Computing Flow-Inducing Network Tolls

Tobias Harks* Guido Schäfer† Martin Sieg*

* Institut für Mathematik, Technische Universität Berlin, Germany
{harks,msieg}@math.tu-berlin.de

† Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
g.schaefer@cwi.nl

Abstract

We consider the problem of computing tolls in non-atomic network routing games such that a predetermined flow is realized as Nash flow. It is a well-known fact that marginal cost tolls give rise to a Nash flow that minimizes the total travel time. In this paper, we study the problem of computing such tolls such that an additional toll-dependent objective function is optimized. We consider a broad class of objective functions, including convex and min-max functions, and show that such tolls can be computed in polynomial time. We also consider the problem of computing tolls such that the number of tolled arcs is minimized. We prove that this problem is NP-hard and APX-hard, even for very restricted single-commodity networks, and give first approximation results. Finally, we empirically evaluate the performance of our approximation algorithm on a set of real-world test instances.

1 Introduction

It is a well known fact that selfish behavior results in outcomes that are inefficient in general. A prime example is the rush-hour phenomenon observed in urban road traffic. 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, as if everybody would have been routed according to a centrally coordinated routing scheme. With the increasing number of traffic participants, the regulation of traffic becomes an increasingly important issue. One of the most promising means to regulate traffic is to impose tolls on roads. The basic idea is to impose tolls that guarantee that the selfish outcome corresponds to a predetermined routing scheme, e.g., one that minimizes the total average travel time. In this paper, we consider the problem of computing such tolls that additionally optimize a toll-dependent objective function.

A potential example scenario that fits into our framework is that a central authority (e.g., the state) aims for installing a toll-system to regulate traffic, thereby minimizing the total tolls charged to the participants. Alternatively, the goal might be to minimize the total installation cost of the facilities (toll-booths) needed to collect the tolls. Yet another application is that of a navigational systems provider that wishes to dynamically reroute some of his customers (e.g., due to some emerging congestion) according to a specific, centrally computed routing scheme. Nowdays, modern navigation devices feature bidirectional data communication and the provider can thus effectuate this routing scheme by sending appropriate "tolls" to the devices of the respective customers. These tolls are then used to update the estimated travel time data stored on the devices. Hereby, the transmission cost is proportional to the total amount of data sent to the customers. A natural objective is thus to compute tolls such that the sum of the tolls is minimized.

Network Routing Games. A standard way to model the selfish behavior of the traffic participants is by means of a *non-atomic network routing game*. We are given a directed network $G = (V,A)$ and k commodities $(s_1,t_1),\dots,(s_k,t_k) \in V \times V$. Let n and m refer to the number of vertices and arcs in G, respectively. Additionally, we are given a *demand* $r_i > 0$ for every commodity $i \in [k]$ which specifies the amount of flow that has to be routed from the origin s_i to the destination t_i. Let $\mathscr{P}_i$ be the set of all (simple) directed s_i,t_i-paths in G and define $\mathscr{P} = \cup_{i\in[k]} \mathscr{P}_i$. It is convenient to express a *flow* as a function $f : \mathscr{P} \to \mathbb{R}_+$ that assigns to every path $P \in \mathscr{P}$ a non-negative flow-value f_P that is routed along P. A flow f is *feasible* (with respect to r) if for every commodity $i \in [k]$, a total of r_i units of flow are routed from s_i to t_i, i.e., for every $i \in [k]$, $\sum_{P\in\mathscr{P}_i} f_P = r_i$. For a given flow f, we define the flow on an arc $a \in A$ as $f_a = \sum_{P\ni a} f_P$. Every arc $a \in A$ has a *latency function* $\ell_a : \mathbb{R}_+ \to \mathbb{R}_+$ associated with it. For each $a \in A$ the latency function ℓ_a is assumed to be nonnegative, nondecreasing and differentiable. If not indicated otherwise, we also assume that ℓ_a is a *standard* latency function (cf. [19]), i.e., ℓ_a is defined on $[0,\infty)$ and $x\ell_a(x)$ is a convex function of x. 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 arcs in the path, i.e., $\ell_P(f) = \sum_{a\in A} \ell_a(f_a)$. The triple (G,r,ℓ) is called an *instance*.

The total *cost* of a flow f is defined as $C(f) = \sum_{P\in\mathscr{P}} f_P \ell_P(f)$ or, equivalently, $C(f) = \sum_{a\in A} f_a \ell_a(f_a)$. A feasible flow of minimum total cost is called *optimal* and denoted by f^*. A feasible flow f is a *Nash flow* iff

$$\forall i \in [k],\ \forall P \in \mathscr{P}_i,\ f_P > 0,\ \forall P' \in \mathscr{P}_i : \quad \ell_P(f) \le \ell_{P'}(f). \tag{1}$$

That is, for every commodity the latency of every path that carries some positive amount of flow is minimum; in particular, this implies that all s_i,t_i-paths to which f assigns a positive amount of flow have equal latency. Under the assumption that all latency functions are standard, the cost of a Nash flow is unique, i.e., if f_1 and f_2 are Nash flows for the same instance then $C(f_1) = C(f_2)$ (see e.g. [20]). The *price of anarchy* is defined as the worst-case ratio (over all instances) of the cost of a Nash flow and the cost of an optimal flow, i.e., $C(f^*)/C(f)$. It is well-known (see [20]) that the price of anarchy is unbounded for general standard latency functions.

Network Toll Problems. An efficient means to reduce the price of anarchy in network routing games is by deploying *network tolls*. Intuitively, every (non-atomic) player that traverses arc $a \in A$ incurs, besides the latency $\ell_a(f_a)$, an additional (non-negative) toll cost. We represent the tolls of a given network by a non-negative vector $\tau = (\tau_a)_{a\in A}$, where τ_a specifies the toll that is imposed on arc $a \in A$. We assume that players are *heterogeneous*. That is, we are given a positive parameter γ_i for every commodity $i \in [k]$ and the total cost of a path $P \in \mathscr{P}_i$ with respect to a feasible flow f is defined as $\ell_P(f) + \gamma_i \tau(P)$, where $\tau(P) := \sum_{a\in P} \tau_a$. The parameter γ_i therefore specifies how the players of commodity i value latency relative to cost. We say that players are *homogeneous* if $\gamma_i = 1$ for all $i \in [k]$.

A question that arises is whether we can efficiently compute non-negative network tolls $\tau = (\tau_a)_{a\in A}$ such that a predetermined feasible flow f can be realized as Nash flow, i.e.,

$$\forall i \in [k],\ \forall P \in \mathscr{P}_i,\ f_P > 0,\ \forall P' \in \mathscr{P}_i : \quad \ell_P(f) + \gamma_i \tau(P) \le \ell_{P'}(f) + \gamma_i \tau(P'). \tag{2}$$

We call such tolls *f-inducing*. The problem of computing tolls that induce an optimal flow f^* is of particular interest and we call such tolls *opt-inducing*.

It is well known (see, e.g., Smith [21]) that if latency functions are standard, opt-inducing tolls are guaranteed to exist for homogeneous players: f^* is an optimal flow iff f^* is a Nash flow with respect to the *marginal latency functions* $\ell^*_a(x) = \ell_a(x) + x \cdot \ell'_a(x)$. Thus, we can simply define the *marginal cost tolls* as

$$\tau_a = f^*_a \cdot \ell'_a(f^*_a) \quad \text{for every arc } a \in A. \tag{3}$$

Note that since we assume that all latency functions are non-decreasing, $\tau_a \geq 0$.

Although marginal cost tolls assure that opt-inducing network tolls always exist, there might be a wide variety of such tolls.

In this paper, we are interested in computing f-inducing network tolls, such that an additional (toll-dependent) objective function $z(\tau)$ is minimized (or maximized). There are several natural objective functions that one may want to consider. Here we mainly concentrate on the following fundamental objective functions:

1. *Min-Convex Toll Problem (MCT)*: The goal is to compute f-inducing network tolls $\tau = (\tau_a)_{a\in A}$ such that a convex objective function $z(\tau)$ is minimized.

2. *Min-Max Toll Problem (MMT)*: The goal is to compute f-inducing network tolls $\tau = (\tau_a)_{a\in A}$ such that the maximum toll on any arc is minimized, i.e., $z(\tau) = \max_{a\in A} \tau_a$.

3. *Min-Toll-Booth Problem (MTB)*: The goal is to compute f-inducing network tolls $\tau = (\tau_a)_{a\in A}$ such that the sum of the weights of arcs with positive tolls is minimized. Formally, for a given toll vector τ define $Z(\tau) := \{a \in A \ : \ \tau_a > 0\}$ as the *support* of τ. The task then is to compute f-inducing network tolls τ such that $z(\tau) := \sum_{a\in Z(\tau)} w_a$ is minimized, where $(w_a)_{a\in A}$ are some given weights.

We remark that the min-convex toll problem subsumes several natural objective functions. For example, in the *min-revenue toll problem* the goal is to compute opt-inducing network tolls such that the total collected revenue $\sum_{a\in A} f_a^* \tau_a$ is minimized (see also [9, 10]). Another example is the *min-total toll problem*, where the goal is to compute tolls such that the sum of all tolls $\sum_{a\in A} \tau_a$ is minimized.

Our Results. In this paper, we study different variants of the network toll problems defined above. In particular, our main contributions are as follows:

1. We prove that a very restricted special case of the min-toll-booth problem for single-commodity instances is polynomial time equivalent to the *minimum L-length bounded cut problem* (definitions will be given in Section 2). This result enables us to prove that the min-toll-booth problem is NP-hard and APX-hard, even for very restricted single-commodity instances. While constant approximation algorithms may still be obtainable in the single-commodity case, we rule out there existence for the multi-commodity min-toll-booth problem. Via a reduction from the directed multicut problem, we show that the min-toll-booth problem cannot be approximated within a factor of $2^{\Omega(\log^{1-\varepsilon} n)}$ for every $\varepsilon > 0$.

2. We derive a very simple approximation algorithm for the min-toll-booth problem. The algorithm achieves an (instance-dependent) approximation factor that depends on the largest toll in an optimal solution, which might thus be difficult to quantify. However, this is the first approximation algorithm for the general problem and our experiments show that it performs quite well on real-world instances.

3. We present experimental findings on real-world instances for some special cases of the min-convex toll problem and the min-toll-booth problem. The experiments show that our approximation algorithms perform much better in practice than their worst-case approximation guarantees suggest. For most of the test instances, our algorithms compute solutions whose cost is at most a factor 4 worse than that of an optimal solution.

Related Work. Pigou [18] already suggested in 1920 that in order to obtain a system optimal traffic pattern vehicles should be charged taxes 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 [16], Beckmann et al. [3] and Smith [21].

There is a large body of work in the transportation literature (see, among others, Bergendorff et al. [5], Hearn and Ramana [14], Larsson and Patriksson [17]) that characterized the set of feasible arc tolls supporting a system optimal flow as a user equilibrium by a non-empty polyhedron expressed in terms of a linear inequality and equality system. These works, however, did not derive a compact representation of this polyhedron for heterogeneous users.

Hearn and Ramana [14] also proposed secondary optimization problems, where the objective is to minimize (maximize) a toll dependent function over the respective toll polyhedron. In particular, they were the first to study the min-toll-booth problem. Dial [9, 10] proposed efficient algorithms for finding tolls that minimize the total revenue.

Much recent work addressed the setting of heterogeneous users, where different users may have different trade-offs for delay versus toll. In this setting, one can exploit linear-programming duality to obtain tolls that induce an optimal flow, even for multi-commodity flows, see Cole et al. [8], Fleischer et al. [11] and Swamy [22].

Fleischer et al. [11] give a description of f-inducing tolls via linear programming techniques and derive sufficient and necessary conditions for their existence. They encode the Nash flow conditions in (1) by means of the complementary slackness conditions of a linear program and its dual. Independently, Karakostas et al. [15] proved existence of opt-inducing tolls for heterogeneous users and elastic demands.

2 Path-Raising Problem

Given a network routing game instance $I = (G, r, \ell)$, parameters $(\gamma_i)_{i\in[k]}$ and a feasible flow f for I, we can characterize the set of all feasible f-inducing network tolls for heterogeneous players by using the Nash flow characterization in (2). That is, a toll vector τ is f-inducing iff for every commodity $i \in [k]$, every s_i,t_i-path P carrying a positive amount of flow $f_P > 0$ is a shortest path with respect to the cost function $\ell + \gamma_i \tau$. This observation gives rise to the following optimization problem: Suppose we are given a directed network $G = (V, A)$ with non-negative arc-costs $(c_a)_{a\in A}$, k commodities $(s_i,t_i)_{i\in[k]}$, parameters $(\gamma_i)_{i\in[k]}$, and a set $\bar{\mathscr{P}}_i$ of designated (simple) s_i,t_i-paths for every commodity $i \in [k]$. The *path-raising problem (PR)* is to compute non-negative *arc-offsets* $\alpha = (\alpha_a)_{a\in A}$ minimizing a given objective function $z(\alpha)$ such that for every commodity $i \in [k]$, all paths in $\bar{\mathscr{P}}_i$ are shortest paths (among the paths in $\mathscr{P}_i$) with respect to the cost function $c + \gamma_i \alpha$. More formally, we want to solve the following program:

$$\min\ z(\alpha) \quad \text{subject to} \quad \alpha \in \mathscr{L}, \tag{4}$$

where $\mathscr{L}$ is the set of feasible arc-offsets defined as

$$\mathscr{L} := \{\alpha \in \mathbb{R}^m_{\geq 0} \,:\, c(P) + \gamma_i \alpha(P) \leq c(P') + \gamma_i \alpha(P')\ \ \forall P \in \bar{\mathscr{P}}_i,\ \forall P' \in \mathscr{P}_i,\ \forall i \in [k]\}. \tag{5}$$

This program generalizes all toll problems introduced in Section 1: For a given instance $I = (G, r, \ell)$, parameters $(\gamma_i)_{i\in[k]}$ and a feasible flow f for I, define $c_a := \ell_a(f_a)$ and let $\bar{\mathscr{P}}_i := \{P \in \mathscr{P}_i \,:\, f_P > 0\}$ be the set of all paths of commodity $i \in [k]$ that carry positive flow. Clearly, the computed arc-offsets $(\alpha_a)_{a\in A}$ correspond to the network tolls $(\tau_a)_{a\in A}$ associated with the arcs.

Compact Formulation. We next develop a compact formulation of the set $\mathscr{L}$ of feasible arc-offsets for the path-raising problem.

For every $i \in [k]$, we define

$$\bar{A}_i := \{a \in A \,:\, a \in P \text{ for some } P \in \bar{\mathscr{P}}_i\}.$$

We prove that the set $\mathscr{L}$ of feasible arc-offsets defined in (5) can be described by the following system of linear constraints:

$$\begin{aligned} \mathscr{F} := \{\alpha \in \mathbb{R}^m_{\geq 0} \,:\, & \delta_{i,v} \leq \delta_{i,u} + c_{uv} + \gamma_i \alpha_{uv} \quad \forall uv \in A \setminus \bar{A}_i,\ \forall i \in [k] \\ & \delta_{i,v} = \delta_{i,u} + c_{uv} + \gamma_i \alpha_{uv} \quad \forall uv \in \bar{A}_i,\ \forall i \in [k] \\ & \delta_{i,u} \quad \text{free} \qquad\qquad\qquad \forall u \in V,\ \forall i \in [k]\}. \end{aligned} \tag{6}$$

Proposition 1. *The descriptions of feasible arc-offsets given in* (5) *and* (6) *are equivalent. In particular,* $\mathscr{F}$ *is a compact formulation of the path-raising problem.*

Proof. Suppose $\alpha = (\alpha_a)_{a \in A} \in \mathscr{L}$. Fix an arbitrary commodity $i \in [k]$. Let $\delta_{i,u}$ be the shortest-path distance in G from s_i to $u \in V$ with respect to the cost function $c + \gamma_i \alpha$. Note that these distances must exist since G does not contain any negative cost cycles and $\gamma_i \alpha_a \geq 0$ for all $a \in A$. One can easily verify that $\alpha \in \mathscr{F}$ with $(\delta_{i,u})_{i \in [k], u \in V}$ being the corresponding labels.

Conversely, suppose $\alpha \in \mathscr{F}$ and let $(\delta_{i,u})_{i \in [k], u \in V}$ be the corresponding labels. Note that these labels remain feasible with respect to α if we add the same constant to all labels of commodity i. We can thus adjust these labels such that $\delta_{i,s_i} = 0$ for every $i \in [k]$. Fix an arbitrary commodity $i \in [k]$. The constraints of commodity i now imply that $(\delta_{i,u})_{u \in V}$ define shortest-path distances from s_i to every vertex $u \in V$ and that all arcs of the paths in $\bar{\mathscr{P}}_i$ are tight. That is, all paths in $\bar{\mathscr{P}}_i$ are shortest paths with respect to $c + \gamma_i \alpha$ and thus $\alpha \in \mathscr{L}$.

The number of constraints and variables needed to define $\mathscr{F}$ is polynomially bounded in the size of the input instance and thus $\mathscr{F}$ is a compact description of the set of all feasible arc-offsets. □

Cole et al. [8] give a similar description of $\mathscr{F}$ for single-commodity instances. Fleischer et al. [11] give a compact description of $\mathscr{F}$ for multi-commodity instances by relying on linear programming duality. They also proved that the set $\mathscr{F}$ is nonempty for flows f minimizing a function $w : \mathbb{R}^{|A|} \to \mathbb{R}_+$, which is nondecreasing in each of its arguments. Thus, it follows that opt-enforcing flows exist.

Polynomial-Time Solvability. We conclude that the path-raising problem can be solved in polynomial time whenever the objective function z is *convex*, e.g., by using the ellipsoid method [13]. Moreover, efficient combinatorial algorithms exist if the objective function is linear: By dualizing the (then) linear program (4), one obtains a multi-commodity min-cost flow problem for which efficient combinatorial algorithms exist (see [1]). Also min-max (or max-min) objective functions can be solved efficiently by using a standard approach to formulate the respective problem as a linear program.

Since the toll problems introduced in Section 1 can be formulated as path raising problems, the above theorem immediately gives rise to the following corollary.

Corollary 1. *The min-convex toll problem and the min-max toll problem can be solved in polynomial time.*

3 Min-Support Path-Raising Problem

In the previous section, we have shown that the set of feasible arc-offsets for the path-raising problem can be described by a compact system of linear inequalities. We next consider a variant of this problem where we want to minimize the sum of the weights of the arcs with positive arc-offset. More formally, we are given a directed network $G = (V,A)$ with non-negative arc-costs $(c_a)_{a\in A}$ and non-negative arc-weights $(w_a)_{a\in A}$, k commodities $(s_i,t_i)_{i\in[k]}$, parameters $(\gamma_i)_{i\in[k]}$ and a set $\bar{\mathscr{P}}_i$ of designated (simple) s_i,t_i-paths for every commodity $i \in [k]$. The *min-support path-raising problem (MSPR)* is to compute non-negative *arc-offsets* $\alpha = (\alpha_a)_{a\in A}$ such that for every commodity $i \in [k]$, all paths in $\bar{\mathscr{P}}_i$ are shortest paths with respect to the cost function $c + \gamma_i\alpha$ and such that $\sum_{a\in Z(\alpha)} w_a$ is minimized, where $Z(\alpha) := \{a \in A \,:\, \alpha_a > 0\}$ is the support of α. Note that the min-toll-booth problem is a special case of the min-support path-raising problem.

Hardness and Inapproximability. We consider the single-commodity case first. Let s and t be the origin and destination of the commodity, respectively. We use $\mathscr{P}$ to denote the set of directed s,t-paths. Moreover, let $\bar{\mathscr{P}} \subseteq \mathscr{P}$ be the subset of directed s,t-paths in G that we would like to make shortest paths. Throughout this section, we assume that $\gamma = 1$.

As we will prove below, the min-support path-raising problem is hard even for the very restricted case when $\bar{\mathscr{P}}$ consists of a single arc $\bar{a} \in A$ from s to t, i.e., $\bar{\mathscr{P}} := \{\bar{a}\}$. We call this restricted variant the *single-arc min-support path-raising problem*. Intuitively, the goal is to raise the cost of all paths $P \in \mathscr{P} \setminus \{\bar{a}\}$ with $c(P) < c(\bar{a})$ to at least $c(\bar{a})$, thereby minimizing the sum of the weights of the arcs in the support. We will show that this problem is equivalent to the *minimum L-length bounded cut problem*.

An instance of the minimum L-length bounded cut problem is given by a directed graph $G = (V,A)$ with non-negative arc-lengths $(\ell_a)_{a\in A}$, non-negative arc-capacities $(u_a)_{a\in A}$, an origin s, a destination t, and a length bound $L \geq 0$. Let $\mathscr{P}$ refer to the set of all (simple) directed s,t-paths in G. Moreover, define $\mathscr{P}(L) \subseteq \mathscr{P}$ as the set of directed s,t-paths whose length is at most L, i.e., $\mathscr{P}(L) := \{P \in \mathscr{P} \,:\, \ell(P) \leq L\}$. We call a subset $C \subseteq A$ an *L-length bounded cut* if there is no s,t-path of total length at most L left in the graph induced by the arc set $A \setminus C$ or, equivalently, for every $P \in \mathscr{P}(L)$ we have $C \cap P \neq \emptyset$. The capacity of a cut C is defined as $u(C) := \sum_{a\in C} u_a$. The *minimum L-length bounded cut problem* is to find an L-length bounded cut of minimum capacity.

Theorem 1. *Every instance of the minimum L-length bounded cut problem can be reduced to an instance of the single-arc min-support path-raising problem, and vice versa. Moreover, these reductions preserve the objective function values.*

Proof. Suppose we are given an instance $\hat{I} = (\hat{G},\ell,u,(s,t),L)$ of the minimum L-length bounded cut problem. We derive an instance $I = (G,c,w,(s,t),\gamma = 1,\{\bar{a}\})$ of the min-support path-raising problem as follows: We first identify $G = \hat{G}$, $c = \ell$, $w = u$ and then augment G by adding a single arc $\bar{a}$ from s to t with $c_{\bar{a}} = L+\varepsilon$ and $w_{\bar{a}} = 1$. Here $\varepsilon > 0$ is chosen sufficiently small such that $\{P \in \mathscr{P} : c(P) < L+\varepsilon\} = \mathscr{P}(L)$. Consider a feasible L-length bounded cut C for $\hat{I}$. By defining $\alpha_a = M$ for a sufficiently large M if $a \in C$ and $\alpha_a = 0$ otherwise, we obtain feasible arc-offsets α for I. Moreover, $u(C) = \sum_{a\in Z(\alpha)} w_a$. Suppose $\alpha = (\alpha_a)_{a\in A}$ are feasible arc-offsets for I. The support $Z(\alpha)$ of α defines a feasible L-length bounded cut C satisfying $\sum_{a\in Z(\alpha)} w_a = u(C)$.

The reverse transformation is defined analogously: Let $I = (G,c,w,(s,t),\gamma = 1,\{\bar{a}\})$ be an instance of the single-arc min-support path-raising problem. We construct an instance $\hat{I} = (\hat{G},\ell,u,(s,t),L)$ of the minimum L-length bounded cut problem as follows: We first identify $\hat{G} = G$, $\ell = c$, $u = w$ and then remove arc $\bar{a}$ from $\hat{G}$. We also define $L = c(\bar{a}) - \varepsilon$ for a sufficiently small ε. Now it is easy to verify that a solution $\alpha = (\alpha_a)_{a\in A}$ is feasible for I iff $Z(\alpha)$ defines a feasible L-length bounded cut C for $\hat{I}$. Moreover, $\sum_{a\in Z(\alpha)} w_a = u(C)$. $\square$

The decision problem of the minimum L-length bounded cut problem is to determine whether for a given integer k there is a length bounded cut of capacity at most k. Baier et al. [2] proved that this problem is NP-hard for series parallel graphs. Moreover, the authors show that for any $L \in \{4,\dots,\lfloor n^{1-\varepsilon}\rfloor\}$ and arbitrary small $0 < \varepsilon < 1$, the minimum L-length bounded cut problem is NP-hard to approximate within a factor of 1.1377. Given the equivalence between the minimum L-length bounded cut problem and the single-arc min-support path-raising problem (Theorem 1), we conclude:

Corollary 2. *The single-arc min-support path-raising problem is NP-hard to approximate within a factor of* 1.1377 *for instances with unit arc-weights. Moreover, the decision problem of the single-arc min-support path-raising problem is NP-hard for series-parallel graphs with arbitrary arc-costs and arbitrary arc-weights.*

We next show that the multicut problem can be reduced to the multi-commodity min-support path-raising problem. Via this reduction, we are able to show that no algorithm can achieve an approximation factor better than $2^{\Omega(\log^{1-\varepsilon} n)}$ for arbitrary $\varepsilon > 0$.

An instance of the *directed multicut problem* is given by a directed graph $G = (V,A)$ and k commodities $(s_i,t_i)_{i\in[k]}$. A multicut $C \subseteq A$ is a subset of arcs, such that by removing C from G there is no s_i,t_i-path left for all $i \in [k]$, or equivalently $C \cap P \neq \emptyset$ holds for each path $P \in \mathscr{P}_i$ and each commodity $i \in [k]$. The *minimum directed multicut problem* is to find a multicut of minimum cardinality.

Theorem 2. *Every instance of the directed multicut problem can be reduced to an instance of the multi-commodity min-support path-raising problem such that the objective function value is preserved.*

Proof. Consider an arbitrary instance $\hat{I} = (\hat{G},(s_i,t_i)_{i\in[k]})$ of the directed multicut problem. We construct an instance $I = (G,c,w,(s_i,t_i)_{i\in[k]},\gamma = 1,(\bar{A}_i)_{i\in[k]})$ of the multi-commodity min-support path-raising problem as follows: We first augment $G = \hat{G}$ by adding for each commodity $i \in [k]$ an auxiliary arc $\bar{a}_i$ from s_i to t_i. For each original arc we set $c_a = 0$ and for the auxiliary arcs we set $c_{\bar{a}} = 1$. Further we set $w = 1$ and $\bar{A}_i = \{\bar{a}_i\}$.

Let C be a directed multicut for $\hat{I}$. Since C hits each s_i,t_i-path, we obtain feasible arc-offsets α for I by setting $\alpha_a = 1$ for every arc $a \in C$ and $\alpha_a = 0$ otherwise. Clearly $w(Z(\alpha)) = |C|$ holds.

Conversely, let α^* be optimal arc offsets for I. Since we added to G the auxiliary arcs $\bar{a}_i$, we possibly generated additional s_i,t_i-paths. By construction $c_{\bar{a}_i} = 1$ holds, therefore the cost of the additional paths is at least one because they must contain at least one auxiliary arc. Accordingly, α^* satisfies $\alpha^*_{\bar{a}_i} = 0$ for each $i \in [k]$. If a path $P \in \mathscr{P}_i$ contains none of the auxiliary arcs $\bar{a}_j$, then $c(P) = 0$ holds and P is also contained in $\hat{G}$. Since α is feasible for I, there must exist some $a \in P$ with $\alpha_a > 0$. Thus, the support $Z(\alpha^*)$ defines a multicut C for $\hat{G}$ with $|C| = w(Z(\alpha^*))$. $\square$

Chuzhoy and Khanna [7] showed that the directed multicut problem cannot be approximated within a factor of $2^{\Omega(\log^{1-\varepsilon} n)}$ for any constant $\varepsilon > 0$, even for directed acyclic graphs, unless $\mathsf{NP} \subseteq \mathsf{ZPP}$. Given the result in Theorem 2, we conclude that the hardness result carries over to the multi-commodity min-support path-raising problem.

Corollary 3. *The multi-commodity min-support path-raising problem cannot be approximated within a factor of* $2^{\Omega(\log^{1-\varepsilon} n)}$ *for any constant* $\varepsilon > 0$ *for instances with unit-arc weights, unless* $\mathsf{NP} \subseteq \mathsf{ZPP}$.

Approximation Algorithms. Using the results of the previous section, we can formulate the min-support path-raising problem as a mixed integer program:

$$\min \sum_{a\in A} w_a \cdot z_a \quad \text{subject to} \quad \begin{aligned} \alpha &\in \mathscr{F} & \\ z_a \cdot M &\geq \alpha_a & \forall a \in A \\ z_a &\in \{0,1\} & \forall a \in A. \end{aligned} \tag{7}$$

A subtle point in this formulation is the choice of the parameter M. Consider an optimal solution α to a given instance of the min-support path-raising problem and let $M(\alpha)$ be the maximum arc-offset in α. Clearly, choosing $M = M(\alpha)$ is sufficient since this enforces that $z_a = 1$ if $\alpha_a > 0$ and $z_a = 0$ otherwise. Ideally, we would like to choose M as small as possible, i.e., as the minimum of $M(\alpha)$ over all possible optimal solutions α. However, despite some efforts, we are not able to bound the value of $M(\alpha)$ in general. Our negative result (Theorem 3) implies that in general M cannot be smaller than $2^{\Omega(\log^{1-\varepsilon} n)}$.

We present a very simple LP rounding approach (LPR) for the min-support path-raising problem. The idea is to simply return an optimal solution $\alpha^* = (\alpha_a^*)_{a\in A}$ of the following linear program:

$$\min \frac{1}{M} \sum_{a\in A} \alpha_a w_a \quad \text{subject to} \quad \alpha \in \mathscr{F}. \tag{8}$$

Note that α^* can be computed efficiently.

Theorem 3. LPR *is an M/λ-approximation algorithm for the multi-commodity min-support path-raising problem, where $\lambda := \min_{a\in Z(\alpha^*)} \alpha_a^*$ is the minimum positive arc-offset of an arc in the computed solution α^*.*

Proof. Consider the LP relaxation of the mixed integer program in (7). Note that for fixed arc-offsets $\alpha = (\alpha_a)_{a\in A}$ it is clear how to choose $(z_a)_{a\in A}$ in order to minimize the objective function, namely $z_a := \alpha_a/M$ for every arc $a \in A$. Thus, the LP relaxation of (7) is equal to the linear program in (8).

Let Z^* be the support of an optimal solution to the mixed integer program (7). Clearly, $\frac{1}{M}\sum_{a\in A} \alpha_a^* w_a \leq w(Z^*)$. Exploiting that for every arc $a \in A$ with $\alpha_a^* > 0$, $\alpha_a^*/\lambda \geq 1$, we obtain

$$w(Z(\alpha^*)) = \sum_{a\in Z(\alpha^*)} w_a \leq \sum_{a\in A} \frac{\alpha_a^*}{\lambda} w_a = \frac{M}{\lambda} \cdot \frac{1}{M} \sum_{a\in A} \alpha_a^* w_a \leq \frac{M}{\lambda} \cdot w(Z^*).$$

□

It is not hard to see that the arc-offsets of an optimal solution for (8) are integral if all arc-costs are integral.

Corollary 4. *Suppose all arc-costs are integral. Then,* LPR *is an M-approximation algorithm for the multi-commodity min-support path-raising problem.*

Although we are not able to quantify M in general, the above approximation algorithm turns out to perform quite well on real-world instances, as described in the next section.

4 Experimental Results

In this section, we present our experimental findings on real-world instances. We considered network toll problems that fall into the class of problems defined in Section 1. More specifically, we want to compute opt-inducing tolls such that the following objectives are met:

- *Min-Revenue Toll Problem (MRT)*: A special case of the min-convex toll problem with objective function $z(\tau) = \sum_{a \in A} f_a^* \tau_a$, where f^* denotes an optimal flow. That is, the goal is to minimize the total tolls charged to the players.

- *Min-Total Toll Problem (MTT)*: A special case of the min-convex toll problem with objective function $z(\tau) = \sum_{a \in A} \tau_a$.

- *Min-Num-Toll-Both Problem (MNTB)*: A special case of the min-toll-booth problem, where all arcs have unit weight. That is, the goal is to compute feasible tolls such that the number of arcs with positive tolls is minimized.

Our focus in the empirical evaluation is twofold: on the one hand side we study the structural differences of toll vectors optimizing the above different objectives on real-world instances. One would expect that optimal solutions (with respect to MRT, MTT, MNTB) do not differ significantly if the set $\mathscr{F}$ of feasible tolls is small. It turns out, however, that on most of our test instances, the quality of optimal solutions with respect to different objectives exhibits significant varieties.

On the other hand, we are particularly interested in the quality of the LP rounding algorithm (LPR) for the min-num-toll-both problem on real-world instances. While the provable worst-case performance guarantee of LPR is quite weak (it depends on instance related parameters M and λ in Theorem 3, the actual (instance-dependent) approximation ratio achieved on real-world instances might be significantly better. Indeed, it turned out that for *all* instances for which we could assess the value of an optimal solution for MNTB, the worst approximation ratio that we observed was below 4. The average approximatio ratio of LPR was even below 1.5.

Network		nodes	arcs	commodities
Anaheim		416	914	1406
Berlin –	Friedrichshain	224	523	506
	Mitte	398	871	1260
	Prenzlauer Berg	352	749	1406
	Tiergarten	361	766	644
Sioux Falls		24	76	528

Table 1: Assortment of TNTP data sets.

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 only for academic research purposes and consist of several networks for different cities. Also a trip file is given, specifying the commodities and demands. We selected the instances listed in Table 4, see the appendix. The network file specifies parameters such as the length, free flow travel time and the capacity of every arc. These parameters are used to determine the link performance function. We used non-linear link performance functions, proposed by the *Bureau of Public Roads (BPR)* [6]: $\ell_a(x) := \text{free flow time} \cdot \left(1 + \text{bias} \cdot \left(\frac{x}{\text{capacity}}\right)^4\right)$.

We solved the underlying traffic assignment problem with up to 0.01% precision using a variant (CMCF) of the Frank-Wolfe algorithm [12].

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

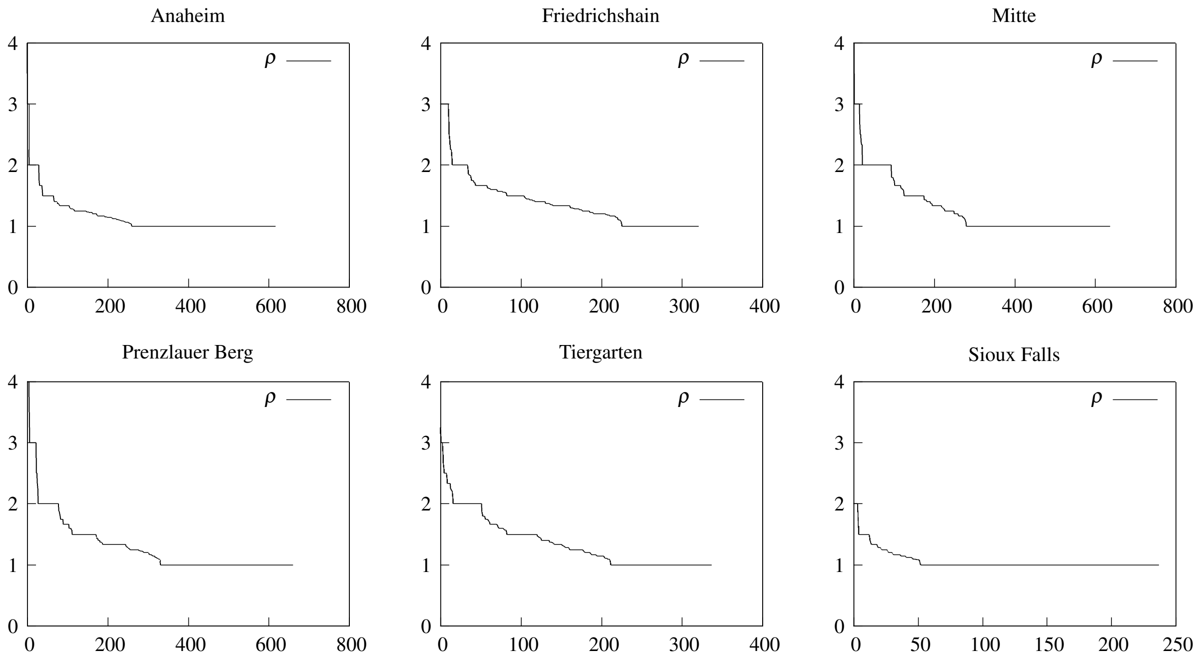

Figure 1: Approximation ratio of LPR for single-commodity instances and $|Z(\alpha)|$-objective.

Toll Solutions and Performance Measures. We compare four different toll solution concepts in our empirical studies, one of which are the marginal cost tolls (MC) as defined in (3), while the other three refer to optimal solutions of the respective optimization problems (MRT, MTT, MNTB):

The computations have been carried out with CPLEX as solver running on a DualCore 64bit-Opteron CPU with 2.6Ghz and 16GB RAM. We computed the respective tolls using CPLEX 11.0, except for the multi-commodity MNTB instances, for these we used the algebraic modelling language AMPL linked to CPLEX 10.0 as solver.

Results for Single-Commodity Instances. For the single-commodity case, we used the following procedure; we split the original trip file into several single commodity instances. For example the Sioux Falls instance has 528 commodities, which we separately used as single-commodity instances. We multiplied for each commodity the demand with a factor of 100, since the original demand would result in an all-or-nothing assignment on the shortest path. For each instance, we solved the traffic assignment problem and used CPLEX as the solver for those solutions where the optimal routing consisted of at least two flow carrying paths. The resulting solutions for MC, MRT, LPR, MNTB with respect to the three objective functions $\sum_{a\in A} f_a^* \alpha_a$, $\sum_{a\in A} \alpha_a$ and $|Z(\alpha)|$ are listed in Table 2, see the appendix. All entries are taken as the average over all instances. We also tracked the average approximation factors (denoted by ρ_{MRT}, ρ_{MTT} and ρ_{MNTB}) for all toll solutions with respect to the respective optimal solution.

Remarkably, the four different solution concepts exhibit significant differences with respect to the considered objectives. While MC and MRT exhibit large average approximation factors with respect to $\sum_{a\in A} \alpha_a$ and $|Z(\alpha)|$, the performance guarantee of LPR does not exceed 1.4 for *all* three objectives.

From an approximation-theoretic point of view, the most interesting question is, how well the considered solution concepts perform (in particular our rounding scheme LPR) with respect to the MNTB objective. Recall that the corresponding minimization problem is NP-hard and APX-hard, see Corollary 2.

In Figure 1, the distribution of the approximation ratio of our rounding scheme LPR is illustrated. For every network, the instances are ordered in decreasing order of the achieved approximation factor. The worst-case approximation factor for all instances stays below 4. Most notably, in more than 90% of all cases the approximation factor of LPR is even below 2.

Multi-Commodity Instances. Due to space limitations, we moved the results for multi-commodity instances to the appendix.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993.

[2] G. Baier, T. Erlebach, A. Hall, E. Köhler, H. Schilling, and M. Skutella. Length-bounded cuts and flows. In *In Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, pages 679–690, 2006.

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

[4] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[5] P. Bergendorff, D. Hearn, and M. Ramana. Congestion Toll Pricing of Traffic Networks. In D.W. Hearn P. M. Pardalos and W.W. Hager, editors, *Network Optimization*, volume 450 of *LNCS*, pages 51–71. Springer-Verlag, Berlin, 1997.

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

[7] J. Chuzhoy and S. Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 179–188. ACM New York, NY, USA, 2007.

[8] R. Cole, Y. Dodis, and T. Roughgarden. Pricing network edges for heterogeneous selfish users. *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 521–530, 2003.

[9] R. B. Dial. Minimal-revenue congestion pricing part i: A fast algorithm for the single-origin case. *Transportation Research Part B: Methodological*, 33(3):189 – 202, 1999.

[10] R. B. Dial. Minimal-revenue congestion pricing part ii: An efficient algorithm for the general case. *Transportation Research Part B: Methodological*, 34(8):645 – 665, 2000.

[11] L. Fleischer, K. Jain, and M. Mahdian. Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 277–285, 2004.

[12] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[13] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[14] 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.

[15] G. Karakostas and S. G. Kolliopoulos. Edge pricing of multicommodity networks for heterogeneous selfish users. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 268–276, 2004.

[16] F. H. Knight. Some fallacies in the interpretation of social cost. *The Quarterly Journal of Economics*, 38(4):582–606, 1924.

[17] T. Larsson and M. Patriksson. Side constrained traffic equilibrium models—analysis, computation and applications. *Transportation Research*, 33B:233–264, 1999.

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

[19] T. Roughgarden. *Selfish Routing*. PhD thesis, Cornell University, 2002.

[20] T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM (JACM)*, 49(2):236–259, 2002.

[21] M. J. Smith. The marginal cost taxation of a transportation network. *Transportation Research Part B: Methodological*, 13(3):237–242, September 1979.

[22] 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.

A Results for Single-Commodity Instances

Marginal Cost Tolls (MC)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ_{MTB}
Anaheim	3144048.5	1778.1072	75.7	1031.7926	50.0	13.5786
Friedrichshain	481307.5	4.3605	273.9	3.7018	26.6	6.6193
Mitte	962331.9	33.3132	414.6	31.0113	26.2	11.6327
Prenzlauer Berg	667645.9	10306.3146	247.9	5185.7609	26.4	10.1413
Tiergarten	335826.2	3730.1557	216.8	1913.5795	34.9	11.0141
Sioux Falls	3439785.3	4.6048	173.4	4.0697	22.1	4.1886

Minimum Revenue Tolls (MRT)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ_{MTB}
Anaheim	21768.2	1.0	123.6	566063.86053	37.8	10.9773
Friedrichshain	82596.2	1.0	1507.4	133.3584	39.9	11.7170
Mitte	22236.4	1.0	1511.3	12724.9912	40.1	19.6010
Prenzlauer Berg	48326.0	1.0	1426.0	3965186.1138	39.1	16.9198
Tiergarten	99172.6	1.0	1903.1	8671408.3594	41.6	13.9845
Sioux Falls	334282.6	1.0	60.4	4.5380	10.5	2.3211

Minimum Total Tolls (LPR)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ_{MTB}
Anaheim	21940.8	1.0101	3.7	1.0	6.3	1.1562
Friedrichshain	82596.2	1.0000	69.8	1.0	7.1	1.3867
Mitte	22236.4	1.0000	24.5	1.0	3.6	1.298
Prenzlauer Berg	48335.8	1.0006	41.3	1.0	4.4	1.3120
Tiergarten	99175.2	1.0001	73.3	1.0	6.3	1.3689
Sioux Falls	334282.6	1.0000	27.4	1.0	6.6	1.0680

Minimum Number of Toll-Booths Tolls (MNTB)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ_{MTB}
Anaheim	22466.7	1.0228	26.6	3.3722	5.6	1.0
Friedrichshain	83615.7	1.0075	134.7	1.7963	5.3	1.0
Mitte	22425.3	1.0029	46.4	1.1157	2.8	1.0
Prenzlauer Berg	48414.2	1.0015	86.1	1.8164	3.5	1.0
Tiergarten	99949.8	1.0084	124.8	1.5363	4.6	1.0
Sioux Falls	334852.3	1.0040	36.4	1.4142	6.4	1.0

Table 2: Results on single-commodity instances with respect to the three different objectives; all entries represent the average over the number of trials (commodities) per instance.

B Multi-Commodity Instances

While all linear programs for the single-commodity case have been solved up to optimality, see Table 4, the mixed integer program (7) for MNTB for multi-commodity instances turned out to be very hard to solve. Our first approach for MNTB has been to put a limit on running time of 24 hours for the multi-commodity instances. In each instance we provided the LPR-solution as an initial solution for the mixed integer program. Yet, CPLEX was neither able to prove optimality for these instances nor to deliver good lower bounds. We therefore adapted Bender's algorithm [4] to the min-support path-raising problemas described below.

B.1 Lower Bounds by Bender Cuts

Suppose we fix in the mixed integer program (7) the binary variables z to some value $\bar{z} \in \{0,1\}^m$. Then we have to decide if it is possible to find feasible arc-offsets $\alpha = (\alpha_a)_{a\in A}$ such that $\alpha_a > 0$ implies $\bar{z}_a = 1$, i.e. to decide whether the following purely linear program is feasible:

$$\min\ 0\cdot\alpha \quad \text{subject to} \quad \begin{array}{rcll} \alpha & \in & \mathscr{F} & \\ \bar{z}_a \cdot M & \geq & \alpha_a & \forall a \in A \end{array} \tag{9}$$

Since (9) is clearly bounded, Strong Duality implies that the primal problem (9) is infeasible iff the dual linear program to (9) is unbounded.

Lemma 1. *Given fixed values $\bar{z}_a \in \{0,1\}$ for all $a \in A$, the linear program* (9) *is feasible if and only if the following linear program* (10) *is bounded.*

$$\begin{array}{rrcll} \max & -\sum_{a\in A}\big(M\cdot\bar{z}_a\cdot g_a + c_a\sum_{i\in[k]} f_a^i\big) & & & \\ \text{subject to} & \sum_{a\in\delta^+(v)} f_a^i - \sum_{a\in\delta^-(v)} f_a^i & = & 0 & \forall v\in V, \forall i\in[k] \\ & \sum_{i\in[k]} f_a^i - g_a & \leq & 0 & \forall a\in A \\ & f_a^i & \geq & 0 & \forall a\in A\setminus\bar{A}_i, \forall i\in[k] \\ & f_a^i & \in & \mathbb{R} & \forall a\in\bar{A}_i, \forall i\in[k] \\ & g_a & \geq & 0 & \forall a\in A \end{array} \tag{10}$$

Note that the feasible region of (10) does not depend on the concrete values of $\bar{z}$. Suppose (10) is unbounded, then we can can find an extreme ray r that leads to unboundedness, more precisely denote by $r(f_a^i)$ the value of the variable f_a^i respectively by $r(g_a)$ the value of g_a. Therefore, we can restate (7) with the constraints that none of the rays in (10) leads to unboundedness. This yields the integer program (11), where each ray corresponds to a cut.

$$\begin{array}{rrcll} \min & \sum_{a\in A} w_a z_a & & & \\ \text{subject to} & \sum_{a\in A}\big(M\cdot r(g_a)\cdot z_a + c_a\sum_{i\in[k]} r(f_a^i)\big) & \geq & 0 & \forall \text{ rays } r \\ & z_a & \in & \{0,1\} & \forall a\in A \end{array} \tag{11}$$

In general there are exponentially many rays, but we may be lucky that we only need to generate a subset of these, in order to obtain an optimal solution. Benders' algorithm for the min-toll-booth problem works essentially like this: Starting with no cuts at all, in each iteration we solve the integer program (11) and obtain a solution vector z. According to z we solve the linear program (10). In case (10) is bounded, we return z as solution, otherwise we obtain a ray and generate a cut for the next iteration. In each iteration, the value $\sum_{a\in A} z_a$ is a lower bound

instance	cuts	lower bound
Anaheim	53	24
Friedrichshain	213	23*
Mitte	62	19
Prenzlauer Berg	65	19
Tiergarten	139	17
Sioux Falls	1779	31

Table 3: Achieved lower bounds with Benders' algorithm. Friedrichshain has been solved up to optimality.

for the optimal solution. Since there are only finitely many rays, the procedure terminates in the worst case after enumerating all of them.

The obtained lower bounds as also the number of generated cuts for each instance are listed in Table 3. The Friedrichshain instance has been solved up to optimality. Remarkably, there were just 213 iterations needed to obtain the optimal solution of 23 toll booths. t

For the larger instances Anaheim, Mitte and Prenzlauer Berg about 60 cuts have been generated. The obtained lower bounds are 24 resp. 19. To reduce the number of cuts needed, it would be helpful to provide the obtained cuts by combinatorial means in order to exploit the underlying network structure of the problem. It would be an interesting direction of future research, to give a combinatorial characterization of the generated cuts.

B.2 Results for Multi-Commodity Instances

In Table 4 we compare the toll solutions for multi-commodity instances. The entry ρ^*_{MTB} refers to the approximation ratio with respect to the obtained lower bounds in Table 3. In terms of approximation.theory, this factor is larger than the true value, yet it indicates that LPR performs quite well compared to MRT and MC as solution concept. Despite the above mentioned computational deficiencies, the picture regarding the performance of the considered toll solutions does not change when moving from single- to multi-commodity instances. LPR's worst-case performance factor with respect to MRT is bounded by 1.5. For MNTB it is proveably bounded by 2.5, though the true value is most likely smaller. Thus, regardless of which performance measure is targeted, LPR delivers a close to optimal performance on all multi-commodity instances considered in this paper. Conversely, MC and MRT exhibit significantly larger approximation factors with respect to the objectives $\sum_{a\in A} \alpha_a$ and $|Z(\alpha)|$. In contrast to the single-commodity case, MC performs quite well with respect to the minimum revenue objective; the approximation factor stays below 8.

Marginal Cost Tolls (MC)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ^*_{MTB}
Anaheim	444338.61	7.8050	57.74	3.0294	749	31.2083
Friedrichshain	188451.99	1.2200	294.21	1.1715	156	6.7826
Mitte	141011.62	1.5666	245.15	1.4098	269	14.1579
Prenzlauer Berg	256800.84	2.0045	352.42	1.7143	251	13.2105
Tiergarten	77082.92	2.0881	129.96	1.4556	277	16.2941
Sioux Falls	14457124.56	6.9996	1280.03	6.6720	76	2.4516

Minimum Revenue Tolls (MRT)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ^*_{MTB}
Anaheim	56930.19	1	1115.75	61.6868	266	11.0833
Friedrichshain	154477.87	1	5888.718	23.4489	156	6.7826
Mitte	90012.20	1	18514.88	106.4747	272	14.3158
Prenzlauer Berg	128113.88	1	15780.84	76.7663	252	13.2632
Tiergarten	36915.27	1	18049.11	202.1630	251	14.7647
Sioux Falls	2065417.12	1	197.46	1.0292	39	1.2581

Minimum Total Tolls (LPR)

instance	$\sum_a f_a^* \alpha_a$	ρ_{MRT}	$\sum_a \alpha_a$	ρ_{MTT}	$\lvert Z(\alpha)\rvert$	ρ^*_{MTB}
Anaheim	82275.39	1.4452	19.06	1	123	5.125
Friedrichshain	165449.86	1.0710	251.13	1	41	1.7826
Mitte	96639.32	1.0736	173.89	1	46	2.4211
Prenzlauer Berg	151950.43	1.1861	205.57	1	47	2.4737
Tiergarten	49756.32	1.3479	89.28	1	39	2.2941
Sioux Falls	2077467.65	1.0058	191.85	1	39	1.2581

Table 4: Results on multi-commodity instances with respect to the three different objectives; values are rounded.