OPTIMISING DIFFERENTIATED TOLLS ON LARGE SCALE NETWORKS, BY USING AN INTELLIGENT SEARCH ALGORITHM.

Ties Brands and Dirk H. van Amelsfort
Goudappel Coffeng BV
PO Box 161, 7400 AD Deventer, The Netherlands

Eric C. van Berkum
University of Twente, Centre for Transport Studies
PO Box 217, 7500 AE Enschede, The Netherlands

ABSTRACT

The design of an optimal road pricing scheme is not a trivial problem. Following the Dutch government's kilometre charge plans, this paper focuses on the optimisation of link based toll levels differentiated in space and time. The optimal toll level design problem is formulated as a bi-level mathematical program. In the upper level we minimise an object function, e.g. the average travel time in the network, using a fixed number of price categories. At the lower level a dynamic traffic assignment model is used to determine the effects of differentiated road pricing schemes on the traffic system. Focus of the paper is on the upper-level where optimal toll levels are approximated. In the optimisation procedure different variants of a pattern search algorithm are tested in a case study. Inspection of the solution space shows that many local minima exist, so the selection of the initial solution becomes important. In the case study however it appears that in all local minima the value of the objective function is almost the same, indicating the fact that many different toll schemes result in the same average travel time. The case study is also used to test the performance of the different variants of the pattern search algorithm. It appears that it is beneficial to change more variables at a time and to use a memory to remember where improvement of the objective function has been made. First tests on a medium scale network showed that it is possible to apply the framework on this network, though further computational improvements are needed to apply the framework to large scale networks, for example by parallel processing.

1 INTRODUCTION

Within the world of traffic engineering, road pricing is considered as a measure that may alleviate several problems in the current transport system: congestion, environmental damage, use of unsustainable recourses, use of space, etc. Some successful practical cordon based applications of road pricing exist (for instance Singapore, London, Stockholm). The Dutch government plans to develop a link based, time, space, and vehicle type differentiated toll system. Thus the amount of toll a car driver has to pay depends a. o. on the number of kilometres driven, but also on the time of travel, the route chosen, and the vehicle. Main goal of this system is to achieve a fairer system where heavy users of the transport system pay more than occasional users, and where the toll level follows demand: the higher demand, the higher the price. Further, with a toll system as proposed also other objectives may be met. Model studies have shown that as a result of toll measures problems concerning congestion, CO_2 and air quality can be alleviated. Now this paper addresses an important question: given a network and a demand, what is an optimal toll scheme to reach a certain policy objective.

This paper is structured as follows. Based on a concise review of the literature of optimising road pricing with dynamic traffic assignment models, the problem is defined and formulated mathematically, using a limited solution space. A solution approach is then discussed, which results in a solution algorithm with different variants. The variants of the solution algorithm are tested two case studies: a small network with different initial solutions and a bigger network with one initial solution. We describe the setup of these tests and the results in succeeding sections. Finally, conclusions are presented, including possible future improvements of the framework.

1.1 Literature review

The problem of congestion pricing has been studied from different modelling perspectives and under various assumptions: marginal cost pricing / second best pricing, different policy objectives, static / dynamic, fixed / elastic demand, link-based / path-based / zone-based pricing. In this study we did not aim to carry out an extensive literature review on the history of road pricing research. Such a review is for example given in Joksimovic *et al.* (2005). We focussed on dynamic models and bi-level modelling approaches for optimising road pricing measures.

In Yang and Bell (1997) a static elastic demand model with queues is given and a bi-level programming approach is used to select the first best tolling policy that replaces delays with an equivalent level of the tolls. Verhoef (2002) studied static second best pricing with perfect driver information and elastic demand.

Dynamic models with time varying network conditions and link tolls have been addressed. Arnott *et al.* (1990) compares the effect of various pricing policies (uniform, time-varying, and step tolls). Huang and Yang (1996) and Wie and Tobin (1998) developed dynamic first best pricing models for general transportation networks, with the important drawback that application of the model is limited to destination specific tolling. In Joksimovic *et al.* (2005) and Viti *et al.* (2003) the problem of optimal tolling is formulated as a bi-level mathematical program. Supply, i.e. the transport network is modelled as a directed weighted graph, where the weights are a combination of travel time and toll, which may differ in time. Demand is modelled as a given OD-matrix and is input to the problem. In the upper level the policy objective (e.g. minimisation of congestion or total travel time) is formulated as objective function, which depends on the value of the design variables: the space and time differentiated toll levels. Some of the stakeholders' demands (e.g. minimum and maximum price levels) are formulated as constraints. Travel times depend on the amount of traffic in the network. These are determined in the lower level, where some form of a dynamic user equilibrium is assumed. Thus in the upper level tolls are set to minimise for instance travel times, in the lower level the travel times are determined, given the tolls. In order to solve the problem, an optimisation was carried out for a small hypothetical network and a straightforward pricing scheme (constant toll or two different tolls in two time periods, only at one tolling location). The search algorithm that was used is a straightforward exhaustive search algorithm, that can only be applied in very simple networks.

In this paper we follow this approach and we develop an optimisation method that can be used for larger networks, and also allows for more space and time differentiation in toll schemes. Because the computation times involved to evaluate the lower level, more intelligent search algorithms must be used. Furthermore, we consider distance based tolls instead of tolls per passage.

2 PROBLEM DEFINITION

As was mentioned in the introduction besides improving the fairness of the transport system another objective of the road authority for the introduction of tolls might be to improve system performance (for example to minimise average travel time). This is achieved by choosing optimal tolls within realistic constraints and subject to the traffic assignment. The road authority selects feasible values for tolls to optimise its own objective function, while network users face these tolls and adapt their route and departure time decisions to minimise their individual travel cost, resulting in changes in the dynamic flow pattern. In response, the road authority will adapt the tolls, and travellers will respond again.

We now consider an ordered set of predefined prices $P = \{p_1, ..., p_m\}, p_1 < p_2 < ... < p_m$, where $p_1 \geq 0$ and *p* is in €/km. For each time window *t* a price $\pi_{at} \in P$ is assigned to each link *a* in the network. The order of the price in set *P* is defined in the following function: $o(\pi_{at}) = v$ if $\pi_{at} = p_v$. An initial assignment is based on the level of service of the link, e.g. when the flow-capacity ratio is high, the price of the link will also be high. Additionally, the location of the link (rural versus urban) is important. In principle, every link could be assigned its own price. In this paper however we reduced the solution space because each model evaluation is time consuming. To achieve this, the links and time windows are categorised in groups which will have the same toll level. Starting from the initial solution, we try to improve the toll setting further. From a mathematical viewpoint, we chose to use a discrete solution space, because gradient based methods, like steepest descent or Powel’s method, require a lot of computation time. In every iteration a numerical gradient has to be computed and the line search sub problem has to be solved. For more information on these search techniques, see Fletcher (1987).

2.1 Notation

Sets and 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*
$g \in G$	Groups of links
$h \in H$	Groups of time windows

Variables

k_t	Length time window *t* (h)
l_a	Length link *a* (km)
c_a	Capacity link *a* (veh/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)
u_{at}	Average inflow on link *a* during time period *t* (veh/h)
τ_{at}	Average travel time on link *a* during time period *t* (h)
C_{at}	Congestion indicator: equals 1 if $\upsilon_{at} \leq 0,6 s_a$ and equals 0 otherwise
s_a	Free flow speed link *a* (km/h)
υ_{at}	Average speed on link *a* during time period *t* (km/h)
N	Total number of travellers in the network

Decision variable

π_{at}	The link price link *a* during time period *t*: $p_1 = \pi_{\min}, p_n = \pi_{\max}$ (€/km)

2.2 Objective functions

Three different objective functions have been researched. Minimisation of average travel time is used to test the optimisation procedure. The total congestion and total revenue are also investigated. The objective functions are:

$$\min_{\pi_{at}} \sum_a \sum_t (u_{at} \tau_{at} k_t) / N \qquad (1)$$

$$\min_{\pi_{at}} \sum_a \sum_t C_{at} l_a c_a \qquad (2)$$

$$\max_{\pi_{at}} \sum_a \sum_t \sum_p u_{at} \pi_{at} k_t l_a \qquad (3)$$

2.3 Constraints

To reduce the solution space we use link groups and time window groups. All links are assigned to a unique group and within a group all links get the same price. The same holds for the time windows. This is enforced by the following constraints:

$$\pi_{\max}(1-\sum_g \eta_{ag}\eta_{bg}) + \pi_{at} \geq \pi_{bt} \quad \forall\, a, b > a, t$$
$$-\pi_{\max}(1-\sum_g \eta_{ag}\eta_{bg}) + \pi_{at} \leq \pi_{bt} \quad \forall\, a, b > a, t \tag{4}$$

$$\pi_{\max}(1-\sum_h \theta_{th}\theta_{wh}) + \pi_{at} \geq \pi_{aw} \quad \forall \quad a, t, w > t$$
$$-\pi_{\max}(1-\sum_h \theta_{th}\theta_{wh}) + \pi_{at} \leq \pi_{aw} \quad \forall \quad a, t, w > t \tag{5}$$

In this formulation the toll level is not free, but has to be chosen from a limited number of price categories:

$$\pi_{at} \in P = \{p_1, ..., p_m\} \tag{6}$$

In the lower level it is determined how the travellers respond to the tolls that were set in the upper level. It is assumed that users of the system may alter their routes and departure times. This is modelled as a dynamic stochastic user equilibrium (DSUE), where users minimise their individual perceived generalised costs (a weighted sum of toll, travel time, and schedule delays). 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.). This results in the constraint:

$$u_{at} \text{ satisfies } DSEU \quad \forall \quad a, t \tag{7}$$

2.4 Solution space

The mathematical formulation is such that the solution space is discretised in several ways. First in the upper level, the toll level is discretised by stating that the price is an element of *P*, links are divided into link group from a set *G* , and the time is divided into intervals in *H*. Thus we are trying to find optimal values for matrix Π, with elements $\pi_{hg} \in P$. Thus the number of possible solutions for Π is $|P|^{|G|*|H|}$, which is huge (e.g. 5 price categories, 4 time windows, and 3 link groups yield $2.44*10^8$ possible solutions).

3 SOLUTION APPROACH

For the lower level DSUE we have used a macroscopic dynamic equilibrium model (INDY), see Bliemer *et al.*, 2004; Bliemer, 2004), To INDY a departure time choice model was added as described in Bliemer and van Amelsfort (2008), see Figure 1. Input for the model are a network, an OD-matrix, a PAT-profile, a fixed route set, and a toll setting.

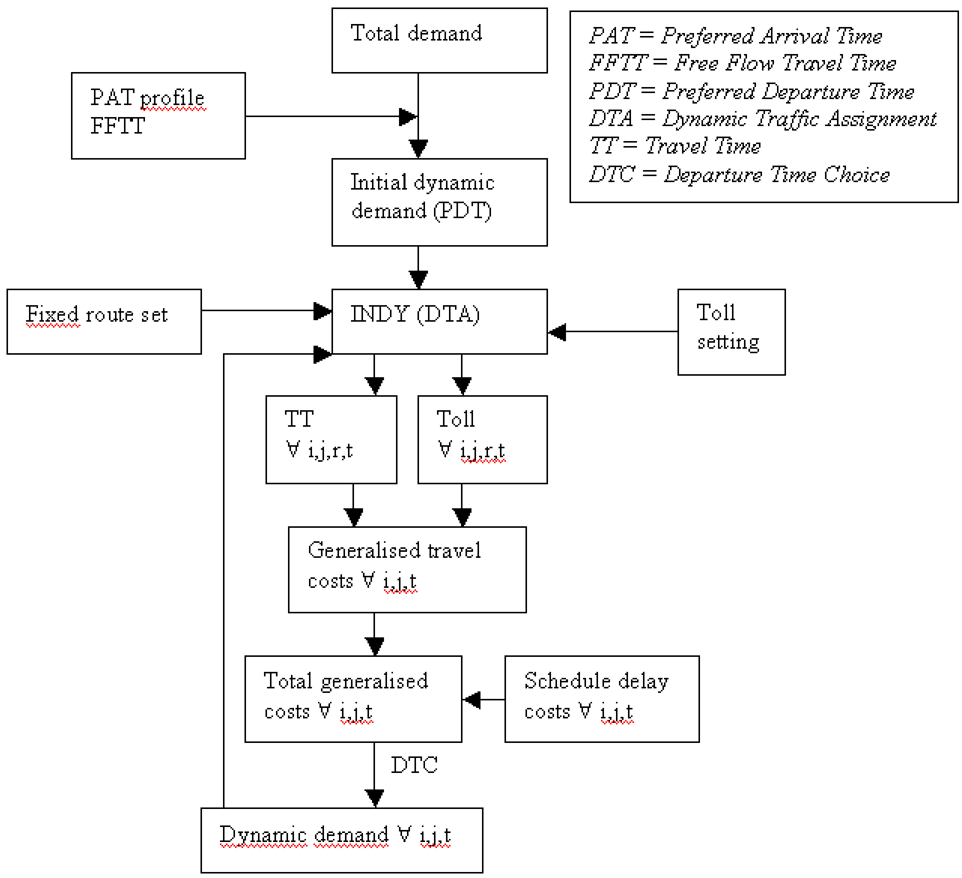

Figure 1: Computation of the effects of a toll setting. The procedure terminates when DSUE is reached.

One run of the lower level model is time consuming and an exhaustive grid search of all possible solutions becomes already infeasible with a only a limited number of price categories, link groups and time windows. As an alternative, a local search algorithm is used, which is called pattern search (Michalewicz and Fogel, 2000). Such an algorithm starts with an initial solution, and considers whether neighbours of this solution give improvement or not. In our case a neighbour is defined as follows:

Definition

Π^1 is called a (h,g)-neighbour of Π^2 if $\pi^1_{fe} = \pi^2_{fe}, \forall (f,e) \neq (h,g)$ and $\left| o(\pi^1_{hg}) - o(\pi^2_{hg}) \right| = 1$ where $o(\pi_{hg}) = v$ if $\pi_{hg} = p_v$, as defined earlier.

Furthermore, Π^1 is the right (h,g)-neighbour of Π^2 if they are (h,g)-neighbours and $o(\pi^1_{hg}) - o(\pi^2_{hg}) = 1$ and the left-(h,g) neighbour if $o(\pi^1_{hg}) - o(\pi^2_{hg}) = -1$

When no neighbour gives an improvement anymore, the algorithm terminates. When the objective function is convex, this is the global minimum. However, in Joksimovic *et al.* (2005) was showed that the objective function can already be non convex in a simple three link network, so it is likely that the objective function is non convex in a general network. The optimal toll setting from the optimisation algorithm will therefore likely be a local minimum. There exist search algorithms that are capable of escaping from a local minimum, like tabu search, simulated annealing and genetic algorithms, see for example Michalewicz and Fogel (2000). However, these algorithms use many function evaluations (model runs on the lower level) which is computational expensive so, we have chosen to concentrate on the pattern search algorithm.

3.1 Variants of pattern search

The design variables of this problem consist of the toll matrix Π. The basic pattern search algorithm in this research is given in Algorithm 1. The objective function value can be seen as a function of the toll setting $\underline{\pi}$: $z(\underline{\pi})$.

n = iteration number, $\underline{\pi}^n$ = toll vector in iteration n, $\underline{\pi}^0$ = initial toll vector, d = dummy variable

Algorithm 1

Initialise: n=1, $\underline{\pi}^1 := \underline{\pi}^0$

FOR h = 1 to |H|

    FOR g = 1 to |G|

        Now suppose (w.l.o.g.) $\pi_{hg}^n = p_v$

        Define $d_{hg} := p_{v+1}$ (if the right (h,g)-neighbour of $\underline{\pi}^n$ exists)

        IF $z(\pi_{11}^n,...,d_{hg},...,\pi_{HG}^n) < z(\underline{\pi}^n)$

        THEN $\pi_{hg}^{n+1} := p_{v+1}$, $n := n+1$

        ELSE $d_{hg} := p_{v-1}$ (if the left (h,g)-neighbour of $\underline{\pi}^n$ exists)

        IF $z(\pi_{11}^n,...,d_{hg},...,\pi_{HG}^n) < z(\underline{\pi}^n)$

        THEN $\pi_{hg}^{n+1} := p_{v-1}$, $n := n+1$

        ELSE $\pi_{hg}^{n+1} := p_v$, $n := n+1$

    END

END

This loop is repeated until no improvement in $z(\underline{\pi})$ occurs anymore.

3.2 The way to select the next variable

When little is known about the shape of the objective function, it is hard to determine the best order in which to select variables. However, the order of the variables can have influence on the results and the speed of the search algorithm, because the order determines the route of the algorithm through the solution space. In Algorithm 1 this order is determined by the structure of the FOR loop. The order can as well make the algorithm terminate in another local minimum. 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 Table 1 an overview is given of the experiments on this topic in this research.

Table 1: Properties of the variants of pattern search tested in this research

Patternsearch	*Way to select next variable*	*When improvement*
P1	Like in algorithm 1	Like in algorithm 1
P2	Change *g* and *h* in the for loops in algorithm 1	Idem as patternsarch1
P3	Change all variables within a group at the same time *i.e.* compute $z(\pi_{11}^n,...,d_{h1},d_{h2}...,d_{hG},...,\pi_{HG}^n)$ or compute $z(\pi_{11}^n,...,d_{1g},d_{2g}...,d_{Hg},...,\pi_{HG}^n)$	Idem as patternsarch1
P4	Randomly select a π_{hg} from the set of variables without a label. When a variable gives no improvement, add a label to it.	Idem as patternsarch1
P5	Idem as patternsearch2.	While $z(\pi_{11}^n,...,d_{hg},...,\pi_{HG}^n) < z(\underline{\pi}^n)$, define $d_{hg} = p_{v+2}$ etc. or $d_{hg} = p_{v-2}$ etc.
P6	Idem as Patternsearch1, but after the first execution of the FOR loop, skip variables which did not give improvement in the former FOR loop.	Save the new value of the variable, add variable to the improvement list, and select the next variable.
Former iterations are stored in each variant		

3.3 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 (P5, see Table 1). This strategy has the danger that it ignores other directions in which more improvement is possible. 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 (P1 to P4). Finally, in patternsearch6 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.

4 CASE STUDY 1

The modelling framework is first applied to a small case study, in order to gain information on the shape of the objective function and the behaviour of the variants of the search algorithm.

The test network based on the real network of the town of Delft in the western part of the Netherlands. It contains two main highways: the A13 and the A4. The rest of the network consists of urban roads (see Figure 2). The network further consists of 12 centroids, 137 links and 90 nodes.

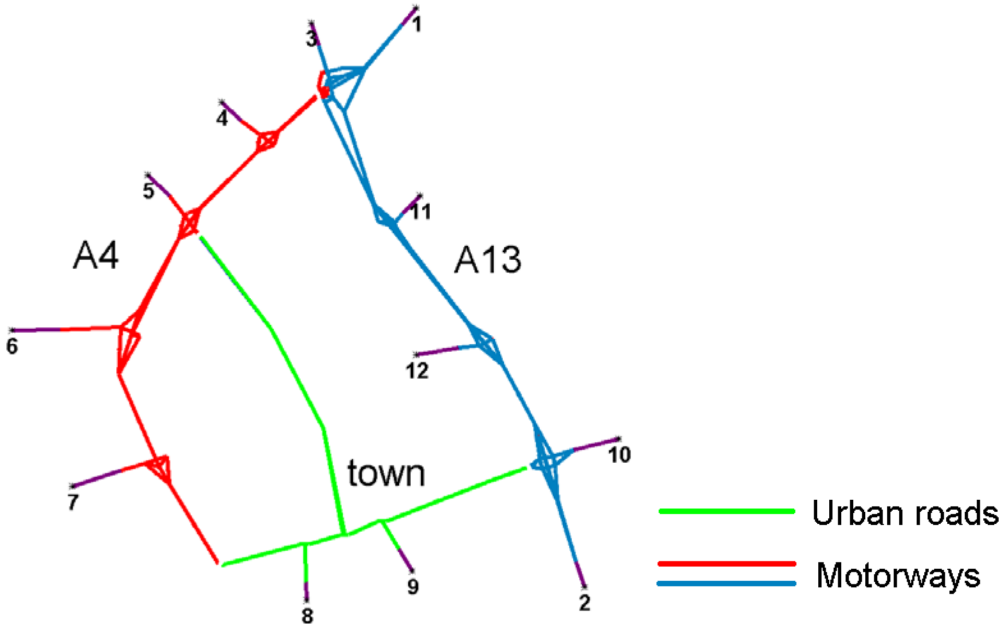

Figure 2: The first test network (Delft).

The time period is an AM peak, from 7:00 AM to 9:00 AM. In order to include some warm-up and cool down time to fill and empty the network, the modelled time period is from 6:00 AM to 10:00 AM. Some fixed preferred arrival time profile is used, corresponding to this peak period. In the situation without tolls ($\Pi^{0,0}$ is a matrix filled with zero's), the average travel time in the network is $z(\Pi^{0,0}) = 28.1\,\text{min}$. Most traffic travels along the A13: at 7.30AM queues start to form there in front of on- and off-ramps. Around 8PM smaller queues develop on the A4 and on the urban roads. So a clear distinction exists between busy highways (A13), quiet highways (A4), and other roads (town).

Now the sets *H*, *G*, and *P* are defined. In this test network it is chosen to create three link groups: $G = \{town, \text{A13}, \text{A4}\}$. The morning peak is divided in 4 time-intervals of each 30 minutes, so $H = \{7:00-7:30, 7:30-8:00, 8:00-8:30, 8:30-9:00\}$. The warm-up and cool-down period have no toll. Then five price categories ranging from €0.00 per km to €0.20 per km are defined, with a step size of €0.05, so $P = \{0.00, 0.05, 0.10, 0.15, 0.20\}$.

4.1 The initial solution

A carefully chosen, initial solution can strongly contribute to the fast achievement of a good solution. In Table 2 three different values for Π^0 are presented. Most analyses in this research have been executed with $\Pi^{0,1}$. This initial toll vector is based on the reference run, with busy roads getting a higher price. The other two initial solutions are chosen such that they differ significantly from $\Pi^{0,1}$.

Table 2: Three initial toll solutions

	Initial solution 1: $\Pi^{0,1}$ $z(\Pi^{0,1}) = 25.25\,\text{min}$				*Initial solution 2:* $\Pi^{0,2}$ $z(\Pi^{0,2}) = 27.83\,\text{min}$				*Initial solution 3:* $\Pi^{0,3}$ $z(\Pi^{0,3}) = 31.95\,\text{min}$		
Time period	*Toll level (€/km)*				*Toll level (€/km)*				*Toll level (€/km)*		
	town	*A13*	*A4*		*town*	*A13*	*A4*		*town*	*A13*	*A4*
6:00-7:00	0	0	0		0	0	0		0	0	0
7:00-7:30	0	0	0		0.20	0.20	0.20		0	0.20	0
7:30-8:00	0.20	0.20	0.20		0.20	0.20	0.20		0.20	0	0.20
8:00-8:30	0.15	0.20	0.15		0.20	0.20	0.20		0	0.20	0
8:30-9:00	0.05	0.10	0.05		0.20	0.20	0.20		0.20	0	0.20
9:00-10:00	0	0	0		0	0	0		0	0	0

4.2 Experimental results

In the following sections the results of the numerical experiments are presented. First, all 6 variants of the pattern search algorithm are tested with initial solution $\Pi^{0,1}$. The three best variants are selected, and their behaviour using the other initial solutions is then tested. Finally the effect on two different objective functions is investigated.

Table 3: Results achieved with the different pattern search algorithms using initial solution $\Pi^{0,1}$

Search algorithm	Π^*	$z(\Pi_1^*)$ *(min)*	*Number of iterations to termination of algorithm*
P1	Π_5^*	23.11	80
P2	Π_4^*	23.07	95
P3	Π_7^*	23.34	38
P4-1	Π_2^*	22.90	74
P4-2	Π_{10}^*	24.22	22
P4-3	Π_{10}^*	24.22	37
P4-4	Π_{10}^*	24.22	51
P5	Π_{10}^*	24.22	42
P6	Π_3^*	22.90	65

4.3 Performance of the variants of pattern search

In Table 3 the results of all 6 variants of pattern search with $\Pi^{0,1}$ are presented. P4 has a random component, so it is executed 4 times with different random seeds (sub-variants P4-1 – P4-4). In this section and in the next section 10 different local minima were found (Π_1^* to Π_{10}^*), so indeed the objective function is not convex. The best objective function value is an average travel time of 22.90 minutes, compared to $z(\Pi^{0,1}) = 25.25\,\text{min}$. This value is achieved by two different search algorithms at two different local minima Π_2^* and Π_3^*. Both computation time and objective function value are used to assess the variants. The best result is achieved by P6: this variant only uses 65 iterations to find the best value. P4 also achieved this value in P4-1, but in P4-2 to P4-4 a much worse local minimum has been found, so overall this is not a good variant. P1 and P2 achieved a little worse objective function value, but used considerably more iterations to reach that value, so the performance is worse. P3 has a little worse objective function value again, but uses less iterations to reach this value. P5 does not achieve a good average travel time value in this case and it uses many iterations, so it is not a good variant. The development of the average travel time throughout iterations when executing the best three variant (P1, P3, and P6) is compared in Figure 3, which illustrates the performance differences of these algorithms.

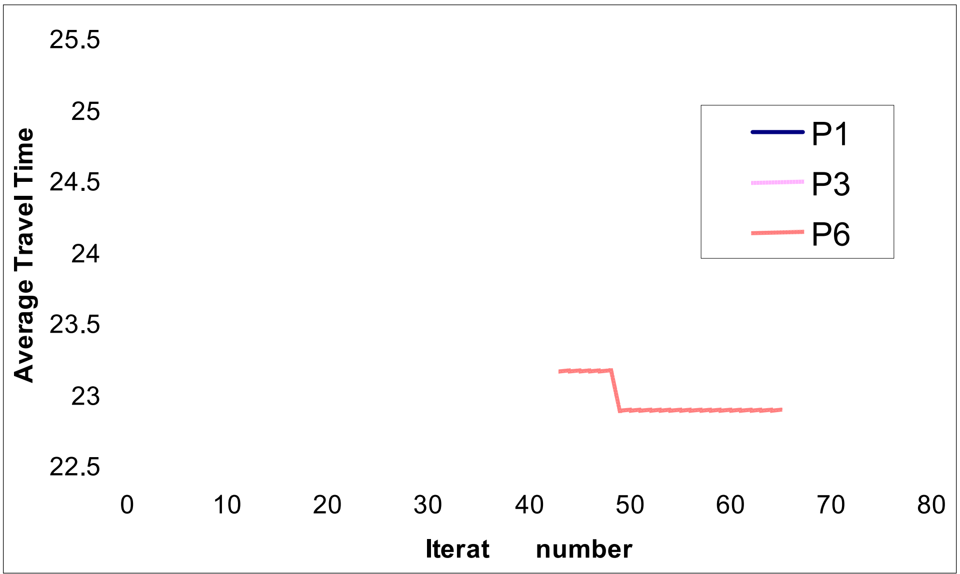

Figure 3: The development of the average travel time value throughout iterations, using the pattern search algorithms P1, P3 and P6.

4.4 Different initial solutions

Since the objective function is not convex and because we use a local search algorithm like pattern, the chance of ending up in a local minimum is high. The search algorithm variants are therefore tested with 2 other initial solutions. In Table 4 the results are presented and it can be concluded that all three variants perform quit good. In fact the impact of another initial solution on the end solution of the objective function is relatively small. Yet, this is also the result of the flat shape of the objective function since each combination of initial solution and variant of pattern search results in another end-solution but the values of the objective function are comparable. Thus, the objective function is such that many local minima (toll-settings) exist, that produce almost equal average travel times. The only difference is the number of iterations that is required. Obviously the worse the initial solution is the longer it takes to reach an optimum. Further it appears that variant P3 needs the least amount of iterations for all initial solutions

Table 4: The effect of different initial solutions

Search algorithm	*Initial toll solution*	*Resulting toll setting*	*Objective function value (min)*	*Number of iterations to termination of algorithm*
P1	$\Pi^{0,1}$	Π_5^*	23.11	80
P1	$\Pi^{0,2}$	Π_8^*	23.46	125
P1	$\Pi^{0,3}$	Π_6^*	23.30	95
P3	$\Pi^{0,1}$	Π_7^*	23.34	38
P3	$\Pi^{0,2}$	Π_5^*	23.11	74
P3	$\Pi^{0,3}$	Π_9^*	23.68	91
P6	$\Pi^{0,1}$	Π_3^*	22.90	65
P6	$\Pi^{0,2}$	Π_1^*	22.72	139
P6	$\Pi^{0,3}$	Π_6^*	23.30	110

4.5 Differences between local minima

As mentioned earlier, 10 different local minima were found (Π_1^* to Π_{10}^*). In every resulting toll setting, a similar structure can be observed, which follows the peak in the traffic demand: the tolls start low, then increase, and finally decrease again. In each of these local minima, $\pi_{22} = 0.20$, so here it is clear that the toll should be on the maximum level. For the other variables different combinations occur, within the mentioned rough structure. Furthermore, in most local minima the toll values in link group 'town' are lower than in the other two link groups.

In order to illustrate these observations, Table 5 shows three examples of these local minima: the best found solution Π_1^*, a solution with relatively low toll values, Π_3^*, and a solution with a relatively bad average travel time value, Π_{10}^*, which quite differs from Π_1^* and is close to initial solution $\Pi^{0,1}$.

Table 5: Three local minima

	Π_1^* $z(\Pi_1^*) = 22.72\,\text{min}$				Π_3^* $z(\Pi_3^*) = 22.90\,\text{min}$				Π_{10}^* $z(\Pi_{10}^*) = 24.22\,\text{min}$		
Time period	*Toll level (€/km)*				*Toll level (€/km)*				*Toll level (€/km)*		
	town	*A13*	*A4*		*town*	*A13*	*A4*		*town*	*A13*	*A4*
6:00-7:00	0	0	0		0	0	0		0	0	0
7:00-7:30	0	0	0.05		0	0	0		0	0	0
7:30-8:00	0.15	0.20	0.15		0.20	0.20	0.10		0.20	0.20	0.20
8:00-8:30	0	0.05	0.20		0	0.15	0.10		0.05	0.20	0.15
8:30-9:00	0	0.10	0.20		0	0.10	0.10		0.05	0.15	0.05
9:00-10:00	0	0	0		0	0	0		0	0	0

4.6 Effects on total revenue and total congestion

Until now all tests were performed where the object function was the average travel time. Earlier other objective functions were defined, i.e. the 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 total congestion level also varies differently than the average travel time. 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. Optimisation with respect to another objective function would thus result in different solutions, as could be expected.

5 CASE STUDY 2

After testing the optimisation framework in case study 1, we apply the resulting framework on a more realistic network to test the practical feasibility of the framework.

The network includes the city of The Hague and the two surrounding towns of Delft and Zoetermeer. It contains several motorways, urban roads, and rural roads (see Figure 4). The network further consists of 168 centroids, 1891 links and 1133 nodes.

The same time period is modelled as in case 1 [6:00AM – 10:00AM] and again a fixed preferred arrival time profile is used. The preferred arrival time profiles were calibrated by comparing historical traffic counts and resulting equilibrium assignment results. In the situation without tolls, the average travel time in the network is $z(\Pi^{0,0}) = 17.5\,\text{min}$. Around 7.30 AM queues start to form on motorways running into the city of the Hague. Around 8PM smaller queues develop on several locations on other roads.

Again the sets *H*, *G*, and *P* are defined. Because one iteration needs more computation time here, only two link groups and only three time intervals are defined. $G = \{urban, other\}$, so the group *other* contains rural roads and motorways. 3 time-intervals of each 60 minutes are defined, so $H = \{6:30-7:30, 7:30-8:30, 8:30-9:30\}$. The periods before and after this period have no toll. Further, three price categories instead of five are defined: $P = \{0.00, 0.10, 0.20\}$.

Figure 4: The second test network (Den Haag, Zoetermeer, Delft)

5.1 The initial solution

In Table 6 the initial solution for case 2 is shown. Again, this initial toll vector is based on the reference run, with busy link group and time combinations getting a higher price.

Table 6: The initial solution and the resulting solution

	Initial solution: Π^0 $z(\Pi^0)=17.0\,\text{min}$			*Resulting solution:* Π^* $z(\Pi^*)=16.3\,\text{min}$	
Time period	*Toll level (€/km)*			*Toll level (€/km)*	
	urban	*other*		*urban*	*other*
6:00-6:30	0	0		0	0
6:30-7:30	0	0.20		0	0.10
7:30-8:30	0.10	0.20		0.10	0.10
8:30-9:30	0.10	0.10		0.20	0.10
9:30-10:00	0	0		0	0

5.2 Experimental results

Since we are interested to test the optimisation on larger networks, only the variants P1 and P2 of pattern search are used. Both algorithms end up in the same solution $z(\Pi^*)=16.3\,\text{min}$ (see Table 6), so the relative improvement w.r.t. no tolls is 6.7%. Both algorithms use approximately the same number of iterations to reach this value: P1 uses 22 iterations and P2 uses 24 iterations. The number of iterations is approximately 3 times smaller than in case 1, because the number of variables is smaller. An optimisation in case 1 with only 6 variables resulted in 24 iterations as well. One iteration takes 12 times more computation time than in case 1, this is directly caused by the bigger network. So a larger network does not imply more iterations in the optimisation algorithm, it only implies longer computation times in the lower level model. In total, the computation time in case 2 is about 4 times longer than in case 1.

The network of case 2 is still not a very big network. When this framework is applied to bigger networks, it is not computationally feasible anymore in the current setting. One possibility is to further reduce the number of link groups and time intervals, which makes the framework less useful, because little differentiation is left. Another possibility is to introduce parallel computing. In that case the search algorithm has to be slightly adapted, in order to evaluate different toll settings in parallel on different computers, which will reduce computation times.

6 CONCLUSIONS

The optimal toll level design problem is formulated as a bi-level mathematical program and an approximation approach is presented for finding the optimal toll levels in space and time differentiated, link based pricing, with the objective to minimise average travel time. Different variants of the search algorithm have been compared and the effect of a different initial solution is treated.

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. The risk to end up in a bad local minimum is small, because different initial solutions all gave acceptable solutions.

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.

This paper showed that it is possible to apply a pattern search algorithm in this context on a large scale network. When this framework is applied, it is yet only computationally feasible with little space and time differentiation and a few price categories, so this definition is important. Furthermore, the definition of the objective function strongly determines the resulting toll setting, so it should be carefully considered. The second, realistic case study showed that a positive effect of road pricing on average travel time exists, but it is not big. In the case studies, elastic demand was not included, but this will in reality increase the effect of road pricing. It is likely that general findings in this research also apply for other networks, though this is not shown in this paper. Future application of this framework to other networks is recommended. When this framework is applied to bigger networks, further improvements in the lower level are needed, as many possibilities for parallelisation are still unused. The behaviour of other search algorithms in this context like simulated annealing is as well an interesting topic for future research.

7 REFERENCES

Arnott, R., Palma, A. de, and Lindsey, R. (1990) Departure Time and Route Choice for the Morning Commute. **Transportation Research Part B, 24** (3) 209–228.

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

Bliemer, M. C. J., van Amelsfort, D. H. (2008) Rewarding instead of charging road users: a model case study investigating effects on traffic conditions. **Proceedings** *of the 3rd Kuhmo-Nectar Summer School and Conference*, Amsterdam, The Netherlands.

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

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

Huang, H. J., Yang, H. (1996) Optimal Variable Road-Use Pricing on a Congested Network of Parallel Routes with Elastic Demand. **Proceedings** *of the 13th International Symposium on the Theory of Traffic Flow and Transportation* (J.-B. Lesort, ed.), Elsevier Science Ltd, Lyon, France, 479–500.

Joksimovic, D., Bliemer, M. J. C., and Bovy, P. H. L. (2005) Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. **Transportation Research Record, Journal of the Transportation Research Board, No. 1923**, Washington, D.C., 61-72.

Michalewicz, Z., Fogel, D. B. (2000) *How to solve it: Modern Heuristics.* Springer, Berlin, Germany.

Verhoef, E. T. (2002) Second-Best Congestion Pricing in General Static Transportation Networks with Elastic Demands. **Regional Science and Urban Economics, 32** 281–310.

Viti, F., Catalano, S. F., Li, M., Lindveld, C., and Zuylen, H. J. van (2003) Optimisation Problem with Dynamic Route–Departure Time Choice and Pricing. **Presented** *at 82nd Annual Meeting of the Transportation Research Board*, Washington, D.C.

Wie, B., Tobin, R. L. (1998) Dynamic Congestion Pricing Models for General Traffic Networks. **Transportation Record Part B, 32** (5) 313–327.

Yang, H., Bell, M. G. H. (1997) Traffic Restraint, Road Pricing, and Network Equilibrium. **Transportation Research Part B, 32** (1) 303–314.

