On Dynamic Optimal Toll Design Problem with Traffic-flow Dependent Tolls and Drivers' Joint Route and Departure Time Choices

Kateřina Staňková

***Abstract*— In this paper the dynamic optimal toll design problem as a game of the Stackelberg type is investigated, with the road authority as leader and drivers on the road network as followers. The road authority sets dynamic traffic-flow dependent tolls on some links in order to minimize its objective function, while the drivers choose their routes and departure times so as to minimize their own perceived travel costs, which include a travel time component, tolls, and penalties for deviation from their preferred arrival and departure times. The drivers' behavior is modeled using a dynamic user equilibrium model. We define the problem in a general form so that a wide class of objective functions for the road authority and a wide class of the dynamic traffic equilibria can be employed. We also discuss the problem properties.**

The problem to find optimal traffic-flow dependent tolls in a general setting introduced in this paper is NP-hard. One of the ways of tackling such problems is to use advanced heuristic methods. In this paper a neurosimulation-based approach is proposed, using the neurosimulator FAUN 1.0.

The proposed solution method is illustrated on case studies with the so-called Chen network.

I. Introduction & Literature Overview

Traffic congestion has become a big problem, especially in heavily populated metropolitan areas. One of the ways to reduce traffic congestion is to impose appropriate tolls, in a road pricing scheme. Road pricing has received a lot of attention in research ([1], [2], [3], [4]) and practice ([5]).

The *dynamic optimal toll design problem,* discussed in this paper ([2], [6], [7]) is a problem of the Stackelberg type ([8], [9]), applied to the traffic environment with a road authority as a leader and drivers (also referred to as travelers or users) as followers. The aim is to minimize the objective function of the road authority by imposing appropriate tolls on some links, while the drivers minimize their perceived travel costs by their travel choices, taking the tolls into account. In an equilibrium state, a *dynamic user equilibrium* applies ([10], [11]).

A number of studies have focused on the so-called *first-best* pricing, in which all the links in the network can be tolled ([12], [13], [14]). In our research we consider the so-called *second-best* pricing ([1], [15]), in which only a proper subset of all links can be tolled, as this concept seems to be more suitable for practical applications.

Most publications dealing with the optimal toll design problem with second-best pricing ([6], [1]) aim to find the optimal link tolls as values independent of the traffic flows in the network (uniform or time-varying toll). We have recently introduced ([16], [17]) the optimal toll design problem, in which the link tolls are computed as functions of link or route traffic flows in the network. This fits within the theoretical framework of the so-called *inverse Stackelberg problems* ([18], [19]). We have shown that the traffic-flow dependent toll can improve the traffic system performance remarkably, while it can never perform worse than the traffic-flow invariant toll.

In our previous work we have assumed that travelers decide only about their routes, while their departure times are fixed. This paper introduces an extension of our recent research into the situation in which the drivers choose their departure times, too. Moreover, we generalize the problem so that a wide range of objective functions for the road authority and user equilibrium models for the drivers can be considered. We present some of the properties of this generalized problem. A neural-networks based approximation algorithm is used to solve the problem, as the problem is NP-hard. The proposed algorithm can be used for general networks and for a wide range of pricing problems. The performance of the algorithm is presented on the so-called *Chen network* ([20]).

This paper is organized as follows: In Section II, the optimal toll design problem is defined and its properties are discussed. In Section III, the algorithm that is used for solving the problem is presented. Its advantages and pitfalls are discussed. In Section IV we perform experiments on the Chen network. Conclusions and possible future research are given in Section V.

II. THE DYNAMIC OPTIMAL TOLL DESIGN PROBLEM

A. Preliminaries

Let $G = (\mathscr{N}, \mathscr{A})$ be a strongly connected road network, where $\mathscr{N}$ and $\mathscr{A}$ are finite nonempty sets of nodes and directed links, respectively. Let $\mathscr{T} \subseteq \mathscr{A}$ be the set of tollable links. There is a finite, nonempty set of origin-destination pairs $\mathscr{RS} \subset \mathscr{N} \times \mathscr{N}$. Let $\mathscr{K} = \{1, 2, \ldots, |\mathscr{K}|\}$ be a time index set. Here each $k \in \mathscr{K}$ refers to

- the interval $[(k-1.5)\Delta, (k-0.5)\Delta)$ if $k \geq 2$,
- the interval $[0, 0.5\Delta)$ if $k = 1$,

where Δ [h] is the length of each time interval.

For an ordered pair of nodes $(r,s) \in \mathscr{RS}$, where r is an origin and s is a destination, there is a positive number of drivers traveling from r to s, the so-called travel demand $d^{(r,s)}$ [veh], during the time interval $[0, (|\mathscr{K}|-0.5)\Delta]$.

Let $\mathscr{P}$ be the set of all simple paths in the network and let $\mathscr{P}^{(r,s)} \subset \mathscr{P}$ be the set of simple paths between from r to s. Each path is formed by one or more directed links. The

K. Staňková is with NeCS (INRIA Grenoble Rhône-Alpes / Gipsa-lab), Grenoble, France. katerina.stankova@inrialpes.fr.

route flow on path $p \in \mathscr{P}$ during the k-th time interval will be denoted by $f_p^{(k)}$ ([veh/h]), the link flow on the link a during the k-th time interval will be denoted by $q_a^{(k)}$ ([veh/h]).

Let $c_p^{(k)}$ ([euro]) denote the route travel costs on the route $p \in \mathscr{P}$ for a driver entering this route during the k-th time interval. Let $\varsigma_a^{(k)}$ ([euro]) denote the average link travel cost on a link a during the k-th time interval.

The traffic, times, costs, flows, and dynamic tolls are assumed to be additive. See [18] for a detailed description via the so-called *dynamic route-arc incidence indicator*. [1]

For each route $p \in \mathscr{P}^{(r,s)}$ the dynamic route travel cost $c_p^{(k)}$ for the travelers entering the route during the k-th time interval is defined as follows:

$$
\begin{aligned}
c_p^{(k)} &= \alpha_1 \theta_p^{(k)} + \theta_p^{(k)} + \alpha_2 \left(k - \hat{k}_{(r,s),\text{dep}}\right) \\
&\quad + \alpha_3 \left(k + \tau_p^{(k)} - \hat{k}_{(r,s),\text{arr}}\right). \qquad (1)
\end{aligned}
$$

where $t_p^{(k)}$ is the actual route travel time, $\theta_p^{(k)}$ is the actual route toll, $\hat{k}_{(r,s),\text{dep}}$ is the preferred departure time for a driver traveling from origin r to destination s, starting his/her trip during the k-th time interval, $\hat{k}_{(r,s),\text{arr}}$ is the preferred arrival time for a driver traveling from origin r to destination s, starting his/her trip during the k-th time interval; coefficients α_1, α_2, and α_3 ([euro/h]) are the value of time, the penalty for deviating from the preferred departure time, and the penalty for deviating from the preferred arrival time, respectively.

Let q, t, and ς denote the column vectors of the link flows, the link travel times, and the link travel costs for all links and all time intervals and let f, τ, and c denote the column vectors of the route flows, the route travel times, and the route travel costs for all link and all time intervals (For more details, see [18]).

For each link from $\mathscr{T}$ and each time interval a traffic-flow dependent toll can be imposed. The traffic-flow dependent toll on link $a \in \mathscr{T}$ will be denoted by $\theta_a^{(k)}(\cdot)$. This toll will be defined for each k-th time interval as a polynomial function of link flow for the same time interval and on the same link, i.e.,

$$
\theta_a^{(k)}\left(q_a^{(k)}\right) = \sum_{m=0}^{M} w_a^{(m),(k)} \left(q_a^{(k)}\right)^m, \qquad (2)
$$

where

$$
w_a^{(m),(k)} = \begin{cases} 0 & \text{for} \quad a \in \mathscr{A} \setminus \mathscr{T}, \\ \in \mathbb{R} & \text{for} \quad a \in \mathscr{T}, \end{cases}
$$

with $M \in \mathbb{N}_0$. By definition,[2]

$$
\theta_a^{(k)}\left(q_a^{(k)}\right) \begin{cases} = 0 & \text{for} \quad a \in \mathscr{A} \setminus \mathscr{T}, \\ \geq 0 & \text{for} \quad a \in \mathscr{T}. \end{cases} \qquad (3)
$$

Let θ be a vector of link toll functions for all links and all time intervals. Similarly, the coefficient vector will be defined as w. Let $w \in W$, where W is a compact set, i.e., $w_a^{(m),(k)} \in \left[w^{(m),\min}, w^{(m),\max}\right]$ for all $a \in \mathscr{T}$ and for all m, $w^{(m),\min}, w^{(m),\max} \in \mathbb{R}$, $w^{(m),\min} < w^{(m),\max}$. Note that while coefficients $w_a^{(m),(k)}$ can be negative, the toll has to be nonnegative on all links, as stated in (3).

With $M = 0$ in equation (2) the toll level becomes time-varying, but not directly dependent on traffic flow.

B. Drivers' behavior – dynamic traffic assignment

The so-called *dynamic traffic assignment (DTA) model* describes user-optimal flows over a network in which each driver chooses his/her preferred route and his/her preferred departure time from origin to destination, based on the time-varying conditions in the network.

The standard DTA models consist of a *dynamic travel choice* (DTC) model and a *dynamic network loading* (DNL) model ([22], [20]).

The DTC contains a path choice model in which all travelers are distributed on all available routes so that some kind of dynamic user equilibrium is achieved.

In the problem of traffic assignment with given total traffic demand, each driver chooses both a departure time and a certain route from his/her origin to his/her destination in order to minimize his/her perceived travel costs. We assume that some *Dynamic traffic equilibrium* applies.[3]

The dynamic network loading (DNL) model is formulated as a system of equations expressing link dynamics, flow conservation, flow propagation, and boundary constraints. The DNL model simulates the progression of the route flows on the network, yielding dynamic link flows, link volumes, and link travel times developing over time. The DNL model used in this paper is adapted from [22] and can be found in, e.g., [21].

C. The problem formulation

The goal of the road authority is to choose an optimal w^*, minimizing his/her objective function. We will denote this function by $Z = Z(q(w), w)$. The problem to be dealt with can be formulated as follows:

$$
(P) \quad \begin{cases} \text{Find} \\ \quad w^* = \arg\min_{w \in W} Z(q(w), w), \\ \text{subject to } q \in DUE(w), \end{cases}
$$

where (2) and (3) hold.

We assume that the function Z is defined for each w, $q(w)$ without any additional requirements. Therefore, a wide range of functions can be used as Z.[4]

The expression $q \in DUE(w)$ reads as "link flow vector q is a result of a used dynamic user equilibrium (DUE) model when a polynomial toll function with coefficient vector w is used." . Note that the function Z in the problem (P) is not specified, we only assume that it is a function of w of $q(w)$.

The "standard" Stackelberg problem would be defined as a subproblem of (P), with $M = 0$.

[1] Since some of the variables have to be rounded off, additional discussion about the consistency of these equations is needed. Such a discussion can be found in, e.g., [21].

[2] More advanced toll functions can include, e.g., traffic flows from previous time periods, but we are looking for a very simple scheme improving the system performance. Therefore we will restrict ourself to toll functions in the form (2). See [18] for discussions on this topic.

[3] At this moment no further specification of such an equilibrium is required.

[4] Typically, the function Z is defined as the total travel time of the system, negative of the total toll revenue, or negative of the surplus of the system.

D. General problem properties

Note that problem (P) is a nonlinear programming problem. Also, the problem (P) has at least one solution if the DUE constraint represents a compact set of $(w,q(w))$. If for any given w the set $DUE(w)$ is a singleton, $w \to q$ is a one-to-one mapping. In this case, the continuity of q with respect to w will guarantee that the constrained set of (P) is closed, which implies the solution existence of (P) since q and w are bounded. In general, $DUE(w)$ may have multiple solutions in terms of q and thus $DUE(w)$ may not be a singleton. In this case, $DUE(w)$ is a point-to-set mapping of w ([23]). The solution existence of (P) will depend on the compactness of the graph $DUE(w)$, defined as

$$\Psi(w,q) = \{(w,q) \,|\, q \in DUE(w), \forall w \in W\}. \tag{4}$$

Theorem 2.1: The problem (P) has at least one solution if the following conditions are satisfied:

i. The set $DUE(w)$ is nonempty and compact for $\forall w \in W$.

ii. Let $w, \overline{w} \in W$ and let $q \in DUE(w)$, $\overline{q} \in DUE(\overline{w})$. For each $\varepsilon > 0$, there exists $\delta > 0$ such that if $||w-\overline{w}|| < \delta$, then

$$\max_{\forall q \in DUE(w)} \min_{\forall \overline{q} \in DUE(w)} ||q-\overline{q}|| < \varepsilon.$$

iii. The link travel cost functions on all links are continuous functions of the link flows on the same links.

Proof: Let $R(0,\varepsilon)$ be an open ball with radius ε. Then $\mathscr{Y} \overset{\text{def}}{=} DUE(w) + R(0,\varepsilon)$ is an open set containing $DUE(w)$. Let us define an other open set $\mathscr{X} \overset{def}{=} \{w : ||w-\overline{w}|| < \delta\}$ containing $\overline{w}$. According to condition *ii.* in Theorem 2.1, for any $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\max_{\forall q \in DUE(w)} \min_{\forall \overline{q} \in DUE(w)} ||q-\overline{q}|| < \varepsilon,$$

which is equivalent to $\cup_{w \in \mathscr{X}} DUE(w) \subseteq \mathscr{Y}$. Thus, under *ii.*, the point-to-set mapping of $DUE(w)$ is upper-semicontinuous. Together with condition *i.* it implies that the point-to-set mapping $DUE(w)$ is closed on set W. Thus the graph $\Psi(w,q)$ is closed. Also, under *i.*, $DUE(w)$ is bounded for any $w \in W$. Since W is a bounded set, the graph $\Psi(w,q)$ is bounded as well. Thus, graph $w \in W$ is compact. Together with *iii.* and the fact that W is compact, we can conclude that (P) has at least one solution, since it is a nonlinear programming problem with a continuous objective function defined on a compact set. ■

Theorem 2.2: Problem (P) is strongly NP-hard.

Proof: See [18]. ■

Note that Theorems 2.1 and 2.2 extend theory about "standard" Stackelberg games (with $M = 0$), which can be found in, e.g., [24], [25]. Additionally, the problem (P) can have a nonunique solution (See [18] for details).

III. PROBLEM SOLUTION

The solution method for (P) that we propose is a combination of a neural networks approach for the upper level of the problem (finding optimal traffic-flow dependent toll for the road authority) and a method of sufficient averages (MSA) for the lower level of the problem (finding optimal traffic flows of the travelers according to the chosen dynamic user equilibrium). The concept of neural networks is closely related to the concept of supervised learning, which will be explained below. In Figure 1 the flow diagram of the solution process is depicted.

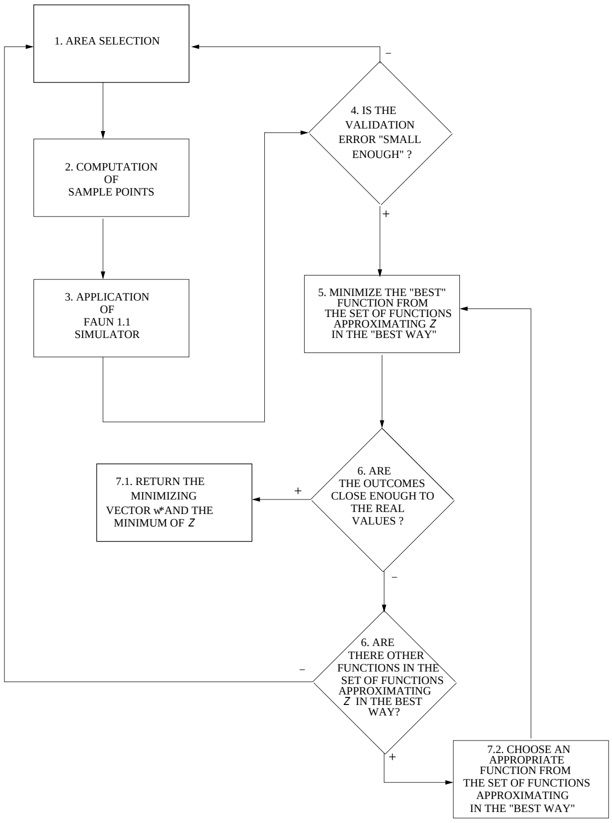

Fig. 1. Flow diagram of the solution process

A. Supervised learning

Let function $g : \mathbb{R}^n \to \mathbb{R}^m$ assign a vector $\mathbf{y}^i \in \mathbb{R}^m$ to each vector $\mathbf{x}^i \in \mathbb{R}^n$, i.e., $\mathbf{y}^i = g(\mathbf{x}^i)$. We will refer to the pair $(\mathbf{x}^i, \mathbf{y}^i)$ as the i-th *pattern* of the function g. The vector $\mathbf{x}^i$ will be called the *input* vector (of g) and the vector $\mathbf{y}^i$ will be called the *output* vector (of g). Supervised learning is a way to approximate the function g given a set of o patterns [26].

An artificial neural network (ANN) can be thought of as a simple mathematical formula with parameters called weights [26]. The result of supervised learning is an approximation function g^{app} with an appropriately chosen vector of weights $\mathbf{s}$. The goal of supervised learning with ANN is therefore to find a function $g^{\text{app}} : \mathbb{R}^n \to \mathbb{R}^m$, that approximates the function g in the best way. Moreover, it is required that g^{app} has derivatives of all finite orders in the components of $\mathbf{x}$.

Several criteria can be used to validate whether the function g^{app} is "close enough" to g. In our approach the so-called *validation error* for each pattern $(\mathbf{x}^i, \mathbf{y}^i)$, $i = 1, 2, \ldots, o$, is minimized.

The set of o patterns is divided into a set of t training patterns and a set of $o-t$ validation patterns. For a given vector of weights $\mathbf{s}$ the training and the validation errors are calculated by

$$\begin{aligned}\varepsilon_t(\mathbf{s}) &\stackrel{\text{def}}{=} \frac{1}{2}\sum_{i=1}^{t}\sum_{k=1}^{m}(g_k^{\text{app}}(\mathbf{x}^i;\mathbf{s})-y_k^i)^2,\\ \varepsilon_v(\mathbf{s}) &\stackrel{\text{def}}{=} \frac{1}{2}\sum_{i=t+1}^{o}\sum_{k=1}^{m}(g_k^{\text{app}}(\mathbf{x}^i;\mathbf{s})-y_k^i)^2,\end{aligned} \tag{5}$$

where g_k^{app} and y_k^i, $k=1,2,\ldots,m$, refer to the k-th entry of g^{app} and $\mathbf{y}^i$, respectively. The elements of $\mathbf{s}$ are optimized only for t training patterns, while the validation patterns are used to prevent overtraining.

An ANN is trained iteratively, i.e., ε_t is decreased by adaption of $\mathbf{s}$, until ε_v increases for two consecutive iterations (prevention of overtraining). Note that the training stops before a local minimum of ε_t is reached. Weight upgrades $\mathbf{s}^{\text{iter}+1}-\mathbf{s}^{\text{iter}}$ can be calculated with any minimization algorithm, e.g., a first derivative method such as steepest descent, or a second derivative method such as Newton's method. For the first derivative methods the iterative sequence

$$\mathbf{s}^{\text{iter}+1}=\mathbf{s}^{\text{iter}}+\eta\left(\varepsilon_t\left(\mathbf{s}^{\text{iter}}\right),\nabla_{\mathbf{s}}\varepsilon_t\left(\mathbf{s}^{\text{iter}}\right)\right)\Delta\mathbf{s}\left(\varepsilon_t(\mathbf{s}^{\text{iter}}),\nabla_{\mathbf{s}}\varepsilon_t\left(\mathbf{s}^{\text{iter}}\right)\right), \tag{6}$$

with the search direction $\Delta\mathbf{s}$ and with step length η takes place. For the neurosimulation the FAUN[5] 1.0 simulator is used ([26]). Numerical methods implemented within FAUN 1.0 for constrained nonlinear least-squares problems ([27]) are sequential quadratic programming (SQP) methods and generalized Gauss-Newton (GGN) methods. These methods can exploit the special structure of the Hessian matrix of ε_t ([28], [29], [30]). The SQP and GGN methods can automatically overcome most of the training problems of ANN such as flat spots or steep canyons of the error function ε_t. See [26], [31], [32] for more information about neurosimulator FAUN 1.0.

B. Solution of problem (P)

The solution process consists of the following steps:

1) For every selection of the toll vector w the so-called *C-load* algorithm ([22]) is used in order to obtain the DUE traffic flows $q \in DUE(w)$. The value of the objective function $Z(q(w),w)$ is computed.
2) The set of sample points $(w, Z(q(w),w))$ is divided into training patterns and validation patterns and the neurosimulation introduced in Section III-A takes place in order to approximate Z.
3) This approximation function of Z is minimized, optimal w minimizing Z is found.

For description of the solution process in more detail, see [18].

IV. CASE STUDIES

In this section the performance of the algorithm introduced in Section III will be tested on more specific problem of the type (P). We will consider the so-called *Chen network*,

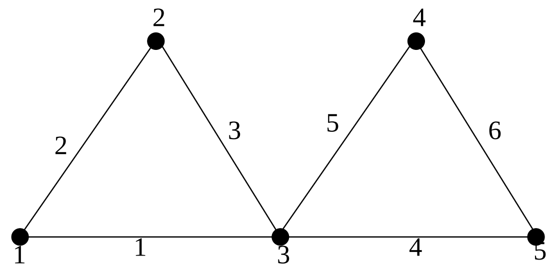

Fig. 2. Chen network.

depicted in Fig. 2. In the case studies we will confine ourselves by the following assumptions:

- The objective function for the road authority is the total travel time, i.e., $Z \stackrel{\text{def}}{=} \tau^T \cdot \mathbf{f}$.
- We consider two origin-destination pairs: $(1,5)$, $(3,5)$.
- The input parameters are set as follows: $\mathscr{K} = \{1,\ldots,16\}$, $\alpha_1 = 10$ [euro/h], $\alpha_2 = 0.8$, [euro/h], $\alpha_3 = 2$ [euro/h], $\hat{k}_{(1,5),\text{dep}} = 4$, $\hat{k}_{(1,5),\text{arr}} = 8$, $\hat{k}_{(3,5),\text{dep}} = 6$, $\hat{k}_{(3,5),\text{arr}} = 8$, $d^{(1,5)} = 8000$ [veh], $d^{(3,5)} = 4000$ [veh], $\varepsilon_v = 0.001$, $w^{(m),\min} = -10$, $w^{(m),\max} = 10$.
- We consider the so-called *dynamic logit-based user equilibrium* as a specific dynamic user equilibrium (see [18] for more details), with $\mu = 0.2$. Convergence of the algorithm is verified using the *relative dynamic duality gap* (see [18] for its definition). If the relative duality gaps of two consecutive iterations are close enough, i.e., if $|\varepsilon^{(\text{iter})} - \varepsilon^{(\text{iter}-1)}| < \varepsilon_{\max}$, with a given small positive number $\varepsilon_{\max}$, the algorithm is terminated. We set $\varepsilon_{\max}$ to 10^{-3}.
- For each directed arc $a \in \mathscr{A}$ the following parameters are initially given: link length s_a [km], maximum speed $\vartheta_a^{\max}$ [km/h], minimum speed $\vartheta_a^{\min}$ [km/h], critical speed $\vartheta_a^{\text{crit}}$ [km/h], jam density J_a^{jam} [pcu/km], where pcu denotes passenger car units, and the unrestricted link capacity C_a [pcu/h]. Dynamic link travel time for an individual user entering link a during the k-th time interval ($k \in \mathscr{K}$) is defined as $\tau_a^{(k)} = \frac{s_a}{\vartheta_a^{(k)}}$, where the link speed $\vartheta_a^{(k)}$ [km/h] is defined as *Smulders speed-density function* (see [33]):

$$\vartheta_a^{(k)} = \begin{cases} \vartheta_a^{\max} + \frac{\vartheta_a^{\text{crit}}-\vartheta_a^{\max}}{J_a^{\text{crit}}} J_a^{(k)}, \\ \qquad \text{if} \quad J_a^{(k)} \le J_a^{\text{crit}}, \\ J_a^{\text{jam}} + \left(\vartheta_a^{\text{crit}} - \vartheta_a^{\min}\right) \frac{\left(J_a^{(k)}\right)^{-1} - \left(J_a^{\text{jam}}\right)^{-1}}{\left(J_a^{\text{crit}}\right)^{-1} - \left(J_a^{\text{jam}}\right)^{-1}}, \\ \qquad \text{if} \quad J_a^{\text{crit}} \le J_a^{(k)} \le J_a^{\text{jam}}, \\ \vartheta_a^{\min}, \quad \text{if} \quad J_a^{(k)} \ge J_a^{\text{jam}}, \end{cases} \tag{7}$$

with critical density J_a^{crit} [pcu/km] defined as $J_a^{\text{crit}} = C_a / \vartheta_a^{\text{crit}}$.
- Initial link parameters are depicted in Table I.
- Link tolls are defined by (2). The results of the problems with $M = 0,1,2,3$ were compared.

The algorithm introduced in Section III was applied to solve the problem.

[5]Fast Approximation with Universal Neural networks

TABLE I

INITIAL LINK PARAMETERS

a	s_a	$\vartheta_a^{\max}$	$\vartheta_a^{\text{crit}}$	$\vartheta_a^{\min}$	J_a^{jam}	C_a
1	7.5	150	90	20	50	3500
2	15	120	70	10	150	1500
3	15	120	70	10	150	1500
4	10	150	90	20	50	3500
5	15	120	70	10	150	1500
6	15	120	70	10	150	1500

TABLE II

THE RESULTING TOLL VALUES (FOR $k \in \{1,\ldots,8\}$) AND TOTAL TRAVEL TIMES FOR CASE STUDY WITH ONE LINK TOLLED

M	$\theta_1^{(k)}$ [euro]				Z [h]
0	2.4	2.7	4.2	6.6	53667
	6.5	5.9	5.4	5.2	
1	1.9	2.3	5.3	7.2	48142
	7.3	6.2	5.2	4.9	
2	1.8	2.3	5.4	7.5	45130
	7.4	6.4	5.2	4.7	
3	1.5	1.9	5.9	7.6	43825
	7.8	6.4	5.0	4.8	

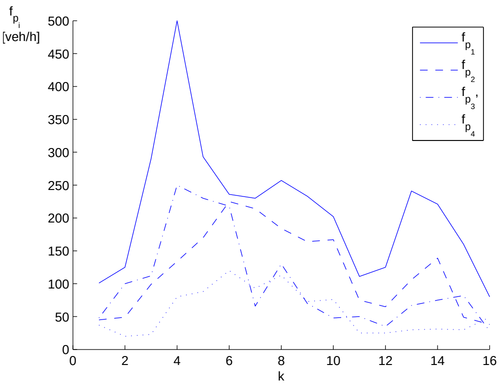

Fig. 3. The resulting traffic flows for case study with two links tolled, $M = 3$.

A. Case study with one link tolled

We will first consider the situation with only link 1 tolled, M will vary from 0 to 3. For the solution process introduced in Section III 40000 samples $(w, Z(q(w), w))$ were collected. For each computations, these samples were splitted into 30000 training data and 10000 validation data.

The resulting toll values for first 8 time intervals and values of total travel times for the situation with $M = 0$, $M = 1$, $M = 2$, and $M = 3$ are depicted in Table II. Clearly, the polynomial tolls result in a better outcome for the road authority. For all variants of M the approximation function had many local minima, but only one global minima.

The computation time on 16 processors was approximately 10.5 [min]. This time can be geometrically decreased by further parallelization.

TABLE III

THE RESULTING TOLL VALUES (FOR $k \in \{1,\ldots,8\}$) AND TOTAL TRAVEL TIMES FOR THE CASE STUDY WITH TWO LINKS TOLLED

M	toll values - link 1				toll values - link 4				Z [h]
0	1.3	1.8	3.5	4.7	0.9	0.9	1.5	0.5	45221
	4.8	4.7	4.5	4.0	1.4	2.8	2.2	2.4	
1	1.2	1.4	4.1	4.4	0.8	0.9	1.2	0.8	43790
	4.9	4.5	4.0	4.1	2.6	2.6	2.9	2.9	
2	1.2	1.5	4.3	4.6	0.8	0.8	1.3	1.2	42091
	4.6	4.4	3.8	3.8	2.7	2.8	2.8	2.5	
3	1.2	1.4	4.4	4.7	0.7	0.8	1.2	1.4	41349
	4.7	4.7	4.1	3.8	2.5	2.9	2.7	2.5	

B. Case study with two links tolled

Let both links 1 and 4 be tolled. We will use 80000 samples, with 60000 training data and 20000 validation data. In Table III the resulting toll values for first 8 time intervals and resulting total travel times for the situations with $M = 0$, $M = 1$, $M = 2$, and $M = 3$ are depicted. Again, the polynomial tolls improved the system performance remarkably.

Let p_1 denote the route consisting of links 1 and 4, let p_2 denote the route consisting of links 1, 5, and 6, let p_3 denote the route consisting of links 2, 3, and 4, and let p_4 denote the route consisting of links 2, 3, 5, and 6. In Figure 3 the traffic flows on the routes p_1, p_2, p_3, and p_4 are depicted. Note that quite many travelers use the route p_1, as its costs are lower even after imposing tolls on both its links.

The computation time on 16 processors was approximately 52.3 [min].

C. Discussion

After application of the neurosimulation-based algorithm to solve the problems presented in this section we applied a very detailed grid search on the same problems. It showed that the obtained minimal values differed by 0.5 percent maximally (by means of L_2-norm) and the global minima were located in the same points as those computed using the neurosimulation-based algorithm. One has to be very careful with choosing the right value of the validation error as well as the sample area, though, to obtain correct results.

The tolls set as polynomial functions of traffic flows improved the system performance remarkably. Note that while these tolls are computed as functions of traffic flows, they are imposed as positive values.

V. CONCLUSIONS & FUTURE RESEARCH

We formulated a very general version of the dynamic optimal toll design problem and discussed its properties. As the problem is NP-hard and the objective function of the road authority generally has many local minima, we proposed a neurosimulation-based approximation algorithm using Neurosimulator FAUN 1.0 to deal with the problem.

The algorithm that we proposed is applicable to a wide range of optimal toll design problems, with different DUE formulations, and with a wide range of objective functions.

The performance of the proposed algorithm was tested on a case study on the Chen network.

As the proposed solution method is an approximation method, it is important to choose the initial parameters for neurosimulation very carefully to ensure sufficient accuracy, as well as to avoid overtraining. These issues are discussed in [26], [18]. With a proper choice of the criteria for neurosimulation the algorithm performs very well, as we showed in the case studies.

Even solving a problem defined on a small network was quite time consuming. However, the algorithm can be parallelized, and, therefore, this pitfall can be easily eliminated and the algorithm can be applied on real-size networks.

We showed that the traffic-flow dependent tolls can perform remarkably better than the traffic-flow invariant tolls, even with simple polynomial toll functions. Use or more sophisticated toll functions may bring even better results.

ACKNOWLEDGEMENTS

Research presented in this paper was carried out during author's stay at the Delft Technical University, the Netherlands, and was supported by the Next Generation Infrastructures Foundation and the TRAIL Research School, the Netherlands.

The author would like to thank the reviewers for the constructive comments which helped to improve the quality of the paper.

REFERENCES

[1] E. T. Verhoef, "Second-best congestion pricing in general networks. heuristic algorithms for finding second-best optimal toll levels and toll points," *Transportation Research Part B*, vol. 36, no. 8, pp. 707–729, 2002.

[2] M. Patriksson and R. T. Rockefellar, "A mathematical model and descent algorithm for bilevel traffic management," *Transportation Science*, vol. 36, no. 3, pp. 271–291, 2002.

[3] A. D. May and D. S. Milne, "Effects of alternative road pricing systems on network performance," *Transportation Research Part A*, vol. 34, no. 6, pp. 407–436, 2000.

[4] W. Vickrey, "Congestion theory and transport investment," *The American Economic Review*, vol. 59, no. 2, pp. 251–260, 1969.

[5] J. Supernak, T. Golob, C. Kaschade, C. Kazimi, E. Schreffler, and D. Steffey, "San Diego's Interstate 15 congestion pricing project: Traffic-related issues," *Transportation Research Record*, vol. 1812, pp. 43–52, 2002.

[6] D. Joksimovič, M. C. J. Bliemer, and P. H. L. Bovy, "Optimal toll design problem in dynamic traffic networks with joint route and departure time choice," *Transportation Research Record*, vol. 1923, 2004.

[7] K. Staňková, G. J. Olsder, and M. C. J. Bliemer, "Bilevel optimal toll design problem solved by the inverse Stackelberg games approach," *Urban Transport*, vol. 12, pp. 871–880, 2006.

[8] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Philadelphia, Pennsylvania: SIAM, 1999.

[9] A. Bagchi, *Stackelberg Differential Games in Economical Models*. Berlin, Germany: Springer-Verlag, 1984.

[10] M. Patriksson, *The Traffic Assignment Problem: Models and Methods*. Utrecht, The Netherlands: VSP, 1994.

[11] C. Fisk, "Some developments in equilibrium traffic assignment," *Transportation Research Part B*, vol. 14, no. 3, pp. 243–255, 1980.

[12] M. B. Yilidirim and D. W. Hearn, "A first best toll pricing framework for variable demand traffic assignment problems," *Transportation Research Part B*, vol. 39, pp. 659–678, 2005.

[13] L. Zhang and D. Levinson, "Road pricing with autonomous links," *Journal of the Transportation Research Board*, vol. 1932, pp. 147–155, 2005.

[14] T. Larson and M. Patriksson, "Traffic management through link tolls – An approach utilizing side constrained traffic equilibrium models," *Rendiconti del Circolo Matematico di Palermo, Serie II*, vol. 48, pp. 147–170, 1997.

[15] E. T. Verhoef, "Second best congestion pricing in general static transportation networks with elastic demands," *Regional Science and Urban Economics*, vol. 32, no. 3, pp. 281–310, 2002.

[16] K. Staňková, H.-J. Von Mettenheim, and G. J. Olsder, "Dynamic optimal toll design problem with second-best-flow-dependent tolling solved using neural networks," in *Proceedings of the 10th International Conference on Applications of Advanced Technologies in Transportation (submitted also for a special issue of Transportation Research A)*, Athens, Greece, 2008.

[17] K. Staňková, G. J. Olsder, and M. C. J. Bliemer, "Road pricing games in a two-link dynamic network," Delft University of Technology, Delft, The Netherlands, Tech. Rep., 2009, to appear in the European Journal of Transportation and Infrastructure Research.

[18] K. Staňková, "On Stackelberg and Inverse Stackelberg Games & Their Applications in the Optimal Toll Design Problem, the Energy Market Liberalization Problem, and in the Theory of Incentives," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, 2009.

[19] G. J. Olsder, "Phenomena in inverse Stackelberg problems," Mathematisches Forschungsinstitut Oberwolfach, Oberwolfach, Germany, Tech. Rep. 11/2005, 2005.

[20] K. H. Chen, *Dynamic Travel Choice Models: A Variational Inequality Approach*. Oxford, UK: Springer, 1999.

[21] K. Staňková and M. C. J. Bliemer, "Dynamic road pricing with traffic-flow dependent tolling," in *Proceedings of the 87th Transportation Research Board Annual Meeting*, Washington, District of Columbia, 2008, paper 08-1016.

[22] M. C. J. Bliemer, "Analytical dynamic traffic assignment with interacting user-classes," Ph.D. dissertation, Delft University of Technology, Delft, The Netherlands, 2001.

[23] F. Fachinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems: Vol. I, II*. New York, New York: Springer-Verlag, 2003.

[24] A. Chow, "Properties of dynamic system optimal assignment and its solution method," *Transportation Research B*, vol. 43, no. 3, pp. 325–344, 2009.

[25] T. Friesz, C. Kwon, and R. Mookherjee, "A computable theory of dynamic congestion pricing," in *Transportation and Traffic Theory*, R. Allsop, M. Bell, and B. Heydecker, Eds. Amsterdam, The Netherlands: Elsevier, 2007, pp. 1–26.

[26] S. Koenig, F. Koeller, and M. H. Breitner, "FAUN 1.1 User Manual," Institut für Wirtschaftsinformatik, Gottfried Wilhelm Leibniz Universitt, Hannover, Germany, Tech. Rep., 2005.

[27] U. Nowak and L. Weimann, "A family of Newton codes for systems of highly nonlinear equations," Konrad Zuse Zentrum, Berlin, Germany, Tech. Rep. TR 91-10, 1998.

[28] P. Deuflhard, *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Berlin, Germany: Springer, 2004.

[29] R. Fletcher, *Practical Methods of Optimization*. New York, New York: John Wiley & Sons, 2000.

[30] P. E. Gill, W. Murray, and M. H. Wright, *Practical optimization*. London, UK: Academic Press, 2004.

[31] M. H. Breitner, "Usage of artificial neural networks for the numerical solution of dynamic games," in *Proceedings of the 11th International Symposium on Dynamic Games and Applications*, Tuscon, Arizona, 2004, pp. 62–79.

[32] M. H. Breitner and M. Hans-Jrg, "Coarse-grained parallelization of the advanced neurosimulator FAUN 1.0 with PVM and the enhanced cornered rat game revisited," *International Game Theory Review*, vol. 7, pp. 347–365, 2005.

[33] S. Smulders, "Modelling and filtering of freeway traffic flow," Centre of Mathematics and Computer Science, Amsterdam, The Netherlands, Tech. Rep. OS-R8706, 1988.