To appear in Networks

Computing Network Tolls with Support Constraints

Tobias Harks* Ingo Kleinert‡ Max Klimm†§ Rolf H. Möhring†

Abstract

Reducing traffic congestion via toll pricing has been a central topic in the operations research and transportation literature and, recently, it has been implemented in several cities all over the world. Since in practice it is not feasible to impose tolls on every edge of a given traffic network, we study the resulting mathematical problem of computing tolls on a predefined subset of edges of the network so as to minimize the total travel time of the induced equilibrium flow. We first present an analytical study for the special case of parallel edge networks highlighting the intrinsic complexity and non-convexity of the resulting optimization problem. We then present algorithms for general networks for which we systematically test the solution quality for large-scale network instances. Finally we discuss the related optimization problem of computing tolls subject to a cardinality constraint on the number of edges that have tolls.

1 Introduction

Today's traffic situations in large cities are still far from being satisfactory. Traffic jams at rush hour or at special events (sport or music events) occur frequently and drivers suffer from increased total travel times. Moreover, traffic congestion significantly increases exhaust gas pollution. The situation is particularly dramatic in the rising mega-cities in Asia and South-America. The traffic volume in China, for instance, has increased in 2005 by 15% and is expected to grow faster in the next years [20], making good network planing and traffic control indispensable. It is a well known fact that selfish behavior of traffic participants is one of the main reasons that leads to inefficient traffic situations [7, 14]. Since every traffic participant solely aims at minimizing her individual travel time, the overall outcome is less efficient, e.g., in terms of the total average travel time, than when everybody would have been routed according to a centrally coordinated scheme.

Modeling selfish behavior in traffic networks has been a central topic in the operations research and transportation literature for decades, see Beckmann et al. [4] and Sheffi [33].

*Maastricht University, Operations Research Group, School of Business and Economics, PO Box 616, 6200 MD, Maastricht, The Netherlands, Email: t.harks@maastrichtuniversity.nl

†Technische Universität Berlin, Institut für Mathematik, Straße des 17. Juni 136, 10623 Berlin, Germany. Email: {kleinert,klimm,moehring}@math.tu-berlin.de.

‡Research supported by the Federal Ministry of Education and Research (BMBF grant 03MOPAI1).

§Research supported by the Deutsche Forschungsgemeinschaft within the research training group 'Methods for Discrete Structures' (GRK 1408).

The classical theory of selfish behavior in traffic networks started with the traffic model of Wardrop [40]. The basic idea is to model the interaction among the selfish network users as a *non-cooperative game*. We are given a directed graph with latency functions on the edges and a set of origin-destination pairs, called *commodities*. Every commodity is associated with a *demand*, which specifies the rate of flow that needs to be sent from the respective origin to the destination. In the non-atomic variant, every demand represents a continuum of agents, each controlling an infinitesimal amount of flow. The latency that an agent experiences when traversing an edge is given by a (non-decreasing) function of the total flow on that edge. Agents are assumed to act selfishly and route their flow along a minimum-latency path from their origin to their destination. A solution in which no agent can switch to a path with smaller travel time corresponds to a Wardrop equilibrium [11, 40]. It is well known that a Wardrop flow in general does not minimize the total travel time; or said differently, selfish behavior may cause a performance degradation in the network. Koutsoupias and Papadimitriou [26] introduced a measure to quantify the inefficiency of equilibria which is now known as the *price of anarchy*. It is defined as the worst-case ratio of the cost of an equilibrium over the cost of a system optimum (also called social optimum). By now, the price of anarchy is well understood for specific classes of latency functions (affine latencies or polynomial latencies with non-negative coefficients), see Roughgarden and Tardos [32], Roughgarden [30] and Correa et al. [10]. Despite the bounds for specific classes of latency functions, the price of anarchy is unbounded for general latency functions even on networks of parallel edges [32].

Due to this large efficiency loss, researchers have proposed congestion pricing strategies for over 80 years, see Pigou [29] and Beckmann et al. [4]. The basic idea is to impose tolls on network edges that guarantee that the selfish outcome corresponds to a predetermined routing scheme, e.g., one that minimizes the total average travel time. Assuming that one can possibly collect tolls on every edge of the network and that the value of time is the same for all users, it is a classical result in the economic theory of transportation that tolls equal to the marginal edge cost of the system optimal solution (*marginal cost pricing*) induce a socially optimal equilibrium flow [4].

Over the last decades, congestion pricing strategies have been implemented in various cities. Examples include London [38] (London congestion charge), Stockholm [37], Bergen [39], and Singapore [34] (Electronic Road Pricing). All these applications have in common that only designated areas of the transport network are amenable to tolls. In London [38] and Stockholm [37], a congestion fee is charged only for access to the center of the city. Bergen [39] implemented a toll ring where congestion fees are charged. These features makes the application of classical marginal cost pricing impossible.

Since in practice it may not be feasible to impose tolls on every edge of a given traffic network, we study in this article the resulting mathematical problem of computing tolls on a predefined subset of edges of the network so as to minimize the travel time of the induced equilibrium flow. We first present an analytical study for the special case of networks consisting of parallel edges that highlights the intrinsic complexity and non-convexity of the resulting optimization problem. We also present algorithms for general networks and test the quality of our algorithms on large-scale networks. We finally discuss the related optimization problem of computing tolls subject to a cardinality constraint on the number of tollable edges.

1.1 Related Work

Already in 1920, Pigou [29] suggested that, in order to obtain a system optimal traffic pattern, vehicles should be charged tolls equal to the difference between marginal social and marginal private cost (marginal cost pricing). The theoretical foundation of marginal cost pricing has been further explored by many researchers, see for example Knight [25], Beckmann et al. [4] and Smith [35]. Bergendorff et al. [5], Hearn and Ramana [21] and Larsson and Patriksson [27] showed that the set of feasible edge toll vectors supporting a system optimal flow as a user equilibrium can be characterized by a non-empty polyhedron expressed in terms of a system of linear inequalities and equations. Hearn and Ramana [21] also studied secondary optimization problems that minimize or maximize a toll dependent function over the toll polyhedron. Dial [12, 13] and Bai et al. [1] proposed efficient algorithms for computing tolls in the toll-polyhedron that minimize the total revenue collected from the users. Bai and Rubin [3] and Bai et al. [2] developed algorithms for computing tolls in the toll-polyhedron so as to minimize the number of tolled edges.

Cole et al. [9] considered the case of heterogeneous users, in which users value latency and monetary cost differently. For single-commodity networks, they showed the existence of tolls that induce an optimal flow as a Wardrop flow. Using a mathematical programming approach, Fleischer et al. [17], Karakostas and Kolliopoulos [24], and Yang and Huang [41] proved that there are tolls inducing an optimal flow for heterogeneous users even in general networks. The resulting mathematical program also yields a characterization of equilibrium flows that can be enforced by tolls. Swamy [36] and Yang and Zhang [42] proved the existence of optimal tolls for the atomic splittable model with fixed demands.

Hoefer et al. [22] studied the problem of finding optimal tolls on a *subset* of edges of a network. They showed that it is NP-hard to compute optimal tolls in a 2-commodity network even if all latency functions are linear. For a network of parallel edges with linear latencies, they gave a polynomial algorithm for computing optimal tolls.

1.2 Results and Paper Outline

We study the problem of computing tolls on a given subset of network edges so as to minimize the total travel time of the induced equilibrium flow. Our contribution can be summarized as follows.

After introducing the basic model in Section 2, we consider single-commodity networks consisting of parallel edges in Section 3. This setting describes situations in which access roads to the central district of a city are the bottlenecks for the inbound or outbound traffic. These roads may be either tollable (bridges or highways) or non-tollable and the goal is to devise tolls so as to minimize congestion on the bottleneck edges. Another application arises when a highway is divided into tollable and non-tollable lanes (as in Tel Aviv [28]). We devise an algorithm that approximates an optimal solution within an additive error of $\varepsilon > 0$. Our algorithm runs in $\mathsf{poly}(m, K, \kappa, d, 1/\varepsilon)$-time, where m denotes the number of edges, K is an upper bound on the latency functions, κ is a common Lipschitz constant of the latency functions, their derivatives, and inverse functions, d is the flow demand, and ε is the precision. Note that previously, a polynomial algorithm was known only for affine latencies [22].

The running time of our algorithm proposed in Section 3 is only quasi-polynomial, as it depends linearly on K, κ and d. Section 4 is devoted to the design of a polynomial algorithm. We identify conditions on the latency functions guaranteeing that the objective function is a piecewise convex function (with a polynomial number of breakpoints) of the total demand that is routed along the toll-free edges. We use this convexity property to devise a polynomial algorithm that approximates the optimal objective value within a precision $\varepsilon > 0$ in time $\mathsf{poly}(m, \log K, \log \kappa, \log d, \log 1/\varepsilon)$. We demonstrate that our conditions on the latency functions are satisfied by practically relevant functions such as the popular $M/M/1$-latency functions arising in queuing networks [6] or certain polynomials with non-negative coefficients that can be used for modeling latency functions in transportation networks [33].

In Section 5 we turn to the design of algorithms that work for general networks. Given the complexity of the toll problem even on networks of parallel edges, we consider only heuristics. We present three algorithms that are inspired by the gradient descent method. They iteratively increase the toll on those feasible edges on which the edge flow of the current Wardrop flow exceeds the system optimal edge flow, and they decrease the tolls otherwise. The rationale behind this iterative process is to follow a solution trajectory along a gradient descent direction of the objective function. We provide a computational study evaluating the quality and convergence behavior of our algorithms on large-scale network instances. It turns out that for most test instances, already a small number of tollable edges suffices to significantly reduce the total travel time.

Finally, Section 6 is devoted to the problem of actually selecting the set of edges on which tolls are imposed. We introduce the cardinality constrained toll problem, where the objective is to compute tolls subject to a cardinality constraint on the set of tollable edges so as to minimize the travel time of the induced equilibrium flow. We prove that this optimization problem is strongly NP-hard and inapproximable by any constant $c \geq 1$, unless P = NP. We then present a computational study for several edge subset selection algorithms in combination with our algorithms for computing tolls on the selected edges. On most of the test instances, our combined algorithms perform very well and significantly reduce the overall travel time already for small cardinalities.

2 Preliminaries

A standard way to model the selfish behavior of traffic participants is by means of a *non-atomic network routing game*. We are given a directed network $G = (V, E)$ and k commodities $(s_1, t_1), \dots, (s_k, t_k) \in V \times V$. Let n and m denote the number of vertices and edges in G, respectively. Additionally, we are given a *demand* $d_i > 0$ for every commodity $i \in \{1, \dots, k\}$ which specifies the amount of flow that must be routed from the origin s_i to the destination t_i. Let $\mathcal{P}_i$ be the set of all (simple) directed (s_i, t_i)-paths in G and define $\mathcal{P} = \bigcup_{i=1}^{k} \mathcal{P}_i$. It is convenient to express a *flow* as a function $f : \mathcal{P} \to \mathbb{R}_{\geq 0}$ that assigns to every path $P \in \mathcal{P}$ a non-negative flow-value f_P that is routed along P. A flow f is *feasible* (with respect to $d = (d_1, \dots, d_k)$) if d_i units of flow are routed from s_i to t_i for every $i \in \{1, \dots, k\}$, i.e., $\sum_{P \in \mathcal{P}_i} f_P = d_i$. For a given flow f, we define the flow on an edge $e \in E$ as $f_e = \sum_{P \ni e} f_P$. Every edge $e \in E$ has a nonnegative, increasing, convex and differentiable *latency function* $\ell_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The latency $\ell_P(f)$ of a path P with respect to a flow f is defined as the sum of the latencies of the edges in the path, i.e., $\ell_P(f) = \sum_{e \in P} \ell_e(f_e)$.

The total *cost* of a flow f is defined as $C(f) = \sum_{P \in \mathcal{P}} f_P \ell_P(f)$ or, equivalently, $C(f) = \sum_{e \in E} f_e \ell_e(f_e)$. A feasible flow of minimum total cost is called *optimal* and denoted by f^*. A feasible flow f is a *Wardrop equilibrium* (or simply an equilibrium flow) if

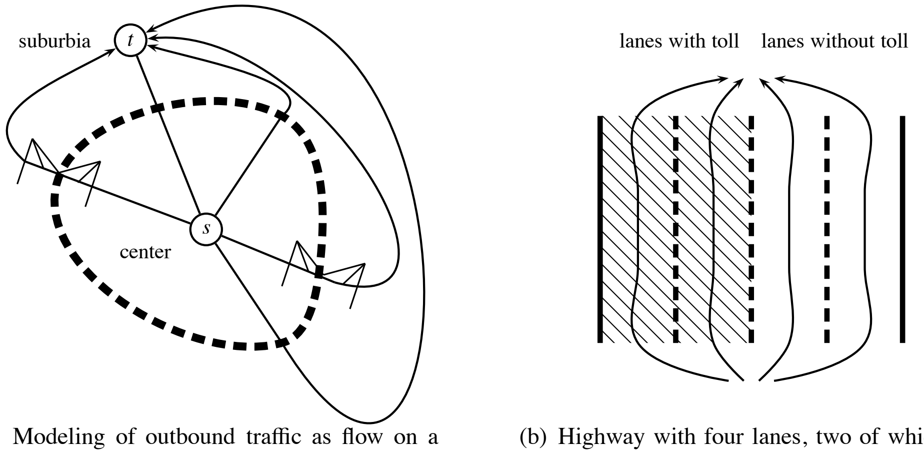
$$\ell_P(f) \leq \ell_{P'}(f) \quad \text{for all } i \in \{1, \dots, k\} \text{ and } P, P' \in \mathcal{P}_i \text{ with } f_P > 0. \tag{1}$$

For every commodity, the latency of every path that carries a positive amount of flow is minimum. This implies that all flow carrying (s_i, t_i)-paths have equal latency. Under the assumption that all latency functions are convex, the cost of a Wardrop equilibrium is unique (see e.g. [32]). The *price of anarchy* is defined as the worst-case ratio (over all instances) of the cost of a Wardrop equilibrium f and the cost of an optimal flow f^*, i.e., $C(f)/C(f^*)$. It is well-known (see [32]) that the price of anarchy is unbounded for general convex latency functions even on parallel edge networks.

Network tolls are an efficient means to reduce the price of anarchy in network routing games. Intuitively, every (non-atomic) player that traverses edge $e \in E$ experiences, besides the latency $\ell_e(f_e)$, an additional (non-negative) toll τ_e. Because in practice, it might not be feasible to impose a toll on every edge of the network in practice, we introduce the *toll problem with support constraints*. An instance I_s of this problem is a tuple $I_s = (G, d, \ell, T)$, where $G = (V, E)$ is a directed network with k commodities, $d_i, i \in \{1, \dots, k\}$ are the corresponding demands, $\ell_e, e \in E$ is the latency function of edge e, and $T \subsetneq E$ is the set of edges on which tolls can be imposed. A non-negative vector $\tau = (\tau_e)_{e \in E}$ with the property that $\tau_e = 0$ for all $e \in E \setminus T$ will be called a *feasible toll vector* for I_s. Every toll vector τ induces a unique Wardrop equilibrium f^τ satisfying $\ell_P(f^\tau) + \sum_{e \in P} \tau_e \leq \ell_{P'}(f^\tau) + \sum_{e \in P'} \tau_e$ for all $i \in \{1, \dots, k\}, P, P' \in \mathcal{P}_i$ with $f_P > 0$. A toll vector τ^* will be called *optimal* for I_s if $C(f^{\tau^*}) \leq C(f^\nu)$ for all feasible toll vectors ν. We also consider the *toll problem with cardinality constraints*. Here, we are given a tuple $I_c = (G, d, \ell, b)$, where G, d, and ℓ are defined as above but $b \in \mathbb{N}$ is a *cardinality bound* on the set of edges that may have tolls. A non-negative vector $\tau = (\tau_e)_{e \in E}$ is a feasible toll vector for I_c if $|\{e \in E : \tau_e > 0\}| \leq b$.

3 A Pseudo-Polynomial Algorithm for Parallel Edges

We first consider toll problems with support constraints for parallel edge networks. Parallel edges model situations in which access roads to the central district of a city are the bottlenecks for the inbound or outbound traffic. In such a situation, access roads may be either tollable (bridges or highways) or non-tollable and the goal is to devise tolls so as minimize congestion on the bottleneck edges, see Figure 1 (a). Very recently, in the area of Tel Aviv a fast lane on highways has been opened. While the use of the regular lanes is not charged, the toll on the fast lane is adapted to the traffic on that lane so as to guarantee a speed of 70 km/h [28]. The problem of finding the socially optimal toll on the fast lane can be formulated as a toll problem with support constraints on parallel edges (Figure 1 (b)). Parallel edges also model scheduling problems involving selfish players [31]. Our model includes the case where only some machines (edges) are allowed to charge tolls.



(a) Modeling of outbound traffic as flow on a parallel edge graph. Edges with bridges are assumed to be tollable.

(b) Highway with four lanes, two of which are with tolls.

Figure 1: Two applications of parallel edge graphs in traffic models.

It is not hard to see that there are instances involving quadratic latency functions with non-negative rational coefficients for which the optimal toll vector is unique and irrational. Therefore, we focus on efficiently computing ε-approximate toll vectors. That is, for given $\varepsilon > 0$ we want to compute in polynomial time a toll vector τ^*_ε with $C(f^{\tau^*_\varepsilon}) \leq C(f^\nu) + \varepsilon$ for all toll vectors ν. We first show that this problem can be reduced to a one-dimensional optimization problem in terms of the demand that is routed over the non-tollable edges. We will solve this problem by discretizing the demand interval. For a given demand value, the main difficulty of this approach stems from approximating the objective value, which itself is defined via optimal solutions of two related optimization problems defined below.

3.1 Problem Parameterization

An instance of the toll problem with support constraints on parallel edge networks is given by a tuple $I_s = (G, d, \ell, T)$, where the graph $G = (V, E)$ has only two nodes $s, t \in V$ and every edge $e \in E$ goes from s to t. Since there is only one commodity starting in s and ending in t, we will denote the demand of this commodity by d. The set $E \setminus T$ of non-tollable edges will be denoted by N. In addition to the assumption that every latency function is non-negative, increasing, convex and differentiable, we make the following assumptions on the latency functions.

Assumption 3.1 (Lipschitz assumption)**.** *For every edge $e \in E$, the latency function ℓ_e, its derivative ℓ'_e and its inverse function ℓ_e^{-1} are Lipschitz continuous with constant $\kappa \in \mathbb{N}$. More formally, there is $\kappa \in \mathbb{N}$ such that*

$$\ell_e(y) - \ell_e(x) \leq \kappa(y - x), \qquad \ell'_e(y) - \ell'_e(x) \leq \kappa(y - x), \qquad \ell_e^{-1}(y) - \ell_e^{-1}(x) \leq \kappa(y - x),$$

for all $e \in E$ and for all $0 \leq x < y \leq d$.

Note that this assumption implies that $1/\kappa \leq \ell_e'(x) \leq \kappa$ for all $e \in E$ and $x \in [0, d]$. In addition, we assume that the latencies are bounded.

Assumption 3.2. *There is $K \in \mathbb{N}$ such that $\ell_e(d) \leq K$ for all $e \in E$.*

These assumptions are satisfied by a large class of latency functions. For example, every convex and twice continuously differentiable function $\ell_e : [0, d] \to \mathbb{R}_{\geq 0}$ with $\ell_e'(0) > 0$ satisfies Assumptions 3.1 and 3.2. To see this, note that because ℓ'' is continuous, it attains its maximum on $[0, d]$. Let

$$\kappa = \max\Bigl\{\max_{x \in [0,d]} \ell_e''(x), 1/\ell_e'(0), \ell_e'(d)\Bigr\}.$$

Then, ℓ, ℓ' and ℓ^{-1} are Lipschitz continuous with constant κ. In addition, Assumption 3.2 is satisfied for $K = \ell_e(d)$.

Remark 3.3. *All our results in this section continue to hold for instances with convex and twice continuously latencies with $\ell_e'(0) > 0$ for all $e \in E$, where each edge $e \in E$ has a capacity $c_e \in \mathbb{R}_{>0}$ and $\ell_e(x) \to \infty$ for $x \to c_e$. To see this, note that if $d \geq \sum_{e \in E} c_e$, there is no feasible flow vector with finite cost, thus, we may assume that $\sum_{e \in E} c_e > d$. Let*

$$f_e = \max\Bigl\{0, c_e - \frac{\sum_{e \in E} c_e - d}{|E|}\Bigr\}$$

and let $M = \max_{e \in E: f_e > 0} \ell_e(f_e)$. Then, dM is an upper bound on the cost of a Wardrop equilibrium on E and, hence, an upper bound on $C(f^{\tau^*})$. Let $e \in E$ with $f_e > 0$ be arbitrary. By construction, we have $\ell_e(f_e)f_e \leq Mf_e \leq dM$. As $\ell_e(f_e)f_e \to \infty$ for $f_e \to c_e$, there is $f_e \in [f_e, c_e)$ with $\ell_e(f_e)f_e = dM$. Thus, we know that any flow f with $f_e > f_e$ has cost larger than $C(f^{\tau^*})$. As a consequence, we may effectively restrict the latency functions to the domain $[0, f_e]$. The restricted latency functions $\ell_e : [0, f_e] \to \mathbb{R}_{\geq 0}$ then satisfy Assumptions 3.1 and 3.2.

It follows that important classes of latency functions such as $M/M/1$ functions used in queueing networks or latency functions used by the Bureau of Public Roads [8] satisfy the above assumptions. For $d_N, d_T \in [0, d]$, we call a pair (d_N, d_T) a *demand distribution* if $d_N + d_T = d$. A demand distribution is called optimal if there is an optimal toll vector τ^* such that the corresponding equilibrium flow f^{τ^*} satisfies $\sum_{e \in N} f_e^{\tau^*} = d_N$ and $\sum_{e \in T} f_e^{\tau^*} = d_T$. The following lemma has already been proven by Hoefer et al. [22] for the special case of affine latencies.

Lemma 3.4. *Let (d_N, d_T) be an optimal demand distribution. Then there exists an optimal flow f^{τ^*} such that $f_N^{\tau^*} = (f_e^{\tau^*})_{e \in N}$ is a Wardrop equilibrium on N with demand d_N and $f_T^{\tau^*} = (f_e^{\tau^*})_{e \in T}$ solves*

$$\begin{array}{lll} \min & \sum_{e \in T} \ell_e(f_e) f_e & \\ s.t. & \sum_{e \in T} f_e = d_T & \textit{(Flow demand)} \\ & \ell_e(f_e) \leq L(d_N) \quad \textit{for all } e \in T \textit{ with } f_e > 0, & \textit{(Latency restriction)} \end{array} \tag{2}$$

where $L(d_N)$ denotes the common latency of all flow-carrying edges in N. An optimal toll vector τ^* is given by $\tau_e^* = L(d_N) - \ell_e(f_e^{\tau^*})$ for all $e \in T$ with $f_e^{\tau^*} > 0$ and $\tau_e = 0$, otherwise.

Proof. Let (d_N, d_T) be an optimal demand distribution. This implies that there is an optimal toll vector τ^* and an induced optimal equilibrium flow f^{τ^*} with the property that $\sum_{e\in N} f_e^{\tau^*} = d_N$ and $\sum_{e\in T} f_e^{\tau^*} = d_T$. Since $\tau_e^* = 0$ for all $e \in N$, the flow $f_N^{\tau^*}$ constitutes a Wardrop equilibrium on N. Thus, there is a constant $L(d_N)$ such that $\ell_e(f_e^{\tau^*}) = L(d_N)$ for all $e \in N$ with $f_e^{\tau^*} > 0$. Being in equilibrium, the optimal flow f^{τ^*} also fulfills $\ell_e(f_e^{\tau^*}) \leq L(d_N)$ for all $e \in T$ with $f_e^{\tau^*} > 0$. Furthermore, since the demand distribution (d_N, d_T) is optimal, the flow $f_T^{\tau^*}$ minimizes $\sum_{e\in T} \ell_e(f_e)f_e$. We derive that $f_T^{\tau^*}$ is a solution of (2). The optimal flow f^{τ^*} is induced by the tolls $\tau_e^* = L(d_N) - \ell_e(f_e^{\tau^*})$ for all $e \in T$ with $f_e^{\tau^*} > 0$ and $\tau_e^* = 0$, otherwise. □

So far, we observed that the problem of finding optimal tolls essentially reduces to the problem of finding an optimal demand distribution (d_N, d_T). Since $f_N^{\tau^*}$ is a Wardrop equilibrium on the non-tollable edges N, every edge $e \in N$ that carries flow has a unique latency $\ell_e(f_e^{\tau^*}) = L(d_N)$. The total social cost of the non-tollable edges then writes as $C_N(d_N) = L(d_N)d_N$, where we define $L(0) = \min_{e\in N} \ell_e(0)$. Moreover, for $d_T \in [0, d]$ and $L \in \mathbb{R}_{\geq 0}$, let us denote by $F_T(d_T, L)$ the set of flows on T with demand d_T for which the latencies of the used edges are bounded by L, that is,

$$F_T(d_T, L) = \Big\{ f_T \geq 0 : \sum_{e\in T} f_e = d_T \text{ and } \ell_e(f_e) \leq L \text{ for all } e \in T \text{ with } f_e > 0 \Big\}.$$

In addition, let $C_T(d_T) = \min_{f_T \in F_T(d_T, L(d-d_T))} C(f_T)$ be the optimal cost of the tollable edges when routing a demand of $d_T = d - d_N$. Then, an optimal demand distribution and hence an optimal toll vector can be found solving the following one-dimensional optimization problem

$$\begin{array}{ll} \min & C_N(d_N) + C_T(d - d_N) \\ \text{s.t.} & d_N \in [d_{\min}, d], \end{array} \tag{3}$$

where $d_{\min} = \min\{d_N \in [0, d] : F_T(d_T, L(d_N)) \neq \emptyset\}$. We will call the function $C_E : [d_{\min}, d]$, $d_N \mapsto C_N(d_N) + C_T(d_N)$ the *combined cost function*. By construction, for every $d_N \in [d_{\min}, d]$ there is a toll vector τ with $\sum_{e\in N} f_e^\tau = d_N$ and $C(f^\tau) = C_E(d_N)$. In the next section, we will show how to solve (3) approximately by discretizing the demand interval $[d_{\min}, d]$.

3.2 Discretizing the Demand Interval

In light of Lemma 3.4, for a given $\varepsilon > 0$ we are interested in computing a demand $d_N \in [d_{\min}, d]$ with $C_E(d_N) \leq C_E(d_N') + \varepsilon$ for all $d_N' \in [d_{\min}, d]$. We will solve this problem by discretizing the demand interval $[d_{\min}, d]$ with a sufficiently small step size $\delta > 0$ and by approximating the combined cost function in every subinterval with a sufficiently small error. In order to define the proper step size, we need to calculate the Lipschitz constant of the combined cost function.

Lemma 3.5. *The combined cost function C_E is Lipschitz continuous with constant $|E|^2\kappa^2(K+\kappa d)$.*

Proof. We first show that $C_N(d_N) = L(d_N)\, d_N$ is Lipschitz continuous with constant $K + \kappa d$ on $[0, d]$. Let $e_0 \in \arg\min_{e\in N} \ell_e(0)$. Then, e_0 carries flow for all values of $d_N \in (0, d]$. For $e \in N$ we denote by $f_e(d_N)$ the flow on e when a total demand of d_N is sent over the non-tollable edges.

We have $\sum_{e\in N} f_e(d_N) = d_N$, and, thus, $\sum_{e\in N} f_e'(d_N) = 1$. As $f_e'(d_N) \geq 0$, we obtain in particular $f_e'(d_N) \leq 1$ for all $e \in N$ and $d_N \in [0, d]$. For all $0 \leq d_N < d_N' \leq d$, we calculate

$$\begin{aligned} C_N(d_N') - C_N(d_N) &= \ell_{e_0}(f_{e_0}(d_N'))\, d_N' - \ell_{e_0}(f_{e_0}(d_N))\, d_N \\ &= \ell_{e_0}(f_{e_0}(d_N'))\, d_N' - \ell_{e_0}(f_{e_0}(d_N'))\, d_N + \ell_{e_0}(f_{e_0}(d_N'))\, d_N - \ell_{e_0}(f_{e_0}(d_N))\, d_N \\ &\leq K(d_N' - d_N) + \kappa d(f_e(d_N') - f_e(d_N)) \leq (K + \kappa d)(d_N' - d_N), \end{aligned}$$

where we use $f_e(d_N') - f_e(d_N) \leq \max_{\xi \in [d_N, d_N']} f_e'(\xi)(d_N' - d_N) \leq d_N' - d_N$.

We now turn to the proof of the Lipschitz continuity of $C_T(d_T)$. For $d_T \in [0, d - d_{\min}]$ and $e \in T$ let $f_e(d_T)$ denote the flow on edge e when a total demand of d_T is sent over the tollable edges. Note that for every $e \in T$, the flow $f_e(d_T)$ is increasing in d_T unless the latency restriction $\ell_e(f_e(d_T)) \leq L(d - d_T)$ becomes tight. When an edge is tight at d_T, it is also tight for all $d_T' > d_T$. In particular, the flow of a tight edge e equals $f_e(d_T) = \ell_e^{-1}(L(d - d_T))$. It is a useful observation, that $L(d_N)$ as a function of d_N is Lipschitz continuous with constant κ. To see this, note that $L'(d_N) = \ell_{e_0}'(f_{e_0}(d_N))\, f_{e_0}'(d_N) \leq \kappa$ for all $d_N \in [d_{\min}, d]$. Thus, when increasing the flow sent over the tollable edges from d_T to $d_T' > d_T$, the flow released by the tight edges can be bounded by

$$\begin{aligned} \sum_{e\in T} \Big(\ell_e^{-1}(L(d - d_T)) - \ell_e^{-1}(L(d - d_T'))\Big) &\leq \sum_{e\in T} \kappa\,(L(d - d_T) - L(d - d_T')) \\ \leq \sum_{e\in T} \kappa^2\,(d - d_T - (d - d_T')) &\leq (|E| - 1)\,\kappa^2(d_T' - d_T), \end{aligned}$$

where we use $T \subsetneq E$. Let $d_T, d_T' \in [0, d - d_{\min}]$ with $d_T' > d_T$ and let $e \in T$ be an edge that is not tight at d_T, i.e., $\ell_e(f_e(d_T)) < L(d - d_T)$. Since we can bound the flow released by the edges that become tight in $[d_T, d_T']$, the inequality $f_e(d_T') \leq |E|\,\kappa^2(d_T' - d_T) + f_e(d_T)$ holds. We obtain

$$\begin{aligned} C_T(d_T') - C_T(d_T) &= \sum_{e\in T} \ell_e(f_e(d_T'))f_e(d_T') - \sum_{e\in T} \ell_e(f_e(d_T))f_e(d_T)) \\ &\leq \sum_{e\in T} \Big(\ell_e\big(f_e(d_T) + |E|\kappa^2(d_T' - d_T)\big)\big(f_e(d_T) + |E|\kappa^2(d_T' - d_T)\big) - \ell_e(f_e(d_T))f_e(d_T)\Big) \\ &= \sum_{e\in T} \Big(\ell_e\big(f_e(d_T) + |E|\kappa^2(d_T' - d_T)\big)\big(|E|\kappa^2(d_T' - d_T)\big) \\ &\quad + \ell_e(f_e(d_T) + |E|\kappa^2(d_T' - d_T))f_e(d_T) - \ell_e(f_e(d_T))f_e(d_T)\Big) \\ &\leq \kappa^2(|E| - 1)|E|K(d_T' - d_T) + \kappa^3 d(|E| - 1)|E|(d_T' - d_T) = (K + \kappa d)(|E| - 1)|E|(d_T' - d_T)\kappa^2, \end{aligned}$$

for all $0 \leq d_T < d_T' \leq d - d_{\min}$. Together with the estimations for C_N we obtain that the Lipschitz constant of C_E does not exceed $|E|^2\kappa^2(K + kd)$. □

The Lipschitz assumptions on the latencies and their inverse are necessary in the sense that if one of these assumptions is dropped, then there are instances for which the combined cost function is not Lipschitz continuous.

As a corollary of this result, we obtain that it suffices to calculate the combined cost function for a polynomial number of sampling points in order to get a good approximation of its maximum. In the remainder of this section, we will show how to approximately calculate $C_N(d_N)$ and $C_T(d_T)$.

Approximating $C_N(d_N)$. We first focus on the computation of the cost of the non-tollable edges. In order to approximate their total cost, we are interested in calculating an ε-approximate Wardrop equilibrium on N with demand d_N, which we define below for the special case of parallel edges.

Definition 3.6 (ε-approximate Wardrop equilibrium)**.** *For $\varepsilon > 0$, a flow f is called an* ε-approximate Wardrop equilibrium*, if $\ell_e(f_e) \leq \ell_{e'}(f_{e'}) + \varepsilon$ for all $e, e' \in E$ with $f_e > 0$.*

As shown by Beckmann et al. [4], the problem of computing a Wardrop equilibrium with given demand in general networks can be reformulated as a convex optimization problem and hence can be solved within arbitrary precision by the ellipsoid method. However, the approximate computation of a Wardrop equilibrium with the Ellipsoid method does not necessarily give an ε-approximate Wardrop equilibrium, see the discussion in [15]. We here devise a combinatorial algorithm that computes an ε-approximate Wardrop equilibrium and runs in polynomial time.

Our first observation is that the common latency L of all flow carrying edges in a Wardrop equilibrium with demand d lies between $L_- = \min_{e\in E} \ell_e(0)$ and $L_+ = \min_{e\in E} \ell_e(0) + \kappa d$, where the upper bound is due to the joint Lipschitz constant κ of all latency functions. The main algorithmic idea for the computation of an ε-approximate Wardrop equilibrium is to guess the correct value of L via a binary search between L_- and L_+. For a given value of L, we compute for every edge $e \in E$ a flow value f_e so that the induced latency $\ell_e(f_e)$ approximately matches the common latency L, that is, we approximate $\ell_e^{-1}(L)$. This can be done with binary search of the flow value f_e in the interval $[0, d]$. When the sum of the flow values approximately matches the demand d_N, we have computed an approximate Wardrop equilibrium. The details of this procedure can be found in Algorithm 1.

Proposition 3.7. *Algorithm 1 terminates in time $\mathsf{poly}(|N|, \log\kappa, \log d_N, \log 1/\varepsilon)$ and computes an ε-approximate Wardrop equilibrium on N with demand d_N.*

Proof. If $d_N \leq \frac{\varepsilon}{2\kappa}$, one can simply compute an ε-approximate Wardrop equilibrium by assigning the total demand to the edge with the lowest offset cost. So we assume throughout this proof that $d_N > \frac{\varepsilon}{2\kappa}$.

We start showing that Algorithm 1 terminates in time $\mathsf{poly}(|N|, \log\kappa, \log d_N, \log 1/\varepsilon)$. Since a Wardrop equilibrium always exist (cf. Beckmann et al. [4]), there are $L \in [L_-, L_+]$ and a flow g_N with demand d_N such that $\ell_e(g_e) = L$ for all $e \in N$ with $g_e > 0$. We claim that $L \in [L_-, L_+]$ is an invariant during the run of the algorithm. This is obviously true when the algorithm starts. To see that this property is preserved in each iteration of the algorithm, let L_-^i, L_+^i, L^i and f_e^i denote the values of L_-, L_+, L, and f_e after the i-th iteration. Suppose that the invariant is satisfied after the $(i-1)$-st iteration and consider the i-th iteration. We distinguish two cases. If $d' = \sum_{e\in N} f_e^i < d_N$, then, there is an edge $e \in N$ with $f_e^i < g_e$ and, thus, also

Algorithm 1: Computation of an ε-approximate Wardrop equilibrium on parallel edges.

```
   Input: Latency functions $(\ell_e)_{e\in N}$ with Lipschitz constant $\kappa$, demand $d_N$, accuracy $\varepsilon$.
   Output: $\varepsilon$-approximate Wardrop equilibrium $f$.
1  $L_- \leftarrow \min_{e\in N} \ell_e(0)$, $L_+ \leftarrow L_- + \kappa d$, $d' \leftarrow 0$;
2  while $d' \notin [d_N, d_N + \frac{\varepsilon}{2\kappa}]$ do
3      $L \leftarrow \frac{L_+ - L_-}{2}$;
4      foreach $e \in N$ do $f_e \leftarrow 0$, if $\ell_e(0) \geq L$, and $f_e \in [\ell_e^{-1}(L), \ell_e^{-1}(L) + \frac{\varepsilon}{4\kappa^3|N|}]$, otherwise;
5      $d' \leftarrow \sum_{e\in N} f_e$;
6      if $d' > d_N + \frac{\varepsilon}{2\kappa}$ then $L_+ \leftarrow L$;
7      if $d' < d_N$ then $L_- \leftarrow L$;
8  end
9  Choose some edges $N' \subseteq N$ and reduce the flow on them in total by $d' - d_N$;
```

$\ell_e(f_e^i) < \ell_e(g_e) = L$. Using $f_e^i \in [\ell_e^{-1}(L^i), \ell_e^{-1}(L^i) + \frac{\varepsilon}{4\kappa^3|N|}]$ and the Lipschitz continuity of ℓ_e, we obtain $L^i \leq \ell_e(f_e^i) \leq L^i + \frac{\varepsilon}{4\kappa^2|N|}$. Hence, $L^i < L$ and since we set $L_-^i = L^i$ in line 7, we also have $L \in [L_-^i, L_+^i]$.

If, on the other hand, $\sum_{e\in E} f_e^i > d_N + \frac{\varepsilon}{2\kappa}$, then by the pigeon hole principle, there is at least one edge $e \in N$ with $f_e^i > g_e + \frac{\varepsilon}{2\kappa|N|}$. Hence, $\ell_e(f_e^i) > \ell_e(g_e + \frac{\varepsilon}{2\kappa|N|}) \geq \ell_e(g_e) + \frac{\varepsilon}{2\kappa^2|N|} \geq L + \frac{\varepsilon}{2\kappa^2|N|}$. Together with $L^i \leq \ell_e(f_e^i) \leq L^i + \frac{\varepsilon}{4\kappa^2|N|}$, we obtain $L^i \geq L + \frac{\varepsilon}{4\kappa^2|N|}$. As we set $L_+^i = L^i$ in line 6, we have $L \in [L_-^i, L_+^i]$, as claimed.

We now show that the algorithm terminates for any $L \in [L, L + \frac{\varepsilon}{4\kappa^2|N|}]$. Suppose that in the i-th iteration, we have $L^i \in [L, L + \frac{\varepsilon}{4\kappa^2|N|}]$. For the flow f_N^i with $f_e^i \in [\ell_e^{-1}(L^i), \ell_e^{-1}(L^i) + \frac{\varepsilon}{4\kappa^3|N|}]$ for all $e \in N$, we get

$$f_e^i \leq \ell_e^{-1}(L) + \frac{\varepsilon}{4\kappa^3|N|} \leq \ell_e^{-1}\Big(L + \frac{\varepsilon}{4\kappa^2|N|}\Big) + \frac{\varepsilon}{4\kappa^3|N|} \leq g_e + \frac{\varepsilon}{2\kappa|N|}$$

for all $e \in N$ with $f_e > 0$ using that ℓ_e^{-1} is Lipschitz continuous with constant κ. Therefore, $d_N \leq \sum_{e\in N} f_e^i \leq d_N + \frac{\varepsilon}{2\kappa}$ and we derive that the while loop is left.

To bound the number of iterations, note that $L_-^0 = \min_{e\in N} \ell_e(0)$ and $L_+^0 = L_- + \kappa d$, thus, $L_+^0 - L_-^0 = \kappa d$. In each iteration, the length of the interval $L_+^i - L_-^i$ is halved, therefore $L_+^i - L_-^i = \kappa d_N \,/\, 2^i$. Because of the above invariant, we have $[L, L + \frac{\varepsilon}{4\kappa^2|N|}] \subseteq [L_-^i, L_+^i]$ in each iteration of the algorithm. Thus, the binary search stops at the latest when $L_+^i - L_-^i \leq \frac{\varepsilon}{4\kappa^2|N|}$, that is, after at most $\lceil \log(4\kappa^3 d_N|N| \,/\, \varepsilon) \rceil$ iterations. Each iteration involves a binary search for each edge, thus the total time complexity is in $\mathsf{poly}(|N|, \log\kappa, \log d_N, \log 1/\varepsilon)$.

We proceed showing that the thus computed flow constitutes an ε-Wardrop equilibrium. To this end, note that $f_e \in [\ell_e^{-1}(L), \ell_e^{-1}(L) + \frac{\varepsilon}{4\kappa^3|N|})$ for all $e \in N$ with $f_e > 0$, and $\ell_e(0) > L$ for all $e \in N$ with $f_e = 0$ when the while loop is left. The final normalization in line 9 reduces the flow on edges $e' \in N'$ by at most $\frac{\varepsilon}{2\kappa}$, thus, the latency of every edge $e' \in N'$ is at least $L - \frac{\varepsilon}{2}$. Let $e \in N \setminus N'$ with $f_e > 0$ be arbitrary. Then, $L \leq \ell_{e'}(f_{e'}) \leq L + \frac{\varepsilon}{4\kappa^2|N|}$. Using that the latency of every other edge is at least $L - \frac{\varepsilon}{2}$, we have established that f is an ε-approximate Wardrop equilibrium. □

The runtime $\mathsf{poly}(|N|, \log \kappa, \log d_N, \log 1/\varepsilon)$ of Algorithm 1 is a fundamental improvement compared to the algorithm proposed by Fabrikant et al. [15] which runs on arbitrary single-commodity networks but requires time $\mathsf{poly}(|N|, \kappa, d_N, 1/\varepsilon)$. Moreover, the authors consider a slightly weaker notion of ε-approximate Wardrop equilibria. We continue showing that the cost of an ε-approximate Wardrop equilibrium is indeed an approximation of the cost of the exact Wardrop equilibrium.

Proposition 3.8. *Let g and f be a Wardrop equilibrium and an ε-approximate Wardrop equilibrium on N with demand d_N, respectively. Then, $C(g) - \varepsilon d_N \leq C(f) \leq C(g) + \varepsilon d_N$.*

Proof. Since $\sum_{e \in N} f_e = \sum_{e \in N} g_e = d_N$, there is an edge $e' \in N$ with $g_{e'} > 0$ and $g_{e'} \geq f_{e'}$ and an edge $e'' \in N$ with $f_{e''} > 0$ and $f_{e''} \geq g_{e''}$. We obtain

$$C(f) = \sum_{e \in N} \ell_e(f_e) f_e \leq (\ell_{e'}(f_{e'}) + \varepsilon) d_N \leq \ell_{e'}(g_{e'}) d_N + \varepsilon d_N = C(g) + \varepsilon d_N,$$
$$C(f) = \sum_{e \in N} \ell_e(f_e) f_e \geq (\ell_{e''}(f_{e''}) - \varepsilon) d_N \geq \ell_{e''}(g_{e''}) d_N - \varepsilon d_N = C(g) - \varepsilon d_N,$$

which proves the result. □

As a corollary of this result, we can calculate a flow f_N on N with $C(f_N) - \varepsilon d_N \leq C_{WE} \leq C(f_N) - \varepsilon d_N$ in $\mathsf{poly}(|N|, \log \kappa, \log d_N, \log 1/\varepsilon)$ time, where C_{WE} is the cost of the Wardrop equilibrium. Using that the common latency L of the flow carrying edges in the Wardrop equilibrium fulfills the equation $L = C \,/\, d_N$, we can compute the common latency with an additive error of ε.

Approximating $C_T(d_T)$. In this section, we show how to approximately compute for given $d_N \in [d_{\min}, d]$ the cost of the optimal flow on T with demand $d_T = d - d_N$ that obeys the latency restriction $\ell_e(f_e) \leq L(d_N)$ for all $e \in T$. That is, for a given demand distribution (d_N, d_T) we find an approximate solution of $\min_{f_T \in F_T(d_T, L(d-d_T))} C(f_T)$. Since the demand distribution is fixed, we will use the L as shorthand for $L(d_N)$ in the sequel.

It is well known that a flow is an optimal solution of the problem $\min_{f_T \in F_T(d_T, \infty)} C(f_T)$ *without* latency restriction if and only if it is at Wardrop equilibrium with respect to the marginal edge latencies ℓ_e^* defined as $\ell_e^*(f_e) = \ell_e(f_e) + \ell_e'(f_e)\, f_e$ for all $f_e \geq 0$ and all $e \in E$, see Beckmann et al. [4] and Roughgarden and Tardos [32]. In particular, in an optimal solution, the marginal latencies of all flow carrying edges are equal. As the problem $\min_{f_T \in F_T(d_T, L)} C(f_T)$ *with* latency restriction remains a convex program, we obtain the following necessary and sufficient conditions (KKT conditions) for an optimal solution.

Lemma 3.9. *f_T is an optimal solution of $\min_{f_T \in F_T(d_T, L)} C(f_T)$ if and only if there is $L^* \in \mathbb{R}_{\geq 0}$ with*

1. *$\ell_e^*(f_e) = L^*$ for all $e \in T$ with $f_e > 0$ and $\ell_e(f_e) < L$,*
2. *$\ell_e^*(f_e) \leq L^*$ for all $e \in T$ with $f_e > 0$ and $\ell_e(f_e) = L$,*
3. *$\ell_e^*(f_e) \geq L^*$ for all $e \in T$ with $f_e = 0$.*

Algorithm 2: Computation of an approximate solution of $\min_{f_T \in F_T(d_T, L)} C(f_T)$.

```
Input: Marginal cost functions $(\ell_e^*)_{e \in T}$ with Lipschitz constant $\kappa_*$ and bound $L$,
       demand $d_T$, accuracy $\varepsilon$.
Output: Approximate solution of $\min_{f_T \in F_T(d_T, L)} C(f_T)$.
$L_-^* \leftarrow \min_{e \in T} \ell_e^*(0)$, $L_+^* \leftarrow L_-^* + 2\kappa_* d_T$, $d' \leftarrow 0$;
while $d' \notin [d_T - \frac{\varepsilon}{2\kappa_*}, d_T]$ do
    $L^* \leftarrow \frac{L_+^* - L_-^*}{2}$;
    $T \leftarrow \emptyset$;                  // almost tight edges for latency bound $L^*$
    foreach $e \in T$ do
        if $\ell_e^*(0) < L^*$ then
            Find $x \in [(\ell_e^*)^{-1}(L^*) - \frac{\varepsilon}{2\kappa_*^3 |T|}, (\ell_e^*)^{-1}(L^*)]$ and set $f_e \leftarrow x$;
            if $\ell_e(f_e) > L - \frac{\varepsilon}{4\kappa_*^3 |T|}$ then
                $T \leftarrow T \cup \{e\}$;
                Find $x$ with $L - \frac{\varepsilon}{2\kappa_*^3 |T|} < \ell_e(x) \leq L - \frac{\varepsilon}{4\kappa_*^3 |T|}$ and set $f_e \leftarrow x$;
            end
        else
            $f_e \leftarrow 0$;
        end
    end
    $d' \leftarrow \sum_{e \in T} f_e$;
    if $d' > d_T$ then $L_+^* \leftarrow L^*$;
    if $d' < d_T - \frac{\varepsilon}{2\kappa_*}$ then $L_-^* \leftarrow L^*$;
end
Distribute the remaining $d_T - d'$ units of flow on the edges $e \in T \setminus T$ such that
$f_e \leq L - \frac{\varepsilon}{4\kappa_*^3 |T|}$ for all $e \in T \setminus T$;
```

Our basic algorithmic idea is similar to that of Algorithm 1; we use binary search to find L^* such that the above optimality conditions are almost satisfied. There is, however, a subtle difficulty as the common latency L belongs to the input of the definition of C_T, while we are only able to compute an approximate value for L. We resolve this difficulty by satisfying the latency restriction $\ell_e(f_e) \leq L$ with some gap to be specified later. Note that, since the latency functions and their derivatives are Lipschitz continuous with constant κ, the marginal latency functions are Lipschitz continuous with constant $\kappa_* = (2 + d)\kappa$. To see this, note that

$$|\ell_e^*(x) - \ell_e^*(y)| \leq |\ell_e(x) - \ell_e(y)| + |\ell_e'(x)x - \ell_e'(x)y + \ell_e'(x)y - \ell_e'(y)y| \leq 2\kappa|x - y| + d\kappa|x - y|.$$

We proceed proving correctness of Algorithm 2.

Proposition 3.10. *If $F_T(d_T, L - \frac{\varepsilon}{2\kappa_*^3 |T|}) \neq \emptyset$, Algorithm 2 terminates in time $\mathsf{poly}(|T|, \log \kappa, \log d_T, \log 1/\varepsilon)$ and computes a flow $f_T \in F_T(d_T, L - \frac{\varepsilon}{4\kappa_*^3 |T|})$ with social cost $C(f_T) \leq C_{opt} + \varepsilon d_T$, where $C_{opt} = \min_{g_T \in F_T(d_T, L)} C(g_T)$.*

Proof. We first show the claimed runtime. Using Lemma 3.9, there are $L^* \in [L^*_-, L^*_+]$ and an optimal solution g_T satisfying $\ell^*_e(g_e) = L^*$ for all $e \in T$ with $g_e > 0$ and $\ell_e(g_e) < L$, $\ell^*_e(g_e) \leq L^*$ for all $e \in T$ with $g_e > 0$ and $\ell_e(g_e) = L$, and $\ell^*_e(g_e) \geq L^*$ for all $e \in T$ with $g_e = 0$. We claim that $L^* \in [L^*_-, L^*_+]$ is an invariant during the run of the algorithm. This is obviously true before the first iteration. We denote by $L^{*i}_-, L^{*i}_+, L^{*i}, f^i_e, T^i$ the values of $L^*_-, L^*_+, L^*, f_e, T$ after the i-th iteration. Let us assume that $L^* \in [L^{*i-1}_-, L^{*i-1}_+]$. For the i-th iteration, we distinguish two cases. If $d' = \sum_{e\in T} f^i_e > d_T$, then there is an edge $e \in T$ with $f^i_e > g_e > 0$. Using that the marginal cost functions are non-decreasing, we have $L^{*i-1} \geq \ell^*_e(f_e) \geq \ell^*_e(g_e) \geq L^*$. As we set $L^{*i}_+ = L^{*i-1}$ in line 17, the invariant remains valid after the i-th iteration.

If, on the other hand, we have $d' = \sum_{e\in T} f^i_e < d_T - \frac{\varepsilon}{2\kappa_*}$, there is $e \in T$ with $f^i_e < g_e - \frac{\varepsilon}{2\kappa_*|T|}$ and thus $\ell_e(f^i_e) < \ell_e(g_e) - \frac{\varepsilon}{2\kappa_*\kappa|T|} < \ell_e(g_e) - \frac{\varepsilon}{2\kappa_*^2|T|}$. This implies in particular that $e \notin T^i$ since $\ell_e(f^i_e) < \ell_e(g_e) - \frac{\varepsilon}{2\kappa_*^2|T|} \leq L - \frac{\varepsilon}{2\kappa_*^2|T|}$. As $e \notin T^i$, we have $f_e \in [(\ell^*_e)^{-1}(L^*) - \frac{\varepsilon}{2\kappa_*^3|T|}, (\ell^*_e)^{-1}(L^*)]$, and thus, $L^* - \frac{\varepsilon}{2\kappa_*^2|T|} \leq \ell^*_e(f_e) \leq L^*$. We obtain $L^* \leq \ell^*_e(f_e) + \frac{\varepsilon}{2\kappa_*^2|T|} < \ell^*_e(g_e) \leq L^*$. As we set $L^{*i}_- = L^{*i-1}$ in line 18, the invariant remains valid after the i-th iteration.

We proceed showing that the algorithm terminates for any $L^* \in [L^* - \frac{\varepsilon}{4\kappa_*^2|T|}, L^*]$. To this end, suppose that $L^{*i} \in [L^* - \frac{\varepsilon}{4\kappa_*^2|T|}, L^*]$ is arbitrary in the i-th iteration. For the flow f^i_e on edge e, we calculate

$$f^i_e \geq (\ell^*_e)^{-1}(L^*) - \frac{\varepsilon}{4\kappa_*^3|T|} \geq (\ell^*_e)^{-1}\Big(L^* - \frac{\varepsilon}{4\kappa_*^2|T|}\Big) - \frac{\varepsilon}{4\kappa_*^3|T|} \geq g_e - \frac{\varepsilon}{2\kappa_*|T|}$$

for all $e \in T \backslash T$ with $f_e > 0$. Moreover, for all $e \in T$, we obtain $f_e > L - \frac{\varepsilon}{2\kappa_*^3|T|} \geq g_e - \frac{\varepsilon}{2\kappa_*^3|T|}$. Hence, $d_T \geq \sum_{e\in T} f_e \geq d_T - \frac{\varepsilon}{2\kappa_*}$ and the algorithm terminates. Analogously to the proof of Theorem 3.7, this observation proves the number of iterations is in $\mathsf{poly}(\log \kappa, \log d_T, \log 1/\varepsilon)$. Since each iteration requires binary searches for every edge, the total time complexity of Algorithm 2 is in $\mathsf{poly}(|T|, \log \kappa, \log d_T, \log 1/\varepsilon)$.

It remains to show that f_T approximates C_{opt} with the claimed precision. First, note that the final redistribution of flows in line 20 is feasible as we require $F_T(d_T, L - \frac{\varepsilon}{2\kappa_*^3|T|}) \neq \emptyset$ and as all edges $e \in T$ carry more flow than in each solution $g_T \in F_T(d_T, L - \frac{\varepsilon}{2\kappa_*^3|T|})$. Thus, at termination we obtain a constant L^* and a flow f_T with demand d_T that satisfies the inequalities $L^* - \frac{\varepsilon}{2\kappa_*^3|T|} \leq \ell^*_e(f_e) \leq L^* + \frac{\varepsilon}{2}$, if $e \in T \backslash T$, and $\ell^*_e(f_e) < L^*$, if $e \in T$. We will show that f_T is optimal with respect to some shifted latencies. To this end, we define $\tilde{L}^* = L^* + \frac{\varepsilon}{2}$. Furthermore, we set $\delta_e = \tilde{L}^* - \ell^*_e(f_e)$ for all $e \in T \setminus T$ and $\delta_e = L - \ell_e(f_e)$ for all $e \in T$, and define $\tilde{\ell}_e(f_e) = \ell_e(f_e) + \delta_e$ for all $e \in T$ with $f_e > 0$.

By construction, we have $0 < \delta_e \leq \varepsilon$ for all $e \in E$. In addition, we obtain $\tilde{\ell}^*_e(f_e) = \tilde{\ell}_e(f_e) + f_e\ell'_e(f_e) = \tilde{L}^*$ for all $e \in T \setminus T$ and $\tilde{\ell}^*_e(f_e) \leq \tilde{L}^*$ for all $e \in T$. Thus, f minimizes $\sum_{e\in E} \tilde{\ell}_e(f_e) f_e$ under the constraints $\sum_{e\in E} f_e = d_T$ and $\ell_e(f_e) \leq L$. Then,

$$C_{\text{opt}} \;=\; \min_{g_T \in F_T(d_T, L)} \sum_{e\in E} (\tilde{\ell}_e(g_e) \;-\; \delta_e) g_e \;\geq\; \min_{g_T \in F_T(d_T, L)} \sum_{e\in E} \tilde{\ell}_e(g_e) g_e \;-\; \varepsilon d_T \;=\; C(f) \;-\; \varepsilon d_T,$$

as claimed. □

Approximating $C_E(d_N)$. We are now ready to combine Algorithms 1 and 2 in order to approximate the combined cost function $C_E(d_N)$ for fixed d_N. We proceed by providing an algorithm that takes $\tilde{\varepsilon} > 0$ as input, runs in time $\mathsf{poly}(|E|, \log\kappa, \log d, \log 1/\tilde{\varepsilon})$ and computes $C_E(d_N)$ for any $d_N \in [d_{\min} + \frac{\tilde{\varepsilon}}{8d\kappa_*^2}, d]$ with an additive error of $\frac{9}{32}\tilde{\varepsilon}$.

Lemma 3.11. *For any $\tilde{\varepsilon} > 0$ and $d_N \in [d_{\min} + \frac{\tilde{\varepsilon}}{8d\kappa_*^2}, d]$ one can compute flows f_N and f_T with*

1. $C(f_N) + C(f_T) \le C_E(d_N) + \frac{9}{32}\tilde{\varepsilon}$,
2. $\ell_e(f_e) \le \ell_{e'}(f_{e'})$ *for all $e \in T$ with $f_e > 0$ and $e' \in N$,*
3. $\ell_e(f_e) \le \ell_{e'}(f_{e'}) + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$ *for all $e, e' \in N$ with $f_e > 0$,*

in time $\mathsf{poly}(|E|, \log\kappa, \log d, \log 1/\tilde{\varepsilon})$.

Proof. Using Algorithm 1, we compute an $\frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$-approximate Wardrop equilibrium f_N on the non-tollable edges $e \in N$ with demand d_N in time $\mathsf{poly}(|N|, \log\kappa, \log d_N, \log 1/\tilde{\varepsilon})$. As we have shown in Proposition 3.8, $C(f_N) - \frac{\tilde{\varepsilon}d_N}{32d\kappa_*^3|T|} \le C(g_N) \le C(f_N) + \frac{\tilde{\varepsilon}d_N}{32d\kappa_*^3|T|}$, where $C(g_N)$ is the cost of a Wardrop equilibrium with the same demand. In particular, the common latency L of the Wardrop equilibrium g_N satisfies the inequalities

$$\frac{C(f_N)}{d_N} - \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|} \le L \le \frac{C(f_N)}{d_N} + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}.$$

We define $\tilde{L} = \frac{C(f_N)}{d_N} + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$ and obtain $L \in [\tilde{L} - \frac{\tilde{\varepsilon}}{16d\kappa_*^3|T|}, \tilde{L}]$. By definition of $d_{\min}$, we have $F_T(d - d_{\min}, L(d_{\min})) \neq \emptyset$. Let $g_T \in F_T(d - d_{\min}, L(d_{\min}))$ and consider the flow $\tilde{g}_T$ defined as $\tilde{g}_e = \max\{0, g_e - \frac{\tilde{\varepsilon}}{8d\kappa_*^2|T|}\}$ for all $e \in T$. Then,

$$\ell_e(f_e) \le L(d_{\min}) - \frac{\tilde{\varepsilon}}{8d\kappa\kappa_*^2|T|} \le L(d_{\min}) - \frac{\tilde{\varepsilon}}{8d\kappa_*^3|T|}$$

for all $e \in T$ with $f_e > 0$. We derive that for $d_N \ge d_{\min} + \frac{\tilde{\varepsilon}}{8d\kappa_*^2}$, the set of flows $F_T(d_N, L(d_N) - \frac{\tilde{\varepsilon}}{8d\kappa_*^3|T|})$ is non-empty as well. Thus, when we call Algorithm 2 with accuracy $\frac{\tilde{\varepsilon}}{4d}$ and latency restriction $\tilde{L}$ it returns a flow f_T that satisfies $f_T \in F_T(d_T, \tilde{L} - \frac{\tilde{\varepsilon}}{16d\kappa_*^3|T|}) \subseteq F_T(d_T, L)$ and is thus also feasible with respect to the latency restriction $L(d_N)$ of the exact Wardrop equilibrium. In addition, f_T approximates the minimum cost of a flow $g_T \in F_T(d_T, \tilde{L})$ with an additive error of $\frac{\tilde{\varepsilon}}{4}$ (see Proposition 3.10). Thus, the total error when computing $C_E(d_N)$ is at most $\frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|} + \frac{\tilde{\varepsilon}}{4} \le \frac{9\tilde{\varepsilon}}{32}$ and 1. follows. To see 2., note that by construction $\ell_e(f_e) \le \tilde{L} - \frac{\tilde{\varepsilon}}{16d\kappa_*^3|T|}$ for all $e \in T$ with $f_e > 0$. Moreover, since $\tilde{L} - \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|} \le \ell_{e'}(f_{e'}) + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$ for all $e' \in N$, we have $\ell_e(f_e) \le \ell_{e'}(f_{e'})$ for all $e \in N$ with $f_e > 0$. The third claim follows from the fact that f_N is an $\frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$-approximate Wardrop equilibrium. □

We are now ready to state the main result of this section.

Theorem 3.12. *Let $I_s = (G, d, \ell, T)$ be an instance of the toll problem with support constraints, where G is a network of parallel edges. Then, one can compute a toll vector τ with $C(f^{\tau}) \leq C(f^{\tau^*}) + \varepsilon$ in time* $\mathsf{poly}(|E|, K, \kappa, d, 1/\varepsilon)$.

Proof. Let $\tilde{\varepsilon} = \frac{\varepsilon}{|E|^2(K+\kappa d)}$, $\delta = \frac{\varepsilon}{2|E|^2\kappa^2(K+\kappa d)}$ and define the k-th sampling point as $x_k = k\delta$ for $k \in \{0, \ldots, \lceil d/\delta \rceil\}$. Referring to Lemma 3.11, for every sampling point $x_k \geq d_{\min} + \frac{\tilde{\varepsilon}}{8d\kappa_*^2}$, we can approximate the combined cost function $C_E(x_k)$ with an additive error of at most $\frac{9}{32}\tilde{\varepsilon}$ in polynomial time. Let $\tilde{C}_E(x_k)$ denote the result of these computations and let $k^* = \arg\min_{k \in \{0,\ldots,\lceil d/\delta \rceil\}} \tilde{C}_E(x_k)$ be the index of the sampling point that minimizes $\tilde{C}_E$. Moreover, let $x^* = \arg\min_{x \in [d_{\min}, d]} C_E(x)$. Clearly, there is a sampling point x_j with $|x_j - x^*| \leq \max\{\frac{\delta}{2}, \frac{\tilde{\varepsilon}}{8d\kappa_*^2}\}$. By the choice of $\tilde{\varepsilon}$ and δ, we get $C_E(x_j) \leq C_E(x^*) + \frac{\varepsilon}{4}$. Since $\tilde{C}_E$ approximates C_E with an additive error of at most $\frac{9}{32}\tilde{\varepsilon} \leq \frac{9}{32}\varepsilon$, we have

$$\tilde{C}_E(x_{k^*}) \leq \tilde{C}_E(x_j) \leq C_E(x_j) + \frac{9}{32}\varepsilon \leq C_E(x^*) + \frac{17}{32}\varepsilon.$$

Moreover, the flows f_N and f_T used in Lemma 3.11 to calculate $\tilde{C}_E(x_{k^*})$ satisfy the inequalities $\ell_e(f_e) \leq \ell_{e'}(f_{e'})$ for all $e \in T$ with $f_e > 0$ and $e' \in N$, and $\ell_e(f_e) \leq \ell_{e'}(f_{e'}) + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|}$ for all $e, e' \in N$ with $f_e > 0$. We define $\ell_{\min} = \min_{e \in N : f_e > 0}\{\ell_e(f_e)\}$ and set $\tau_e = \ell_{\min} - \ell_e(f_e)$ for all $e \in T$ and $\tau_e = 0$, otherwise. Using the properties of f_N and f_T shown in Lemma 3.11, we have $\tau_e \geq 0$ for all $e \in E$ and

$$\begin{aligned} \ell_e(f_e) + \tau_e &\leq \ell_{e'}(f_{e'}) + \tau_{e'} && \text{for all } e \in T, e' \in N \text{ with } f_e > 0, \\ \ell_e(f_e) + \tau_e &\leq \ell_{e'}(f_{e'}) + \tau_{e'} + \frac{\tilde{\varepsilon}}{32d\kappa_*^3|T|} && \text{for all } e, e' \in N \text{ with } f_e > 0. \end{aligned}$$

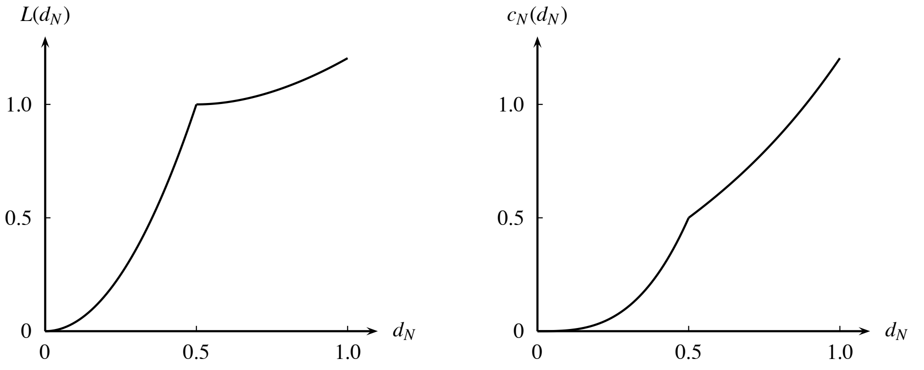
Let g^{τ} denote the Wardrop equilibrium corresponding to toll vector τ. Then, $C(g^{\tau}) \leq \tilde{C}_E(x_{k^*}) + \frac{\tilde{\varepsilon}}{32\kappa_*^3|T|}$. Finally, we obtain $C(g^{\tau}) \leq C_E(x^*) + \frac{18}{32}\varepsilon < C(f^{\tau^*}) + \varepsilon$. □

4 A Polynomial Algorithm for Parallel Edges

In the previous section, we have solved the one-dimensional search problem (3) by discretizing the demand interval *uniformly*. This procedure leads to a runtime that is polynomial in K, κ and d and, thus, exponential in the encoding length of these parameters. In this section, we derive a condition on the latency functions that guarantees that the combined cost function C_E is piecewise convex with at most $|E|$ breakpoints on $[d_{\min}, d]$. We will use this convexity property to derive a polynomial algorithm based on *binary search* on the demand interval (giving a runtime polynomial in $\log K, \log \kappa$, and $\log d$).

4.1 Piecewise Convexity of the Combined Cost Function

In this section, we will derive a condition that guarantees that the combined cost function C_E is piecewise convex on $[d_{\min}, d]$. We start with an example showing that the common latency function function L and hence also the cost of the non-tollable edges can be non-convex.



(a) Common latency L

(b) Cost of the non-tollable edges C_N

Figure 2: Common latency and cost of the non-tollable edges of the network considered in Example 4.1.

Example 4.1. *Consider the instance* $I_s = (G, d, \ell, T)$ *of the toll problem with support constraints, where G is a graph of parallel edges with only two non-tollable edges* $E = \{a, b\}$. *The latencies are set to* $\ell_a(x) = 4x^2$ *and* $\ell_b(x) = x^2 + 1$, *respectively. We observe that for* $d_N \leq 1/2$ *only the first edge is used by the equilibrium flow. For* $d_N > 1/2$ *both edges carry flow such that the equation* $4f_a^2(d_N) = f_b^2(d_N) + 1$ *holds. Using* $f_a(d_N) + f_b(d_N) = d_N$ *and solving for* $f_b(d_N)$ *gives* $f_a(d_N) = d_N - f_b(d_N)$ *and* $f_b(d_N) = 4d_N/3 - (4d_N^2/9 + 1/3)^{1/2}$ *for* $d \geq 1/2$. *The common latency function L, thus, equals*

$$L(d_N) = \begin{cases} 4d_N^2, & \text{if } d_N \leq 1/2 \\ \frac{20}{9}d_N^2 - \frac{8}{3}d_N\sqrt{\frac{4}{9}d_N^2 + \frac{1}{3}} + \frac{4}{3}, & \text{if } d_N > 1/2. \end{cases}$$

The common latency functions L and the cost of the non-tollable edges C_N *are shown in Figure 2.*

While Example 4.1 illustrates that the cost of the non-tollable edges need not be convex even if all latencies are quadratic, we remark that the non-convexity is due to a single *breakpoint*, where a new edge starts to carry flow. Formally, for each edge $e \in N$ the breakpoint d_N^e of edge e is defined as $d_N^e = \max\{d_N \in [0, d] : f_e(d_N) = 0\}$.

Lemma 4.2. *The breakpoint of any* $e \in N$ *fulfills* $d_N^e = \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0))$.

Proof. For a contradiction, assume that there is $e \in N$ with $d_N^e \neq \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0))$. We first consider the case $d_N^e < \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0))$. Fix $d_N \in (d_N^e, \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0)))$ arbitrarily. As $d_N > d_N^e$ we have $f_e(d_N) > 0$. Moreover, since $d_N < \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0))$ and latencies are strictly increasing (cf. Assumption 3.1), there is at least one edge $e' \in N$ with $\ell_{e'}(0) < \ell_e(0)$ and $\ell_{e'}(f_{e'}) < \ell_e(0)$, which contradicts the Wardrop equilibrium conditions.

For the second case, assume $d_N^e > \sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0))$. For an arbitrary demand $d_N \in (\sum_{e' \in E : \ell_{e'}(0) < \ell_e(0)} \ell_{e'}^{-1}(\ell_e(0)), d_N^e]$, we have $f_e(d_N) = 0$. Since latency functions are strictly increasing, there is an edge $e' \neq e$ with $\ell_{e'}(f_{e'}) > \ell_e(0)$, which again contradicts the equilibrium conditions. □

If two edges $e, e' \in N$ have the same latency offset values, that is $\ell_e(0) = \ell_{e'}(0)$, then their breakpoints are equal. In particular, there are at most $|N|$ distinct breakpoints. In the following, we let $N = \{e_1, \dots, e_{|N|}\}$ and assume that the edges of N are ordered according to their latency offsets, that is, $\ell_{e_1}(0) \leq \dots \ell_{e_{|N|}}(0)$. This implies in particular that $d_N^{e_1} \leq \dots \leq d_N^{e_{|N|}}$.

Note that using Lemma 4.2 we can compute the breakpoints with an additive error of ε in time $\mathsf{poly}(\log d, \log 1/\varepsilon)$ using binary search. We proceed showing that the accumulated cost $C_N(d_N) = L(d_N)\, d_N$ of the flow on the non-tollable edges is a convex function of d_N between two neighboring breakpoints. To this end, we first show the following result.

Lemma 4.3. *For every $i \in \{1, \dots, |N| - 1\}$, the common latency function $L : [0, d] \to \mathbb{R}_{\geq 0}$ is convex on $[d_N^{e_i}, d_N^{e_{i+1}}]$.*

Proof. Note that by the definition of breakpoints, the set of edges that carry a positive amount of flow is constant on $(d_N^{e_i}, d_N^{e_{i+1}}]$. We claim that for $d_N \in [d_N^{e_i}, d_N^{e_{i+1}}]$, the flow $f_e(d_N)$ on edge $e \in N$ is described by the following system of differential equations:

$$f'_e(d_N) = \frac{1 \,/\, \ell'_e(f_e(d_N))}{\sum_{a \in N_{i+1}} 1 \,/\, \ell'_a(f_a(d_N))}, \qquad \text{for all } e \in N_{i+1}, \tag{4}$$

$$f'_e(d_N) = 0, \qquad \text{for all } e \in N \setminus N_{i+1}, \tag{5}$$

where $N_{i+1} = \{e \in E : f_e(d_N^{e_{i+1}}) > 0\}$. The initial values $f_e(d_N^{e_i})$ of the above system of differential equations are given by the unique Wardrop equilibrium flow with demand $d_N^{e_i}$. Note that $\ell'_e(x) \geq 1/\kappa$ since we assume that ℓ_e^{-1} is Lipschitz continuous with constant κ; therefore the right hand sides of equations (4) and (5) are well-defined, continuous in d_N and bounded between 0 and 1 for all $d_N \in [d_N^{e_i}, d_N^{e_{i+1}}]$. Thus, the system (4)-(5) has a solution $f_e : [d_N^{e_i}, d_N^{e_{i+1}}] \to \mathbb{R}_{\geq 0}$.

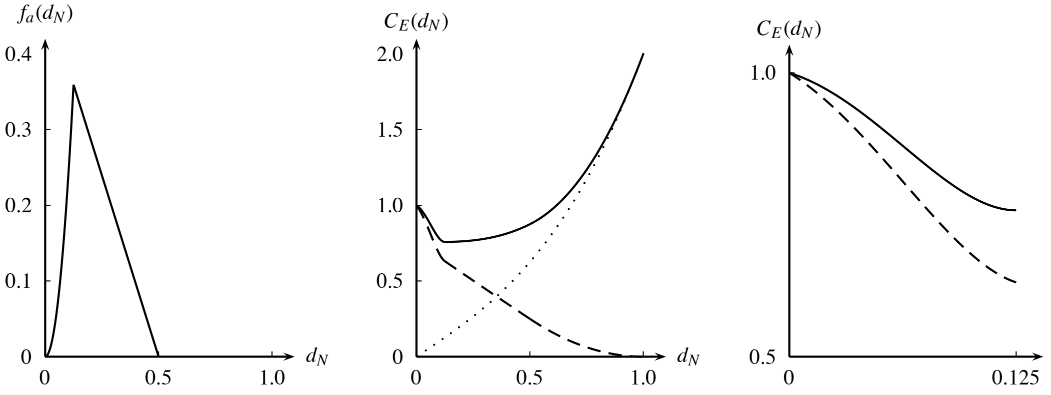
We argue that the solution f_N of the system (4)-(5) constitutes a Wardrop equilibrium flow with demand d_N. We first show that $\sum_{e \in N} f_e(d_N) = d_N$ for all $d_N \in [0, 1]$. To see this, note that $\sum_{e \in N} f_e(0) = 0$ by definition and that $\sum_{e \in N} f'_e(d_N) = 1$ for all $d_N \in [0, 1]$. It remains to show that $\ell_e(f_e(d_N)) = \ell_{e'}(f_{e'}(d_N))$ for all $d_N \in [0, 1]$ and all $e, e' \in N$ with $f_e(d_N) > 0, f_{e'}(d_N) > 0$. Consider the function $h_{e,e'} : [d_N^{e_i}, d_N^{e_{i+1}}] \to \mathbb{R}$ defined as $h_{e,e'}(d_N) = \ell_e(f_e(d_N)) - \ell_{e'}(f_{e'}(d_N))$. Deriving h we obtain $h'(d_N) = \ell'_e(f_e(d_N))f'_e(d_N) - \ell'_{e'}(f_{e'}(d_N))f'_{e'}(d_N) = 0$ for all $d_N \in [d_N^{e_i}, d_N^{e_{i+1}}]$, where we use that f is a solution of the system (4)-(5). We conclude that h is constant. Clearly, for $d_N = 0$ no edge carries flow. This implies together with the observation that $h_{e,e'}$ is constant that $\ell_e(f_e(d_N)) = \ell_{e'}(f_{e'}(d_N))$ for all $d_N \in [0, 1]$ and $e, e' \in N$ with $f_e(d_N) > 0$ and $f_{e'}(d_N) > 0$.

We calculate that $L'(d_N) = \ell'(f_e(d_N)) \cdot f'_e(d_N) = 1 \,/\, (\sum_{a \in N_{i+1}} \frac{1}{\ell'_a(f_a(d_N))})$ and obtain

$$L''(d_N) = -\frac{1}{\left(\sum_{a \in N_{i+1}} \frac{1}{\ell'_a(f_a(d_N))}\right)^2} \cdot \sum_{b \in N_{i+1}} -\frac{\ell''_b(f_b(d_N)f'_b(d_N)}{(\ell'_b(f_b(d_N))^2} = \sum_{b \in N_{i+1}} f'_b(d_N)^3 \ell''_b(f_b(d_N)),$$

which is non-negative for every $d_N \in [d_N^{e_i}, d_N^{e_{i+1}}]$. □

It follows that $C_N(d_N) = L(d_N)\, d_N$ is convex in d_N. Thus, the cost accumulated on the non-tollable edges is convex in d_N between successive breakpoints. Such a result does not hold in general for the cost on the tollable edges as we will show in the following example.



(a) Flow on edge a

(b) Cost

(c) Zoomed cost

Figure 3: (a) Flow on edge a, (b) Cost on the non-tollable edges (dotted), cost on the tollable edges (dashed), and combined cost (solid) of the instance considered in Example 4.4, (c) zooms into the region of non-convexity.

Example 4.4. *Consider the instance $I_s = (G, d, \ell, T)$ on the network G of three parallel edges with demand $d = 1$. Only the first edge e with latency $\ell_e(x) = x^2 + 1$ is non-tollable. The latency functions of the other two (tollable) edges equal $\ell_a(x) = x/23 + 1$ and $\ell_b(x) = x$, respectively. An optimal flow with demand $1 - d_N$ satisfies $2f_b = (2/23)f_a + 1$, if $f_a > 0$ and $f_b > 0$. We conclude that $f_a = 0$ for $d_N \geq 1/2$. For $d_N < 1/2$, we obtain the equation $f_a(d_N) = \frac{23}{48} - \frac{46}{48}d_N$. Since $L(d_N) - \ell_a(d_N) = d_N^2 + \frac{2}{48}x - \frac{1}{48} = (d_N - \frac{1}{8})(d_N + \frac{1}{6})$, the latency restriction of edge a becomes tight for $d_N \leq \frac{1}{6}$. This implies that $f_a(d_N) = \ell_a^{-1}(\ell_e(d_N)) = 23d_N^2$ for $d_N < \frac{1}{8}$. Thus, the optimal flows equal*

$$f_e(d_N) = d_N, \quad f_a(d_N) = \begin{cases} 23d_N^2, & \text{if } 0 \leq d_N < 1/8, \\ 23/48 - 46d_N/48, & \text{if } 1/8 \leq d_N < 1/2\ , \\ 0, & \text{if } 1/2 \leq d_N, \end{cases} \quad f_b(d_N) = 1 - d_N - f_a.$$

The combined cost as a function of d_N equals

$$C_T(d_N) = \begin{cases} (d_N^2 + 1)d_N + 23d_N^2(d_N^2 + 1) + (1 - d_N - 23d_N^2)^2, & \text{if } d_N \leq 1/8, \\ (d_N^2 + 1)d_N + (\frac{23}{48} - \frac{46}{48}d_N)(\frac{49}{48} - \frac{2}{48}d_N) + (\frac{25}{48} - \frac{2}{48}d_N)^2, & \text{if } 1/8 \leq d_N < 1/2, \\ (d_N^2 + 1)d_N + (1 - d_N)^2, & \text{if } 1/2 \leq d_N. \end{cases}$$

This function is depicted in Figure 3. Clearly, neither the cost of the tollable edges nor the combined cost function are convex for small values of d_N.

In light of Example 4.4, we are interested in providing general conditions ensuring piecewise convexity of the cost of the tollable edges (and thus also the combined cost function). Before we give an abstract sufficient condition for (piecewise) convexity below, we first explain the non-convexity arising in Example 4.4. By decreasing d_N, at some point a formerly attractive edge e becomes tight and starts loosing flow according to $f_e(d_N) = \ell_e^{-1}(L(d_N))$. If the function $\ell_e^{-1}(L(d_N))$ is convex, the total flow of non-tight edges grows sublinearly resulting in concave cost (see Fig. 3 for an illustration of this effect). To resolve this issue, we introduce an additional property defined below.

Definition 4.5. *An instance $I_s = (G, d, \ell, T)$ of the toll problem with support constraint satisfies the* inverse concavity property *if $\ell_e^{-1}(L(d_N))$ is concave in d_N for all $e \in T$.*

In the following lemma, we prove that the inverse concavity property is sufficient for the convexity of $C_T(1 - d_N)$.

Lemma 4.6. *Let I_s be an instance of the toll problem with support constraints that satisfies the inverse concavity property. Then, $C_T(1 - d_N)$ is convex in d_N.*

Proof. We show that the cost of the solution of the convex optimization problem $C_T(d_N) = \min_{f_T \in F_T(1-d_N, L(d_N))} C(f_T)$ is convex in d_N. Clearly, the objective $\sum_{e \in T} \ell_e(f_e) f_e$ is convex in f. Referring to the work of Fiacco and Kyparisis [16, Prop.2.1] it is sufficient to show that the set of flows $F_T(1 - d_N, L(d_N))$ is convex in d_N, that is,

$$\lambda F_T(1-d_N, L(d_N)) + (1-\lambda) F_T(1-\Delta_N, L(\Delta_N)) \subseteq F_T(1-\lambda d_N - (1-\lambda)\Delta_N, L(\lambda d_N + (1-\lambda)\Delta_N)$$

for all $d_N, \Delta_N \in [0, 1]$ and all $\lambda \in (0, 1)$. To see this, let $f_T \in F_T(1 - d_N, L(d_N))$ and let $g_T \in F_T(1 - \Delta_N, L(\Delta_N))$. In particular, $\sum_{e \in T} f_e = 1 - d_N$, $\sum_{e \in T} g_e = 1 - \Delta_N$, which implies that $\sum_{e \in T} \lambda f_e + (1 - \lambda) g_e = 1 - \lambda d_N - (1 - \lambda)\Delta_N$.

Let $e \in T$ be arbitrary and let us first assume that $f_e > 0$ and $g_e > 0$. Then, $\ell_e(f_e) \leq L(d_N)$ and $\ell_e(g_e) \leq L(\Delta_N)$. Thus, $f_e \leq \ell_e^{-1}(L(d_N))$ and $g_e \leq \ell_e^{-1}(L(d_N))$, and we obtain

$$\lambda f_e + (1 - \lambda) g_e \leq \lambda \ell_e^{-1}(L(d_N)) + (1 - \lambda) \ell_e^{-1}(L(\Delta_N)) \leq \ell_e^{-1}(L(\lambda d_N + (1 - \lambda)\Delta_N))),$$

where we use the concavity of $\ell_e^{-1}(L(\cdot))$.

If $f_e = g_e = 0$, then also $\lambda f_e + (1 - \lambda) g_e = 0$ and there is nothing left to show. So let us finally assume that $f_e > 0$ and $g_e = 0$. We have $\lambda f_e + (1 - \lambda) \cdot 0 \leq \lambda \ell_e^{-1}(L(d_N))$ which implies $\lambda f_e + (1 - \lambda) g_e \leq \ell_e^{-1}(L(\lambda d_N)) \leq \ell_e^{-1}(L(\lambda d_N + (1 - \lambda)\Delta_N))$, where we again use the concavity of $\ell_e^{-1}(L(\cdot))$. □

As a corollary of this result, we obtain a polynomial algorithm that solves the toll problem with support constraints for instances in which the graph has only parallel edges and satisfies the inverse concavity property.

Algorithm 3: Binary Search on $[u_N^{e_i}, l_N^{e_{i+1}}]$

```
Input: interval $[u_N^{e_i}, l_N^{e_{i+1}}]$, convex combined cost function $C_E$, accuracy $\tilde{\varepsilon}$, precision $\delta$
Output: $x_0 \in [u_N^{e_i}, l_N^{e_{i+1}}]$ with $C_E(x_0) \leq C_E(x^*) + \frac{\tilde{\varepsilon}}{2}$
$x_0 \leftarrow u_N^{e_i}$, $x_1 \leftarrow l_N^{e_{i+1}}$, $k \leftarrow 0$;
repeat
    $x_{1/2} \leftarrow \frac{x_0 + x_1}{2}$, $x_{1/4} \leftarrow \frac{x_0 + x_{1/2}}{2}$, $x_{3/4} \leftarrow \frac{x_{1/2} + x_1}{2}$, $k \leftarrow k + 1$;
    compute $\frac{\tilde{\varepsilon}}{2^{k+5}}$-approximations $\tilde{C}_E(x_0), \tilde{C}_E(x_{1/4}), \ldots, \tilde{C}_E(x_1)$ of
    $C_E(x_0), C_E(x_{1/4}), \ldots, C_E(x_1)$;
    choose $j = \arg\min_{i=0,1/4,\ldots,1} \tilde{C}_E(x_i)$, and set $x_0 \leftarrow x_{\max\{j-1/4,0\}}$, $x_1 \leftarrow x_{\min\{j+1/4,1\}}$;
until $|x_0 - x_1| < \delta$ ;
```

Theorem 4.7. *Let $I_s = (G, d, \ell, T)$ be an instance of the toll problem with support constraints, where G is a graph of parallel edges that satisfies the inverse concavity property. Then, one can compute a toll vector τ with $C(f^\tau) \leq C(f^{\tau^*}) + \varepsilon$ in time* $\mathsf{poly}(|E|, \log K, \log \kappa, \log d, \log 1/\varepsilon)$.

Proof. Let $\tilde{\varepsilon} = \frac{\varepsilon}{|E|^2(K+\kappa d)}$ and let $\delta = \frac{\tilde{\varepsilon}}{4|E|^2\kappa^2(K+\kappa d)}$. Using the formula shown in Lemma 4.2, for every breakpoint $d_N^e, e \in N$ we calculate a lower bound $l_N^e \in [d_N^e - \delta, d_N^e]$ and an upper bound $u_N^e \in [d_N^e + \frac{\tilde{\varepsilon}}{8d\kappa_*^2}, d_N^e + \delta]$ in time $\mathsf{poly}(\log |E|, \log K, \log \kappa, \log d, \log 1/\varepsilon)$. Let $e_i, e_{i+1} \in N$ be such that $d_N^{e_i} < d_N^{e_{i+1}}$ and there is no other breakpoint between them. Lemmas 4.3 and 4.6 establish that the combined cost function C_E is convex between $d_N^{e_i}$ and $d_N^{e_{i+1}}$, thus, C_E is also convex between $u_N^{e_i}$ and $l_N^{e_{i+1}}$. Note that since $u_N^e > d_N^e \geq d_{\min}$, for every $\tilde{\varepsilon}' \leq \tilde{\varepsilon}$, we can use Lemma 3.11 to approximate C_E with additive error $\frac{9}{32}\tilde{\varepsilon}' < \tilde{\varepsilon}'$. We will use this idea to minimize C_E approximately with the binary search procedure shown in Algorithm 3. We claim that $C_E(x_0) \leq C_E(x^{e_i}) + \tilde{\varepsilon}/2$ at termination of Algorithm 3, where $x^{e_i} = \arg\min\{C_E(x) : x \in [u_N^{e_i}, l_N^{e_{i+1}}]\}$. To prove this result, let $x_0^k, x_{1/4}^k, x_{1/2}^k, x_{3/4}^k, x_1^k$ denote the values of $x_0, x_{1/4}, x_{1/2}, x_{3/4}, x_1$ when line 3 of Algorithm 3 is visited the k-th time. Since $x_1^k - x_0^k \leq (x_1^{k-1} - x_0^{k-1})/2$, the algorithm terminates, thus, there is $T \in \mathbb{N}$ such that line 3 is visited exactly T times. We first show by complete induction that for each $k \in \{1, \ldots, T\}$ there is $\tilde{x}^k \in [x_0^k, x_1^k]$ such that $C_E(\tilde{x}^k) \leq C_E(x^{e_i}) + (1 - \frac{1}{2^k})\frac{\tilde{\varepsilon}}{4}$. This is obviously true for the first time where line 3 is visited. Let us assume that the statement holds for fixed $k \in \mathbb{N}$. We will show that $\min_{x \in [x_0^{k+1}, x_1^{k+1}]} C_E(x) \leq C_E(x^{e_i}) + (1 - \frac{1}{2^{k+1}})\frac{\tilde{\varepsilon}}{4}$. We show this claim only for the case that $\tilde{C}_E(x_{1/2}^k) \leq \min\{\tilde{C}_E(x_0^k), \tilde{C}_E(x_{1/4}^k), \tilde{C}_E(x_{3/4}^k), \tilde{C}_E(x_1^k)\}$, the other four cases are similar. We have $C_E(x_{1/2}^k) \leq \min\{C_E(x_0^k, C_E(x_{1/4}^k), C_E(x_{3/4}^k, C_E(x_1^k))\} + \frac{\varepsilon}{2^{k+4}}$, as $\tilde{C}_E$ is an $\frac{\tilde{\varepsilon}}{2^{k+5}}$-approximation of C_E. Using the convexity of C_E, we obtain $C_E(x) \geq C_E(x_{1/2}^k) + \frac{\tilde{\varepsilon}}{2^{k+4}}$ for all $x \in [x_0^k, x_1^k]$. Note that $x_0^{k+1} = x_{1/4}^k$ and $x_1^{k+1} = x_{3/4}^k$. Thus,

$$\min_{x \in [x_0^{k+1}, x_1^{k+1}]} C_E(x) = \min_{x \in [x_{1/4}^k, x_{3/4}^k]} C_E(x) \leq \min_{x \in [x_0^k, x_1^k]} + \frac{\tilde{\varepsilon}}{2^{k+3}}$$
$$\leq C_E(x^{e_i}) + \Big(1 - \frac{1}{2^k}\Big)\frac{\tilde{\varepsilon}}{4} + \frac{\tilde{\varepsilon}}{2^{k+3}} = C_E(x^{e_i}) + \Big(1 - \frac{1}{2^{k+1}}\Big)\frac{\tilde{\varepsilon}}{4},$$

where the second inequality follows from the induction hypothesis. For the final iteration T, we obtain $|C_E(\tilde{x}^T) - C_E(x_0^T)| \leq \tilde{\varepsilon}/4$ by the Lipschitz constant of C_E proven in Lemma 3.5. Thus, $C_E(x_0^T) \leq C_E(x^{e_i}) + \tilde{\varepsilon}/2$.

Using that $u_N^e - d_N^e < \delta$ and $d_N^e - l_N^e < \delta$ for all $e \in N$, we obtain $C_E(x_0^T) \leq C_E(x^*) + \frac{3}{4}\tilde{\varepsilon}$, where $x^* = \arg\min_{x \in [d_N^{e_i}, d_N^{e_{i+1}}]} C_E(x)$. Moreover, as $\tilde{C}_E(x_0^T)$ is an $2^{-T-5}\tilde{\varepsilon}$-approximation on $C_E(x_0^T)$, we have $\tilde{C}_E(x_0^T) \leq C_E(x^*) + \frac{3}{4}\tilde{\varepsilon} + 2^{-5}\tilde{\varepsilon}$. Referring to Lemma 3.11, the flows f_T and f_N used to compute $\tilde{C}_E(x_0^T)$ satisfy $\ell_e(f_e) \leq \ell_{e'}(f_{e'})$ for all $e \in T$ with $f_e > 0$ and $e' \in N$ and $\ell_e(f_e) \leq \ell_{e'}(f_{e'}) + \frac{1}{2^6} \cdot \frac{\tilde{\varepsilon}}{32dk_*^3|T|}$. We define $\ell_{\min} = \min_{e \in N : f_e > 0}\{\ell_e(f_e)\}$ and set $\tau_e = \ell_{\min} - \ell_e(f_e)$ for all $e \in T$ and $\tau_e = 0$, otherwise. Applying the same arguments as in the proof of Theorem 3.12 that the Wardrop equilibrium g^τ induced by τ satisfies the inequality

$$C(g^\tau) \leq C_E(x^*) + \frac{3}{4}\tilde{\varepsilon} + 2^{-4}\tilde{\varepsilon}.$$

Finally, computing the toll vector τ between every two consecutive breakpoints and taking the one that gives the best approximation on the combined cost function establishes the result.

It is left to show that the computation of the ε-optimal toll vector can be done in time $\mathsf{poly}(|E|, \log K, \log \kappa, \log d, \log 1/\varepsilon)$. Note that we need at most $|E|$ binary searches between $d_N^{e_i}$ and $d_N^{e_{i+1}}$ and that for each binary search with Algorithm 3 a logarithmic number of iterations suffice. Thus, in each binary search we have $1/\tilde{\varepsilon} \in \mathsf{poly}(|E|, K, \kappa, d, 1/\varepsilon)$. Referring to Lemma 3.11, each approximation of C_E can be done in time $\mathsf{poly}(|E|, \log K, \log \kappa, \log d, \log 1/\varepsilon)$. This observation together with the fact that we call Algorithm 3 at most $|E|$ times gives the claimed runtime. □

Remark 4.8. *Along the same lines we obtain that $C_E(1 - d_N)$ is piecewise convex with polynomially many breakpoints as long as $\ell_e^{-1}(L(d_N))$ is piecewise concave with polynomially many breakpoints for all $e \in T$. We will refer to this weaker condition as the piecewise inverse concavity property.*

4.2 Instances with the Piecewise Inverse Concavity Property

We proceed identifying sets of cost functions $\mathcal{L}$ such that every instance $I = (G, d, \ell, T)$ with $\ell_e \in \mathcal{L}$ for all $e \in E$ satisfies the piecewise inverse concavity property. To this end, let $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a convex, strictly increasing and twice differentiable function, and define

$$\mathcal{L}(h) = \{\ell : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} : \ell(x) = h(\alpha x + \beta) \text{ for all } x \geq 0, \text{ where } \alpha > 0, \beta \geq 0\}.$$

Theorem 4.9. *Let $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a convex, strictly increasing and twice differentiable function and let $I_s = (G, d, \ell, T)$ be an instance of the toll problem with support constraints, where G is a graph of parallel edges and $\ell_e \in \mathcal{L}(h)$ for all $e \in E$. Then, I satisfies the piecewise inverse concavity property.*

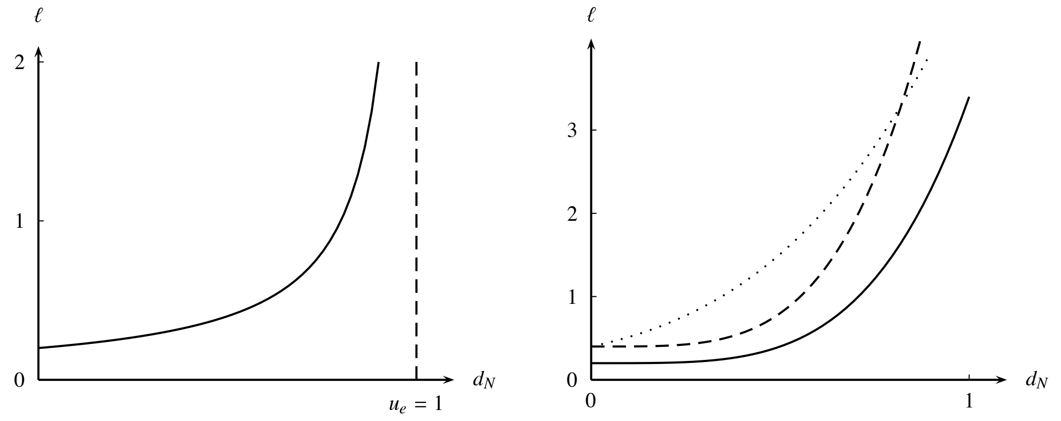
Proof. It suffices to prove that $\ell_e^{-1}(L(d_N))$ is piecewise concave for all $e \in T$. In fact, we will show that $\ell_e^{-1}(L(d_N))$ is concave between two neighbored breakpoints $d_N^{e_i} < d_N^{e_{i+1}}$, $e_i, e_{i+1} \in N$. By definition, the set of non-tollable edges that carry flow is constant on $(d_N^{e_i}, d_N^{e_{i+1}}]$. We will denote this set by N_{i+1}. Referring to Lemma 4.3, the flow of edge $e \in N_{i+1}$ as a function of d_N is described by the differential equation

$$f_e'(d_N) = \frac{1/\ell_e'(f_e(d_N))}{\sum_{e' \in N_{i+1}} 1 \,/\, \ell_{e'}'(f_{e'}(d_N))} = \frac{1\,/(h'(\alpha_e f_e(d_N) + \beta_e)\,\alpha_e)}{\sum_{e' \in N_{i+1}} 1\,/\,(h'(\alpha_{e'} f_{e'}(d_N) + \beta_{e'})\,\alpha_{e'})}. \tag{6}$$

Since h is strictly increasing, the equilibrium condition $h(\alpha_e f_e(d_N) + \beta_e) = h(\alpha_{e'} f_{e'}(d_N) + \beta_{e'})$ implies $\alpha_e f_e(d_N) + \beta_e = \alpha_{e'} f_{e'}(d_N) + \beta_{e'}$ for all $e, e' \in N_{i+1}$ and all $d_n \in [d_N^{e_i}, d_N^{e_{i+1}}]$. Together with (6), we obtain $f_{e'}'(d_N) = \left(\sum_{e \in N_{i+1}} \alpha_{e'} \,/\, \alpha_e\right)^{-1}$ for all $e' \in N_{i+1}$. Thus, the flow of every edge $e' \in N_{i+1}$ is linear in d_N for $d_N \in [d_N^{e_i}, d_N^{e_{i+1}}]$. Now, consider an arbitrary tollable edge $e \in T$. Since $\ell_e \in \mathcal{L}(h)$, we can find $\alpha_e, \beta_{e'} \geq 0$ such that $\ell_e(x) = h(\alpha_e x + \beta_e)$ for all $x \geq 0$. This implies that $\ell_e^{-1}(y) = (h^{-1}(y) - \beta_e) \,/\, \alpha_e$. We have established

$$\ell_e^{-1}(L(d_N)) = \ell_e^{-1}(\ell_{e'}(f_{e'}(d_N)) = \frac{h^{-1}(h(\alpha_{e'} f_{e'}(d_N) + \beta_e)) - \beta_e}{\alpha_e} = \frac{\alpha_{e'} f_{e'}(x) + \beta_{e'} - \beta_e}{\alpha_e},$$

where $e' \in N_{k+1}$ is arbitrary. Since $f_{e'}$ is linear, the function $\ell_e^{-1}(L(d_N))$ is linear as well. □



(a) $M/M/1$ function

(b) BPR functions

Figure 4: Cost functions frequently used in the transportation literature. (a) $M/M/1$ function with parameters $t_e = 0.2$ and $u_e = 1$. (b) Plotted with solid line BPR function ℓ_e with parameters $t_e = 0.2, u_e = 0.5$, and $b_e = 1$; in dashed line the same function with $t_e = 0.4$. The dotted function shows a latency in $\mathcal{L}(\ell_e)$ with free flow travel time 0.4.

As an application of this result, we show that two important classes of functions considered in the networking and transportation literature (cf. Sheffi [33]) satisfy the assumptions of Theorem 4.9. The important class of $M/M/1$ functions is defined in terms of the free flow travel time $t_e > 0$ and a positive capacity $u_e > 0$. The latency is then given as $\ell_e(x) = \frac{t_e u_e}{u_e - x}$, see Figure 4 (a). Let $I_s = (G, d, \ell, T)$ be an instance of the toll problem with support constraints, where all latencies are $M/M/1$ functions, possibly with edge-specific free flow travel times and capacities. It is not hard to see that $\{\ell_e : e \in E\} \subseteq \mathcal{L}(h)$, where $h(x) = 1 / (T - x)$ and $T = \max_{e \in E} 1 / t_e$. Thus, any instance of the toll problem with support constraints with $M/M/1$ latency functions has the piecewise inverse concavity property. Note that we can effectively bound the flows on every edge in order to obtain a finite Lipschitz constant and a bound on the latencies as required in Assumptions 3.1 and 3.2, see the discussion in Remark 3.3.

Another important class of latency functions are the Bureau of Public Roads (BPR) functions defined as $\ell_e(x) = t_e \cdot (1 + b_e \cdot (\frac{x}{u_e})^4) = t_e + \frac{t_e b_e}{u_e^4} x^4$, where $b_e > 0$ is an edge-specific bias, see [8]. It is not possible to find a function h such that the set of BPR functions is contained in $\mathcal{L}(h)$. However, with the choice $h(x) = \ell_e$ for some edge $e \in E$ we obtain the freedom of choosing the latencies of the other edges within the set $\mathcal{L}(h) = \{\ell : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0} : \ell(x) = (\alpha x + \beta)^4, \alpha, \beta \geq 0\}$, which allows the modeling of edge-specific latency offsets (also called free flow travel times), see Figure 4 (b). Thus, with slight restrictions on the choice of the bias and capacity, also BPR functions have the piecewise inverse concavity property.

5 Algorithms for General Networks

In this section, we consider the problem of computing tolls on predefined subsets of a multi-commodity network. Multi-commodity instances model arbitrary traffic distributions over a given traffic network. The algorithmic approach that we used for parallel edge instances breaks down for the multi-commodity case. In fact, there is no hope for a polynomial time algorithm as

Algorithm 4: Marginal Cost Toll Computation Algorithm (MCT)

```
Input: instance $I = (G, d, \ell, T)$, minimum toll change $\Delta$
Output: toll vector $\tau = (\tau_e)_{e \in T}$
$f^* \leftarrow$system optimum flow, $\tau_e^{(1)} \leftarrow \ell'_e(f_e^*) f_e^*$ for all $e \in T$, $i \leftarrow 1, c \leftarrow 1$;
repeat
    $f^\tau \leftarrow$ Wardrop equilibrium flow with tolls $\tau^{(i)}$;
    foreach $e \in T$ do
        if $f_e^\tau > f_e^*$ then $\tau_e^{(i+1)} \leftarrow \tau_e^{(i)} + c \cdot \ell'_e(f_e^\tau) f_e^\tau$;
        if $f_e^\tau < f_e^*$ then $\tau_e^{(i+1)} \leftarrow \max\{0, \tau_e^{(i)} - |\tau_e^{(i)} - \tau_e^{(i-1)}|\}$;
    end
    $i \leftarrow i + 1, c \leftarrow c \cdot 0.9$
until $\max_{e \in T} |\tau_e^{(i)} - \tau_e^{(i-1)}| < \Delta$ ;
```

Hoefer et al. [22] proved that the resulting problem is NP-hard even for instances with only two commodities and affine latencies.

In view of this intrinsic hardness, we present in this section three heuristic descent algorithms (for which we do not prove a worst-case performance guarantee) that compute network tolls on a predefined subset of edges of a given instance. These algorithms are based on the algorithmic idea to iteratively increase the toll on those edges $e \in T$ on which the edge flow of the current Wardrop flow exceeds the system optimal edge flow and to decrease the tolls of edges on which it is smaller. The rationale behind this iterative process is to follow a solution trajectory along a gradient descent direction of the objective function. For our algorithms we need to compute a system optimal flow once. In every iteration we then compute a Wardrop equilibrium with respect to the current toll vector. The algorithms (specified below) differ in their specific rules how to adopt the tolls. We will discuss their differences in terms of convergence behavior and solution quality in the subsequent paragraphs.

Marginal Cost Toll Computation Algorithm (MCT). This algorithm initializes the toll values according to the marginal cost pricing strategy, that is $\tau_e = \ell'_e(f_e^*) f_e^*$ for all $e \in T$. Given this initial toll vector, we compute a Wardrop equilibrium. If an edge carries more flow in equilibrium than in the system optimum, the algorithm increases the currently imposed tolls by the marginal cost of the current equilibrium flow on that edge. The actual value by which the toll is increased is multiplied by a *cooling* factor c which decreases exponentially over time ensuring convergence of the algorithm. If the edge carries more flow in equilibrium than in the system optimum, then the absolute difference between the old tolls and new ones is subtracted. Due to the cooling factor c the toll changes vanish over the iterations and the algorithm finally stops when the maximum toll change over all edges is less than a predefined limit Δ. A description of the algorithm is given in Algorithm 4.

Exponential Marginal Cost Difference Toll Computation Algorithm (EMCD). The Exponential Marginal Cost Difference Algorithm (EMCD-Algorithm) calculates the toll value of one

Algorithm 5: Exponential Marginal Cost Difference Toll Computation (EMCD)

```
Input: instance $I = (G, d, \ell, T)$, minimal toll change $\Delta$
Output: toll vector $\tau = (\tau_e)_{e \in T}$
$f^* \leftarrow$system optimum flow, $\tau_e^{(1)} \leftarrow \max\{\Delta, \ell'_e(f_e^*) f_e^*\}$ for all $e \in T$, $i \leftarrow 1, c \leftarrow 1$;
repeat
    $f^\tau \leftarrow$ Wardrop equilibrium flow with tolls $\tau^{(i)}$, $\alpha \leftarrow \max_{e \in T}\{\ell'_e(f_e^\tau) f_e^\tau\}$;
    foreach $e \in T$ do
        $\tau_e^{(i+1)} \leftarrow \tau_e^{(i)} \cdot \exp\Big(\frac{c}{\max\{1,\alpha\}}(\ell'_e(f_e^\tau) f_e^\tau - \ell'_e(f_e^*) f_e^*)\Big)$;
    end
    $i \leftarrow i + 1, c \leftarrow c \cdot 0.9$;
until $\max_{e \in T} \big|\tau_e^{(i)} - \tau_e^{(i-1)}\big| < \Delta$ ;
```

edge by considering the difference between the marginal cost of the actual equilibrium flow and the marginal cost of the optimal flow on that edge. We use this difference as the argument for an exponential function that, due to its special characteristics, yields smooth convergence. A small difference between the two marginal cost values implies only a small change in the toll value. Similar to the MCT-algorithm (Algorithm 4), we choose marginal cost pricing as initial toll values with a small modification $\tau_e = \max\{\Delta, \ell'_e(f_e^*) f_e^*\}$ for all $e \in T$. EMCD terminates after a finite number of iterations due to the embedded cooling factor c. In this way, the rate of toll changes decreases with the number of iterations until the maximal toll change over all edges in one iteration is less than a predefined termination threshold Δ.

Combinatorial Toll Computation Algorithm (CT). In contrast to the previous algorithms, the following algorithm (denoted by CT, see Algorithm 6) is based on a combinatorial approach. Instead of changing the toll on all tollable edges simultaneously, we iteratively increase the toll of the edge with currently highest marginal costs $\ell'_e(f_e^\tau) f_e^\tau$ for which $f_e^\tau > f_e^*$ by a fixed step size Δ. The performance of this straightforward approach strongly depends on the step size. A higher step-size results in faster convergence, but may produce a poor solution quality of the resulting equilibrium. It will turn out that a smaller step-size finds tolls whose induced equilibrium have less travel time, but might cause slow convergence and high computing times. Note that CT always converges as the tolls are monotonically increased.

5.1 Computational Study

Data Sets. We used data sets from the *Transportation Network Test Problems (TNTP)*[1], a database originally set up to provide realistic data for the traffic assignment problem. The data sets are for academic research purposes and consist of several networks for different cities. In addition to the data sets provided by TNTP, we used a quite large *Swiss* instance with $2,210$ nodes, $6,334$ edges, 20 commodities and a demand of $49,975$ and a set of instances of the city of Berlin. The largest Berlin instance by far is *Berlin-Mitte* with $2,953$ edges and a demand of

[1] `http://www.bgu.ac.il/~bargera/tntp`

Algorithm 6: Combinatorial Toll Computation Algorithm (CT)

```
Input: instance $(G, d, \ell, T)$, step size $\Delta$
Output: toll vector $\tau = (\tau_e)_{e \in T}$
$f^*$ ←system optimum flow, $\tau_e \leftarrow 0$ for all $e \in T$;
while $T \neq \emptyset$ do
    $f^\tau \leftarrow$ Wardrop equilibrium flow with tolls $\tau$;
    choose $e \leftarrow \arg\max_{e \in T}(\ell'_e(f_e^\tau) f_e^\tau)$;
    if $f_e^\tau > f_e^*$ then $\tau_e \leftarrow \tau_e + \Delta$ else $T \leftarrow T \setminus \{e\}$;
end
```

$49,974$, followed by the almost equally sized instances *Berlin-Tiergarten* and *Berlin-Prenzlauer Berg* with each about 355 nodes and 760 edges. *Berlin-Friedrichshain* is the smallest network instance of the four, but still contains nine times the number of nodes and seven times the number of edges of the Sioux Falls network. For each instance, a trip file specifies the commodities and demands. The network file specifies parameters such as the length, free flow travel time and the capacity of every edge from which we constructed the respective BPR functions presented in Section 4.2.

We solved the underlying traffic assignment problems (Wardrop equilibrium and system optimum) with up to a precision of 0.01% using a variant (CMCF developed by Jahn et al. [23]) of the Frank-Wolfe algorithm [18]. The CPU running times for the computation of each system optimum and Wardrop equilibrium were in the range of less than 1 second for the Sioux Falls instance and about 6 minutes for the Swiss instance.

Relative Price of Anarchy. After having computed the system optimum and Wardrop equilibrium for each instance, we obtain different magnitudes of the price of anarchy depending on the instance. We use the *relative price of anarchy*, defined as $(C(f) - C(f^*))/C(f^*)$ as an efficiency measure for selfish flows. The Berlin instances show remarkable differences in the relative price of anarchy. Without tolls, the Berlin-Friedrichshain instance with about 9.41% has the largest relative price of anarchy, followed by Berlin-Mitte (6.72%), Berlin-Prenzlauer Berg (4.85%) and Berlin-Tiergarten (2.78%). An overview about the instances, including network parameters and actual values of the total travel time of the system optimum, Wardrop equilibrium, the resulting relative price of anarchy, and the computation times are depicted in Table 1.

Tollable Edge-Set Selection. We used subsets of tollable edges of cardinality $1, 2, 5, 10, 25$ and 50. The set of tollable edges were chosen as follows. We first computed a system optimal flow f^* and a Wardrop equilibrium f of the instance without tolls and then ordered the edges in descending order in terms of their marginal cost $\ell_e(f_e)f_e$ with $f_e > f_e^*$, $e \in T$. From this list, we chose in decreasing order the first 1, 2, 5, 10, 25 and 50 tollable edges. In total we performed 144 different toll computations. A detailed discussion of the edge selection problem is presented in Section 6, where we formally state and investigate the mathematical problem of selecting the best subset of tollable edges subject to a cardinality constraint.

	network topology				travel times			computation	
Instance	n	m	k	$\sum_{i=1}^{k} d_i$	$C(f^*)$	$C(f)$	ρ	t^*	t
Anaheim	416	914	1406	104,694	1,304,562	1,322,566	1.38%	23s	11s
B.-Friedrichshain	224	523	506	11,205	475,801	520,586	9.41%	53s	24s
B.-Mitte	1782	2935	20	49,974	33,578,426	35,835,784	6.72%	273s	97s
B.-Prenzlauer Berg	352	749	1406	16,660	999,565	1,048,046	4.85%	350s	143s
B.-Tiergarten	359	766	644	10,755	565,746	581,450	2.78%	73s	20s
Sioux Falls	24	76	552	9,537	4,050,027	4,150,454	2.48%	<1s	<1s
Swiss	2210	6334	20	49,975	202,284,911	212,957,076	5.28%	380s	111s
Winnipeg	1040	2836	4344	64,775	808,915	815,441	0.81%	285s	109s

Table 1: Number of nodes n, number of edges m, number of commodities k, sum of the commodities' demand, total travel time of system optimal flow f^*, total travel time of Wardrop equilibrium flow f, relative price of anarchy $\rho = (C(f) - C(f^*))/C(f^*)$, CPU computing time t^* for system optimum, CPU computing time t for Wardrop equilibrium. System optima and Wardrop equilibria were computed with 0.01% CMCF precision.

5.2 Results

For all network instances, we computed tolls with the three algorithms MCT, EMCD and CT. The tables below show the resulting relative prices of anarchy for these algorithms. All algorithms perform quite well and significantly reduce the total travel time of the induced equilibrium flow for instances with at least 10 tollable edges. On almost all instances, MCT and EMCD outperform CT in terms of the final total travel time. For all instances, all algorithms reduce the relative price of anarchy by at least 35% using only 25 tollable edges. Except for Berlin-Mitte and the Swiss instance, this reduction is even more than 70%.

Perhaps the most interesting instances are Berlin-Mitte and Berlin-Friedrichshain, as they exhibit the two largest relative prices of anarchy of 9.41% and 6.72%, respectively. For the Berlin-Friedrichshain instance, the relative price of anarchy is reduced by 50% already using only 5 tollable edges. Berlin-Mitte on the contrary needs at least 10 tollable edges to show a significant reduction of total travel time of approximately 16%. The results for our algorithms on all 8 instances can be found in Table 2.

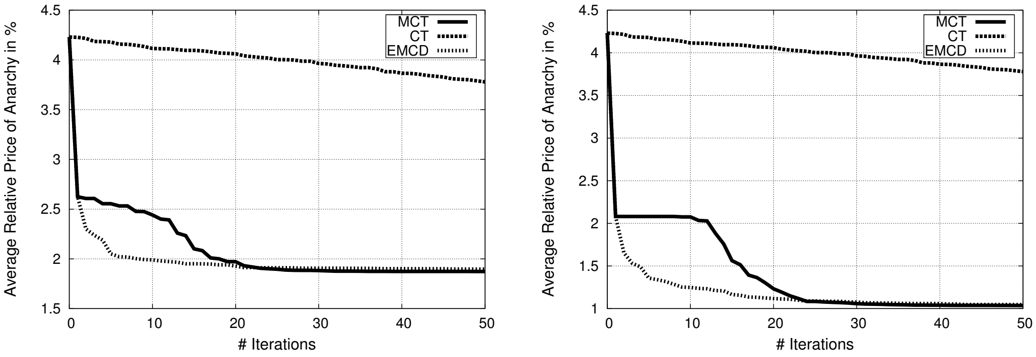
5.3 Convergence Time

We first demonstrate the *average* convergence behavior of the algorithms over all instances. Figure 5 shows the average performance over all instances for the cases $|T| = 10$ and $|T| = 25$, respectively. Here, one can see that the algorithms exhibit significant differences in terms of the speed of convergence. On average, EMCD exhibits the fastest convergence to close-to-optimal solutions, followed by MCT; however, CT needs hundreds of iterations to converge to good solutions. We also demonstrate the convergence behavior on a single instance (Berlin-Tiergarten) with ten tollable edges. Figure 6 shows for each iteration of the algorithms the total travel time of the computed Wardrop equilibrium with imposed tolls. Already after the first toll computation, the total travel time of the induced equilibrium improves significantly. In the first iteration, the toll values are set to the marginal cost prices $\ell_e(f_e^*)f_e^*$, $e \in T$, which seems to be a good choice of initial toll values. The CT-Algorithm progresses almost linearly and thus converges quite slowly despite the relatively large step-size of 0.1. In contrast, MCT and EMCD converge quickly (already after 20 iterations) to a near optimal solution. EMCD exhibits very

Instance	|T|	MCT	# it	EMCD	# it	CT	# it
Anaheim	0	1.38%					
	1	1.24%	65	1.24%	18	1.24%	125
	2	1.09%	66	1.09%	35	1.10%	200
	5	0.80%	70	0.79%	32	0.80%	434
	10	0.57%	69	0.57%	86	0.61%	708
	25	0.19%	69	0.19%	57	0.21%	1,130
	50	0.05%	69	0.04%	131	0.11%	1,346
B.-Friedrichshain	0	9.41%					
	1	6.93%	107	6.93%	39	6.92%	240
	2	5.22%	107	5.21%	74	5.21%	342
	5	2.77%	107	2.92%	51	3.25%	638
	10	2.60%	109	2.65%	92	2.68%	701
	25	0.16%	107	0.17%	317	0.39%	1,086
	50	0.07%	150	0.07%	225	0.23%	1,112
B.-Mitte	0	6.74%					
	1	6.7%	138	6.7%	55	6.7%	40
	2	6.6%	138	6.6%	93	6.5%	142
	5	6.4%	138	6.4%	210	6.4%	331
	10	5.6%	138	5.7%	104	5.5%	612
	25	4.0%	138	4.0%	139	4.0%	1,314
	50	2.5%	138	2.6%	125	2.6%	2,041
B.-Prenzlauer Berg	0	4.90%					
	1	3.5%	101	3.5%	47	3.4%	86
	2	2.2%	107	2.2%	52	2.2%	134
	5	2.0%	105	2.0%	73	2.1%	176
	10	1.1%	106	1.2%	106	1.3%	272
	25	0.3%	105	0.3%	122	0.6%	355
	50	0.1%	106	0.1%	222	0.5%	401
B.-Tiergarten	0	2.8%					
	1	2.3%	90	2.3%	49	2.3%	91
	2	1.6%	91	1.6%	64	1.6%	224
	5	0.7%	90	0.7%	113	0.7%	459
	10	0.1%	93	0.1%	79	0.1%	617
	25	0.02%	95	0.02%	136	0.1%	864
	50	<0.01%	92	<0.01%	136	0.1%	957
Sioux Falls	0	2.48%					
	1	2.48%	112	2.48%	74	2.24%	950
	2	2.48%	12	2.48%	760	2.04%	1,899
	5	1.24%	117	1.25%	88	1.23%	5,586
	10	0.62%	117	0.63%	77	0.58%	8,277
	25	0.04%	117	0.02%	274	0.07%	11,323
	50	<0.01%	369	<0.01%	162	0.03%	11,912
Swiss	0	5.40%					
	1	5.1%	128	5.1%	93	5.1%	501
	2	5.0%	136	5.0%	85	5.0%	1,052
	5	4.5%	143	4.5%	112	4.5%	1,897
	10	3.8%	148	3.8%	248	3.8%	3,354
	25	3.3%	136	3.3%	194	3.3%	4,961
	50	2.3%	135	2.3%	136	2.4%	8,000
Winnipeg	0	0.81%					
	1	0.7%	75	0.7%	34	0.7%	42
	2	0.7%	75	0.7%	25	0.7%	72
	5	0.6%	76	0.6%	94	0.6%	132
	10	0.5%	75	0.5%	63	0.5%	194
	25	0.2%	78	0.2%	74	0.3%	265
	50	0.1%	75	0.1%	81	0.2%	317

Table 2: Performance of our algorithms for different numbers $|T|$ of tollable edges in terms of the achieved relative price of anarchy.

fast convergence yielding a close-to-optimal toll solution after only 10 iterations.



(a) Average Performance for $|T| = 10$

(b) Average Performance for $|T| = 25$

Figure 5: Average performance of the MCT-Algorithm, the EMCD-Algorithm, and the CT-Algorithm with step-size = 0.1 on all instances with 10 and 25 tollable edges.

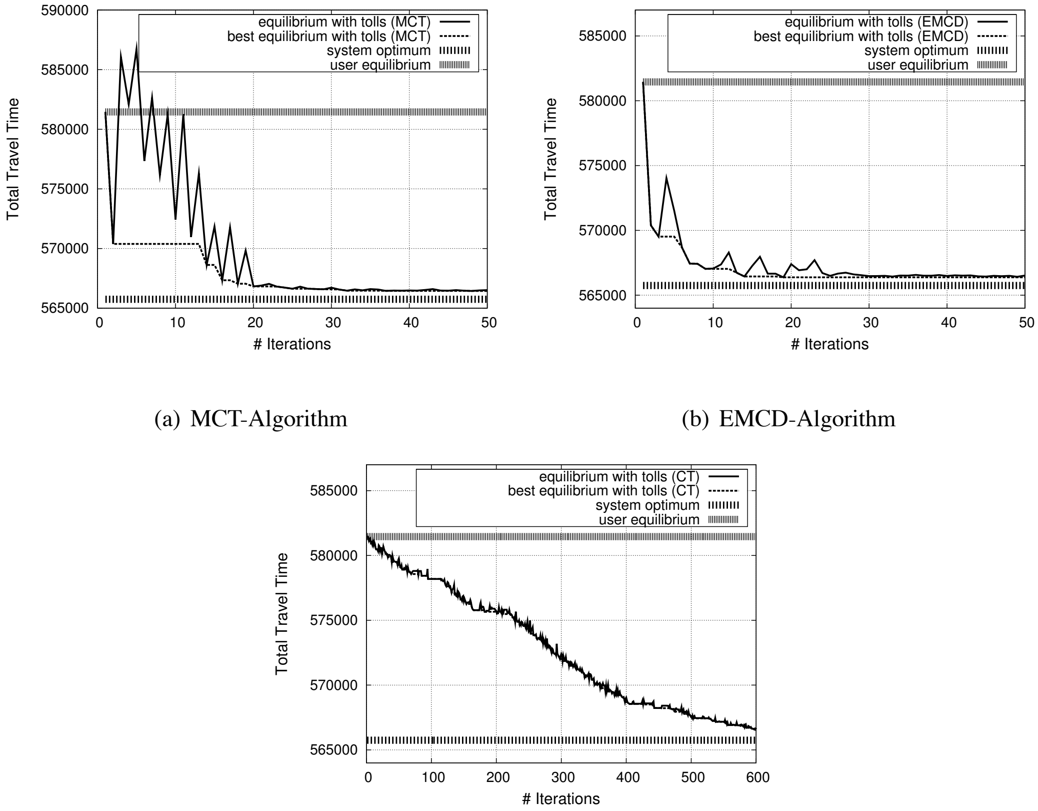
6 The Impact of the Choice of Taxable Edges

A good choice of the set T of tollable edges is critical to achieve a small total travel time. In practice, traffic planers usually select congested roads (for instance the city center) for installing toll schemes to improve the overall traffic throughput. In this section, we discuss the mathematical problem of selecting the *best* subset of tollable edges subject to a cardinality constraint. Formally, we define the following problem that we term the *cardinality constrained toll problem*. Here, we are given a tuple $I = (G, d, \ell, b)$, where G, d, and ℓ are defined as before and the parameter $b \in \mathbb{N}$ specifies a cardinality constraint on the support of the toll vector. That is, we require that the toll vector τ satisfies $|\{e \in E : \tau_e > 0\}| \leq b$.

6.1 Hardness

We next show that the problem of finding an optimal cardinality constrained set of tollable edges is strongly NP-hard and not even approximable by any constant $c \geq 1$. We reduce from the directed multicut problem. An instance of the *directed multicut problem* is given by a directed graph $G = (V, E)$ and k commodities $(s_i, t_i)_{i=1,\dots,k}$. A multicut $S \subseteq E$ is a subset of edges such that after removing S from G there is no (s_i, t_i)-path for all $i \in \{1, \dots, k\}$, or, equivalently, $S \cap P \neq \emptyset$ for each path $P \in \mathcal{P}_i$ and each commodity $i \in \{1, \dots, k\}$. The *minimum directed multicut problem* is to find a multicut of minimum cardinality. The corresponding decision problem is to decide for a given graph G and an integer $b \in [0, |E|]$ whether G has a multicut of cardinality at most b. The directed multicut problem is strongly NP-hard even for $k = 2$, see [19].

Theorem 6.1. *The cardinality constrained toll problem is strongly NP-hard. Moreover, unless $P = NP$, there is no c-approximation algorithm for the cardinality constrained toll problem for any $c \geq 1$. This even holds for instances with at most* 3 *commodities and* $\tau_e \in \{0, 1\}$ *for all* $e \in E$.



(a) MCT-Algorithm

(b) EMCD-Algorithm

(c) CT-Algorithm

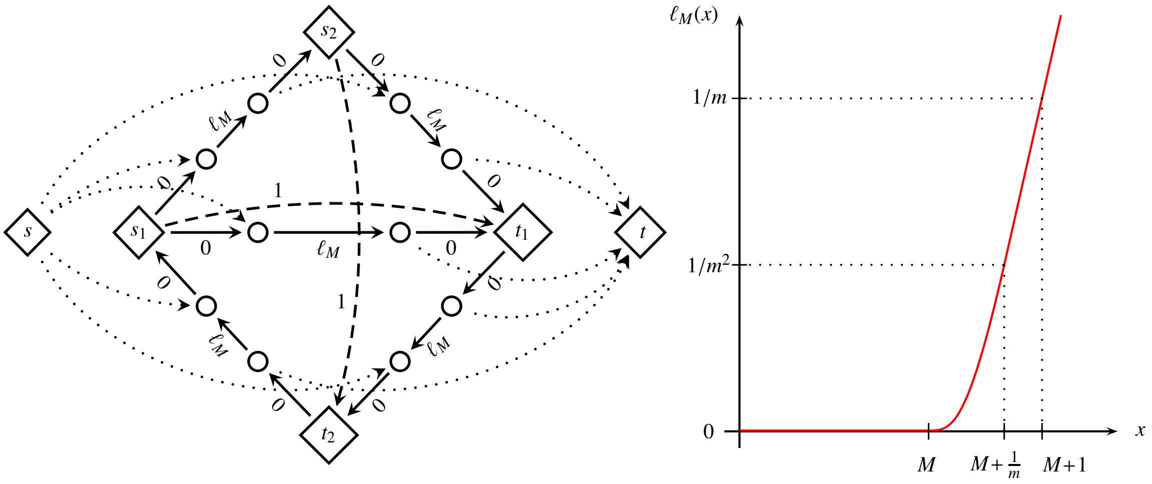
Figure 6: Performance of the MCT-Algorithm, the EMCD-Algorithm, and the CT-Algorithm (step-size = 0.1) on the Berlin-Tiergarten instance with 10 tollable edges.

Proof. Consider an instance $I = (G, (s_i, t_i)_{i=1,\dots,k})$ of the directed multicut problem with $G = (V, E), k = 2, |E| = m$ and $b \in [0, m]$. We will construct an instance $\hat{I} = (\hat{G}, \hat{\ell}, \hat{d}, b)$ of our problem as follows. We first augment G by adding for each commodity $i \in \{1, 2\}$ an auxiliary edge $\hat{e}_i$ from s_i to t_i. For the auxiliary edges we set $\ell_{\hat{e}_i}(x) = 1$. We replace each original edge $e = (u, v) \in E$ by three dummy edges in series, that is, we introduce $e_l = (u, v_e), e_m = (v_e, w_e), e_r = (w_e, v)$ for every $e \in E$ (see also Figure 7 (a) for the graph construction). The right and left dummy edges e_l and e_r have zero latency. For each dummy middle edge, we define the latency function as

$$\ell_M(x) = \begin{cases} 0, \text{ if } x \leq M, \\ \frac{x-M}{m}, \text{ if } x \geq M + 1/m, \end{cases}$$

where we choose $M \geq 2 \cdot m^2$. In the interval $(M, M + 1/m)$ the function ℓ_M is defined arbitrarily so that overall it is a standard and convex function (see also Figure 7 (b)). The commodities $i \in \{1, 2\}$ have a demand of 1 each.

We introduce a super-source s and a super-sink t. We connect s to every start node $v_e, e \in E$



(a) Augmentation of G

(b) Latency function ℓ_M

Figure 7: (a) An example of the augmentation used in the proof of Theorem 6.1. The figure shows the resulting network for the graph $G = (V, E)$ with 4 nodes $V = \{s_1, s_2, t_1, t_2\}$ and five edges $E = \{(s_1, t_1), (s_1, s_2), (s_2, t_1), (t_1, t_2), (t_2, s_1)\}$. All dotted edges have zero latency. (b) The latency function $\ell_M(x)$ used in the proof of Theorem 6.1.

of every middle dummy edge and we connect the end-nodes w_e of every middle dummy edge directly to the super-sink. All these additional edges have zero latency. For an illustration of this construction see Figure 7 (a). We assume a demand from s to t of value $m \cdot M$. Let τ^b be an optimal solution to the instance $\hat{I}$. We claim the following equivalence proving the theorem.

$$C(f^{\tau^b}) \leq 2 \Leftrightarrow \text{there exists a feasible multicut of cardinality } b.$$

We first assume that $C(f^{\tau^b}) \leq 2$. Suppose there is no feasible multicut of cardinality b. This implies that there is a commodity, say $i = 1$, with at least one toll-free path P using only dummy edges. Suppose $\ell_P(f^{\tau^b}) \geq 1$. By the definition of the latencies $\ell_M(x)$, there is at least one edge $e_m \in P$ with $f^{\tau^b}_{e_m} \geq M + 1/m$. We obtain

$$C(f^{\tau^b}) = \sum_{e \in \hat{E}} \ell_e(f_e(\tau^b)) f_e(\tau^b) \geq \frac{M + 1/m - M}{m}(M + 1/m) > 2,$$

a contradiction. Thus, we may assume that $\ell_P(f^{\tau^b}) < 1$. Using the Wardrop conditions, the edge $\hat{e}_1$ from s_1 to t_1 is not used in the induced equilibrium. Thus, there is a total demand of $m \cdot M + 1$ to be distributed over the remaining available paths, each containing at least one of the m middle dummy edge. Thus, there must exist a middle dummy edge e_m with flow $f_{e_m}(\tau^b) \geq (m \cdot M + 1)/m$ and we are in the previous case.

Conversely, assume that there is a feasible multicut S of cardinality b. We define tolls $\tau^b_{e_r} = 1, e \in S$ on the corresponding dummy right edges and zero, otherwise. The flow f^{τ^b} defined by routing 1 flow unit for every $i \in \{1,2\}$ along the direct s_i-t_i-edges $\hat{e}_i$ each with latency 1 and routing $m\,M$ flow units along their direct middle dummy edge (each with flow value M) satisfies the Wardrop conditions. We obtain $C(f^{\tau^b}) \le 2$ proving the claim.

To prove inapproximability, for any constant $c \ge 1$, we define $M \ge 2m^2c$. Then it follows for a c-approximate toll solution τ^b that

$$C(f^{\tau^b}) \le c \cdot 2 \Leftrightarrow \text{there exists a feasible multicut of cardinality } b,$$

which proves the claim. □

6.2 Computational Study for the Cardinality Constrained Toll Problem

As the problem of selecting the best set of tollable edges subject to a cardinality constraint is NP-hard and not approximable, we discuss several heuristics for selecting the edges on which tolls can be imposed. Our heuristics are based on sorting the edges with respect to marginal costs (Heuristic MCT in Algorithm 7), the difference between marginal costs of equilibrium and optimal flow (which gives Heuristic DMCT), and the difference between equilibrium and optimal flow (Heuristic DFT). We combine all edge selection heuristics with the EMCD-algorithm to compute the final tolls on the chosen subsets of edges.

Algorithm 7: Tollable Edge Subset Selection Heuristics MCT, DMCT, and DFT

```
Input: network instance $(G, d, \ell)$ with $G = (V, E)$, maximal number $b$ of tollable edges
Output: set $T$ of tollable edges with $|T| \le b$
$f^* \leftarrow$ system optimum flow, $f \leftarrow$ Wardrop equilibrium flow, $T \leftarrow \emptyset$;
while $|T| \le b$ do
    MCT:  $e \leftarrow \arg\max_{e \in E \setminus T, f_e > f^*_e} (\ell'_e(f_e) f_e)$;
    DMCT: $e \leftarrow \arg\max_{e \in E \setminus T, f_e > f^*_e} (\ell'_e(f_e) f_e - \ell'_e(f^*_e) f^*_e)$;
    DFT:  $e \leftarrow \arg\max_{e \in E \setminus T} (f_e - f^*_e)$;
    if $\{e \in E \setminus T : f_e > f^*_e\} = \emptyset$ then
        MCT:  $e \leftarrow \arg\max_{e \in E \setminus T} (\ell'_e(f_e) f_e)$;
        DMCT: $e \leftarrow \arg\max_{e \in E \setminus T} (\ell'_e(f_e) f_e - \ell'_e(f^*_e) f^*_e)$;
    end
    $T \leftarrow T \cup \{e\}$;
end
```

In addition to the above three heuristics, we also consider selection heuristics with random choices of the tollable edges. For every instance and every cardinality $b \in \{1, 2, 5, 10, 25, 50\}$, edges are selected iteratively and uniformly at random. We then compute tolls for the randomly chosen tollable edges with algorithm EMCD. Using 50 repetitions, the *aRND* solution is computed as the average of the relative difference between the total travel time of the toll-induced user equilibrium (using EMCD) and the system optimum. Furthermore, we picked the best random solution (*bRND*) out of these 50 runs.

Instance	$\lvert T\rvert$	aRND	bRND	MCT	DMCT	DFT
Anaheim	0	1.38%				
	1	1.37%	1.24%	1.24%	1.24%	1.17%
	2	1.36%	1.24%	1.09%	1.09%	1.04%
	5	1.34%	1.10%	0.79%	0.76%	0.88%
	10	1.31%	1.00%	0.57%	0.55%	0.76%
	25	1.24%	0.97%	0.19%	0.19%	0.53%
	50	1.12%	0.81%	0.04%	0.12%	0.23%
B.-Friedrichshain	0	9.40%				
	1	9.41%	9.39%	6.93%	6.93%	9.41%
	2	9.36%	7.77%	5.21%	5.45%	9.41%
	5	9.12%	6.63%	2.92%	2.92%	8.41%
	10	9.05%	6.62%	2.65%	2.65%	8.17%
	25	8.54%	5.10%	0.17%	0.53%	6.78%
	50	7.80%	3.92%	0.07%	0.07%	1.36%
B.-Mitte	0	6.70%				
	1	6.7%	6.2%	6.7%	6.7%	6.2%
	2	6.7%	6.2%	6.6%	6.7%	6.1%
	5	6.7%	6.2%	6.4%	5.9%	5.6%
	10	6.6%	6.2%	5.7%	5.4%	5.5%
	25	6.4%	6.0%	4.1%	3.7%	5.1%
	50	6.1%	5.2%	2.6%	2.2%	4.3%
B.-Prenzlauer Berg	0	4.90%				
	1	4.9%	4.8%	3.5%	3.5%	4.6%
	2	4.8%	3.5%	2.2%	2.2%	4.6%
	5	4.8%	4.5%	2.0%	1.7%	4.5%
	10	4.7%	3.4%	1.2%	1.4%	3.2%
	25	4.5%	3.0%	0.3%	0.3%	1.2%
	50	4.3%	2.6%	0.1%	0.1%	1.2%
B.-Tiergarten	0	2.80%				
	1	2.7%	2.0%	2.3%	2.3%	2.3%
	2	2.7%	2.0%	1.6%	1.8%	2.3%
	5	2.7%	2.0%	0.7%	0.7%	1.9%
	10	2.6%	1.7%	0.1%	0.1%	1.1%
	25	2.4%	1.4%	<0.1%	<0.1%	0.4%
	50	2.3%	1.1%	<0.1%	<0.1%	0.1%
Sioux Falls	0	2.48%				
	1	2.45%	2.31%	2.48%	2.48%	2.48%
	2	2.38%	1.96%	2.48%	2.48%	2.48%
	5	2.30%	1.91%	1.25%	1.66%	1.34%
	10	2.08%	1.19%	0.63%	0.59%	1.14%
	25	1.43%	0.88%	0.02%	0.02%	0.27%
	50	0.39%	0.01%	<0.01%	0.02%	<0.01%
Swiss	0	5.40%				
	1	5.4%	5.3%	5.1%	5.1%	5.2%
	2	5.4%	5.3%	5.0%	4.9%	5.2%
	5	5.3%	5.1%	4.5%	4.3%	5.1%
	10	5.3%	5.2%	3.8%	3.6%	5.0%
	25	5.3%	5.1%	3.3%	3.0%	4.4%
	50	5.2%	4.9%	2.3%	2.1%	3.8%
Winnipeg	0	0.80%				
	1	0.8%	0.8%	0.7%	0.7%	0.7%
	2	0.8%	0.8%	0.7%	0.7%	0.7%
	5	0.8%	0.7%	0.6%	0.6%	0.7%
	10	0.8%	0.7%	0.5%	0.5%	0.7%
	25	0.8%	0.7%	0.2%	0.3%	0.6%
	50	0.8%	0.7%	0.1%	0.1%	0.5%

Table 3: EMCD-Performance in terms of the relative price of anarchy for different selection rules and different numbers $|T|$ of tollable edges, CMCF precision= 0.1%. The best solutions are underlined.

Table 3 contains the results of all heuristics (in combination with EMCD) for the 8 test instances; the best solutions are underlined. Taking for instance a closer look at the results for Berlin-Mitte, we observe that the strategy which chooses 25 edges according to their marginal costs already leads to a reduction of the relative price of anarchy by 50% to 4.1%. Selecting edges in descending order of the flow difference $f - f^*$ between selfish flow and optimal flow (Heuristic DFT) seems to result in a better equilibrium state when only a few edges are tollable. For a larger number of tollable edges (more than 10) Heuristic DFT is significantly weaker than the other two heuristics. For the Berlin-Friedrichshain instance the edge selection rule depending on the flow difference $f - f^*$ does not perform well at all, whereas the two selection strategies which consider marginal costs lead to similar solutions. The strategies using information about marginal costs and the difference between optimal and selfish marginal costs, respectively, lead to small total travel time in almost every case and thus seem to be very good choices. Only in a few cases, in which the cardinality constraint is one or two, the best randomly selected edge set outperforms our heuristics.

7 Conclusions

We have provided a detailed study of network toll problems with constraints on the support set of feasible toll vectors.

For the simplest, but still practically relevant case of single-commodity networks with m parallel edges we obtained a far-reaching generalization of the analysis for affine latencies [22] to standard latency functions with a common Lipschitz constant κ and an upper bound K. In the most general case, we obtained a quasi-polynomial algorithm that computes for a demand of d an ε-optimal solution with an additive error of ε in $\mathsf{poly}(m, K, \kappa, d, 1/\varepsilon)$-time. We improved the runtime to $\mathsf{poly}(m, \log K, \log \kappa, \log d, \log 1/\varepsilon)$, i.e., to a polynomial algorithm, for the case that the total travel time is a piecewise convex function of the demand d. We also identified sufficient conditions when this holds and showed that these conditions cover commonly used latency functions such as the $M/M/1$ functions and functions similar to the Bureau of Public Roads (BPR) functions.

Our analysis showed that there is little hope to extend these results to arbitrary multi-commodity traffic networks. Moreover, even simple cases with two commodities are known to be NP-hard [22]. In order to still handle practically important cases, we devised three algorithms that are motivated by steepest descent approaches. We investigated their performance w.r.t. solution quality and convergence behavior in an extensive computational study on large street networks with up to 6,334 edges. All algorithms perform quite well and significantly reduce the total travel time for only a few tollable edges. The reduction in terms of a decrease of the relative price of anarchy is at least 35% with only 25 tollable edges, and even more than 70% on all but two networks. The algorithms differ, however w.r.t. their convergence behavior. Algorithm EMCD, in which the descent per edge e is based on the exponential marginal cost difference between system-optimal flow and tax induced flow on e, turned out to be the best of the three.

In the last section, we considered the cardinality constrained version of the toll problem. We no longer fix the set of tollable edges, but permit arbitrary subsets up to a fixed size, thus increasing the optimization potential for finding good tolls that significantly reduce the total travel time. We showed via a reduction from the directed multicut problem that the cardinality constrained toll problem is NP-hard and cannot be approximated within a constant factor unless P = NP. Again we devised practical algorithms that combine the three algorithms above with suitable heuristic choices for the tollable edges. Algorithm EMCD combined with an edge selection rule based on their marginal cost showed the best overall performance.

Our results show that the problem of computing tolls with support constraints is mathematically challenging, computationally hard and even not approximable. We presented the first rigorous analysis for single-commodity networks with parallel links and general latency functions, and investigated heuristic algorithms for multi-commodity networks. Our computational results show that, for realistic street networks, these algorithms have the potential to significantly reduce the total travel time with only a small number of tollable edges.

References

[1] L. Bai, D. Hearn, and S. Lawphongpanich. Decomposition techniques for the minimum toll revenue problem. *Networks*, 44(2):142–150, 2004.

[2] L. Bai, D. Hearn, and S. Lawphongpanich. A heuristic method for the minimum toll booth problem. *J. Global Optimization*, 48(4):533–548, 2010.

[3] L. Bai and P. Rubin. Combinatorial benders cuts for the minimum tollbooth problem. *Oper. Res.*, 57(6):1510–1522, 2009.

[4] M. Beckmann, C. McGuire, and C. Winsten. *Studies in the Economics and Transportation*. Yale University Press, 1956.

[5] P. Bergendorff, D. Hearn, and M. Ramana. Congestion Toll Pricing of Traffic Networks. In P. Pardalos, D. Hearn, and W. Hager, editors, *Network Optimization*, volume 450 of *Lecture Notes in Economics and Mathematical Systems*, pages 51–71, 1997.

[6] D. Bertsekas and R. Gallagher. *Data Networks – 2nd Edition*. Prentice-Hall, 1992.

[7] D. Braess. Über ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12:258 – 268, 1968. (German).

[8] Bureau of Public Roads. Traffic assignment manual. *U.S. Department of Commerce, Urban Planning Division, Washington, DC*, 1964.

[9] R. Cole, Y. Dodis, and T. Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proc. of the 35th Annual ACM Symp. on Theory of Computing*, pages 521–530, 2003.

[10] J. Correa, A. Schulz, and N. Stier-Moses. Selfish routing in capacitated networks. *Math. Oper. Res.*, 29:961–976, 2004.

[11] S. Dafermos and F. Sparrow. The traffic assignment problem for a general network. *J. Res. Nat. Bur. Stand.*, 73(B):91–118, 1969.

[12] R. Dial. Minimal-revenue congestion pricing part I: A fast algorithm for the single-origin case. *Transportation Res.*, 33(B):189–202, 1999.

[13] R. Dial. Minimal-revenue congestion pricing part II: An efficient algorithm for the general case. *Transportation Res.*, 34(B):645–665, 2000.

[14] P. Dubey. Inefficiency of Nash equilibria. *Math. Oper. Res.*, 11:1–8, 1986.

[15] A. Fabrikant, C. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proc. 36th Annual ACM Sympos. Theory Comput.*, pages 604–612, 2004.

[16] A. Fiacco and J. Kyparisis. Convexity and concavity properties of the optimal value function in parametric nonlinear programming. *J. Optimiz. Theory App.*, 48(1):95 – 126, 1986.

[17] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *Proc. of the 45th Annual IEEE Symp. on Foundations of Computer Science*, pages 277–285, 2004.

[18] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Nav. Res. Logist.*, 3:95–110, 1956.

[19] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in directed and node weighted graphs. In *Proc. 21st Internat. Colloquium on Automata, Languages and Programming*, pages 487–498, 1994.

[20] Green Choice Beijing. Urban growth and its pressure on traffic, 2010. `http://www.fon.org.cn/greenchoice`.

[21] D. Hearn and M. Ramana. Solving congestion toll pricing models. In P. Marcotte and S. Nguyen, editors, *Equilibrium and Advanced Transportation Modeling*, pages 109–124. Kluwer Academic Publishers, 1998.

[22] M. Hoefer, L. Olbrich, and A. Skopalik. Taxing subnetworks. In C. Papadimitriou and S. Zhang, editors, *Proc. 4th Internat. Workshop on Internet and Network Econom.*, number 5385 in LNCS, pages 286 – 294, 2008.

[23] O. Jahn, R. Möhring, A. Schulz, and N. Stier-Moses. System-optimal routing of traffic flows with user constraints in networks with congestion. *Oper. Res.*, 53(4):600–616, 2005.

[24] G. Karakostas and S. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. In *Proc. 45th Annual IEEE Sympos. Foundations Comput. Sci.*, pages 268–276, 2004.

[25] F. Knight. Some fallacies in the interpretation of social cost. *Q. J. Econ.*, 38(4):582–606, 1924.

[26] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. 16th Internat. Sympos. Theoretical Aspects of Comput. Sci.*, pages 404–413, 1999.

[27] T. Larsson and M. Patriksson. Side constrained traffic equilibrium models: Analysis, computation and applications. *Transportation Res.*, 33(B):233 – 264, 1999.

[28] N. Nisan. Speed-price equilibrium. `http://agtb.wordpress.com/2011/01/20/speed-price-equilibrium/`.

[29] A. Pigou. *The Economics of Welfare*. Macmillan, 1920.

[30] T. Roughgarden. The price of anarchy is independent of the network topology. *J. Comput. Syst. Sci. Int.*, 67:341–364, 2002.

[31] T. Roughgarden. Stackelberg scheduling strategies. *SIAM J. Comput.*, 33(2):332–350, 2004.

[32] T. Roughgarden and É. Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, 2002.

[33] Y. Sheffi. *Urban Transportation Networks*. Prentice-Hall, 1985.

[34] Singapur Government. Electronic road pricing, 2010. `http://www.lta.gov.sg/motoring_matters/index_motoring_erp.htm`.

[35] M. Smith. The marginal cost taxation of a transportation network. *Transportation Res.*, 13(B):237–242, 1979.

[36] C. Swamy. The effectiveness of Stackelberg strategies and tolls for network congestion games. In *Proc. of the 18th ACM-SIAM Symp. on Discrete Algorithms*, pages 1133–1142, 2007.

[37] Swedish Road Administration. Congestion tax in Stockholm. Technical report, Swedish Road Administration, 2007.

[38] Transport for London. Central london congestion charging. Technical report, Transport for London, 2008. `http://www.tfl.gov.uk/assets/downloads/sixth-annual-impacts-monitoring-report-2008-07.pdf`.

[39] A. Vingan, L. Fridstrom, and K. Johansen. Congestion charging in Bergen and Trondheim - an alternative 20 years ahead? Technical report, Norwegian Transportation Institute, 2007.

[40] J. Wardrop. Some theoretical aspects of road traffic research. *Proc. of the Institution of Civil Engineers*, 1(Part II):325–378, 1952.

[41] H. Yang and H. H.-J. The multi-class, multi-criteria traffic network equilibrium and systems optimum problem. *Transportation Res.*, 38(B):1–15, 2004.

[42] H. Yang and X. Zhang. Existence of anonymous link tolls for system optimum on networks with mixed equilibrium behaviors. *Transportation Res.*, 42(B):99–112, 2008.