scalar, and v the search direction. When the minimum is reached along this line, this is taken as the next solution. For more information, see for example Fletcher (1987).

Computation time

This method searches in an efficient direction, so only a few iterations are necessary to reach a local minimum. However, numerical approximation of the gradient is computationally very expensive and has to be executed every iteration. In every iteration a line search algorithm has to be executed, which is computationally expensive, so this method will take a lot of computation time.

Constraints

The tolls are subject to constraints, so it may not be allowed to change in the direction of the gradient, which may make the method less effective.

Solution space

The solution space is continuous, so no solutions are excluded in advance. However, the solution needs to be rounded off when the tolls are implemented in reality. The solution is likely to loose some of its quality after this rounding off.

Local minimum

This algorithm is not able to escape from a local minimum.

5.1.3 Powell's method

Here a line search is executed in every direction of a set of vectors, which has the same size as the number of variables. First, all unity vectors are in the set. After one iteration, the first vector in the set is replaced by the average of all improvements in the previous iteration. This is given by the vector $X_n - X_0$. More information for example in Fletcher (1987).

Computation time
The addition of the average vector to the direction set achieves more efficient searching: it prevents the function from zigzagging. It is not necessary to compute the gradient of the function. However, in every iteration a number of line searches needs to be executed, equal to the number of variables. This makes that this method takes a lot of computation time.

Constraints
The tolls are subject to constraints, so it may not be allowed to change in the desired direction, which may make the method less effective.

Solution space
The solution space is continuous, so no solutions are excluded in advance. However, the solution needs to be rounded off when the tolls are implemented in reality. The solution is likely to loose some of its quality after this rounding off.

Local minimum
This algorithm is not able to escape from a local minimum.

5.1.4 Pattern search

Starting with the initial solution, the pattern search algorithm considers the neighbours of the current solution. A neighbour can be defined in different ways, but in this case it is defined as a change of one variable by one discrete step. This corresponds to a step from the initial solution in the direction of the unity vector with the value of one step size. The function is evaluated in this new point. Depending on the exact form of the algorithm and on whether this gives improvement or not, the new point is saved as the new current solution. Now the next variable is selected and the procedure is repeated. When a complete loop through all variables does not provide improvement, the algorithm is terminated: a local minimum is reached. In this sense a local minimum is defined as a solution, which is lower than every direct neighbour of that solution. More information can also be found in Michalewicz (2000), where the procedure is found within the category of local search.

Computation time
No derivatives or line searches are used, which makes this method computationally feasible. Convergence is not fast, because every variable is changed one by one.

Constraints
Can easily handle a highly constrained solution space: this corresponds to the lower and upper bounds of the tolls.

Solution space
Uses a discrete solution space: this corresponds to the definition of a limited number of price categories. Because the solution space is made discrete, a large part of it is

excluded in advance. From a computational point of view this is an advantage. The accuracy and the convergence speed of the search algorithm can be influenced by the step size, which can be freely chosen. It is also possible to start with a rough step size and end up with a small step size, to speed up the algorithm.

Local minimum
This algorithm is not able to escape from a local minimum.

5.1.5 Other search algorithms

In Table 5.1 a short description of some more search algorithms is given, because the algorithms in the earlier sections all do not have the possibility to escape from a local minimum. From Michalewicz (2000) some algorithms are listed that have the ability to escape from a local minimum. This source can be used to find more information on these search methods. All these algorithms are not implemented, because they use more computation time than pattern search.

Grid search	In a grid search every variable is given a limited number of (usually discrete) values. For every point on this grid the value of the objective function value is computed. Because every combination is computed, the computation time grows exponentially with the number of variables. In our case this is not feasible.
Genetic algorithm	A genetic algorithm starts with a random pair of solutions and combines these solutions to a number of ‘children’. For all children the value of the objective function is computed. The children with higher objective function values have a higher probability to produce descendants in the next generation than other children. After a fixed number of generations the best solution is chosen as the best solution. This search technique has a random component, which is a pity when some information on the shape of the objective function can be known. Furthermore, a genetic algorithm needs a lot of function evaluations.
Simulated annealing (within the category of local search)	Looks like pattern search, but it contains a random component. It starts with an initial solution. Then it computes the value of a neighbour of the current solution. The difference between the value of the neighbour and the value of the current solution determines the chance for the neighbour to be selected as the new current solution. So both when the new solution is better and when it is worse there is a chance of being selected, but in the first case the chance is bigger. When the algorithm develops, the chance of being selected when the solution worsens, becomes smaller. This is called cooling. The presence of a random component makes it necessary to test the algorithm multiple times. Simulated annealing may give better results than pattern search, but is requires more computation time.

Tabu search (within the category of local search)	Looks like pattern search, but when the algorithm is in a local minimum, it is also possible to jump to neighbours that are not better than the present solution. In that case, the algorithm picks the best neighbour. All former iterations are stored in the tabu list. The algorithm stops when all neighbours are in the tabu list, when some predefined time has collapsed, or when the solution has not improved for a long time. Tabu search may give better results than pattern search, but requires more computation time.

Table 5.1: Other search algorithms and their properties

5.1.6 Conclusion

From this short exploration can be concluded that pattern search is the most suitable search algorithm for the problem in this research. The main arguments for this are:

- Gradient based methods are not suitable, because the computation of a numerical gradient is very expensive.
- A method that uses the line search sub problem is not suitable, because this is computationally expensive and because the problem is constrained rather tight, so the line search usually cannot be conducted exactly in the descent direction.
- A genetic algorithm, simulated annealing, and tabu search explore more solutions than pattern search, so the risk of getting stuck in local minima is smaller and the quality of the solution may be better. The drawback of making more evaluations is the need for more computation time. In order to stay within reasonable computation times, these search methods are not used in this research.

5.2 Pattern search

Within the technique of pattern search, there are still a lot of different possibilities for implementation. This section pays attention to these possibilities. First, some general properties are given about when a continuous solution space is converted to a discrete solution space, because in chapter 3 the tolls are incorporated in discrete toll categories. After that, some issues about the use of a local search algorithm are given, in order to illustrate the effect of the use of different variants of pattern search. The section ends with a discussion of all possible properties that can be used to adapt the pattern search algorithm and that determine its exact shape.

5.2.1 Properties of a discrete minimum

The difference between a discrete and a continuous local minimum is illustrated in this section. A discrete local minimum is defined as a solution which is better than all of its direct neighbours. A direct neighbour is defined as the change of one variable of the solution by one step. The fact that only one variable is changed makes that only in these directions a search for improvement will be made, which may cause an error. A continuous model would search a point in all possible directions for improvement. This difference is illustrated in example 1. Another error will occur because of the step size. A discrete solution space only covers a limited number of points, so it can jump over a continuous local minimum. This can work out both positive and negative. This is illustrated in example 2.

Example 1: search direction
The following function of two variables is used as a counterexample: *f(x,y)* = $x^4 + y^4 - 3(xy)^2$. The objective is to minimise this function. In Table 5.2 the values of *f(x,y)* are given for *x* and *y* varying from –2 to 2 with step size = 1. The table shows that the neighbours of (0,0) are all greater than the value of (0,0) itself, so it is considered to be a discrete local minimum. This point is however not a continuous local minimum, but a saddle point: in the diagonal directions the function is decreasing in the ε environment of (0,0). So a discrete algorithm would terminate, while a continuous algorithm would not terminate.

	-2	*-1*	*0*	*1*	*2*
-2	-16	5	16	5	-16
-1	5	-1	1	-1	5
0	16	1	0	1	16
1	5	-1	1	-1	5
2	-16	5	16	5	-16

Table 5.2: function values of : $f(x,y) = x^4 + y^4 - 3(xy)^2$

Example 2: step size
In Figure 5.1 the graph of the function *f(x)* = $x^4 - 3(x\text{-}0{,}35)^2$ is given. The objective is to minimise this function. The function clearly has two local minima, one of which is the global minimum. When the minimum of this function is searched with a discrete algorithm with step size = 1, it will terminate at x = -1, independent of its starting point. A continuous search algorithm will terminate in the right local minimum, or in the left local minimum (which is also the global minimum), depending of its starting point. In the first case, the continuous algorithm performs worse than the discrete algorithm in terms of the objective function value. In the second case, it performs better. This example shows that a discrete approach can contribute to overcome local minima in order to achieve a better estimate of the global minimum. On the other hand, a continuous approach can estimate local minima more accurately, because it does not have a discrete step size.

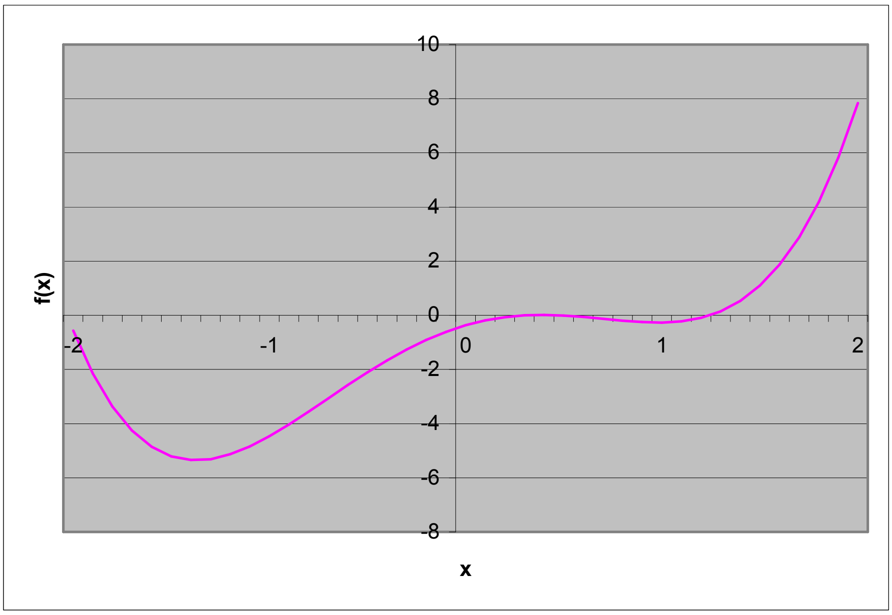


Figure 5.1: The function f(x) = x4 – 3(x-0,35)²

5.2.2 Properties of local search and local minima

Some small examples are used to explain certain issues of the approximation of an optimisation problem with a discrete local search algorithm. In these examples two variables exist, which can both take only two values. So the solution space consists of 4 possible solutions, visualised in the tables below, with one variable in the horizontal direction and one variable in the vertical direction. The values in the table represent the hypothetical objective function values.

25	26
27	24

When the algorithm is at 25, it will only compare itself with its direct neighbours, so it will terminate. The global minimum of 24 will not be found.

27	26
24	25

When the algorithm is at 27, the search direction will determine the speed of the algorithm. When first the horizontal direction is explored, the algorithm will respectively jump to 26, 25, and will terminate in 24, so it needs 4 calculations of the objective function. When first the vertical direction is explored, the algorithm will jump to 24 straightaway. It still checks its direct neighbour 25, but because this presents no improvement, it will terminate after 3 calculations of the objective function.

27	25
24	26

Again the algorithm starts at 27. In this situation the search direction will even determine the solution of the algorithm. When first the horizontal direction is explored, the algorithm terminates at 25, whereas if the vertical direction is explored first, it will terminate at 24.

One possibility to increase the chance to end up in the global minimum of 24 is to first search all neighbours and then jump to the smallest value. In this case there are only two neighbours, but in bigger cases this number grows fast. So this possibility will cost a lot of computation time. Another possibility is to execute the search algorithm with different search orders and test what order is the best.

5.2.3 Different variants of pattern search

In section 5.1.4 a short description of pattern search was already given. Here different possibilities are listed, which determine the details of the search algorithm. A range of these different nuances of the pattern search algorithm have been investigated and tested numerically in chapter 6. In section 6.4 an overview of the different tests of pattern search is presented. In section 6.5 the results of these tests are given.

For the 2 dimensional case, all possible search directions are illustrated in Figure 5.2. Here the step size is 1. It can be seen that for both dimensions it is possible to search in two directions. These 4 possible search directions could be restricted by constraints. For example, when non-negativity constraints would be valid, only v1 and v2 would be available for searching.

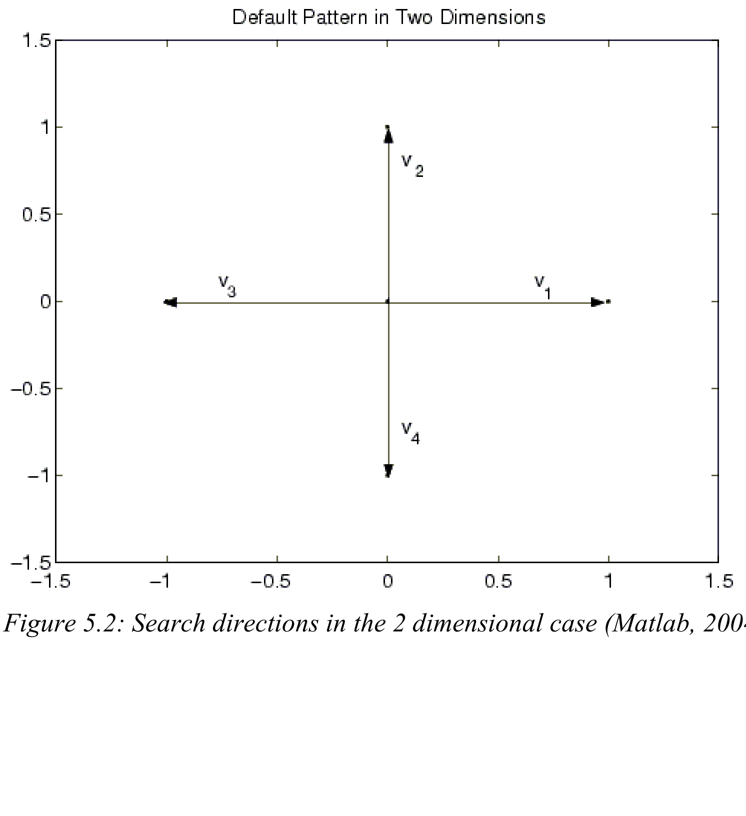


Figure 5.2: Search directions in the 2 dimensional case (Matlab, 2004)

In Matlab (2004) an overview is given of different aspects and options when a pattern search algorithm is executed. Although the pattern search module of this software package has not been used for this research, the aspects and options are valid for all pattern search algorithms. This results in a list of 7 parameters, which together determine the exact shape of the pattern search algorithm.

Complete poll on / off

When the neighbours of the current solution are explored and an improvement is found, there are two possibilities: jump to this new solution straightaway, or finish searching the complete neighbourhood of the current solution and jump to the best neighbour. In the example in Figure 5.2 the first case would mean that when an improvement is found in v2, the algorithm jumps to the point (0,1) straightaway. The second case would mean that when an improvement is found in v2, the algorithm continues to calculate v3 and v4 and jumps to the biggest improvement.

Step size and mesh expansion and contraction

The step size determines the level of detail in the search and the speed of the algorithm. Usually these two are inversely related, so the step size determines the trade-off between search speed and level of detail. However, a small step size can cause the algorithm to terminate in a local minimum, whereas a bigger step size helps the algorithm to escape from this local minimum. It is also possible to have a changing step size. In that case, when a pattern search has conducted a successful iteration, to the step size is increased, because it is likely that more improvement is possible further away from the current solution. When no improvement is detected with the current step size, it is decreased in order to make the search finer. These are called the mesh expansion factor and the mesh contraction factor. When the step size has reached a predetermined accuracy, the algorithm is terminated. Mesh expansion and contraction is a possibility when a fine grid is defined or when a continuous problem is approximated discretely. In this research, this is not useful, because the solution space is defined to be discrete with only a few price categories (see section 3.3), so the step size is fixed.

Different poll methods

The order in which the neighbours of the current solution are explored is called the poll method. Two different methods can be identified. Each method is illustrated by the example in Figure 5.2.

1. First one by one increase all variables with one step, then decrease all variables one by one with one step. In Figure 5.2: (1,0), (0,1), (-1,0), (0,-1), or v1, v2, v3, v4.
2. First increase variable 1 by one step, then decrease variable 1 with one step, then increase variable 2 by one step, then decrease variable 2 with one step, etc. In Figure 5.2: (1,0), (-1,0), (0,1), (0,-1), or v1, v3, v2, v4.

The difference between these two cases is small. More poll methods are possible by defining a different neighbourhood set, for example first one by one increase all variables with one step, then decrease all variables together by one step. In Figure 5.2: (1,0), (0,1), (-1,-1), or v1, v2, v3 and v4 together. In this example, a complete loop through all variables consists of fewer iterations, compared to the first two cases, so this method can make the algorithm faster. This is not without a price: the risk of getting stuck in a local minimum is much bigger, because the algorithm loses some detail. In this way a lot of different poll methods are possible, but are not tested in this research.

Furthermore, a distinction can be made between first increasing a variable or first decreasing a variable. This matters when both directions give improvement, because then the first search direction determines the direction in which the algorithm develops. A solution to this problem is to always evaluate both an increase and a decrease in a variable and move to the direction with the biggest improvement. This is implemented in this research. This is only possible with poll method 2, so this method is chosen to be used.

Cache on: store former iterations

When conducting the algorithm, one can save former iterations or not save them. When the calculation of the objective function is fast, many function values will probably be calculated, which makes it hard to save every iteration. Because the calculations can be made quickly it is not a problem when double calculations are made. However, when a calculation of the objective function is slow, memory requirements for saving are low and a lot of computation speed can be gained with it. In this research the computation of one objective function value costs 48 minutes, so here in all algorithms all former iterations are stored. Every time the algorithm encounters a toll setting which has been calculated before, the algorithm can skip this toll setting, because its objective function value is never better than the present objective function value. In this way cycling is prevented.

The way to select the next variable

In Matlab (2004) is assumed that the order of the variables is already determined, i.e. which variable is defined as v1, which as v2, etc. However, this can be a relevant decision, so here it is added as a relevant topic on pattern search. When nothing is known in advance about the objective function, nothing can be said about the best order in which to select variables. But as was shown by the examples in section 5.2.2, the order of the variables can have influence on the results and the speed of the search algorithm. The only way to find out the best order is to test different orders and compare them. Another aspect of this topic is to use a single order or to use multiple orders when a new loop begins. These multiple orders can be predetermined, random, or use information of former iterations. In section 6.4.1 an overview is given of the experiments on this topic in this research.

5.3 Conclusion

In this chapter a range of approximation methods is presented, which can be found in the literature. The pattern search algorithm is considered to be the most suitable for this research and is used for the case study in chapter 6. Some examples have illustrated the consequences of the use of a discrete algorithm. In this chapter a range of different variants of pattern search is treated as well. A selection of these variants is tested in the case study in chapter 6. These tests give an indication which variant is the most suitable to be used in optimal toll design.

6 Case study

In this chapter we test the framework for optimal toll design from the earlier chapters in a simple network. In essence the results of the preceding three chapters are combined here in a test application. The mathematical formulation of chapter 3 is used as a base for the optimisation approach, which forms the upper level of the model. The calculation of the objective function value based on network data is based on this chapter as well. The traffic model of chapter 4 is used to determine the network effects of the tolls in terms of network flows and speeds. Finally, chapter 5 gave a range of possible search algorithms to approximate the optimisation problem. The results from this chapter reveal something about the quality and speed of these possible search algorithms, when applied to a real, small network.

The objective of this chapter is to demonstrate the effectiveness and efficiency of different search algorithms given an objective function. Given the simplification of the network and modelling framework on the lower we do not aim to draw conclusions on the effectiveness of different road pricing measures.

The outline of the chapter is as follows. First the test network is described. After that, some notes are given on the reference situation without tolls and from that an initial toll solution is determined. In order to gain some insight in the shape of the objective function near the initial toll solution, in this chapter a line search and a grid search are conducted in a part of the solution space. After this exploration of the objective function, enough information has been gained to start the approximation algorithms and present the results in section 6.5. Finally, some concluding notes on the different algorithms are given, because generalisation of the results is not easy.

6.1 Test network

The chosen test network is a greatly simplified version of the real network of the town of Delft in the western part of the Netherlands. It contains two main motorways: the A13 connecting Den Haag and Rotterdam in the eastern part of the network and the A4 in the western part of the network, which acts as a ring road around Delft. Furthermore, there are two main urban roads in the network: the Kruithuisweg, connecting the A4 and the A13 at the south border of Delft and the Provicialeweg, a north-south connection through the town, connecting the Kruithuisweg and the A4. A map of the area is shown in the left part of Figure 6.1. In the right part the schematic network which is used in this research is shown.

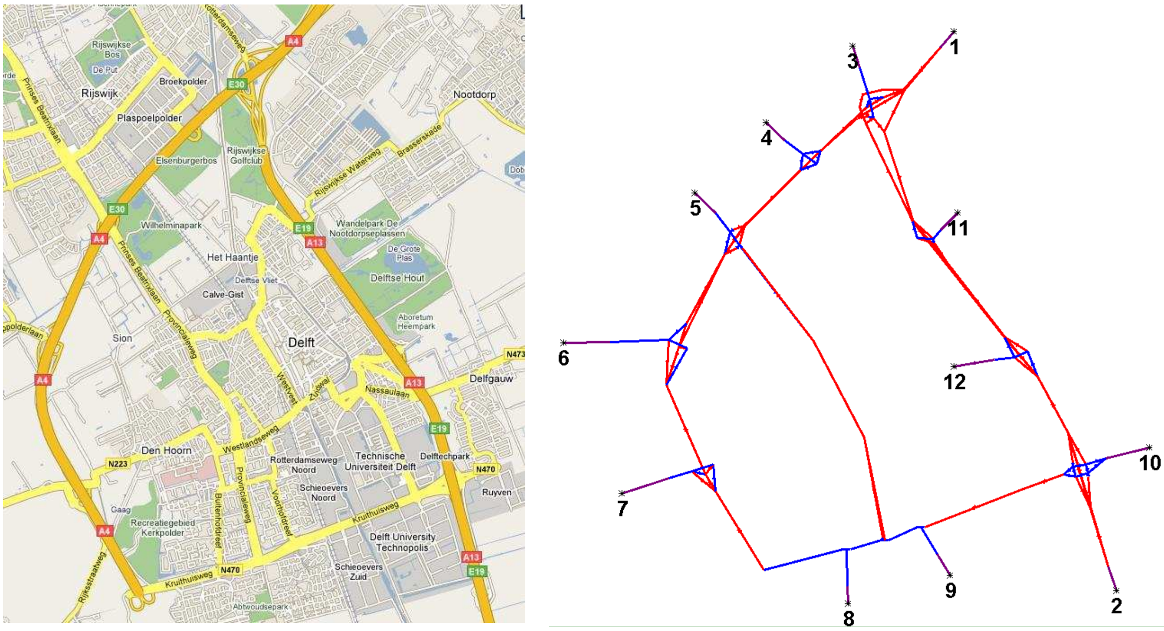


Figure 6.1: map of the test area of Delft and schematic reproduction of the test network

The network consists of 12 centriods, 137 links and 90 nodes. Because the network contains several motorways, many junctions exist in the network. Every junction consists of several links and nodes, so this reduces the actual number of nodes and links. This test network has a limited number of route choice possibilities on some OD pairs, but others have realistic route choices. For example, the main route between centroid 1 and 2 follows the A13, because the alternatives through the town and along the A4 are big detours. The OD pair 8 – 1 has much more flexibility: all three routes (A4, through town, and A13) are reasonable alternatives.

In the real network, distances are quite small: OD pair 1 – 2 is about 10 kilometre, which would be a journey of 5 minutes at free flow speeds and hardly any traveller would experience different toll values when a different time windows is entered. To achieve more interesting results for testing purposes, the test network has been artificially enlarged by a factor 3. In this way, a free flow journey is about 15 minutes. During the modelled peak period, by congestion travel times will reach the order of 30 minutes, which corresponds to the length of the chosen time windows in the toll setting.

According to the mathematical model of chapter 3, the links in the network should be assigned to a limited number of link groups. This number should not be too big and should represent some ‘natural’ categories of roads. In this test network it is chosen to create three link groups: A4, A13, and urban roads. In this way a distinction is made between urban roads and motorways. Furthermore, the A13 is a much busier road than the A4. This can be observed from the reference run (section 6.2). This is an indication that tolls may not be the same on both roads in the optimal situation, so it is desirable to make a distinction between these two motorways. So in total, 3 link groups are defined.

The modelled time period is an AM peak. In order to include some start up time and some time to empty the network, the modelled time period is from 6:00 AM to 10:00 AM. The preferred arrival time profile (PAT profile) is set between 8:00 AM and 9:10 AM, so every traveller in the network prefers to arrive at its destination between these times. The complete PAT profile is specified for each 10 minutes time window (see Table 6.1). According to the mathematical model of chapter 3, the modelled time period should be incorporates in time windows. Most traffic is on the network between 7AM and 9AM. Here is chosen to define windows of half an hour: this gives four time windows. Two more time windows of one hour are added for the periods before and after this busy period: 6AM – 7AM and 9AM – 10AM. These periods need less detail, because they are quiet during the whole period in the reference situation. So in total, 6 time windows are defined.

Time window	8:00-8:10	8:10-8:20	8:20-8:30	8:30-8:40	8:40-8:50	8:50-9:00	9:00-9:10
Percentage of travellers who prefers to arrive during this time window	5%	12,5%	20%	25%	20%	12,5%	5%

Table 6.1: PAT profile

6.2 Reference situation

In this section, the results of the lower level model without tolls are presented. This reference situation serves as a basis of the comparison of different algorithms.

In OmniTRANS it is possible to view the dynamic results of the model run. For every 5 minutes period the flow is shown by the thickness of the link and the speed ratio (actual speed divided by free speed) is shown by the colour of the link. In Figure 6.2 an example is shown for four specific times. This way of visualising makes it possible to develop a good feeling of the traffic situation which is represented by the results. Based on these images, the initial toll setting is determined.

The value of the average travel time in this situation without tolls is equal to 28.09 minutes. The aim of the algorithm is to reduce this value by approximating the optimal tolls.

6.3 Before optimisation

In this section we discuss some important issues concerning the optimisation of toll. In order to start optimising we need a start toll solution. Depending on the approxima-

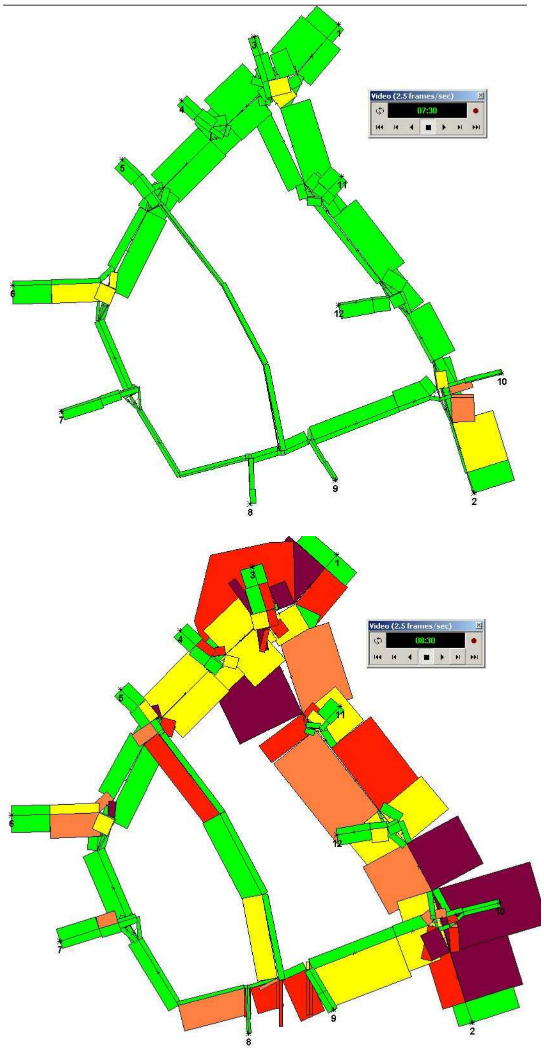


Figure 6.2: State of the network at 7.30 AM, 8.00 AM, 8.30 AM, and 9.00 AM . Thickness represents traffic intensity. Colour represents speed: a darker colour means a lower speed.

tion method this is not a trivial issue, e.g. the resulting toll solution may depend on the start solution. The second issue in this section is the shape of the objective function, which given the 'blackbox' on the traffic model is unknown. In order to better interpret the results of different methods later in the chapter, we examine the shape of the objective function in this section. We can then move forward in the next section to apply different approximation methods.

6.3.1 Determining the initial toll solution

Before a pattern search algorithm can start, it is necessary to determine an initial solution. This initial solution consists of a toll value for all time periods and all link groups, as defined in section 6.1. First, the assumption is made that during the first and the sixth time period (before 7 AM and after 9AM) no toll is charged: the toll value is fixed to €0.00. These periods are clearly quiet in this model, so pushing away traffic from these times will not contribute to a reduction of the objective function value. So an initial solution consists of values for the remaining 12 variables of the problem (see section 6.1). These variables are for example called 700Town (representing time period 7:00-7:30 and link group 'Town') and 830A4 (representing time period 8:30-9:00 and link group 'A4") (see Table 6.2). Most analyses in this research have been executed with a single initial solution, thus neglecting the influence of the initial solution on the resulting optimal toll setting. This initial solution has been determined based on the dynamic results of the situation without tolls. The reference conditions show that the busiest link group is the A13 motorway and it is shown that queues start to form at approximately 7:30 AM and stay in the network until approximately 8:30AM. In order to push the traffic to earlier time periods, the toll is set on the maximum level in the third time period in all link groups and on the minimum level in the second time period. After 8:00 AM the traffic flow slightly diminishes, so the toll can go down slightly in time period 4. Time period 5 is quieter again, so tolls can go down further in this period. Because the A13 is busier than the two other link groups in the reference case, tolls are set one step higher on the A13. These considerations result in the initial solution in Table 6.22.

Nr	*Time period*	*Town*	*A13*	*A4*
1	6:00-7:00	Fixed on 0	Fixed on 0	Fixed on 0
2	7:00-7:30	700Town	700A13	700A4
3	7:30-8:00	730Town	730A13	700A4
4	8:00-8:30	800Town	800A13	800A4
5	8:30-9:00	830Town	830A13	830A4
6	9:00-10:00	Fixed on 0	Fixed on 0	Fixed on 0

Table 6.2: Variable names

Time period	*Toll level (€/km)*		
	town	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.15	0.20	0.15
8:30-9:00	0.05	0.10	0.05
9:00-10:00	0	0	0

Table 6.3: The initial solution

When the model is run with this toll setting, the average travel time in the network is reduced from 28.09 to 25.25 minutes. So this initial toll setting already gives a travel time reduction of 10.1% compared with the situation without tolls.

6.3.2 Exploring the objective function

In section 3.2.3 we discussed alternative specifications of objective functions for road pricing. As argued, in this research we will use the minimisation of average travel time on the network. Since the lower level of the mathematical program is considered to be a black box, the objective function value is an unknown function of the toll setting. Mathematically speaking, the shape of this function is totally unknown. In order to gain some insight in the shape of the function, in this section some calculations are carried out. First, some line searches are carried out in section 6.3.3. Each of the 12 variables is varied individually, starting from the same toll setting. This is equivalent with exploring the function in the 12 unity vector directions. So 12 line searches are carried out, beginning with the initial toll solution. For more theory on line search, see section 5.1 or for example Fletcher (1987). In this way it can be shown whether the function is approximately convex, ascending, descending with respect to one variable and whether it has a unique local minimum with respect to this one variable. Second, a small grid search is conducted for three variables in order to get some insight in interaction effects between variables (section 6.3.4). As shown in section 3.3 on the solution space, a total grid search for all 12 variables is computationally infeasible.

6.3.3 Line search

The results of the line search consist of the 12 graphs shown in Figure 6.3. In each of these graphs one variable is varied from its minimum to its maximum (from €0.00 to €0.20 per kilometre) on the horizontal axis in steps of €0.05. The other variables are fixed on the value of the initial solution (table 6.2). These values are bold at the axes of the graphs in . On the vertical axis the value of the objective function (average travel time in minutes) is plotted. Note that changing the value of any fixed variable can change the curve of the varied variable. Some comments on these graphs:

- In most cases only one local minimum exists. So in these cases local search along this line will result in the global minimum along this line.
  - Some of these cases have shapes of a convex function (800A4, 830A4), so when the curve is changed by changing other variables there is a small risk that the algorithm gets stuck in a local minimum when only this variable is varied.
  - Others have a unique minimum, but the shape of the function is not convex (700Town, 730Town), so when the curve changes by external variables there is a risk that local minima appear and the algorithm gets stuck in a local minimum at this variable.
  - Some variables are clearly bound by the minimum and maximum values (700Town, 700A13, 700A4, 730A13). For these variables it is likely that deviating from this value will not be optimal, so the algorithm has a small risk of getting stuck in a local minimum.
- Some variables have a small influence on the objective function value (for example 700Town and 830Town) Other variables have a big influence on the objective function value (especially 730A13). As would be expected, these variables correspond to quiet respectively busy roads and time periods. In some search algorithms this is taken into account by optimising the variables with a big influence first (see section 6.4).
- In two cases (800Town and 800A13) the function has two local minima on the line. When the initial solution is in the wrong local minimum, in these cases a local search algorithm will not find the global minimum along this line. Furthermore, when the initial solution is at €0,15 and when the local search algorithm first increases the value of the variable, it will also terminate in the wrong local minimum. In order to reduce this risk, the implemented search algorithm (section 6.4.4) evaluates both an increase and a decrease of every variable and jumps to the change which gives the most improvement in the objective function value.

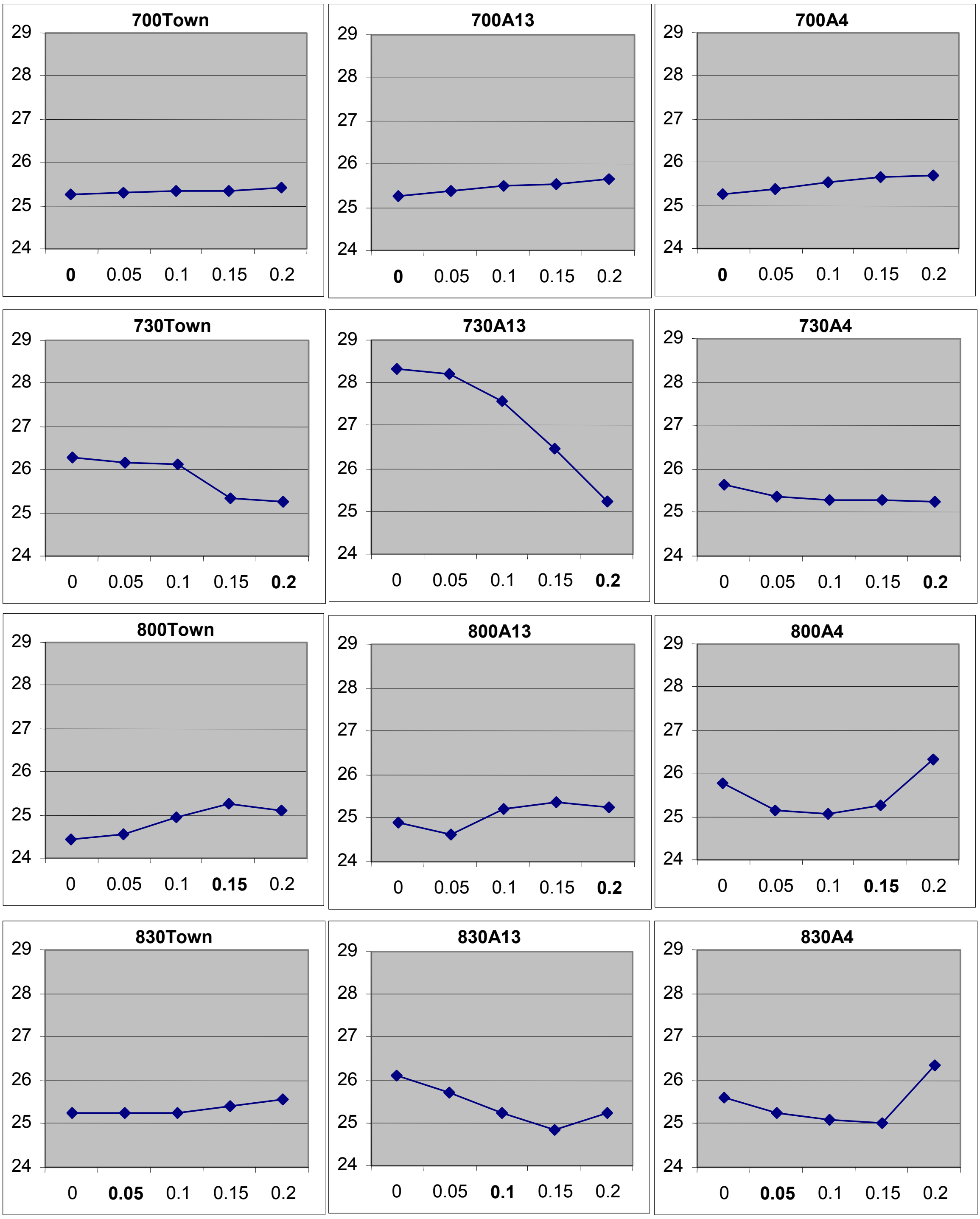


Figure 6.3: Results of the 12 line searches. Horizontal axis: toll level. Vertical axis: value of the objective function (average travel time in minutes)

Although in this specific case the global minimum will be found, no guarantee can be given for this to happen in a general case.

6.3.4 Grid search

In the conducted grid search, three variables are selected which are assumed to have a lot of interaction with each other, because the corresponding time intervals or the corresponding link groups can easily exchange traffic. These variables are 730Town, 800Town and 800A13. All other variables are kept fixed on the values of the initial solution. The first two variables show the interaction between time period 7.30 – 8.00AM and 8.00 – 8.30AM. The last two variables show the interaction between urban roads and the A13 motorway. When 730Town and 800A13 are compared, some insight may be gained on cross interaction effects of implementation of both time and space differentiation of tolls.

In a grid search all points on the defined grid are evaluated. In this grid search 3 variables are varied and all variables can take 5 different values. So the result of this grid search is a data set of 5^3 = 125 results. A table with all these results is added in appendix 9.4. Here the interaction effects of each of the three combinations of variables are viewed by 3D graphs. In these graphs one of the three variables is fixed. The other two variables are plotted on the two horizontal axes and the objective function value is plotted on the vertical axis. In general, these graphs give quite different results of the different cases. In some cases a very clear relation exists between the two variables (730Town – 800Town). In other cases, the function is very complicated. This indicates that in interaction between some variables, local minima exist a lot, and in interaction between other variables none of them exist. This indicates a risk of getting stuck in a local minimum when the optimal solution is approximated.

730Town – 800Town

In Figure 6.4 the value of the objective function is plotted, depending on the values of 730Town and 800Town. 800A13 is fixed at a value of €0.10. So at the same location (town) tolls are varied during two different time periods (7.30 – 8.00AM and 8.00 – 8.30AM). A clear relation can be identified: the higher 730Town and the lower 800Town, the lower the objective function value is. There is not much difference between the value at 730Town = 800Town = €0.00 and the value at 730Town = 800Town = €0.20. This is an indication that the absolute value of the toll is not really important: the difference between the toll values (or other costs) determines the network effects. The most important thing to improve the objective function value is to maximise the difference 730Town – 800Town. Note that in this research no elastic demand is modelled, which probably causes this observation. The observation is an indication that different local minima with the same objective function value exist.

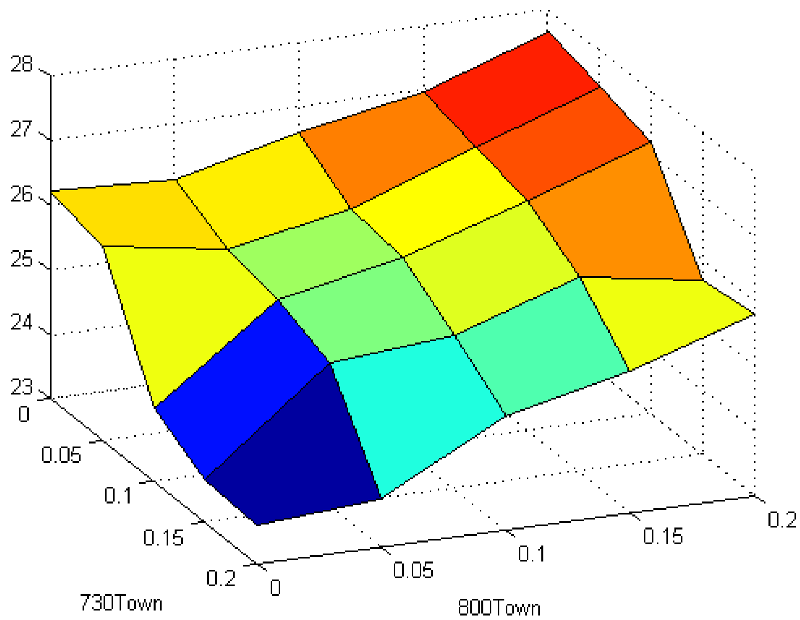


Figure 6.4: Interaction between 730Town and 800Town. 800A13 is fixed at a value of €0.10.

800Town – 800A13

In Figure 6.5 the value of the objective function is plotted, depending on the values of 800Town and 800A13. 730Town is fixed at a value of €0.00. So during the same time period (8.00 – 8.30AM), tolls are varied on two different locations (town and A13). This plot shows a less clear relationship: when 800A13 = €0.00 varying 800Town results in an increasing line, but all other values of 800A13 result in strange functions with multiple minima. No trend is visible for the relation between the two variables. An explanation for this may be the value of 730Town, because big differences exist between 730Town and its neighbour variables, 730A13, 730A4 and 800Town, so big cost differences occur between routes and departure times. These relatively big differences may lead to strong shifts in driver behaviour.

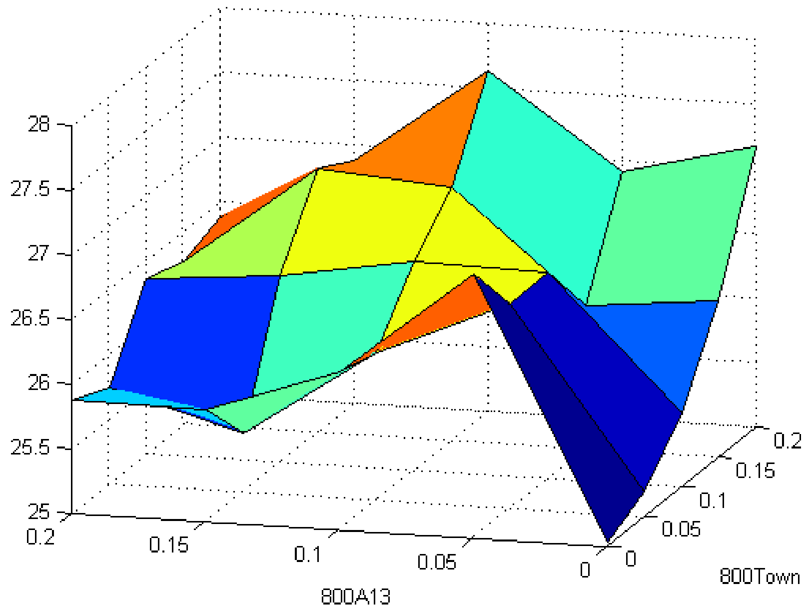


Figure 6.5: Interaction between 800Town and 800A13. 730Town is fixed at a value of €0.00

In Figure 6.6 the effect of the fixed value of 730Town is plotted. Two surfaces are shown: the upper one has 730Town = €0.00 (which is the same as in Figure 6.5) and the lower one has 730Town = €0.20. It is clearly visible that 730Town = €0.20 gives a better (lower) objective function value on all values of 800Town and 800A13. This is an example of a positive characteristic of the shape of the objective function. Some other plots, which are not shown here, are less well-ordered, and are another warning for the existence of local minima.

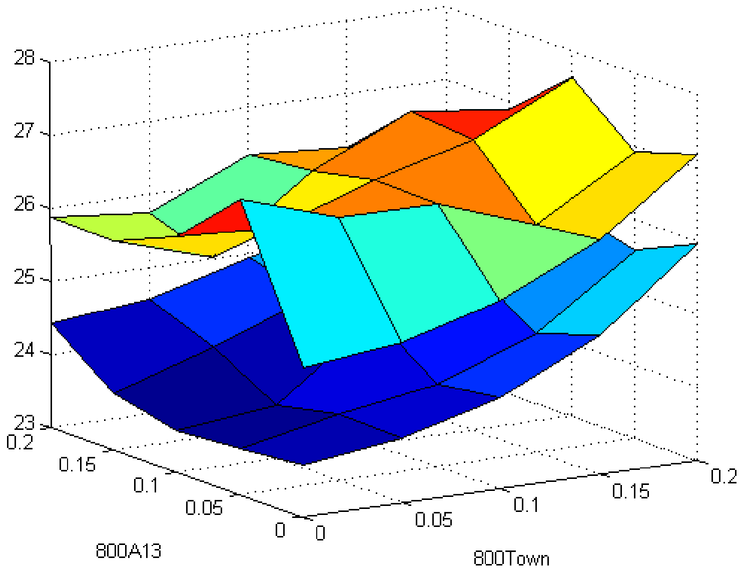


Figure 6.6: Interaction between 800Town and 800A13, when 730Town is changed from €0.00 (upper surface) tot €0.20 (lower surface).

730Town – 800A13

When 730Town and 800A13 are compared, the interaction effects of a change in both location and in time is investigated. In Figure 6.7 the value of the objective function is plotted, depending on the values of 730Town and 800A13. 800Town is fixed at a value of €0.00. When viewing this plot, some interaction effects can be observed, but they do not seem big. Varying 730Town individually has a big effect on the average travel time in a clear direction. Varying 800A13 individually has a smaller effect on the average travel time and the direction of the effect is dependent on the value of 800A13. Connected to this, a big difference can be observed between the slope of the function in this plot. For the speed of the search algorithm, it makes a big difference in what order these slopes are calculated: when the height gain of a big slope is taken in one iteration, the algorithm can be much faster than when a big detour is made to gain the same height.

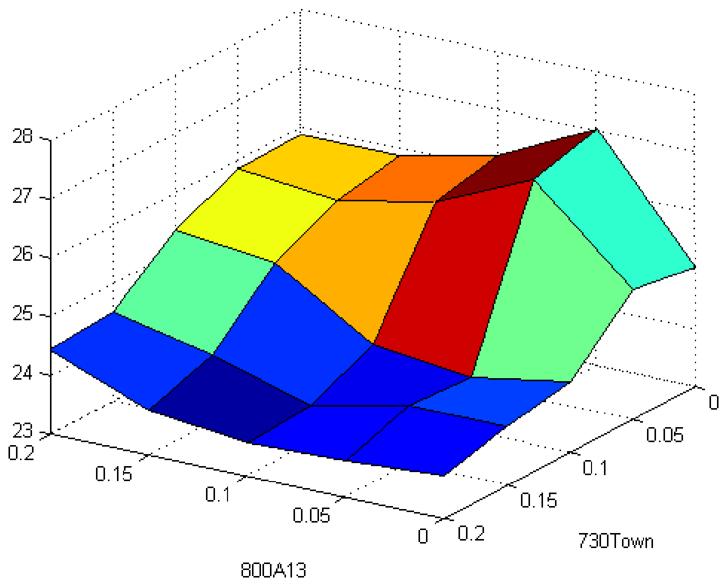


Figure 6.7: Interaction between 730Town and 800A13. 800Town is fixed at a value of €0.00

6.3.5 Conclusion

The exploration of the function resulted in the following observations:

- The function is not completely irregular, so it makes sense to conduct a pattern search on the function to find better solutions.
- The function still has non convex characteristics and local minima occur occasionally. So the results of the different search algorithms should be interpreted with care. Furthermore, the effect of different initial solutions should be studied.
- The absolute value of the tolls does not seem very important, while the difference between the value of variables is important (in the case of no elastic demand). Because the constraints to the toll values are quite tight bounds, it will also be possible to determine the absolute toll level in this case.
- The order in which the search algorithm evaluates objective function values amongst other things determines the speed of the algorithm.

6.4 Investigated variants of pattern search

As we know from chapter 5, pattern search can be executed in a lot of different variants. In this section the different variants of pattern search are given. First the properties of the different tested pattern search variants are given. In section 5.2.3 some properties were already defined to be the same for all variants. The section ends with an overview of the names and properties of the tested variants end a figure to illustrate the behaviour of the variants. In the next section the results for different variants of pattern search are treated.

6.4.1 The way to select the next variable

As we know from section 5.2.3, it is not known in advance which order of variables is the most efficient. From a traffic engineering point of view, the variables all have a meaning. In this way, different orders can be identified. In this research the following ways to select the next variable are investigated:

- A variable corresponds to both a time window and a link group. From here two different strategies arise: first leave the link group constant and change the time window, or keep the time window constant and change the link group. Both strategies are investigated in the research (patternsearch1 respectively patternsearch2).
- The next strategy is to first select a variable first, which corresponding time window and link group contributes most to the objective function value. Mathematically, this is easy to calculate, because the objective function is a sum over all links and time groups (equation (3.3)), so it is easy to take the sum over a limited set of time windows and links. This is called the divided objective function value. From traffic engineering point of view, this is logical, because these time windows and link groups are the busiest in the network, so probably optimisation of these variables will yield the most improvement in the total objective function value (patternearch3).
- Another possibility is to change a few variables at the same time. Mathematically this can be interesting to speed up the algorithm, because changing every variable one by one does not seem very efficient. From traffic engineering point of view it is chosen to change all variables within one link group or all variables within one time interval at the same time (patternsearch4).
- Finally an algorithm is added that makes a random selection of the next variable (patternsearch5). This is what happens in the fifth algorithm. The chance of selecting a certain variable is proportional to the value of the corresponding divided objective function value (see patternsearch3 above).

6.4.2 Step size

In this research a step size of €0.05 per km is used in all tested search algorithms. This is motivated by the relatively tight constraints on one hand and by the comprehensibility of the system on the other hand. The minimum toll is €0.00 per km and the maximum toll is €0.20 per km, so with this step size only 5 possible tolls exist. When the step size is made bigger, even fewer possible tolls exist, so there is not much left to optimise. Furthermore, a quick test without constraints and a big step size of €0.20 per km has shown that in this case tolls will not become higher than €0.20 per km, because of big behavioural changes and border effects corresponding to such a big step size. On the other hand, the step size has not been made smaller, because it is desirable that there are not too many different tolls, to keep the system understandable for the user. Furthermore, a quick test with a step size of €0.01 has shown that a danger exists to end up in a local minimum far from better solutions, because the search is much finer. This also makes the algorithm undesirably slow.

6.4.3 When a solution gives improvement

What to do after improvement is an important question, because it can influence the direction in which the algorithm develops. One strategy is to stay with a variable when improvement is occurred with the argument that it is likely that more improvement is possible in this variable. This strategy has the danger that the algorithm 'walks' into a gorge in the function landscape, and it ignores other direction in which more improvement is possible. For example, in the situation in Table 6.4, when the algorithm starts in point (3,2) and the first search direction is changing variable 1, an algorithm with this strategy will terminate in point (1,2) with an objective function value of 1.5, while the global minimum was very close to the initial point in point (3,3). To prevent this phenomenon a strategy can be used in which after improvement in one variable, the new value is saved, but a next variable is selected. In the case of Table 6.4, when again starting in point (3,2), the algorithm would terminate in the global minimum (3,3). This second strategy is used in patternsearch1 to patternsearch5. In patternsearch6 the first strategy implemented. Finally, in patternsearch7 the improved variables are stored in an improvement list. These variables are tried to improve further in the next iterations, until no further improvement is possible. Then, all variables are tried again to be improved, etc.

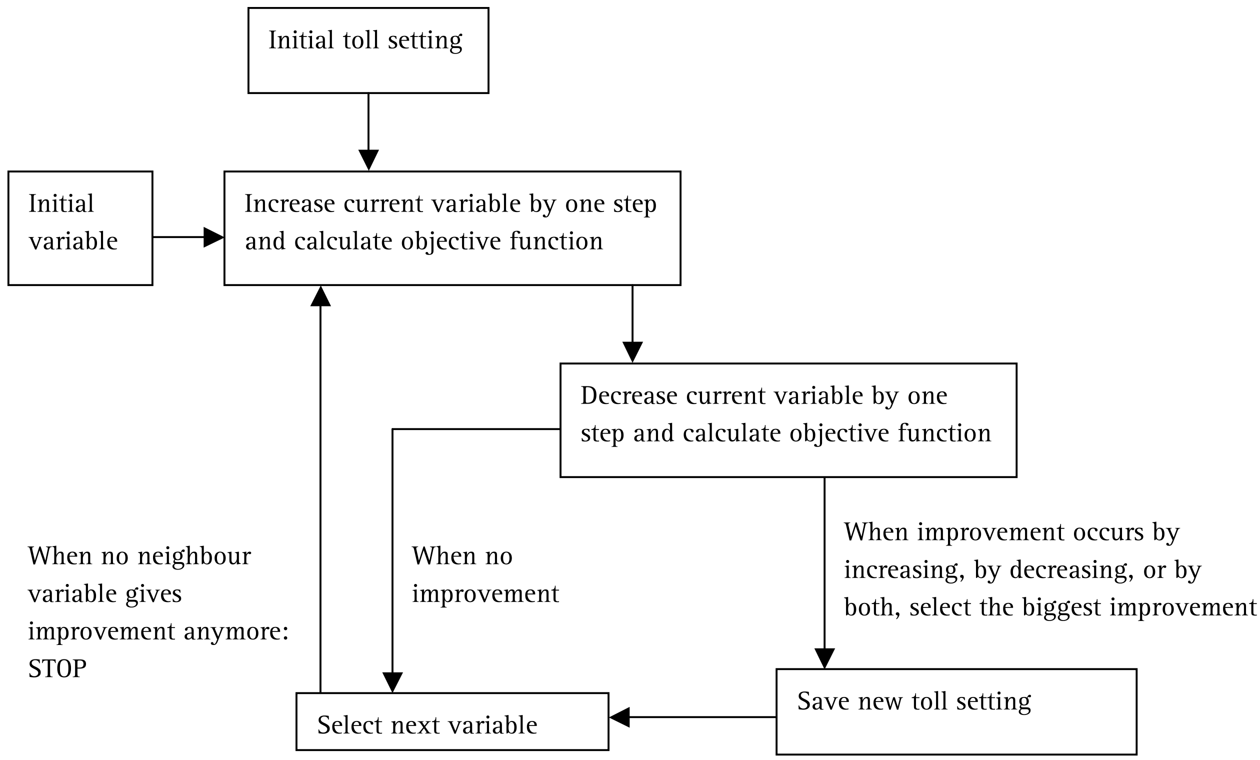


Figure 6.8: behaviour of pattern search algorithms 1 to 5: A variable is always increased by one step and decreased by one step. When there is improvement in both, the direction with the biggest improvement is chosen. The variable is adjusted and the next variable is selected.

		Variable1		
		1	*2*	*3*
Variable2	*1*	2.5	2.5	2.5
	2	1.5	1.8	2
	3	2	1	0.5

Table 6.4: Example with 2 variables which can each take 3 values. So the feasible region consists of 9 points.The objective function value of each feasible point can be read in the table.

6.4.4 Overview of different variants of pattern search

In Figure 6.8 a flowchart of the algorithms 1 to 5 is presented. The only difference between these 5 algorithms is the way to select the next variable, which is not specified in this figure. It can be seen that every variable is both increased and decreased every cycle. Only when an improvement is found, the new variable is saved. Constraints are always respected, so sometimes increase or decrease of a variable is not possible. When no improvement is detected at all neighbours, the algorithm terminates. In Figure 6.9 the behaviour of algorithm 6 is summarised. This algorithm differs from the other 5 algorithms in the way it behaves when improvement is detected: in that case it increases or decreases the variable further, until no more improvement takes place. The algorithms are programmed based on this figures. The pseudo code of the algorithms is added in appendix 9.3. Finally, in Table 6.5 an overview is given of all variants of different pattern search algorithms which are tested during this research.

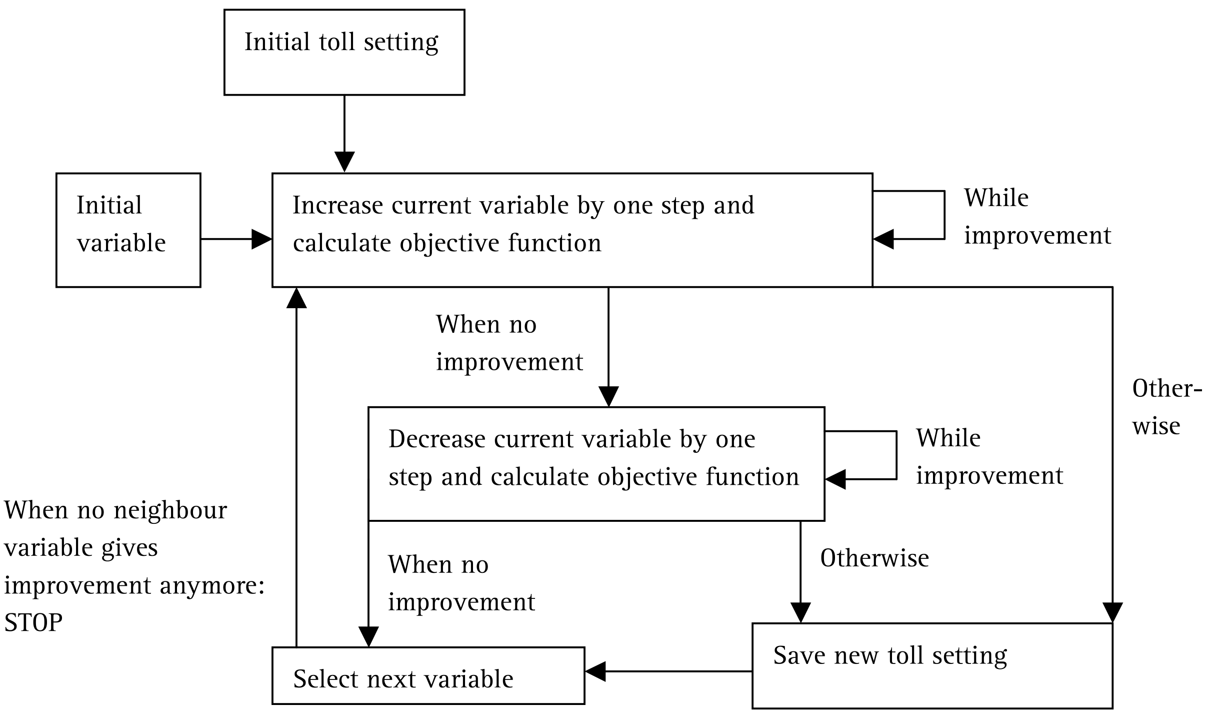


Figure 6.9: the behaviour of pattern search algorithm 6: when a variable is successfully increased by one step, it is increased further, before moving to the next variable

	Way to select next variable	*When improvement*
Patternsearch1	Next *time window* within current link group. After the last time window, select next link group. After the last link group, start again.	Save the new value of the variable and select the next variable.
Patternsearch2	Next *link group* within current time window. After the last link group, select next time window. After the last time window, start again.	Idem as patternsarch1
Patternsearch3	Next variable with the *highest divided objective function value* without a label. When a variable gives no improvement, add a label to it.	Idem as patternsarch1
Patternsearch4	*All link groups* within the next time window are selected. After the last time window, *all time windows* within the next link groups are selected. After the last link group, start again.	Idem as patternsarch1
Patternsearch5	*Randomly* select a variable from the set of variables without a label. The chance to be selected is proportional to the divided objective function value of the variable. When a variable gives no improvement, add a label to it.	Idem as patternsarch1
Patternsearch6	Idem as patternsearch2.	Try to increase / decrease current variable further, until this gives no further improvement. Save the new value of the variable.
Patternsearch7	Idem as Patternsearch1, but after the first round, only select *variables which gave improvement* in the former round (which are in the improvement list). When the improvement list is empty, evaluate all variables again.	Save the new value of the variable and select the next variable. Add variable to the improvement list.
- In this research the step size is kept constant at €0.05 - In this research the value of the toll is both increased and decreased (section 5.2.3) - In this research never a complete poll is executed (section 5.2.3) - In this research former iterations are always stored (section 5.2.3)		

Table 6.5: overview of the different variants of pattern search

6.5 Results of search algorithm tests

In this section the results of the tests of the different variants of pattern search are presented. First the results are given for all variants with the same initial toll solution. The results consist of the objective function value (average travel time) of the local minimum found by the algorithm, the corresponding toll setting, and the number of used iterations required to reach this solution. As stated earlier, computation of one iteration lasts 48 minutes. Second, three variants of the search algorithm are tested in more detail with two different initial toll solutions. Third, the effects of an easier, rougher incorporation in time windows and in link groups on objective function value and on toll setting are treated. Finally, the resulting toll settings are compared with respect to other objective functions.

6.5.1 The effect of the use of different search algorithms

This section gives the results of the application of the different pattern search approximation algorithms. First, in Table 6.6 the values of the average travel time for the free flow situation, for the situation without toll, and for the different initial toll solutions are stated in the table for comparison. In Table 6.7 for every search algorithm several results are given. Except for algorithm 4, for every search algorithm holds that it has terminated in a local minimum. This corresponds to a toll setting, where no improvement is possible by adjusting one variable with one step. Algorithm 4 changes three or four variables at the same time, so the resulting toll settings are not necessarily a local minimum in our definition. The corresponding value of the average travel time is stated as well. The results are ordered on increasing average travel time value. The time and space differentiated tolls corresponding to each toll setting can be found in appendix 9.1. For comparison, the improvement of the average travel time compared to the situation without toll and to the situation with the initial toll solution is also viewed in Table 6.7. For the number of used iterations two values are stated: the iteration number at which the local minimum is reached and the value at which the algorithm has terminated.

Situation	*Objective function value (min)*
Free flow situation	11.53
Reference run without tolls	28.09
Initial solution 1	25.25
Initial solution 2	27.83
Initial solution 3	31.95

Table 6.6: Average travel time without toll and with the initial toll solutions

Search algorithm	Initial toll solution	Resulting toll setting	Objective function value (min)	Relative improvement w.r.t. Reference	Relative improvement w.r.t. start	Number of iterations to final toll setting	Number of iterations to termination of algorithm
P1	Start 1	Toll5	23.11	-17.7%	-8.5%	64	80
P2	Start 1	Toll4	23.07	-17.9%	-8.6%	78	95
P3	Start 1	Toll10	24.22	-13.8%	-4.1%	12	28
P4	Start 1	Toll7	23.34	-17.0%	-7.6%	26	38
P5-1	Start 1	Toll2	22.90	-18.5%	-9.3%	57	74
P5-2	Start 1	Toll10	24.22	-13.8%	-4.1%	6	22
P5-3	Start 1	Toll10	24.22	-13.8%	-4.1%	21	37
P5-4	Start 1	Toll10	24.22	-13.8%	-4.1%	36	51
P6	Start 1	Toll10	24.22	-13.8%	-4.1%	26	42
P7	Start 1	Toll3	22.90	-18.5%	-9.3%	49	65

Table 6.7: Results achieved with the different pattern search algorithms

In Table 6.7 can be seen that the best objective function value achieved is an average travel time of 22.90 minutes. Coincidently, this value is achieved by two different search algorithms at two different local minima. Both computation time and objective function value are used to assess the variants. The best result is achieved by patternsearch7 (P7): this variant only uses 65 iterations to find the best local minimum until now. P5 also achieved this value, but this algorithm has a random component, and in the other three cases a much worse local minimum has been found. So the performance of this random algorithm seems more unsure than the other algorithms. P1 and P2 achieved a little worse objective function value, but used considerably more iterations to reach that value, so the performance is worse. P4 has a little worse objective function value again, but uses even less iterations to reach this value. So when computation time is an important limitation, this variant seems a good choice. The devel-

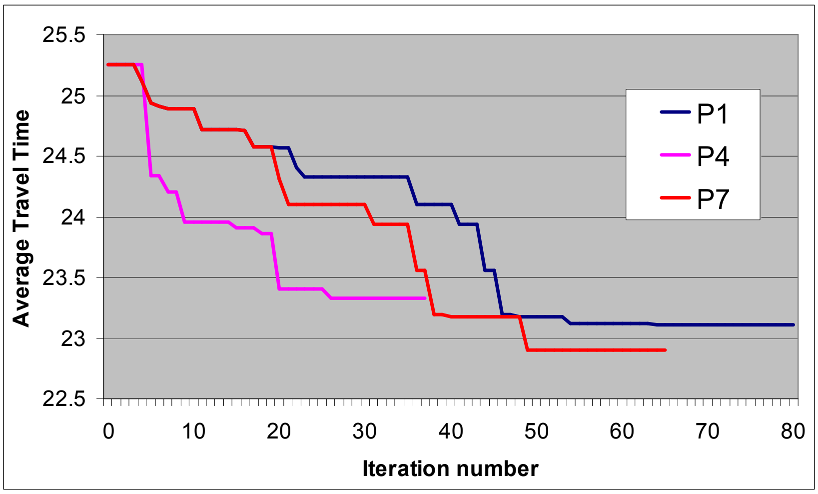


Figure 6.10: development of the value of the average travel time throughout iterations

opment of the average travel time in P1, P4, and P7 is compared in Figure 6.10, which illustrates the performance differences of these algorithms. The remaining two variants P3 and P6 both do not achieve a good average travel time value in this case. P6 uses more iterations for that than P4, so it is not a good variant. P3 uses very few iterations, so when computation time is even more limited, P3 can be chosen.

6.5.2 The effect of a different initial toll solution

As stated earlier, when a local search algorithm like pattern search is used, the chance to end up in a local minimum is high. From the former section, we already know that different search algorithms find different local minima. In this section the algorithms which performed best in the former section, are tested with two different initial toll solutions (see Table 6.8). These two solutions are chosen this way to test if the algorithms can handle strange initial toll solutions. Initial solution 2 has the maximum value of €0,20 per kilometre for every variable, in order to test whether the algorithm can recover from an initial solution which is far away from optimality in some variables (especially the toll during time period 7:00-7:30 is far too high). Initial solution 3 has an irregular form: six variables are set on the minimum level of €0,00 per kilometre and six variables are set on the maximum level of €0,20 per kilometre. In this way is tested whether the algorithm can handle a strange solution without getting stuck in a bad local minimum.

Initial toll solution 2 Objective function value: 27.83 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0.20	0.20	0.20
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.20	0.20	0.20
8:30-9:00	0.20	0.20	0.20
9:00-10:00	0	0	0

Initial toll solution 3 Objective function value: 31.95 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0.20	0
7:30-8:00	0.20	0	0.20
8:00-8:30	0	0.20	0
8:30-9:00	0.20	0	0.20
9:00-10:00	0	0	0

Table 6.8: Two alternative initial toll solutions

Search algorithm	Initial toll solution	Resulting toll setting	Objective function value (min)	Relative improve-ment w.r.t. Reference	Relative improve-ment w.r.t. start	Number of iterations to final toll setting	Number of iterations to termination of algorithm
P1	Start 1	Toll5	23.11	-17.7%	-8.5%	64	80
P1	Start 2	Toll8	23.46	-16.5%	-15.7%	111	125
P1	Start 3	Toll6	23.30	-17.1%	-27.1%	80	95
P4	Start 1	Toll7	23.34	-17.0%	-7.6%	26	38
P4	Start 2	Toll5	23.11	-17.7%	-17.0%	61	74
P4	Start 3	Toll9	23.68	-15.7%	-27.7%	79	91
P7	Start 1	Toll3	22.90	-18.5%	-9.3%	49	65
P7	Start 2	Toll1	22.72	-19.1%	-18.3%	123	139
P7	Start 3	Toll6	23.30	-17.1%	-27.1%	95	110

Table 6.9: Results achieved with three different search algorithms, with two other initial toll solutions

From Table 6.9 can be concluded that all three algorithms do not end up in a bad local minimum: all resulting objective function values are below 24 minutes. P7 even achieves a better objective function value from this different initial solution than from the original initial solution. The main effect to observe is the increase in computation time: because the initial solution is far away from the optimal situation, it takes more steps to reach a local minimum. Another observation is that the computation speed of P4 is relatively bad with initial toll solution 3. This can be explained by the heterogeneity of this initial solution: for P4, which only changes groups of variables together, it is difficult to recover from this strange situation to a situation that performs better in terms of average travel time. The other two variants change one variable at the time, so for these algorithms it is easier to recover from this strange situation. When the results of the three initial solutions are combined, we can conclude that P7 performs slightly better than P1, because it achieves a slightly better average travel time value in about the same number of iterations. P4 performs little worse in terms of average travel time value, but performs considerably better in terms of computation time: it is about 1/3 faster.

6.5.3 The effect of a smaller number of link groups or time windows

In the research two tests are conducted to find out the effect of a smaller number of variables on the computation speed and on the resulting objective function value. The variant of patternsearch1 is implemented in these tests. Two cases were investigated:

1. The case without space differentiation. This corresponds to the situation where all links are incorporated in one link group. Time differentiation is unchanged, so four different time windows are distinguished. This results in four variables.
2. The case with only two time windows. Usually a peak period consists of a busiest period in the middle of the peak and on both sides a less busy period. These two time windows are distinguished in this case: time window 1 consists of the periods 7.00AM – 7.30AM and 8.30AM – 9.00AM and time window 2 consists of the

period 7.30AM – 8.30AM. Space differentiation is unchanged, so three link groups exist. This results in six variables.

Initial toll solution 1 does not have the properties of these two cases. So for this test two new initial toll solutions have to be defined. These solutions are defined in a way they are about the same as initial toll solution 1. Below, for both cases the results are presented. First the initial toll solution and the resulting solution found by the algorithm are presented. Second, a table is given with the performance of the two cases, measured by the objective function value.

Initial toll solution			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.15	0.15	0.15
8:30-9:00	0	0	0
9:00-10:00	0	0	0

Resulting solution from the algorithm			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.10	0.10	0.10
8:30-9:00	0.05	0.05	0.05
9:00-10:00	0	0	0

Table 6.10: Initial and final solution for case 1: no space differentiation

Initial toll solution			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0.05	0
7:30-8:00	0.15	0.20	0.15
8:00-8:30	0.15	0.20	0.15
8:30-9:00	0	0.05	0
9:00-10:00	0	0	0

Resulting solution from the algorithm			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0.15	0.05
7:30-8:00	0.15	0.20	0.15
8:00-8:30	0.15	0.20	0.15
8:30-9:00	0	0.15	0.05
9:00-10:00	0	0	0

Table 6.11: Initial and final solution for case 2: less time differentiation

	Case 1	*Case 2*
Objective function value at start	26.14 min	25.99 min
Objective function value after approximation	24.04 min	25.53 min
Relative improvement w.r.t. Reference	-14.4%	-9.1%
Relative improvement w.r.t. start	-8.0%	-1.8%
Number of iterations to final solution	6	14
Number of iterations to termination of algorithm	11	24

Table 6.12: Performance of the algorithm when fewer link groups and fewer time windows are defined

Case 1 without space differentiation gives good results: it achieves a considerably better average travel time value than the initial solution and the value is not much higher than the values in the case of time and space differentiation (section 6.5.1). Despite the corresponding toll settings in the latter case are quite highly differentiated to space, this test showed that the effect of space differentiation is small. In the case with space differentiation a local minimum exists (Toll10), with a worse objective function value than this local minimum without space differentiation. So in one case, space differentiation has a negative influence on the average travel time value. As could be expected, the computation time goes down rigorously by the reduction in the number of variables when space differentiation is omitted, so in this case it is a very effective way of reducing computation time, without losing much quality of the solution.

Case 2 with less time differentiation gives worse results. Probably the combined time windows do not have the same characteristics in the traffic system, because when these time windows have the same toll value, this is not good for the average travel time value. This indicates that a detailed differentiation in time is good for the average travel time value in this case. This can be explained by the case specific PAT profile, which is concentrated in a small period in this case. In other situations less time differentiation may be a solution for limited computational power.

6.5.4 The effects on different objective functions

In this section the effects of the resulting toll settings on two other possible objective functions are stated (see Table 6.13): the total revenue and total congestion are computed by using objective functions (3.1) and (3.8). To make the value of the total congestion of a practical meaning, a multiplication is made with the length of a time period (1/12 h). The resulting number is the total amount of vehicles which could have travelled on congested links, when the total capacity would have been available. So longer links and links with a higher capacity are weighted higher in this number.

Resulting toll setting	Average travel time (min)	Total revenue (€)	Total congestion (10^3 veh km)
Toll1	22.72	25720	241
Toll2	22.90	23368	264
Toll3	22.90	22131	241
Toll4	23.07	24413	243
Toll5	23.11	23448	238
Toll6	23.30	23588	239
Toll7	23.33	23359	235
Toll8	23.46	28056	246
Toll9	23.68	21016	263
Toll10	24.22	28470	236

Table 6.13: values of average travel time, total revenue and total congestion

The total revenue of the different toll settings varies highly: the highest revenue is 35.5% higher than the lowest revenue, while the corresponding average travel time only differs 6.6%. The corresponding toll settings show that indeed a reasonable difference in toll level exists between these two situations (Table 6.14). Because the difference in average travel time is small between the solutions, the total revenue may be an argument to choose a specific toll setting. This can be considered quantitatively by computing whether constraint (3.15) is satisfied. The total congestion also varies with the different toll settings. The relation with average travel time is not clear. This is probably caused by the indicator formulation of the congestion objective function: a link is either congested or not, while the travel time on a link can vary continuously.

Toll solution 9 Objective function value: 23.68 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0	0.10	0
8:30-9:00	0.05	0.10	0.10
9:00-10:00	0	0	0

Toll solution 10 Objective function value: 24.22 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.05	0.20	0.15
8:30-9:00	0.05	0.15	0.05
9:00-10:00	0	0	0

Table 6.14: toll setting 10 generates a much higher revenue than toll setting 9

Optimisation with respect to another objective function would thus result in different solutions, as could be expected. In the case of average travel time minimisation, the resulting toll settings in most cases have higher tolls on the A4 than on the roads in the town. So apparently, the average travel time decreases when the traffic is guided through the town, which may for example from the viewpoint of liveability or an environmental viewpoint be an undesirable situation. This confirms that the objective function should be chosen with care and the resulting toll setting should be carefully considered, because otherwise undesirable effects on other objectives could occur.

6.6 Conclusions and generalisation

Several variants of the pattern search algorithm have been applied to the case study in this chapter. The variant called Patternsearch7, in which the algorithm saves in what variables improvement has been made, achieves better average travel time values within the same computation time compared with Patternsearch1, the variant which does not save this information. Patternsearch4, in which the algorithm changes more than one variable at a time, achieves much shorter computation times with only slightly worse average travel time values than Patternsearch1. These results are likely to be valid for other networks, because in other networks similarities between different time windows and different locations will probably also exist. However, this is not proved in this research.

The different search variants of pattern search reach multiple local minima, but the average travel time value is comparable in most local minima. So with different toll settings, approximately the same average travel time can be reached. The risk to end up in a bad local minimum is small, because different initial solutions all gave acceptable solutions. The initial solution has influence on the speed of the algorithm: when it is far away from the optimal solution, it takes longer to reach a local minimum than when it is already close. A weak relation exists between computation time and achieved average travel time value: long computation times achieve better average travel time values. The effect of the different local minima (toll settings) on total congestion and on total revenue differs more than the effect on average travel time. Also these results are likely to be valid for other networks, but this is not proved in this research.

A big reduction in computation time can be achieved by reducing the number of variables: a factor 2 resp. factor 3 reduction in the number of variables achieves a factor 3 resp. factor 8 reduction in computation time. In this case, time differentiation has a big positive effect on the average travel time value, so a reduction in the number of time windows causes a worse average travel time value. Space differentiation is less important in this case, because a reduction in the number of link groups only resulted in a slightly worse average travel time value. This result depends strongly on the network and on the PAT profile, so it is likely that results are different for other networks.

7 Conclusion

This chapter contains the main findings of this research. First the resulting framework for optimal toll design is presented. This framework gives a general description of the choices to be made when designing toll levels. After that, conclusions are made from the case specific results and notes are given on generalisation of these results. The chapter ends with a list of recommendations, which can be used as input for further research and for practical implementation.

7.1 Framework optimal toll design

This framework gives a method to design a toll setting with differentiation in time and in space. It is based on the previous chapters. The application to the case study in chapter 6 is a specific application of this framework: some choices can be different in different situations. The framework can be seen as a summary of this report and as a guideline for implementation of this research and for future improvement of the model. Some steps contain different possible actions: the most suitable action to a specific situation should be selected.

Because the solution space is immense and each model evaluation is time consuming, a large part of the framework is used to make the solution space smaller.

1. Determine the lower level traffic model to work with. The traffic model from chapter 4 is suitable to use, but is still rather slow in computation. In chapter 4 the requirements for a potential new model are listed. Furthermore, determine the necessary input data for the used traffic model.
2. Run the model in order to gain information on the situation without tolls. View the dynamic results of this reference run in the network.
3. From the observations in reality and from the model results without tolls, it is possible to identify the problems: what improvements of the traffic system are politically desirable? This defines the objective function. An objective function can be defined for the total network (like in chapter 3) or in a part of the network, can be defined over the total time period (like in chapter 3) or in some particular time interval.
4. Based on the reference results, categorise links and time windows in groups. The link groups should be logical and based on geography or on link type. There should only be a few link groups. The reference run can be used during this step: when no big differences between area's exist in the network one could choose to incorporate all links in one big link group. When differences are observed, one should define area's with the same characteristics in which links are incorporated in the same link group. Such characteristics can for example be level of congestion or land use. Spatially differentiated goals could imply an incorporation based on different link types, for example urban roads

and highways or radial roads and ring roads. The time intervals should be understandable and there should be only a few time intervals. Again the reference run can be used: detailed differentiation is only necessary if the behaviour of the system is very dynamic. When the system is very steady, no differentiation is necessary at all.

5. Determine the constraints to the pricing scheme. At least maximum and minimum toll levels or price categories should be defined. Possible additional constraints are listed in section 3.4.
6. Determine an initial solution. Based on the reference run one can identify the link groups and the time intervals where the problems in terms of the objective function are the biggest. The toll for these variables is set on the maximum level, because it seems desirable to push traffic away from these locations / times. Variables with small or no problems are set on the minimum toll level. The other variables get intermediate values. This approach only uses deduction from the graphical results from the reference run. It is also possible to determine the link groups and the time intervals which contribute most to the objective function value by calculating the objective function value for each link group and each time window. In this way, the numerical structure of the objective function value can contribute to the determination of the initial solution.
7. Determine the solution algorithm to use. The pattern search algorithm from section 5.2 is suitable to use for this case, but in other situations other algorithms can be useful.
8. Execute the solution algorithm.
9. When the search algorithm needs too much time in the present problem definition, it is necessary to adapt some of the choices made in the framework. In step 3 it is possible to define less link groups and /or less time window groups. In step 5 it is possible to add additional constraints: define less price categories or add other constraints from section 3.4 on the solution space. More information on the effect on the reduction of the solution space, can also be found in that section. In step 7 adaptations can be made to the search algorithm (see the different variants of pattern search in section 6.4.4). Usually the effect of such an adaptation has a big uncertainty, because the shape of the objective function is not known.
10. To improve the quality of the solution, the same case can be executed with different initial solutions and with different search algorithms.
    a. When this results in copies of the same solution, it becomes more likely that this is the global minimum.
    b. When this results in different solutions with about the same objective function value, it is not very important which solution is chosen.
    c. When this results in different solutions with different objective function values, it is difficult to say something about the quality of the best solution, and it is advisable to do as much runs as possible, in order to increase the chance of finding a good solution.

7.2 Conclusions

The stakeholder analysis resulted in a specific pricing scheme with the following properties to be investigated in this research: a time differentiated, space differentiated, link based fare per kilometre, which has not too many different prices and has a maximum level.

The optimal toll level design problem is formulated as a bi-level mathematical program. The upper level consists of the design variables (toll level), an objective function, and constraints to the design variables. The objective function in this research is the minimisation of the average travel in the network. For a given set of design variables, the lower level determines the depending variables and thus the value of the objective function.

Application of different variants of the pattern search algorithm to the case study showed that it is possible to achieve considerable improvements in the value of the average travel time compared with the situation without tolls and with an initial toll solution. Multiple local minima have been found, but the average travel time value is comparable in most local minima. So with different toll settings, approximately the same goals can be reached: by means of the policy objective, these toll settings are roughly the same. Other political arguments, like the expected revenue of these toll settings, can determine which exact toll setting is to be implemented.

The risk to end up in a bad local minimum is small, because different initial solutions all gave acceptable solutions. The initial solution has influence on the speed of the algorithm: when it is far away from the optimal solution, it takes longer to reach a local minimum than when it is already close.

When the algorithm saves in what variables improvement has been made, better average travel time values have been achieved within the same computation time. When the algorithm changes more than one variable at a time, considerably shorter computation times can be achieved with only slightly worse average travel time values. When all variants of pattern search are compared, a weak relation exists between computation time and achieved average travel time value: long computation times achieve better average travel time values.

The improvements in terms of average travel time value are bigger when more detailed time and space differentiation of tolls is applied, but the computation time grows fast in that case. In this case study, time differentiation of tolls is more important than space differentiation. Most computation time can be saved by defining less detailed space and time differentiation.

The result of this research is a practical modelling framework for the approximation of optimal toll design. Restrictions to computational power make that at this moment it is not yet possible to approximate highly differentiated tolls in big networks with the proposed modelling framework. The approximation of very roughly differentiated tolls on bigger networks is already possible, which is promising when the space and time differentiation will be designed for the Dutch road pricing system.

It is likely that general findings in this research also apply for other networks, though this is not proved by this research. Furthermore, this research uses a modelling environment, so it is not possible to be sure about the effects of the road pricing measures in reality. Not until the system is operable in the future, it is possible to be sure about the effects of road pricing on the traffic system.

7.3 Recommendations

- The present computation speed is too slow for bigger networks. The following technical measures can contribute to a more feasible computation time for realistic network sizes:
    - Replace INDY by a faster module, for example MADAM. MADAM is a module for dynamic traffic assignment in the OmniTRANS modelling environment.
    - Make the departure time module more efficient. The present code for departure time choice can still be written more efficiently.
    - Improve the lower level model by reducing the necessary number of iterations to reach equilibrium. In this research a lot of improvement has been made, but still it should be possible to reduce the number of iterations and thus the computation time of the equilibrium model.
    - Facilitate parallel processing: use multiple computers and accumulate the data on one central computer. The exact form of the search algorithm should be changed in this case: the algorithm should be able to handle separate computation of a series of solutions, because several corresponding objective function values are computed simultaneously.

- The following steps in the modelling framework are most likely to cause big improvements in computation time:
    - The initial solution strongly determines the speed of the algorithm. Development of a better method to guess the initial solution could speed up the process.
    - Be careful when the time windows and link groups are defined. Only make definitions which seem really necessary in the given case.
    - Be careful when the price categories are defined. In practice it is likely that only a few categories are acceptable and this also makes the computation much faster.

- A lot of assumptions were made in the traffic model. By using a more sophisticated traffic model, the optimal toll designs will be more realistic.
  - Add elastic demand to the model
  - Add spatial redistribution to the model
  - Test the effect of other PAT profiles
  - Define more different user groups
  - Validate behavioural parameters on real life data

- The results in this research are based on a single case study. To make the results more reliable, it is recommended that the different search algorithms are tested on different cases, before this framework is applied in real life.

- Try to use more information on the processes in the lower level model: gain more information on the processes in the 'black box'. When certain computations are made and stored during execution of the lower level, it may be possible to compute numerical gradients in an efficient way (automatic differentiation).

8 References

Amelsfort, D.H., van (forthcoming) *Time varying road pricing measures: behavioural responses and network effects*, PhD Thesis, Delft University of Technology

ANWB (2004) *Op weg naar overmorgen, Vernieuwende kijk van de ANWB op mobiliteit voor de komende drie tot vijf jaar (in Dutch)*, ANWB, Amsterdam, The Netherlands.

Bard, J.F. (1998) *Practical bilevel optimization. Algorithms and applications*, Nonconvex Optimization and its Applications, 30. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Bergenhoff, P., D.W. Hearn, and M.V. Ramana, (1997) *Congestion toll pricing of traffic networks*, in Pardalos, Hearn, Hager (editors), Lecture Notes in Economics and Mathematical Systems 450, Network Optimization, Springer, 51-71.

Bliemer, M. and H. Taale (2006) Route Generation and Dynamic Traffic Assignment for Large Network. *Proceedings of the 1st Symposium on Dynamic Traffic Assignment*, Leeds, UK.

Bliemer, M.C.J. (2001) *Analytical dynamic traffic assignment with interacting user classes: theoretical advances and applications using a Variational inequality approach*, Delft University of Technology, Delft, The Netherlands.

Bliemer, M.C.J. (2004) *DTA Model Specifications*. Working document, Delft University of Technology, The Netherlands.

Bliemer, M.C.J., H.H. Versteegt, and R.J. Castenmiller (2004) DTA: A New Analytical Multiclass Dynamic Traffic Assignment Model. *Proceedings of the TRISTAN V conference*, Guadeloupe, French West Indies.

Fletcher, R. (1987) *Practical methods of Optimization*. John Wiley & Sons, Chichester, UK.

Hau, T.D. (2006a) *Congestion charging mechanisms for roads, Part I – Conceptual framework*. Transportmetrica, 2, 87-116.

Hilbers, H., M. Thissen, P. van de Coevering, N. Limtanakool, and F. Vernooij (2007) *Beprijzing van het wegverkeer, de effecten op doorstroming, bereikbaarheid en de economie (in Dutch)*. Nai Uitgevers, Rotterdam, The Netherlands. Ruimtelijk planbureau, Den Haag, The Netherlands.

Joksimovic, D. (2007) *Dynamic bi-level optimal toll design approach for dynamic traffic networks*. Delft University of Technology, Delft, The Netherlands.

Jou, R.C., S.H. Lam, and P.H. Wu (2007) Acceptance tendencies and commuters' behaviour under different road pricing schemes. *Transportmetrica*, 3, 213-230

Kiel, Jan and Remko Smit (2004) 15 jaar Nieuw Regionaal Model: Tijd voor een nieuw handboek (in Dutch). *Presented at Colloquium Vervoersplanologisch Speurwerk*, Zeist, The Netherlands.

Matlab (2004) *Genetic algorithm and Direct Search Toolbox*. Help file of Matlab version 7.0.0.19920. The Mathworks, inc.

McFadden, D. (1976) Quantal Choice Analysis: A Survey, *Annals of Economic and Social Measurement*, 5, 4, 363-390.

Michalewicz, Zbigniew and David B. Fogel (2000) *How to solve it: Modern Heuristics*. Springer, Berlin, Germany.

Ministry of Finance (2007) *Missie van het ministerie van Financiën (in Dutch)*, http://www.minfin.nl/nl/organisatie/missie, 02-10-2007)

Overheid (2007a) *Wet op de belasting van personenauto's en motorrijwielen 1992* (in Dutch), http://wetten.overheid.nl/ 03-06-2008

Overheid (2007b) *Wet op de motorrijtuigenbelasting 1994 (in Dutch)*, http://wetten.overheid.nl/, 19-9-2007.

Overheid (2007c) *Wet op de Accijns (in Dutch)*, http://wetten.overheid.nl/, 19-9-2007.

Pieper, R. (2001) *MobiMiles. Bewust op weg (In Dutch)*. By order of the Minister of Transport, Public works and Water Management, Bloemendaal, The Netherlands.

Teekamp, R., E.M. Bezembinder, D.H. van Amelsfort, and E.C. van Berkum (2002) Estimation of preferred departure times and changes in departure time patterns, *Proceedings of the 81st Annual meeting of the Transportation Research Board*, Washington D.C., USA.

Tillema, T. (2007) *Road pricing: a transport geographical perspective*. Utrecht University, Utrecht, The Netherlands.

Ubbels, B. (2006) *Road pricing: Effectiveness, Acceptance and Institutional aspects*. Free University of Amsterdam, Amsterdam, The Netherlands.

VenW & VROM (2004) *Nota Mobiliteit, Naar een betrouwbare en voorspelbare bereikbaarheid (in Dutch)*. Ministerie van Verkeer en Waterstaat, Ministerie van Volkshuisvesting, Ruimtelijke Ordening en Milieu beheer, Den Haag, The Netherlands.

Verhoef, Erik, Michiel Bliemer, Linda Steg, and Bert van Wee (2008) *Pricing in Road Transport, a multi-disciplinary perspective*. Edward Elgar Publishing Limited, Cheltenham, UK.

Viti, F., S.C. Fiorenzo, M. Li, Ch.D.R. Lindveld, and H.J. van Zuylen (2003) An optimization problem with dynamic route-departure time choice and pricing, *proceedings of the 82th Transportation Research Board Annual Meeting*, Washington D.C., USA.

Wardrop (1952) J.G., Some theoretical aspects of road traffic research. *Proc. Inst. Civ. Eng. Part II*, 325-378.

Yang, Hai, Hai-Jun Huang (2005) *Mathematical and economic theory of road pricing*. Elsevier Ltd., Oxford, UK

9 Appendices

9.1 Appendix: optimal toll settings

In this appendix an overview is given of all results of the search algorithms: the time and space differentiated toll settings. Each of these toll settings has a number. The corresponding objective function value is shown as well.

Toll solution 1 Objective function value: 22.72 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0.05
7:30-8:00	0.15	0.20	0.15
8:00-8:30	0	0.05	0.20
8:30-9:00	0	0.10	0.20
9:00-10:00	0	0	0

Toll solution 2 Objective function value: 22.90 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.15	0.20	0.15
8:00-8:30	0.05	0.05	0.15
8:30-9:00	0	0.10	0.15
9:00-10:00	0	0	0

Toll solution 3 Objective function value: 22.90 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.10
8:00-8:30	0	0.15	0.10
8:30-9:00	0	0.10	0.10
9:00-10:00	0	0	0

Toll solution 4 Objective function value: 23.07 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.15
8:00-8:30	0	0.10	0.15
8:30-9:00	0.05	0.10	0.10
9:00-10:00	0	0	0

Toll solution 5 Objective function value: 23.11 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.15	0.20	0.20
8:00-8:30	0	0.15	0.10
8:30-9:00	0	0.10	0.10
9:00-10:00	0	0	0

Toll solution 6 Objective function value: 23.30 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0	0.15	0.10
8:30-9:00	0.05	0.05	0.10
9:00-10:00	0	0	0

Toll solution 7 Objective function value: 23.33 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0	0.15	0.10
8:30-9:00	0	0.10	0.05
9:00-10:00	0	0	0

Toll solution 8 Objective function value: 23.46 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0.05	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0	0.15	0.20
8:30-9:00	0	0.05	0.20
9:00-10:00	0	0	0

Toll solution 9 Objective function value: 23.68 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0	0.10	0
8:30-9:00	0.05	0.10	0.10
9:00-10:00	0	0	0

Toll solution 10 Objective function value: 24.22 min			
	Toll level (€/km)		
Time period	*town*	*A13*	*A4*
6:00-7:00	0	0	0
7:00-7:30	0	0	0
7:30-8:00	0.20	0.20	0.20
8:00-8:30	0.05	0.20	0.15
8:30-9:00	0.05	0.15	0.05
9:00-10:00	0	0	0

9.2 Appendix: Convergence of the lower level model

From chapter 4 we know that the lower level model is an equilibrium model, which uses an iterative process to reach this equilibrium. When optimisation of the tolls is taking place in the upper level, it is desirable that at every iteration of the upper level, the equilibrium is approximately reached in the lower level. In this appendix some aspects of this topic are highlighted.

9.2.1 Convergence original algorithm

For this research a model developed by Goudappel Coffeng BV was used. The original departure time choice model uses the method of successive averages for the calculation of the new dynamic OD-matrices. This means that the matrices of the former iteration get a weight of $n/(n+1)$ in the calculation of the present iteration and the newly calculated matrices get a weight of $1/(n+1)$, where n is equal to the iteration number. This method is used to achieve a stable convergence to the equilibrium, or with other words to prevent zigzagging, because at higher iterations the new matrices are weighted lighter than in the earlier iterations. A consequence using this method is a relatively slow convergence. In Figure 9.1 the development of the objective function value is shown when the lower level model is run with the initial toll solution with 150 iterations. This figure shows that at least around 50 iterations are necessary to reach the neighbourhood of the equilibrium. This requires around 5 hours on a modern computer in the case of this simple test network, so the evaluation of one objective function value would require too much computation time.

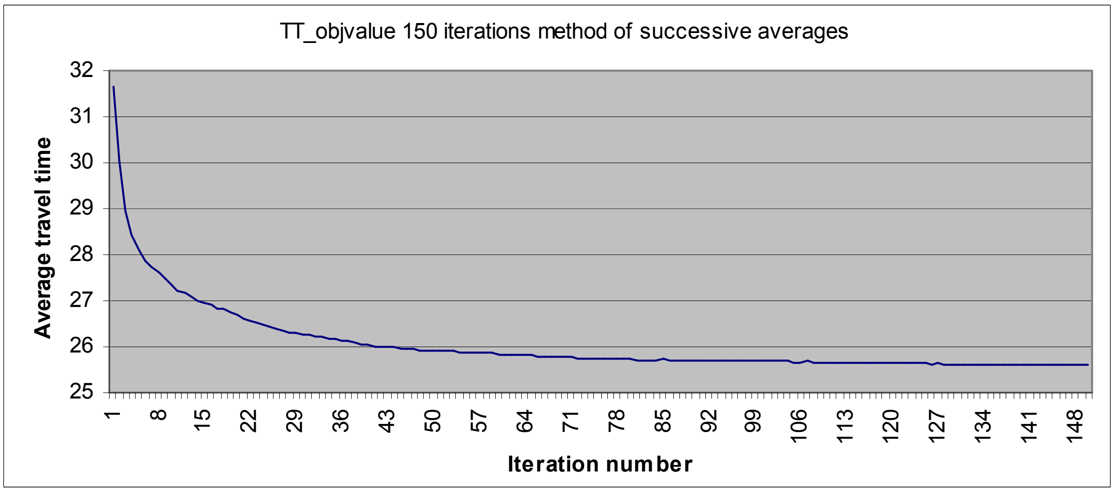


Figure 9.1: Convergence of the lower level model before changes

9.2.2 Initial network conditions

A possibility to speed up the convergence of the lower level is to choose other initial network conditions. In the initial model, the preferred arrival time profiles were used to calculate the first flows on the network. In this situation many people depart at the same time, so a lot of congestion occurs and the model starts with a situation that is nowhere near the equilibrium. In this research, the results of the reference run (section X) are used as starting data for the network conditions. In this way, the algorithm starts closer to the equilibrium and it converges faster to the equilibrium. When both the new weight factors as the new initial solution are used, it is sufficient to do 8 iterations for 1 objective value calculation (see section 9.2.3). This lasts 48 minutes on a modern computer in this case, which is fast enough to execute approximation algorithms within a reasonable computation time. In Figure 9.2 the course of the objective function value is shown when the new method is used (again for the initial toll solution).

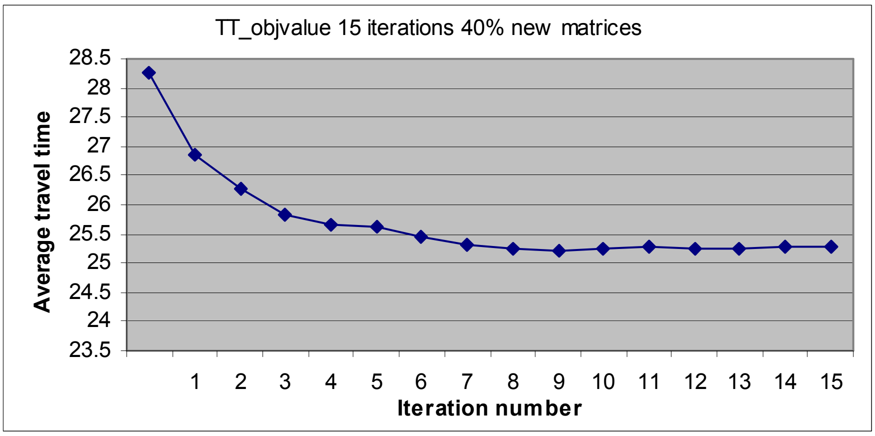
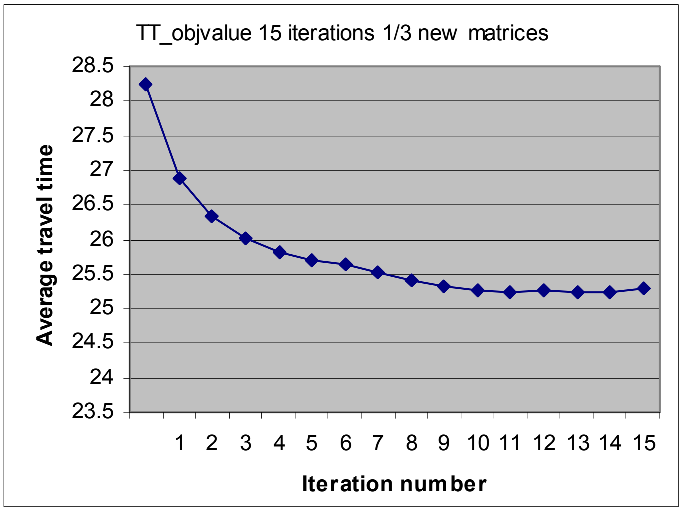


Figure 9.2: Convergence of the lower level model after changes. The first value (iteration 0) is the value of the reference situation without tolls.

9.2.3 Convergence new algorithm

The algorithm used in this research has a constant weight for the new matrices of 40%. Tests have shown that a weight of 50% gives undesirable zigzagging in some cases and a weight of 1/3 still gives slower convergence. First a comparison is made between the convergence speed of the method of successive averages and the new method with weights of 50%, 40% and 1/3 for the new matrices. Second, a comparison is made between the calculation of the objective function value of two different toll settings and the number of used iterations is motivated.



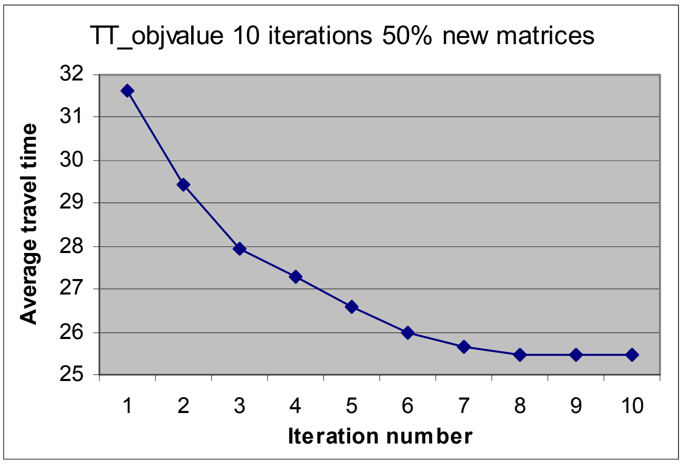
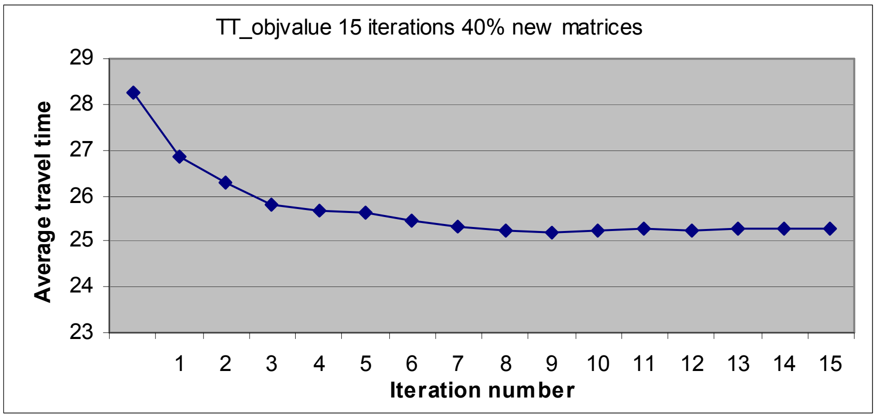


Figure 9.3: Convergence of 50% weight and of 1/3 weight

In Figure 9.3 the two methods to average the new and old dynamic OD matrices are shown. It can be seen clearly that the first method with weight 1/3 converges slower than the second method with a weight of 50%: after 8 iterations the second method reached convergence at the same objective function value as the first method has reached after 11 iterations. This is much faster than the original algorithm, which uses around 100 iterations. In some specific cases which are not shown here a weight of 50% gave undesirable zigzagging and convergence was not that fast, so here the weight of the new matrices is too high. A weight of 1/3 speeds up the algorithm, but still in this case convergence is reached after 11 iterations. To make this number a little smaller a weight of 40% has been implemented. This resulted in a necessary number of iterations of 8 in the case of the initial toll solution (see Figure 9.4). The second graph of this figure gives the convergence process for a toll setting with considerably better performance than the initial toll solution. This figure is given to prove that convergence after 8 iterations is also acceptable in the case of a lower objective function value. So during the approximation algorithms in the lower level 8 iterations are used to determine the objective function value.

So concluding, the weight of 40% is considered to be a good trade off between stability and convergence speed.



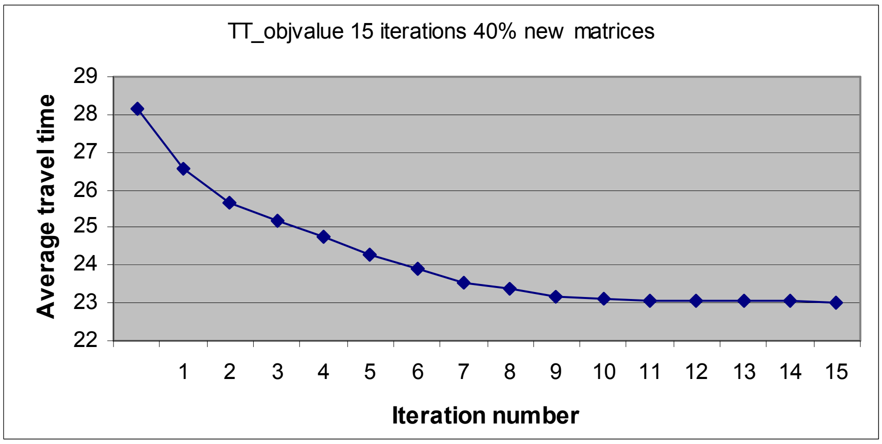


Figure 9.4: Comparison between the initial toll solution and a near optimal toll solution

9.3 Appendix: structure of the code of the algorithms

Patternseach 1 and 2

```
While improvement in objective value
      For all variables
            Increase current variable by one step
            If not calculated earlier
                  If feasible
                        Calculate objective function value
                        If improvement in objective value
                              Save new value of variable
                        Else
                              Go back to original value of
                              variable
                        End
                  Else
                        Go back to original value of variable
                  End
            Else
                  Go back to original value of variable
            End
            Decrease current variable by one step
            If not calculated earlier
                  If feasible
                        Calculate objective function value
                        If improvement
                              Save new value of variable
                        Else
                              Go back to original value of
                              variable
                        End
                  Else
                        Go back to original value of variable
                  End
            Else
                  Go back to original value of variable
            End
      End
End
```

9.4 Appendix: grid search results

In addition to the results in section 6.3.4, in this appendix all 125 results of the grid search run are shown.

	730 Town	800 Town	800 A13	Objvalue Average TT (min)
1	0	0	0	25.03
2	0	0	0.05	25.20
3	0	0	0.1	24.20
4	0	0	0.15	23.99
5	0	0	0.2	23.71
6	0	0.05	0	27.04
7	0	0.05	0.05	26.73
8	0	0.05	0.1	23.92
9	0	0.05	0.15	24.00
10	0	0.05	0.2	23.62
11	0	0.1	0	26.21
12	0	0.1	0.05	26.00
13	0	0.1	0.1	24.13
14	0	0.1	0.15	23.65
15	0	0.1	0.2	23.58
16	0	0.15	0	25.86
17	0	0.15	0.05	25.66
18	0	0.15	0.1	25.15
19	0	0.15	0.15	24.14
20	0	0.15	0.2	23.77
21	0	0.2	0	25.87
22	0	0.2	0.05	25.83
23	0	0.2	0.1	25.37
24	0	0.2	0.15	24.54
25	0	0.2	0.2	24.44
26	0.05	0	0	25.19
27	0.05	0	0.05	24.72
28	0.05	0	0.1	24.19
29	0.05	0	0.15	23.96
30	0.05	0	0.2	23.88
31	0.05	0.05	0	26.58
32	0.05	0.05	0.05	24.28
33	0.05	0.05	0.1	24.16
34	0.05	0.05	0.15	23.85
35	0.05	0.05	0.2	23.90
36	0.05	0.1	0	26.14
37	0.05	0.1	0.05	25.69
38	0.05	0.1	0.1	25.56
39	0.05	0.1	0.15	25.19
40	0.05	0.1	0.2	23.73
41	0.05	0.15	0	25.45
42	0.05	0.15	0.05	25.51
43	0.05	0.15	0.1	25.17
44	0.05	0.15	0.15	24.34
45	0.05	0.15	0.2	24.22
46	0.05	0.2	0	25.74
47	0.05	0.2	0.05	26.07
48	0.05	0.2	0.1	25.77
49	0.05	0.2	0.15	24.78
50	0.05	0.2	0.2	24.56
51	0.1	0	0	25.57
52	0.1	0	0.05	24.76
53	0.1	0	0.1	24.47
54	0.1	0	0.15	24.10
55	0.1	0	0.2	24.24
56	0.1	0.05	0	26.59
57	0.1	0.05	0.05	23.97
58	0.1	0.05	0.1	24.21
59	0.1	0.05	0.15	24.06
60	0.1	0.05	0.2	24.12
61	0.1	0.1	0	26.61
62	0.1	0.1	0.05	26.07
63	0.1	0.1	0.1	25.94
64	0.1	0.1	0.15	25.37
65	0.1	0.1	0.2	24.74
66	0.1	0.15	0	26.44
67	0.1	0.15	0.05	25.97
68	0.1	0.15	0.1	25.79
69	0.1	0.15	0.15	25.53
70	0.1	0.15	0.2	24.71
71	0.1	0.2	0	26.36

72	0.1	0.2	0.05	26.08
73	0.1	0.2	0.1	25.58
74	0.1	0.2	0.15	25.06
75	0.1	0.2	0.2	24.94
76	0.15	0	0	26.22
77	0.15	0	0.05	26.10
78	0.15	0	0.1	25.82
79	0.15	0	0.15	25.17
80	0.15	0	0.2	24.90
81	0.15	0.05	0	26.11
82	0.15	0.05	0.05	25.08
83	0.15	0.05	0.1	25.05
84	0.15	0.05	0.15	24.84
85	0.15	0.05	0.2	24.61
86	0.15	0.1	0	26.96
87	0.15	0.1	0.05	26.75
88	0.15	0.1	0.1	26.55
89	0.15	0.1	0.15	26.02
90	0.15	0.1	0.2	25.20
91	0.15	0.15	0	27.05
92	0.15	0.15	0.05	26.67
93	0.15	0.15	0.1	26.33
94	0.15	0.15	0.15	25.43
95	0.15	0.15	0.2	25.35
96	0.15	0.2	0	26.27
97	0.15	0.2	0.05	26.15
98	0.15	0.2	0.1	26.14
99	0.15	0.2	0.15	25.32

100	0.15	0.2	0.2	25.25
101	0.2	0	0	27.18
102	0.2	0	0.05	26.60
103	0.2	0	0.1	26.47
104	0.2	0	0.15	25.84
105	0.2	0	0.2	25.95
106	0.2	0.05	0	26.91
107	0.2	0.05	0.05	26.75
108	0.2	0.05	0.1	26.46
109	0.2	0.05	0.15	25.67
110	0.2	0.05	0.2	25.57
111	0.2	0.1	0	27.64
112	0.2	0.1	0.05	27.42
113	0.2	0.1	0.1	27.20
114	0.2	0.1	0.15	25.69
115	0.2	0.1	0.2	25.81
116	0.2	0.15	0	26.88
117	0.2	0.15	0.05	26.60
118	0.2	0.15	0.1	26.63
119	0.2	0.15	0.15	25.77
120	0.2	0.15	0.2	25.70
121	0.2	0.2	0	26.38
122	0.2	0.2	0.05	26.97
123	0.2	0.2	0.1	26.09
124	0.2	0.2	0.15	25.20
125	0.2	0.2	0.2	25.12