Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 348 (2005) 217–225

Theoretical Computer Science

www.elsevier.com/locate/tcs

Linear tolls suffice: New bounds and algorithms for tolls in single source networks

Lisa Fleischer[1]

T. J. Watson Research, IBM, Yorktown Heights, P. O. Box 218, NY 10598, USA

Abstract

We show that tolls that are *linear* in the latency of the maximum latency path are necessary and sufficient to induce heterogeneous network users to independently choose routes that lead to traffic with minimum average latency. This improves upon the earlier bound of $O(n^3 l_{\max})$ given by Cole, Dodis, and Roughgarden in STOC 03. (Here, n is the number of nodes in the network; and $l_{\max}$ is the maximum latency of any edge.) Our proof is also simpler, relating the Nash flow to the optimal flow as flows rather than cuts.

We model the set of users as the set [0, 1] ordered by their increasing willingness to pay tolls to reduce latency—their *valuation of time*. Cole et al. give an algorithm that computes optimal tolls for a bounded number of agent valuations, under the very strong assumption that they know which path each user type takes in the Nash flow imposed by these (unknown) tolls. We show that in series parallel graphs, the set of paths traveled by users in any Nash flow with optimal tolls is *independent* of the distribution of valuations of time of the users. In particular, for any continuum of users (not restricted to a finite number of valuation classes) in series parallel graphs, we show how to compute these paths without knowing α.

We give a simple example to demonstrate that if the graph is not series parallel, then the set of paths traveled by users in the Nash flow depends critically on the distribution of users' valuations of time.

Keywords: Nash equilibria; Network flow; Selfish routing; Edge pricing

1. Introduction

In a (transit/Internet/telecommunications) traffic network, the *latency* of a link is the time required to travel from one end of the link to the opposite end. In a simple model of traffic, the latency of an edge is a nonnegative, nondecreasing function of the flow on the edge: Given graph $G = (V, E)$, with $n = |V|$, the latency of edge $e \in E$ is a function $l_e : R^+ \cup \{0\} \to R^+ \cup \{0\}$. We consider such a model in this paper and look at how to induce selfish users of the network to follow a traffic pattern that minimizes the average latency experienced by the users. Such a traffic pattern is called a *system optimal flow*. If we assume that the total flow volume from s to t is 1, then a system optimal flow is equivalently expressed as an s–t flow f of value 1 that minimizes $\sum_{e \in E} l_e(f_e) f_e$.

A selfish user traveling from s to t chooses a path P that minimizes the latency experienced on the path: given that all other network traffic is fixed as f, the traveler minimizes $\sum_{e \in P} l_e(f_e)$. This model is introduced in [13]. If each

E-mail address: lkf@watson.ibm.com

URL: http://www.research.ibm.com/people/l/lkf/.

[1]This research was supported in part by NSF grant CCR-0049071.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.09.014

user succeeds in doing this, we will call the resulting traffic pattern a *Nash flow*, since it is a Nash equilibrium for the routing game where each player is a user with action space the set of all s–t paths. The Nash flow may be far from a system optimal flow [9,11].

Tolls are a well-known method to induce homogeneous users to choose paths that minimize the average latency when users selfishly choose paths that minimize individual latency plus toll. For *marginal cost tolls*: $\tau_e = l'_e(f_e) f_e$, the *Nash flow with tolls* τ is a system optimal flow (see for example [2,10]).

What happens if the users are heterogeneous? To model this, consider for each agent a there is some multiplier $\alpha(a)$ that represents a's valuation of time. User a seeks a path P that minimizes $\sum_{e \in P} \alpha(a) l_e(f_e) + \tau_e$.[2] Early work considers when users pay different tolls on the same edge, according to their multiplier α [6,12]. This is unsatisfying, and also hard to enforce, as it requires knowing individual users' α values, as opposed to a distribution of α-values of users.

Instead, a natural question is, given a distribution α, find a unique toll for each edge that induce users to choose a prespecified flow $\tilde{f}$. We call such tolls *optimal tolls*. Cole et al. [5] show that optimal tolls exist when $\tilde{f}$ is the system optimal flow. Their proof is nonconstructive, and they bound the size of the maximum toll necessary to achieve this by $\alpha_{\max} l_{\max} n^3$, where $\alpha_{\max} = \max_a \alpha(a)$ and $l_{\max} = \max_e l_e(1)$. The proof uses Brouwer's fixed point theorem, and a complicated argument about cuts in the network.

We show that *linear tolls suffice*: the optimal toll on each edge need be no more than the latency of the maximum latency path in the minimum average latency flow times the maximum valuation of time on that edge. In fact, the total toll paid by a user (over all edges in a path) is bounded by this same quantity. This quantity is always less than $\alpha_{\max} l_{\max} n$. This bound is also tight: there are instances that require tolls that are linear in the size of the maximum latency path in the network. Our proof relates the Nash flow to the system optimal flow directly as flows, rather than indirectly through cuts. This linear bound also holds in the multiple source, single sink setting; and to induce any given acyclic flow—not just the system optimal flow.

We consider the set of users as the set [0, 1] ordered by their increasing willingness to pay tolls to reduce latency. Thus $\alpha : [0, 1] \to R^+$ is a nondecreasing function. For the case that α is a step function, Cole et al. [5] show that optimal tolls can be computed by solving a linear program, under the following very strong assumption: The paths of users with valuation α_i in the Nash flow with the optimal tolls is known, even though the optimal tolls are unknown.[3] The correctness of their algorithm relies on their nonconstructive proof of the existence of tolls.

What if these set of paths are not given? We show that in series parallel graphs, the set of paths traveled by users in any Nash flow with optimal tolls is *independent* of the valuations of time of the users: In series parallel graphs, the set of paths is determined by $\tilde{f}$ only. As a consequence, we give the first algorithm that computes tolls for users from a distribution given by *any* increasing function α (not restricted to a finite number of valuation classes), in series parallel graphs. For this we assume that we have access to an oracle that given $a \in [0, 1]$ returns $\alpha(a)$. We compute the tolls using at most $m + 1$ oracle calls.

In general graphs, it is unknown if even verifying that a given set of tolls is optimal for a given α function is in P: Carstensen [4] constructs an example with fixed latencies and tolls where the number of paths that correspond to shortest paths for varying values of α can be exponential in the size of the graph.

We conclude by giving a simple example to demonstrate that if the graph is not series parallel, then the set of paths traveled by users in the Nash flow depends critically on the function α.

2. Preliminaries

Let $G = (V, E, l, s, t)$ denote a directed graph with nonnegative, nondecreasing, continuous latency functions l_e associated with each edge $e \in E$, source node $s \in V$ and sink node $t \in V$. Let $m = |E|$. The latency of edge e is a function solely of the flow on edge e. Given a set of edges F, and a function x defined on E, we denote by x_F the total of

[2] Cole et al. use $l_e(f_e) + \beta(a)\tau_e$ to evaluate edge e [5]. By taking $\alpha(a) = 1/\beta(a)$, our notation is equivalent to theirs.

[3] Cole et al. do not state this assumption in [5], but the LP they use to compute tolls requires the breakdown of the flow into flows per commodity. As we show in Section 4.2, there are simple graphs for which the decomposition of the flow according to commodity depends on the specific α function. Thus, how to obtain this decomposition is an important question, not addressed in [5]. When α is a step function, an algorithm that does not require this decomposition is presented in [7].

x on F: $x_F := \sum_{e \in F} x_e$, where x_e is the function value of x at $e \in E$. For a scalar x, we denote by $[x]^+$ the maximum in $\{x, 0\}$.

A path from s to t is an ordered subset of $V \times E$ of the form $(s = v_0, e_1, v_1, e_2, \ldots, e_k, v_k = t)$ with the property that $e_i = (v_{i-1}, v_i)$. For any subset Γ of $V \times E$, we denote by $E(\Gamma)$ the set $\Gamma \cap E$. Let $\mathcal{P}_{yz}$ be the set of y–z paths in G. For s–t paths, we simply use $\mathcal{P}$.

An *s–t flow* in G is a nonnegative function $f : E \to R^+ \cup \{0\}$ that satisfies *flow conservation* at all nodes of $V \backslash \{s, t\}$: $\sum_v f_{vw} = \sum_v f_{wv}$. The *volume* of a flow is the quantity of flow that leaves s, denoted $|f| := \sum_{v \in V} f_{sv}$. A *path flow* is a flow on a path from s to t in G. A *cycle flow* is a flow around a cycle in G. A *flow decomposition* of a flow f is a set $\Gamma = \{\gamma_1, \ldots, \gamma_r\}$ of path flows and cycle flows whose sum together is f: $\sum_{i=1}^{r} \gamma_i = \Gamma$. Every flow has a flow decomposition into at most $|E|$ path and cycle flows. If f is acyclic, then the flow decomposition consists of path flows only. A flow around a cycle may be *canceled* by sending the flow backward around the cycle, in effect subtracting the flow. A cycle is *canceled* if flow of value equal to the minimum flow value on an edge in the cycle is sent backward around the cycle. For more basic facts on flows, see [1].

The *cost* of an edge e with latency $l(e)$ and toll $\tau(e)$ for agent a is $\alpha(a)l(e) + \tau(e)$. The cost of a path P for agent a is simply the sum of the costs of the edges in the path. While our results apply to any acyclic flow, we discuss the results in the context of the system optimal flow. Thus, the latency of edge e is the latency of the edge in the system optimal flow. Thus, we define the *capacity* of the edge e as the value of flow on e in the system optimal flow: $|\tilde{f}_e|$, and the capacity of a path P to be $\min_{e \in P} |\tilde{f}_e|$.

When l is convex, the system optimal flow can be computed in polynomial time via solving a convex program. While the system optimal flow may not be unique, we will assume throughout the rest of this paper that we are talking about an arbitrary, but fixed system optimal flow $\tilde{f}$.

Given congestion-aversion function $\alpha : [0, 1] \to R^+$ and toll vector $\tau : E \to R^+ \cup \{0\}$, we denote the Nash flow by f_α^τ. When α is clear from context, as it is throughout most of the paper, we will simply use f^τ. Given Nash flow f^τ, we denote by $\gamma(a)$ the path used by user a in f^τ. The Nash flow exists, and has some interesting properties summarized in the following lemma [5].

Lemma 1. *For tolls τ, there exists a Nash flow f^τ with edge latencies l that satisfies*
(i) *For any path $P \in \mathcal{P}$, the agents assigned to P by f^τ form a (possibly empty or degenerate) subinterval of* $[0, 1]$.
(ii) *If $a \leqslant b$, then $l(\gamma(a)) \geqslant l(\gamma(b))$.*
(iii) *If $a \leqslant b$, then $\tau(\gamma(a)) \leqslant \tau(\gamma(b))$.*

Since the latency functions are nonnegative, we assume without loss of generality that the Nash flow and optimal flow induce directed, acyclic graphs. As long as l is nondecreasing and continuous, the Nash flow may be computed in general by solving a convex program.

3. Linear tolls are necessary and sufficient

Let $l_{\max}$ be the maximum latency of an edge in $\tilde{f}$. Clearly $l_{\max} \leqslant \max_e l_e(1)$. Let $L = \max_{P \in \mathcal{P}} \sum_{e \in P} l_e(1)$.

Theorem 2. *Tolls that are bounded by* $1 + \alpha(1)L$ *suffice to induce a minimum latency flow as a Nash flow.*

Proof. Let $T = 1 + \alpha(1)L$. Let $\sigma(\tau_e) = \min\{T, [\tau_e + (f_e^\tau/\tilde{f}) - 1]^+\}$. We show that if σ has a fixed point τ', then $f_e^{\tau'} = \tilde{f}_e$ for all $e \in E$. Then, since σ is continuous [5] and bounded, we can invoke Brouwer's fixed point theorem [3] to obtain the result.

Suppose there is a "bad" fixed point—a fixed point τ of σ with $f^\tau \neq \tilde{f}$. Then every edge e with $f_e^\tau > \tilde{f}_e$ has $\tau_e = T$ (a *taxed* edge); and every edge e with $f_e^\tau < \tilde{f}_e$ has $\tau_e = 0$ (an *untaxed* edge). We create a graph $\hat{G}$ on V with an arc (v, w) with capacity $\tilde{f}(v, w) - f^\tau(v, w)$, if $f^\tau(v, w) < \tilde{f}(v, w)$ (a forward arc), and an arc (w, v) with capacity $f^\tau(v, w) - \tilde{f}(v, w)$, if $f^\tau(v, w) > \tilde{f}(v, w)$ (a backward arc). In words, $\hat{G}$ is the graph of the flow $\tilde{f} - f^\tau$. If $\hat{G}$ is nonempty, then a flow decomposition of $\tilde{f} - f^\tau$ yields only cycles and no paths, since the volume of both flows is the same. Thus if $\hat{G}$ is nonempty, it contains a cycle, with at least one forward and one backward arc, since both f^τ and $\tilde{f}$ are acyclic.

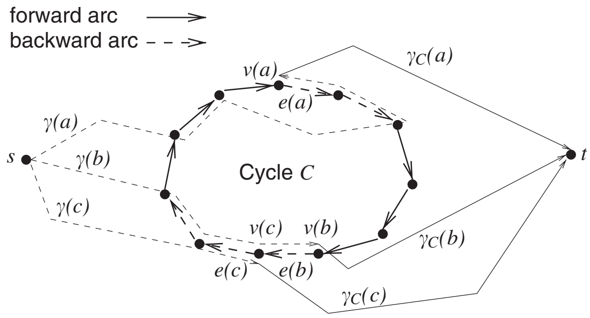

Fig. 1. A cycle in $\hat{G}$ and paths of agents $a, b, c \in A_C$ in f^τ.

Intuition of proof. Suppose the cost of a forward arc for agent a is $\alpha(a)l_e(\tilde{f})$, and the cost of backward arc for agent a is $-\alpha(a)l_e(f^\tau) - T$. Let C be a cycle in $\hat{G}$ such that agent a travels on the counterpart to each backward edge in C. Since f^τ is Nash, C cannot have negative cost. Let $|C|$ be the number of taxed (backward) edges on a cycle C. Since $\tilde{f}$ is acyclic, $|C| \geqslant 1$. Thus, $\alpha(a)l_C(\tilde{f}) \geqslant \alpha(a)l_C(f^\tau) + T|C|$, or $T \leqslant \alpha(a)(l_C(\tilde{f}) - l_C(f^\tau))/|C| \leqslant n * l_{\max}\alpha(1)$, a contradiction if $L := n * l_{\max}$.

To expand this intuition, we show that if we have a "bad" fixed point, then there exists an agent a with incentive to change its path $\gamma(a)$, which contradicts f^τ being a Nash flow:

Consider a cycle C in $\hat{G}$. Let A_C be the set of agents a such that $\gamma(a) \cap C$ contains a taxed edge. Note that by definition of $\hat{G}$ and C, the connected components of $\gamma(a) \cap C$ consist either entirely of taxed (backward) edges, or untaxed (forward) edges (see Fig. 1). This is because all flow in f^τ is going in the same direction as flow in $\tilde{f}$—in the direction of forward arcs, and in the opposite direction of backward arcs. For purposes of this proof, we are just interested in the connected components of $\gamma(a) \cap C$ that are taxed. For each $a \in A_C$, define $e(a)$ to be the last taxed edge on $\gamma(a) \cap C$, define $v(a)$ to be the end point of $e(a)$ closest to t on $\gamma(a)$, and define $\gamma_C(a)$ as the subpath of $\gamma(a)$ from $v(a)$ to t. These definitions are illustrated in Fig. 1. By definition, $\gamma_C(a)$ does not contain a taxed edge on C.

Claim 3. *For each $a_1 \in A_C$, there is an alternate path for some agent $a_2 \in A_C$ from s to t that uses $\gamma_C(a_1)$ instead of $\gamma_C(a_2)$ and at least one fewer taxed edge from C than $\gamma(a_2)$ does.*

We prove this claim: There is a backward arc leaving $v(a_1)$ on C—it corresponds to arc $e(a_1)$ in $\gamma(a_1)$.

Case 1: The arc entering $v(a_1)$ on C is a backward arc. In this case, there is distinct agent $a_2 \in A_C$ such that $v(a_1) \in \gamma(a_2)$. (In Fig. 1, $v(c)$ is an example of this.) Thus a_2 can follow $\gamma(a_1)$ from $v(a_1)$ instead of using $\gamma(a_2)$. In doing so, agent a_2 will use at least one fewer taxed arc from C—it will not use the arc in E that corresponds to the backward arc entering $v(a_1)$. (In Fig. 1, b can take $\gamma_C(c)$ instead of $e(b) \cup \gamma_C(b)$.)

Case 2: The arc entering $v(a_1)$ on C is a forward (untaxed) arc. (In Fig. 1, both $v(a)$ and $v(b)$ are examples of this.) We trace backward around C starting from $v(a_1)$ until we come to the next node y that has an entering backward (taxed) arc. Some agent a_2 has a path $\gamma(a_2)$ that enters C at y. Since all arcs on C from y to $v(a_1)$ are forward arcs, a_2 can be rerouted from y along C to $v(a_1)$ and then onto $\gamma_C(a_1)$ to t. (In Fig. 1, a can take $\gamma_C(b)$, and b can take $\gamma_C(a)$.) This path has fewer taxed edges on C than the path from y to t along $\gamma(a_2)$, since, in particular, it does not include the arc leaving y that corresponds to a backward arc in C. *This establishes the claim.*

Let $\gamma_C(a^*)$ be the least toll path among all $\gamma_C(a)$, $a \in A_C$. By the claim, some agent $b \in A_C$ can replace its current path $\gamma_C(b)$ by using at least one fewer taxed edge in C and the subpath $\gamma_C(a^*)$. No additional edges outside $\gamma(b) - C - \gamma_C(b)$ are added to this new path for b. If somehow a cycle is created in this new path, this cycle is shortcut. Thus, the change in cost that agent b experiences by choosing this path instead is at most the difference in tolls of the two paths, plus the latency of the new path minus the toll of at least one taxed edge on C. This is

$$\tau_{\gamma_C(a^*)} - \tau_{\gamma_C(b)} + \alpha(b)L + [-T] \leqslant \alpha(b)L + [-T].$$

Since f^τ is a Nash flow, this must be $\geqslant 0$. This implies that $T <= \alpha(1)L$, a contradiction. □

Remarks. 1. The bound in Theorem 2 also holds when there are multiple sources and a single sink (or multiple sinks and a single source).

2. Theorem 2 may be strengthened by bounding the toll on each edge separately. In the map γ for edge e we can replace T with $T_e := \min\{\alpha(a) \mid e \in \gamma(a)$ in Nash flow of optimal tolls$\}$. The result is that the toll for agent a is not more than the latency of the maximum latency path times *her* valuation of time.

3. Theorem 2 actually also proves the existence of tolls to induce any pre-specified acyclic flow. Subsequent to this work, and after discussions with Kamal Jain, Mohammad Mahdian, and Tim Roughgarden, we realized there is a very simple proof to show that the *total* toll paid by user a is at most $\alpha(a)L$, if we know that tolls exist. This is discussed in Section 3.2.

3.1. Linear tolls are necessary

The bound in Theorem 2 is trivially tight for uniform α: consider a simple graph that consists of two edges from s to t, one with latency L, the other with latency Lx^r for $r > 0$.[4] The Nash flow will send all flow on the edge with latency x^r. In order to make the both paths attractive to users at the optimal flow, a toll of value $\alpha L\ (1 - 1/(1+r))$ must be imposed on the bottom edge. For r large, this approaches αL.

3.2. Bounding total toll

Theorem 4. *Let τ be a set of tolls that induce the system optimal flow. There exist tolls τ' that induce the system optimal flow such that user a pays at most $\alpha(a)L$.*

Proof. We first note that τ can be modified to τ' such that τ' induces the system optimal flow and that there is a path of edges with 0 toll from s to each sink.[5]

Given tolls τ, let S be the set of nodes reachable from s on paths of edges with 0 toll. Let $\varepsilon > 0$ be the smallest toll on an edge leaving S. Reduce the toll on each edge leaving S by ε and increase the toll on each edge entering S by ε. The sum of tolls on every s–t path decreases by exactly ε: each time the path enters S it must leave S. To enter S the first time, it has to first leave S. Thus the number of edges on the path that leave S is exactly one more than the number that enter. Thus $\alpha(a)l(\gamma) + \tau(\gamma)$ decreases by ε for all γ. Hence the paths traveled by users are still the minimum cost paths for these users, and the new tolls are also optimal. The size of S has increased by at least 1, and so after repeating this at most n times, there is a path of 0 toll edges from s to each sink.

Let γ_0 be a path with 0 tolls. The cost of this path to user a is $\alpha(a)l(\gamma_0) \leqslant \alpha(a)L$. Since user a prefers $\gamma(a)$, we have that $\alpha(a)l(\gamma(a)) + \tau'(\gamma(a)) \leqslant \alpha(a)L$, which implies that $\tau'(\gamma(a)) \leqslant \alpha(a)L$. □

4. Computing tolls for general α

In this section, we assume that a pre-specified flow $\tilde{f}$ is given. We seek tolls τ such that the Nash flow with agent $a \in [0,1]$ seeking to minimize $\alpha(a)l_P(f^\tau) + \tau(P)$ is $\tilde{f}$. We make no assumptions on the nondecreasing function α which reflects the aversion of each agent to congestion. We assume that we have access to α via an oracle that responds to the query $a \in [0,1]$ with $\alpha(a)$.

All latencies in this section refer to the latency of an edge given flow $\tilde{f}$. Thus $l = l(\tilde{f})$ in this section.

4.1. Series parallel graphs

We describe an algorithm that computes optimal tolls if G is series-parallel.

[4] This is a variant of an example illustrated by Pigou [10].
[5] This observation is also made in [5].

4.1.0. Definitions.

A basic series-parallel graph is an edge with terminals a and b. Two series parallel graphs can be joined in a *series composition* by associating terminal b of the first with terminal a of the second. Two series parallel graphs can be joined in a *parallel composition* by associating terminal a of the first with terminal a of the second, and associating terminal b of the first with terminal b of the second. A maximal set of contiguous series compositions is a *series component*. A maximal set of parallel compositions is a *parallel component*.

4.1.1. Algorithm ComputeToll

Step 1. Create a longest-path-first flow decomposition of the minimum latency flow $\tilde{f}$: Find a longest latency path P in $\tilde{f}$, and set the volume of path flow γ along P to be the capacity of P in $\tilde{f}$. Remove γ by setting $\tilde{f} = \tilde{f} - \gamma$, and iterate. Ties are broken among paths by assigning a unique numerical key to each edge, and breaking ties lexicographically. Let η be the number of paths in the decomposition. Note that $\eta \leqslant |E|$, since each path-flow removal reduces the support of $\tilde{f}$ by at least one edge. Let this collection of paths be $\Gamma = \{\gamma_1, \dots, \gamma_\eta\}$, indexed in order of nonincreasing lengths l (so that γ_1 is the longest latency path).

Step 2. Assign agents to the path flows in Γ: The set of agents with the highest α value are assigned to the shortest latency path. That is, agents in $(1 - |\gamma_\eta|, 1]$ are assigned to γ_η. Agents with the next highest α values to next path; and so on, until agents in $[0, |\gamma_1|]$ are assigned to γ_1. In this way, the agents are partitioned into η groups according to the path to which they are assigned. Let $[\alpha_1, \beta_1], \dots, (\alpha_\eta, \beta_\eta]$ be the ranges of α determined by this partition. Thus, $\alpha_i \leqslant \beta_i \leqslant \alpha_{i+1}$ for all i.

Step 3. Assign tolls to edges: From (G, l, α) create a new instance (G, l, α'), where α' is a step function that depends on α, as follows. Let $\alpha'_{2i-1} = \alpha_i$, and let $\alpha'_{2i} = \beta_i$. Let the volume of users of types α'_{2i-1} and α'_{2i}, denoted by r_{2i-1} and r_{2i}, respectively, be each equal to $|\gamma_i|/2$.

Find a feasible solution to the following set of inequalities in variables z and τ. The resulting value τ_e is the toll for edge e.

$$
\begin{aligned}
& z_s^i = 0 \quad \forall\, 1 \leqslant i \leqslant 2\eta, \\
& z_w^i - z_v^i \leqslant \alpha'_i l_{vw}(\tilde{f}_{vw}) + \tau_{vw} \quad \forall\, i, \ \ \forall\, (v, w) \in E(G), \\
& \sum_{i=1}^{2\eta} r_i z_t^i = \sum_{i=1}^{2\eta} \sum_{e \in \gamma_{\lceil i/2 \rceil}} [\alpha_i l_e(\tilde{f}_e) + \tau_e] r_i .
\end{aligned} \tag{1}
$$

4.1.2. Analysis of algorithm ComputeToll

Let $\mathrm{dist}_l(v, w, F)$ be the latency of the least l-latency path between v and w using edges in F. The ordered set of paths $\{\gamma_1, \gamma_2, \dots, \gamma_r\}$ in G with edge-length function l is said to have the *decreasing subpaths property* if for all $i < j$, $\{v, w\} \subset V(\gamma_i) \cap V(\gamma_j)$ implies that $\mathrm{dist}_l(v, w, \gamma_i) \geqslant \mathrm{dist}_l(v, w, \gamma_j)$.

Lemma 1 has the following simple corollary.

Corollary 5. *For any tolls τ, there exists a Nash flow f with edge latencies l such that for all $a \leqslant b$ the ordered set $\{\gamma(a), \gamma(b)\}$ satisfies the decreasing subpaths property.*

We now show that in series parallel graphs, a set of paths has the decreasing subpaths property if an only if it corresponds to a longest path decomposition.

Lemma 6. *Any path decomposition of a series parallel graph G is a longest path decomposition if and only if it has the decreasing subpaths property.*

Proof. Let $\Gamma = \{\gamma_1, \dots, \gamma_\eta\}$ be a longest path decomposition. Consider any two indices $1 \leqslant i < j \leqslant \eta$ and nodes $\{v, w\} \in V(\gamma_i) \cap V(\gamma_j)$ such that $\mathrm{dist}_l(v, w, E(\gamma_i)) < \mathrm{dist}_l(v, w, E(\gamma_j))$. Swapping the subpath of γ_j from v to w with the parallel subpath of γ_i results in a modified path decomposition with γ_i longer than before the swap. This contradicts that γ_i is from a longest path decomposition. Thus Γ satisfies the decreasing subpaths property.

Now suppose $\Gamma = \{\gamma_1, \ldots, \gamma_\eta\}$ is a decomposition of a series parallel graph G that satisfies the decreasing subpaths property. In order to show that Γ is a longest path decomposition it is sufficient to show that γ_1 is a longest path. Then, by induction, since $\Gamma - \gamma_1$ is a decomposition of $G - \gamma_1$ that satisfies the decreasing subpaths property, $\Gamma - \gamma_1$ is a longest paths decomposition of $G - \gamma_1$.

Suppose γ_1 is not a longest path in G. Let Y be the smallest parallel component such that $E(Y)\Gamma_1 \neq \emptyset$ and Γ_1 is not a longer path through Y. Let s' and t' be the end nodes of Y. By definition of Y, a longest path through Y is internally node disjoint with γ_1. Call one such longest path through Y by p. Since Γ obeys the decreasing subpaths property, edges in the subpath p cannot be on just one path in Γ, and thus must appear on at least two paths Γ. Let γ_j be the last such path, and γ_i be some other such path that satisfies the following properties: there is a smallest parallel component X such that γ_i and γ_j intersect X, $\gamma_j \cap E(X) = p$, and $\gamma_i \cap E(X)$ is not a longest path in X. Since both $\gamma_j \cap E(Y)$ and $\gamma_i \cap E(Y)$ must have length less than p by the decreasing subpaths property, such a component must exist if p is not on γ_1. But this contradicts the fact that $\{\gamma_i, \gamma_j\}$ obeys the decreasing subpaths property. Thus γ_1 must be a longest path in G. □

A simple consequence of Lemma 6 is that a longest path decomposition of series parallel graph G is also a shortest path decomposition, since a symmetric argument shows that a shortest path decomposition obeys a symmetric increasing subpaths property.

Together Lemma 6 and Corollary 5 imply that the set of paths used by users in any Nash flow forms a longest path decomposition of G. Thus, even without knowing the exact distribution α, we know that the set of paths traveled in G by any set of selfish users. In addition, if the users are ordered according to α value, we know which users travel on which path.

We now invoke a theorem of Cole et al. [5] that states that for an instance (G, l, α) such that α is a step function, if the path decomposition of the Nash flow with optimal tolls τ is known, then it is possible to compute τ by finding a feasible solution to a set of inequalities. We paraphrase their Theorem 4.2 and the discussion that precedes it below.

Theorem 7 (*Cole, Dodis, Roughgarden*). *Let (G, l, α) be an instance in which α takes on only finitely many distinct values. Let r_i be the volume of users with valuation α_i. Suppose τ induces Nash flow $\tilde{f}$, and let $\tilde{f}^i$ be the flow induced by users with valuation α_i. Then τ and $\tilde{f}$ satisfy the following system of inequalities.*

$$
\begin{aligned}
&z_s^i = 0 \quad \forall i,\\
&z_w^i - z_v^i \leqslant \alpha_i' l_{vw}(\tilde{f}_{vw}) + \tau_{vw} \quad \forall\, i, \ \ \forall\,(v, w) \in E(G),\\
&\sum_i r_i z_t^i = \sum_i \sum_{e \in E} [\alpha_i l_e(\tilde{f}_e) + \tau_e] \tilde{f}_e^i.
\end{aligned}
$$

A corollary of this theorem is that if $\tilde{f}^i$ is known, and the number of distinct values of α is polynomial, then τ can be computed in polynomial time.

Theorem 8. *For any instance (G, l, α) where G is series-parallel, and α is an arbitrary increasing function on* [0, 1], *algorithm* ComputeToll *finds the optimal tolls.*

Proof. By Steps 1 and 2, Corollary 5, and Lemma 6, the instance described in Step 3 of ComputeToll is of the form required by Theorem 7. Thus, the solution to this system of inequalities yields optimal tolls for the problem with valuation function α'. Together Corollary 5 and Lemma 6 imply that the set of paths used with valuation function α is the same set of paths used with valuation function α'. Lemma 1 (i) and (ii) implies that the bounds for the α values of the users on such paths is the same as the bounds for the α' values. Since the tolls are acceptable to users with extreme α-values on each path, they are okay for all users on the paths, and hence the tolls computed for α' are also optimal for α. □

4.2. Other graphs

If G is not series parallel, then for different functions α, flow patterns of agents with optimal tolls may be different. Thus, there is no universal flow decomposition that holds for all α.

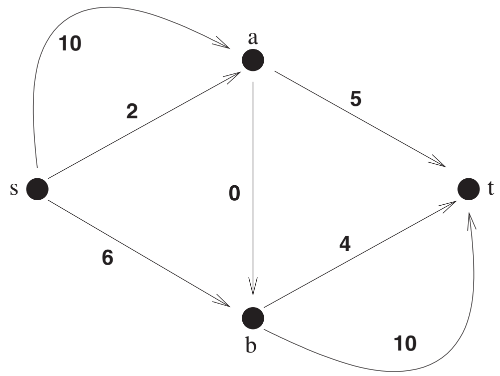

Fig. 2. In this network, the flow patterns of agents in the Nash flow with optimal tolls depend on the distribution α. The number on each arc represents the latency of the arc in the system optimal flow.

For example, consider the graph on 4 nodes $\{s, a, b, t\}$ with arc set and optimal latencies $\{(s, a, 2), (s, a, 10), (s, b, 6), (a, b, 0), (a, t, 5), (b, t, 4), (b, t, 10)\}$ depicted in Fig. 2. This graph is not series-parallel, but would be series-parallel without any one of the arcs (s, b), (a, b), or (a, t).

If $\alpha(a) = 1$ for $a \in [0, \frac{1}{3}]$, $\alpha(a) = \frac{6}{5}$ for $a \in (\frac{1}{3}, \frac{1}{2}]$, $\alpha(a) = \frac{4}{3}$ for $a \in (\frac{1}{2}, \frac{5}{6}]$ and $\alpha(a) = 2$ for $a \in [\frac{5}{6}, 1]$, then the optimal toll vector is $(10, 0, 4, 0, 5, 7, 0)$ and the paths taken by users in the Nash flow are $\{(s, a, 2), (a, t, 5)\}$, $\{(s, a, 10), (a, b, 0), (b, t, 10)\}$, and $\{(s, b, 6), (b, t, 4)\}$.

On the other hand, if $\alpha(a) = 1$ for $a \in [0, \frac{2}{3})$ and $\alpha(a) = 5$ for $a \in [\frac{2}{3}, 1]$, then the optimal toll vector is $(8, 0, 0, 0, 1, 6, 0)$ and the paths taken by users in the Nash flow are $\{(s, a, 2), (a, b, 0), (b, t, 4)\}$, $\{(s, a, 10), (a, t, 5)\}$, and $\{(s, b, 6), (b, t, 10)\}$.

5. Conclusions

In this paper we have provided a improved bound on the size of tolls needed to induce heterogeneous, selfish users to obey the system optimal flow in single source networks; and provided an algorithm to compute such tolls in series parallel networks. This work was motivated by an interest in understanding the tolls problem better so as to address the existence and computation of tolls in multicommodity networks. In joint work with Kamal Jain and Mohammad Mahdian, we have recently proved the existence of tolls for heterogeneous users in multicommodity networks.[6] In fact, we have given the first constructive proof of existence of tolls for not only the system optimal flow, but for any *minimal congestion*. This gives a complete characterization of flows enforceable by tolls. Our proof yields a simple algorithm for computing tolls via solving a linear program. This work extends to general nonatomic congestion games [7].

Acknowledgments

I am grateful to Kamal Jain, Mohammad Mahdian, and Tim Roughgarden for insightful discussions on this topic, and to an anonymous referee for comments that helped to improve the presentation of the paper.

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[2] M. Beckman, C.B. McGuire, C.B. Winsten, Studies in the Economics of Transportation, Yale University Press, 1956.
[3] K.C. Border, Fixed Point Theorems with Applications to Economics and Game Theory, Cambridge, 1985.
[4] P.J. Carstensen, Parametric cost shortest path problems, unpublished Bellcore memo, 1984.
[5] R. Cole, Y. Dodis, T. Roughgarden, Pricing network edges for heterogeneous selfish users, in: Proc. 35th Annu. ACM Symp. on the Theory of Computing, 2003, pp. 521–530.

[6] This first result is also obtained independently by Karakostas and Kolliopoulos [8], and Yang and Huang [14].

[6] S.C. Dafermos, Toll patterns for multiclass-user transportation networks, Transportation Sci. 7 (1973) 211–223.
[7] L. Fleischer, K. Jain, M. Mahdian, Tolls for heterogeneous selfish users in multicommodity networks and generalized congestion games, in: 45th Annu. IEEE Symp. on Foundations of Computer Science, 2004, pp. 277–285.
[8] G. Karakostas, S. Kolliopoulos, Edge pricing of multicommodity networks for heterogeneous users, in: 45th Annu. IEEE Symp. on Foundations of Computer Science, 2004.
[9] E. Koutsoupias, C. Papadimitriou, Worst-case equilibria, in: Proc. 16th Annu. Symp. on Theoretical Aspects of Computer Science (STACS), Lecture Notes in Computer Science, Vol. 1563, 1999, pp. 404–413.
[10] A.C. Pigou, The Economics of Welfare, Macmillan, New York, 1920.
[11] T. Roughgarden, E. Tardos, How bad is selfish routing?, J. ACM 49 (2) (2002) 236–259.
[12] M.J. Smith, The marginal cost taxation of a transportation network, Trans. Res. Ser. B 13 (1979) 237–242.
[13] J.G. Wardrop, Some theoretical aspects of road traffic research, in: Proc. Institute of Civil Engineers, Pt. II, Vol. 1, 1952, pp. 325–378.
[14] H. Yang, H.J. Huang, The multi-class, multi-criteria traffic network equilibria and system optimum problem, Transportation Res. B 28 (2004) 1–15.