Equilibria in Selfish Network Pricing when Paths share Resources and Users route Atomic Splittable Demand

Julia Buwaya and José D. P. Rolim

Computer Science Department, University of Geneva, Switzerland
{julia.buwaya | jose.rolim}@unige.ch

Abstract. We extend the knowledge on equilibrium computation in selfish resource pricing games in congested networks with novel results for atomic splittable flow. We consider specific types of networks where several paths may have an edge in common extending known results for parallel and parallel-series networks. We introduce restrictions that forbid monopolies in the pricing game and force resource owners to set their price to zero when they have zero flow, allowing positive results on equilibrium existence. We present an algorithm that provably converges to an equilibrium for affine latency functions. Our results are non-trivial as standard methods such as Kakutani's fixed point theorem fail to prove equilibrium existence.

1 Introduction

We study *selfish edge pricing* in *congested networks*, a situation that arises in real-world traffic problems when edges are owned by different profit-maximizing parties (as e.g. in the Internet or in multi-national highways). A *congested networks* is a network in which selfish users search for cheapest paths that (besides edge prices) involve load-dependent latencies on the edges (see e.g. [19] for an introduction). Our model is a multi-leader multi-follower *Stackelberg game* that is played in two stages (see e.g. [14] for an introduction to Stackelberg games). In the first stage, edges (leaders) simultaneously set prices per flow unit passing through them, anticipating the reaction of selfish network users (followers). In the second stage, prices are fixed and the network users simultaneously choose cost-minimal paths to route their demand. The users may only route fractional flow, i.e., their subgame corresponds to an atomic splittable routing games.

Games of this form have been studied for at least a decade now, but results for multi-commodity networks with atomic network users are still an open research field. This setting is non-trivial as standard methods such as Kakutani's fixed point theorem fail to prove equilibrium existence. We present an alternative approach using a constructive proof and show that an equilibrium of the pricing game can be computed in *monopoly-free* networks for a larger class network structures and latency functions with specific convexity properties. A network is monopoly-free if the network is *2-edge connected*, i.e., every user has at least two edge-disjunct paths to route its demand. Furthermore, we require:

- If for all $e \in E$, the latency functions l_e are nonnegative, nondecreasing, continuously differentiable, and normalized in the sense that $l_e(0) = 0$. Furthermore, $l'_e(x_e)x_e$ and $l_e(x_e)$ are convex in $x_e \geq 0$ (l'_e being the derivative of latency function l_e of edge $e \in E$). This is satisfied, e.g., by many linear or affine latency functions.
- If an edge e is contained in multiple paths of the network, then all edges that share potential users with e have at most one potential user.

Above conditions will be explained in further detail in the course of this article. We consider a model with inelastic demands and network users have no willingness to pay. Users are homogeneous, i.e., they all experience the same latencies on the edges (latency functions are the same for users). We compute a pure equilibrium, i.e., edges choose a non-negative price (not a probability distribution over prices) and users choose a set of paths to route their demand in fractions.

1.1 Related work

In the following we put our work into context with closely related topics and results. We generally differentiate between results for *congested networks* (latencies are sensitive to the amount of flow on the edges) and *classical flow networks* (latencies are not sensitive to the amount of flow), as well as *selfish pricing* (selfish players control prices) and *social pricing* (a central player controls prices aiming to minimize latencies in the network).

Selfish pricing in congested networks. [1], [2], and [15] study the existence and efficiency of equilibria in selfish edge pricing games in congested networks.

[1] and [2] used the difference between the requesters' willingness to pay and delay costs as efficiency measure, but our model does not support willingness to pay. [15] choose the difference between the edges' profit and the users' total costs from latencies and prices. The authors of [15] note that there are other reasonable measures as for example only considering the edges profit or only the users costs, but argue that the sum of both seems the most reasonable from an economic perspective in which "money is transferable". We follow their argumentation. Efficiency of an equilibrium is measured as the ratio of its social costs and the value of the social optimum of the game instance.

In contrast to our work, the authors of the latter articles consider single-commodity networks with a parallel-paths structure and non-atomic network users.

The authors of [1] investigate a situation where parallel paths are composed of more than one edge and providers own at most one edge in the whole network. Users are homogeneous and inelastic, i.e., all users experience the same latencies (latency functions are the same for each user) and demands are constants (users will not reduce the amount of flow routed through the network to reduce costs). Latency functions are nondecreasing and convex. For mixed strategies over prices, an equilibrium always exists. This is not always the case for pure equilibria. For linear latencies, they prove existence of a (pure) equilibrium applying Kakutani's fixed point theorem and they provide a characterization of prices in an equilibrium which facilitates their efficiency analysis of the game's equilibria. Users have a willingness to pay and they measure efficiency of an equilibrium as the difference between this willingness to pay and the users' delays. They show that efficiency loss relative to the social optimum can be arbitrarily large. For instances with normalized latency functions (zero costs on the edges if there is zero flow) the efficiency of *strong* equilibria, i.e., when every edge plays a strict best response and all traffic is transmitted (because the users' willingness to pay is sufficiently large), can be bounded.

The same authors study networks consisting only of parallel edges with homogeneous inelastic users in [2]. Here, several edges may be owned by the same player in the pricing game. Using a similar approach to the one in [1], they prove existence of an equilibrium and show that efficiency loss can be bounded for normalized linear nondecreasing latency functions.

In [15], networks of parallel paths, each composed of a single edge with linear latencies are studied for inelastic homogeneous demand yielding bounded efficiency of equilibria.

There exists furthermore results on equilibria in selfish pricing game in congested networks with atomic users. [7] study networks consisting of parallel edges with atomic users that control fractions of the total flow. Each user controls the same amount of flow. In contrast to our model of atomic splittable flow, users may arbitrarily split their flow over the paths, we restrict users to split their flow into integral parts over available paths only which complicates the equilibrium existence proof as it results in a best-response function that is not semi-upper continuous (suggesting the application of a fixed point theorem to prove existence). In [7], network users have a willingness to pay and the authors show that worst-case efficiency of equilibria can be bounded.

Unlike to the latter models, we study equilibria in multi-commodity networks with atomic network users. We introduce restrictions that forbid monopolies in the pricing game and force players to set their price to zero when they have zero flow, allowing positive results on equilibrium existence and computation.

Selfish pricing in classical flow networks. There exist various articles on selfish pricing in classical networks. These include [11] and [10], which study efficiency of equilibria in *Bertrand competitions* in networks with maximum capacities on its edges. Their model corresponds to a two-sided market where sellers own network edges and sell bandwidth at fixed prices while consumers buy bandwidth for sending their traffic through the network. Their model is similar to ours while it does not involve congestion in terms of load-dependent delays for the network users. They study efficiency of equilibria (quantifying the ratio of the performance of the equilibrium compared to the performance of an optimal solution) with respect to total costs for the consumers and total profit obtained by the sellers. For single-commodity networks, they give tight bounds of efficiency with respect to the number of monopolistic edges in the network. In multi-source single-sink networks, efficiency may only be bounded under additional assumptions on the network and demand structure.

The authors of [3] extend the work of [11] and [10] by considering convex production costs of the sellers. They characterize the loss in welfare of an outcome as function of the number of monopolies in the network. Besides other results, they show that for multi-commodity networks, if all buyers have uniform and *large* demand and production costs are strictly convex, then an efficient equilibrium exists. In our model, network user demand is fixed.

Also [8] study a network pricing game similar to ours. They consider a Stackelberg game with one *leader* setting prices on a subset of edges of a network. The rest of the edges have fixed costs. The leader is a profit maximizing player that anticipates the reaction of the followers - players that optimize some polynomial-time solvable combinatorial minimization problem based on the prices and costs in the network (e.g. finding a shortest path or minimum spanning tree). In contrast to our game, in this setting the reaction of the followers does not correspond to a subgame where network users play a selfish routing game. The choices of a follower do not depend on the choices of other followers playing simultaneously.

Social pricing in congested networks There are furthermore several works on edge pricing by a central authority with the goal to improve the efficiency of equilibrium flows (the total latency of the equilibrium compared to the optimal latency of a feasible network flow) in congested markets [12] and [18]. Other recent articles, [9] and [6] (for dynamic models), introduce network "taxes" in selfish routing games to optimize the efficiency of allocations. Taxes are set by a central authority.

[20] studies *Stackelberg strategies* and tolls in congested networks. An introduction to the concept of *Stackelberg strategies* with one central authority stipulating the routes of a portion of the flow while the rest of the demand is selfishly routed through the congested network can be found in [19]. In contrast to our problem, here a central authority controls a portion of the network flow (or sets the tolls for edge usage) with the goal to minimize the total latency of a Nash equilibrium. [20] provides effective results for both Stackelberg strategies and tolls in controlling the efficiency of equilibria. For non-atomic routing in general networks, they obtain latency-class specific bounds on the worst-case ratio of the system delay of an equilibrium and the optimal system delay (and they provide even tighter results for parallel and series-parallel networks). The results for general networks give a continuous trade-off between the fraction of flow controlled and the worst-case total latencies ratio between an equilibrium and the social optimum. For network tolls and atomic splittable routing games, they provide a convex program to generate tolls that induce an equilibrium flow that is optimal for general asymmetric games with heterogeneous network users (so in cases

where latency functions on the edges may vary for different users). Bounds on the efficiency of equilibria in congestion games for various Stackelberg strategies have recently been studied in [5] when latency functions are affine.

In this paper, instead of analyzing the setting and options of a central authority controlling flow or edge prices, we study a setting with several selfishly pricing competing individuals.

1.2 Contribution

Referring to the previous subsection on related work, we notice that selfish pricing problems in congested networks have been studied frequently when network users are non-atomic (i.e., when there exists an infinite number of users each controlling an infinitesimal amount of flow) and for networks consisting of parallel paths. We present results for the problem when users are atomic and the network structure is more complex. Our contribution in direct comparison to related results in selfish edge pricing games in congested networks is presented in Table 1. Note that certain results hold for a larger class of latency functions (see our discussion of related work in subsection 1.1). In particular, [2] and [1] present examples with general convex latencies where equilibria are not guaranteed and pure equilibria may be unbounded for parallel serial networks when users have a willingness to pay. Our model differs from those, and in particular, unlike to the latter models, we study equilibria in specific multi-commodity networks with atomic network users. We introduce restrictions that forbid monopolies in the pricing game and force players to set their price to zero when they have zero flow, allowing positive results on equilibrium existence and computation.

Table 1: Equilibria of selfish pricing games in congested networks with normalized linear latency functions.

network structure	user influence	user population	demand	equilibrium existence	willingness to pay
parallel edges	non-atomic	homogeneous	inelastic	✓ [2]	yes
parallel edges	non-atomic	homogeneous	elastic	✓ [15]	no
parallel paths	non-atomic	homogeneous	inelastic	✓ [1]	yes
parallel edges	atomic	homogeneous	inelastic	✓ [7]	yes
multiple paths share edges	atomic	homogeneous	inelastic	✓*	no

* contribution of this paper

Our results are non-trivial as the best-response function of the edges may not be upper semi-continuous when users are atomic and hence, a standard tool like Kakutani's fixed point theorem as, e.g., done in [1] and [2], cannot be applied to prove existence (more details on this are presented in Sec. 3). In this paper, we present an alternative approach using a constructive proof to show existence and to compute equilibria in selfish network pricing games paths share edges and users route atomic splittable flow.

2 Model

Our model is a *Stackelberg game* played on a directed acyclic network $G = (V, E, \mathbf{l} = (l_e)_{e\in E})$ with the set of nodes V, the set of edges E and nonnegative, nondecreasing, continuously differentiable latency functions $l_e : \mathbb{R}_+ \to \mathbb{R}_+$ for each edge $e \in E$. Furthermore, we assume that latency functions are normalized in the sense that for all $e \in E$ we have $l_e(0) = 0$. There exists a finite set N of network users. Network users are selfish players. Each user $i \in N$ has a demand $d_i \in \mathbb{Z}_+$ that needs to be routed from source s_i to sink t_i with $s_i, t_i \in V$. Furthermore, each edge $e \in E$ is a strategic player that sets a price $\pi_e \in \mathbb{R}_+$ per unit of flow passing through them. We denote by $\pi = (\pi_e)_{e\in E} \in \mathbb{R}_+^E$ the vector holding prices chosen by all edges in an outcome of the game.

A Stackelberg game is played in two stages (see e.g. [14] for an introduction on Stackelberg games). There are two types of player, namely *leaders* and *followers*. In the first stage, the leaders simultaneously choose their actions anticipating the reaction of the followers. In the second stage, followers choose their actions while leaders' actions are fixed. In our Stackelberg game, all edges together form the set of leaders in the Stackelberg game. They have full knowledge about the users' routing options and preferences and set and post prices in the first stage of the game anticipating users' reactions. Users are followers in the Stackelberg game that in the second stage route their demand along network paths that minimize their costs.

Atomic splittable flow. The set $\mathcal{P}^i$ denotes the set of all (s_i, t_i)-paths in G. The union of $\mathcal{P}^i$ over $i \in N$ forms the set of all paths $\mathcal{P}$. For the subgame played by the users, we consider an atomic splittable routing model (see e.g. [19] for an introduction to atomic routing games), i.e., when routing demand, a user i will split her demand d_i into integral parts over paths in $\mathcal{P}^i$. Such a combination of paths and allocated fractions of demand corresponds to a strategy of user i. A feasible outcome corresponds to a *feasible flow* $\mathbf{x} = (x_P)_{P\in\mathcal{P}}$ in the multi-commodity network G with

$$\sum_{P\in\mathcal{P}^i} x_P \le d_i \qquad \forall i \in N \;, \tag{1a}$$

$$x_P \in \mathbb{Z}_+ \qquad \forall P \in \mathcal{P} \;. \tag{1b}$$

Variable x_P denotes the amount of flow on path P. Equations (1a) and (1b) indicate that a feasible flow is non-negative, integral and does not exceed the total demand of a user. (Recall that d_i is also a nonnegative integer). Given a feasible flow $\mathbf{x}$, the load $x_e = \sum_{P\in\mathcal{P}\colon e\in P} x_P$ of edge $e \in E$ corresponds to the total flow running through e.

Users' costs. The load-dependent costs from latencies for a user i equal $\sum_{P\in\mathcal{P}^i} \left(\sum_{e\in P} l_e(x_e)\right) x_P$, i.e., the user's allocated demand multiplied with according load-dependent costs of the chosen paths. In addition, a user pays fixed edge prices for routing demand along the edges of the paths, i.e., $\sum_{P\in\mathcal{P}^i} \left(\sum_{e\in P} \pi_e\right) x_P$. So the total costs of a user i are

$$\sum_{P\in\mathcal{P}^i} \left(\sum_{e\in P} (l_e(x_e) + \pi_e)\right) x_P \;, \tag{2}$$

Edges' profits. The profit of an edge $e \in E$ corresponds to the total flow x_e on the edge multiplied with its fixed unite price π_e, i.e.,

$$\pi_e \, x_e \;. \tag{3}$$

Equilibria. In the fist stage of our game edges simultaneously choose prices to maximize their profits. In the second stage, prices are fixed and the users simultaneously route their demand selfishly minimizing theirs costs. The users' subgame corresponds to an *atomic splittable routing game*. An equilibrium of the subgame played by the users, given fixed prices, is an outcome where no user can reduce costs by deviating from her chosen strategy, given the other users' strategies are fixed, more formally:

Definition 1 (Users' subgame equilibrium). *Let $l'(x_e)$ denote the derivative of $l(\cdot)$ at x_e and let $x_{e,i} = \sum_{P\in\mathcal{P}^i : e\in P} x_P$ denote the amount of flow of $\mathbf{x}$ routed by user i on edge e. Given fixed price vector $\pi \in \mathbb{R}_+^E$, a feasible flow $\mathbf{x}$ is a users' subgame equilibrium, if for all $i \in N$ and for all $P, P' \in \mathcal{P}^i$ with $x_P > 0$, we have*

$$\sum_{e\in P}(l_e(x_e)+l'_e(x_e)x_{e,i}+\pi_e) \quad\le\quad \sum_{e\in P'}(l_e(x_e)+l'_e(x_e)x_{e,i}+\pi_e)\,. \tag{4}$$

We denote the set of all user's subgame equilibria given π by NE(π).

The subgame played by the users corresponds to an atomic splittable routing game and the given equilibrium definitions can for example be found in [19] Equation (4) is well-defined as we consider continuously differentiable latency functions. It indicates that in an equilibrium the costs of a user cannot be decreased by shifting flow from one path to another path. As latency functions are nondecreasing and the edge prices are fixed, existence of a users' Nash equilibrium given π is guaranteed (see [19]).

An equilibrium describing a stable outcome of both stages of the game is a *Stackelberg equilibrium* which we define as follows:

Definition 2 (Stackelberg equilibrium). *Given price vector π and a constant $\pi'_e \in \mathbb{R}_+$, let $[\pi_{E\smallsetminus e}, \pi'_e]$ denote a price vector whose entry at position e equals π'_e while the rest of the entries equal the ones of vector π. A vector $(\pi, \mathbf{x})$ is a Stackelberg equilibrium (SE) if $\mathbf{x} \in NE(\pi)$ and for all edges $e \in E$, for all $\pi'_e \in \mathbb{R}_+$, and $\mathbf{x}' \in NE([\pi_{E\smallsetminus e}, \pi'_e])$, we have*

$$\pi_e\, x_e \;\ge\; \pi'_e\, x'_e\,. \tag{5}$$

3 Existence and computation of a Stackelberg equilibrium

Proving existence of an SE of our game is a nontrivial task due to the fact that user responses to prices may not be unique and due to the absence of properties of the best-response function, i.e., the best-response function may not be upper semicontinuous and hence, a standard tool like Kakutani's fixed point theorem cannot be applied to prove existence. A main challenge in computing an SE, is that there could be multiple subgame responses to a set of fixed edge prices as equilibria of the subgame played by the users, an atomic splittable routing game, are not unique (see e.g. [4]). This makes the edges' profit functions not well-defined and hence also the best-response function of the game not well-defined.

In this section, we present a set of conditions that guarantee the existence of an SE and we provide an algorithm to compute it. First conditions are gathered in the following Assumption 1.

Assumption 1 *For the considered game instances the following holds:*

1. *The user network is monopoly-free. A network has a monopoly when there exists an edge and a user that is forced to use this edge due to the lack of alternative routes in the network, otherwise the network is monopoly-free.*
2. *For all $e \in E$, the latency functions l_e are nonnegative, nondecreasing, continuously differentiable, and normalized in the sense that $l_e(0) = 0$. Furthermore, $l'_e(x_e)x_e$ and $l_e(x_e)$ are convex in $x_e \ge 0$ (l'_e being the derivative of latency function l_e of edge $e \in E$).*

We note that 2. of Assumption 1 is satisfied for example for many linear or affine latency functions.

Our approach to generate an equilibrium is as follows: We modify the solution of an atomic routing game without additional prices on the edges such that the result will be the solution of our Stackelberg pricing game. We do this in a way such that we can compute an SE by successively solving a finite series of convex programs. For an intuition on the problem to find an SE in our Stackelberg pricing game and on our solution approach, consider the following two examples.

Example 1. Consider a *parallel-series network*, i.e., all paths are parallel and consist either of one edge or a series of multiple edges. We assume that there is a single user with a demand of 1. Let the latency for one unit of flow on path (e_4) be lower than the total latency on path (e_1, e_2, e_3), e.g. when $l_{e_1}(x) = l_{e_2}(x) = l_{e_3}(x) = l_{e_4}(x) = x$. A user's subgame equilibrium when all prices are zero is shown in Fig. 1, i.e., the user chooses to route demand along edge e_4 resulting in lower costs.

Now with profit-maximizing edges, the edge e_4 will decide to set its price so that the costs from the price and latency on e_4 are just below the costs from latencies on (e_1, e_2, e_3). So in this example, $\pi_e = 3 - \epsilon$ with ϵ being a small positive constant. The other edges have no incentive to set any prices and the user will route its demand along edge e_4 minimizing costs. This solution corresponds to an epsilon-approximation of an SE. In an SE, $\pi_e = 3$ as is shown in Fig. 2. (We note that the user's response to these prices is not unique and we comment on this after Example 2).

The other way around, if the total latency on (e_1, e_2, e_3) is lower than the latency on e_4, in an SE, the edges e_1, e_2, and e_3 together can set prices producing total user costs just a small epsilon below the costs from the latency for one flow unit on e_4. Edge e_4 cannot make any profit and would be indifferent regarding its price; in such a case of "indifference", let edges set their price to zero. Any combination of e_4 having price zero and a distribution of the total price of path (e_1, e_2, e_3) among the edges e_1, e_2, e_3 (with epsilon equal to zero) corresponds to an SE in this case.

In this example, if both paths have the same total latency for one unit of flow, then all edges set their prices to zero in an SE, not making any profit. □

We observe that for the instance of Example 1 and the described scenarios regarding latencies, we can construct an SE by modifying the users' subgame equilibrium when all prices are set to zero by means of increasing the prices of the edges if profitable.

Example 2. We modify the network of Example 1 by adding another path (e_5) as presented in Fig. 3. We furthermore assume there are two network users instead of one, each with a demand of 1. User 1 is routing demand from s_1 to t_1 and user 2 is routing demand from s_2 to t_2. Let the latency functions be given by $l_{e_1}(x) = l_{e_2}(x) = l_{e_3}(x) = 0$, $l_{e_4}(x) = x$, and $l_{e_5} = 3x$. We will construct an SE by starting off again with a users' subgame equilibrium when all prices are zero. With zero prices, the demand of user 1 is routed on path (e_1, e_2, e_3) and the demand of user 2 on path (e_2) as shown in Fig. 3. We will increase prices until an SE is reached. Obviously, starting from this setting, e_4 has no incentive to rise its price as this would not increase its profit. The edges e_1, e_2, and e_3 would all independently have an incentive to increase their prices until the total of their prices is just under 1. This would guarantee that user 1 continues to use their path and making profit from it (the maximum profit per edge from user 1 would be 1). Now on the other side e_2 also makes profit from the demand routed through it by user 2. When e_2 increases its price above 1 it definitely looses demand from user 1, but can still increase its overall profit through user 2 by setting its price just under 3. Once demand from user 1 is shifted to edge e_4, this edge also has an incentive to increase its price. The costs from the price and latency on e_4 has to be smaller for user 1 than routing demand along (e_1, e_2, e_3) with zero latencies and a price just under 3 on edge e_2. We reach an approximate SE, where flow is routed on (e_4) and (e_2) with edge e_2 choosing price $\pi_2 = 3 - \epsilon_2$ and e_4 choosing $\pi_4 = 2 - \epsilon_2 - \epsilon_4$ $(\epsilon_2, \epsilon_4 > 0)$ while the other edges have zero prices. In an SE, $\pi_2 = 3$ and $\pi_4 = 2$. □

We note that in the presented SE of above examples, the users' subgame equilibrium responses are not unique. In Example 2, user 2 could likewise decide to route flow on edge e_5 and user 1 could decide to route its demand on path (e_1, e_2, e_3) instead of e_4. To avoid this, both e_4 and then e_2 can decide to lower their prices by according epsilons. This would be a solution chosen by *risk averse*

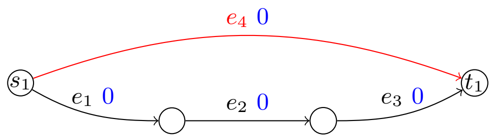

Fig. 1: Subgame equilibrium with zero prices from Ex. 1.

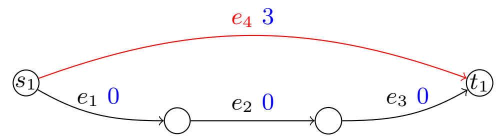

Fig. 2: An SE from Ex. 1.

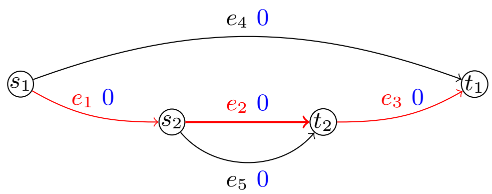

Fig. 3: Subgame equilibrium with zero prices from Ex. 2

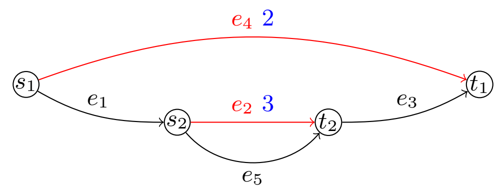

Fig. 4: An SE from Ex. 2.

Fig. 5: Solutions of instances of Ex. 1 and Ex. 2. The prices of the presented solutions are indicated in blue and the paths chosen in the users subgame equilibrium are indicated in red.

edges that play strategies with maximum guaranteed profit even if this profit may be less than the profit it could potentially gain in an SE. Similar arguments can be made for Example 1.

In this article, we will focus on the existence, computation, and efficiency of an SE, leaving further analysis of SE with guaranteed profit for future work.

3.1 Algorithm for Equilibrium Computation

As discussed above, a main challenge in proving existence and in the computation of an SE, is that there could be multiple subgame responses to a set of fixed edge prices which complicates the application of standard tools involving fixed point theorems. To work around this problem, we start off with one particular solution for the users' subgame equilibrium when prices are zero and then modify this solution until we reach an SE. Our approach includes elements of a *best-response dynamic* (see e.g. [17]): We update edge prices over several grounds avoiding cycling. Examples 1 and 2 provide the intuition for our approach; Starting with zero prices, in the beginning, edges cannot decrease their prices, but may only increase them to maximize their profit. We set prices sequentially by solving convex programs converging to an SE. Before we present the pseudo code of our algorithm, we explain the procedure:

For a given game instance, recall that $NE(\pi)$ denotes the set of users' subgame equilibria when prices are fixed and given through π. As latency functions are nondecreasing, existence of a users' subgame equilibrium is guaranteed (see e.g. [19]). A users' subgame equilibrium can be computed via a best-response dynamic (see e.g. [17]). An ϵ-approximation of such an equilibrium can be computed in polynomial time for all $\epsilon > 0$, given that the cost functions on the resources are affine [13]. Under certain symmetry conditions on the network structure, [16] furthermore presents an exact polynomial time algorithm to compute an equilibrium in atomic splittable routing games with affine cost functions.

Starting off with some subgame equilibrium $\mathbf{x} \in NE(\mathbf{0})$ at zero prices $\pi = \mathbf{0}$, we generate the set $\Omega = \{e : x_e = 0\}$ of *edges that have no incentive to change their current price*. We choose some $e \in E \setminus \Omega$ at random and set $\Omega = \Omega \cup e$. First, we check whether we can increase the price and hence, the profit of e without shifting flow from edge e. Then, we check if the edge has an incentive to increase its price even higher, taking in consideration that flow may be shifted, i.e., we check whether if flow is taken from x_e, the profit of e can be increased. We generate the new optimal price π_e^* and an users' subgame equilibrium response by solving the following program (6a)-(6c) for

$x_e - \gamma$ with $\gamma \in [0, x_e] \cap \mathbb{Z}$. We start with with $\gamma = 0$ ($\pi = \mathbf{0}$):

$$\max_{\substack{\pi_e^* \in \mathbb{R}_+ \\ \mathbf{x}^* \in \mathbb{Z}_+^E}} \quad \pi_e^*(x_e - \gamma) \tag{6a}$$

$$\mathbf{x}^* \in NE([\pi_{E\setminus e}, \pi_e^*]) \tag{6b}$$

$$x_e - \gamma = x_e^* \tag{6c}$$

We are computing price π_e^* that maximizes the profit of edge e (6a) while assuring that $\mathbf{x}^*$ is a feasible users' response to the new price (6b). Recall our notation that $[\pi_{E\setminus e}, \pi_e^*]$ denotes the price vector whose entry at position e equals π_e^* while the rest of the entries equal the ones of vector π. Note that the flow $\mathbf{x}$ (and in particular x_e) is an input to this program and that we control the amount of flow that is shifted from edge e to other edges through parameter γ (6c). Problem (6a)-(6c) is equivalent to

$$\max_{\substack{\pi_e^* \in \mathbb{R}_+ \\ \mathbf{x}^* \in \mathbb{Z}_+^E}} \quad \pi_e^* x_e - \gamma \tag{7a}$$

$$\begin{aligned}\sum_{e\in P} \big(l_e(x_e^*) + l'_e(x_e^*)x_{e,i}^*\big) + \sum_{e\in P\setminus e} \pi_e + \sum_{e \in P \cap e} \pi_e^* \leq \\ \sum_{e\in P'} \big(l_e(x_e^*) + l'_e(x_e^*)x_{e,i}^*\big) + \sum_{e\in P'\setminus e} \pi_e + \sum_{e \in P' \cap e} \pi_e^* \\ \forall P, P' \in \mathcal{P}^i \ , \ \forall i \in N\end{aligned} \tag{7b}$$

$$\sum_{P \in \mathcal{P}^i} x_P^* \leq d_i \qquad \forall i \in N \tag{7c}$$

$$x_e - \gamma = x_e^* \tag{7d}$$

Equations (7b) and (7c) replace condition (6b) as defined in Equation (4) in Sec. 2 (here, the condition $x_P > 0$ in (7b) can be dropped as latency functions are normalized in the sense that for all $e \in E$ we have $l_e(0) = 0$). We fix x_e to avoid a quadratic term in the objective function. Note that problem is convex if $l'_e(x_e)x_e$ and $l_e(x_e)$ is convex. The problem has a linear objective function over a compact set with convex constraint.

We solve the same program for $\gamma = 1$, $\gamma = 2$, etc. until $\gamma = x_e$, solving the problem $x_e \in \mathbb{Z}$ times. Out of all the problems, we choose the solution with the highest value for π_e (if there are multiple, we choose the one with lowest γ). Note that for certain $\gamma \in [0, x_e]$ the program could be infeasible, which is fine as at least one of the programs can be solved, i.e., for $\gamma = 0$. Then, we update the optimal price π_e^* for e in the price vector such that $\pi = [\pi_{E\setminus e}, \pi_e^*]$.

We want to make sure that the newly generated equilibrium flow affects the lowest possible number of other edges (by changing their flow). To assures that the change from $\mathbf{x}$ to a new $\mathbf{x}^* \in NE([\pi_{E\setminus e}, \pi_e^*])$ is minimal (in particular if there exists several feasible equilibrium user responses), as new flow, we use the solution of the following program:

$$\min_{\mathbf{x}^* \in NE([\pi_{E\setminus e}, \pi_e^*])} \quad \sum_{e' \in E} ind_{e'} \tag{8a}$$

$$x_e - \gamma = x_e^* \tag{8b}$$

$$\frac{|x_{e'} - x_{e'}^*|}{|x_{e'} - x_{e'}^*| + 1} \leq ind_{e'} \qquad \forall e' \in E \tag{8c}$$

$$\mathbf{x}^* \in \{0, 1\} \tag{8d}$$

Constraint (8c) can be reformulated into two linear constraints: $\frac{x_{e'}-x^*_{e'}}{x_{e'}-x^*_{e'}+1} \leq ind_{e'}$ and $\frac{x^*_{e'}-x_{e'}}{x^*_{e'}-x_{e'}+1} \leq ind_{e'}$ for all $e' \in E$.

Afterwards, we update Ω, the *set of edges that have no incentive to change their price* and choose the next random $e \in E \setminus \Omega$ and continue the procedure with updated solution $(\mathbf{x}, \pi) = (\mathbf{x}^*, [\pi_{E\setminus e}, \pi^*_e])$ until $\Omega = E$. When updating Ω in each round, we have to assure that at some point Ω actually includes all edges of E. Edges $e' \neq e$ may leave the set Ω when the amount of flow going through e' was changed due to the change of price on edge e. A basis for reducing the number of edges that leave Ω is that we minimize the change of flow in each round (by solving program (8a)-(8d)). We make sure that we construct a solution by shifting flow only due to changes on one edge at a time to avoid cycling. In addition, the following property of the considered network assures that our procedure converges.

Assumption 2 *If an edge e is contained in multiple paths of the network, then all edges that share potential users with e have at most one potential user.*

A pseudo code of the SE computation is given in Algorithm 1. Note that the SE that we are creating may not be unique. Other SE may exist, in particular, all edges with zero prices in an SE could choose any price not changing the costs or profit of any other player.

Algorithm 1 SE Computation

Input: *Game Instance*
Output: $SE = (\pi, \mathbf{x})$
Initialize: $\mathbf{x} \in NE(\mathbf{0})$, $\Omega = \{e : x_e = 0\}$, $\pi = \mathbf{0}$
while $\Omega \neq E$ **do**
  Choose $e \in E \setminus \Omega$ at random
  Update $\Omega = \Omega \cup \{e\}$
  for $\gamma \in [x_e, x_e - 1, x_e - 2, \dots, 0]$ **do**
    if (7a)-(7d) feasible **then**
      Update $(\pi_e, \mathbf{x}) = \arg\max(\arg\max((7a)\text{-}(7d))\,;\, \arg\max(\pi_e x_e))$ ▷ set π_e generating highest profit
    end if
  end for
  Update $\mathbf{x} \in NE(\pi)$ as solution from (8a)-(8d) ▷ generate $\mathbf{x}$ with smallest shift
  for edges $e' \neq e$ where flow has shifted **do**
    Set $\pi_{e'} = 0$
    Remove e' from Ω
  end for
end while

Theorem 1 (Computation and existence of an SE). *Under the conditions stated in Assumption 1 and Assumption 2, Algorithm 1 converges to an SE.*

For the proof, we use the following lemma.

Lemma 1. *Algorithm 1 terminates.*

Proof. To prove that Algorithm 1 terminates, we have to assure that at some point $\Omega = E$. In the while-loop of the algorithm, in every loop an edge is added to set Ω. Edges are only potentially removed in the case where $\gamma \neq 0$, corresponding to a case where edge e has set its price such that flow was shifted. Assumption 1 (2.) guarantees that the subroutine programs (7a)-(7d) and (8a)-(8d) are solvable in finite time. As we consider monopoly-free networks (Assumption 1 (1.)), an edge will never unilaterally (without coordination with others) rise its price towards infinity in order to maximize profit.

Let e' be an edge that has not set its price in the current loop, but its flow has changed. Either e' is not yet in Ω (this will not decrease the size of Ω) or e' is in Ω and we have to remove it. Assumption 2 implies that we can have either of the following cases when e' has to be removed:

Case 1: Current e is contained in multiple paths $\Rightarrow$ any affected e' has only one potential user. Here, we have the following possibilities:

1. e' has gained flow: e' can only make profit from flow by a single user. If e' is chosen in the coming round, it will try to keep as much flow as possible. Obviously, it cannot increase its flow, but only keep it while either increasing its price or leaving it as it is in the coming round. This will not affect the flow on any other edge.
2. e' has lost flow (because also e lost flow setting its price so high that some flow was removed).
   (a) e' may leave its price as it is without any affect on any other edge.
   (b) e' has never set its price before (its price equals zero) and still has positive flow after it has lost flow: In this case, if chosen in the coming round, e' will increase its price with potential affects on the flow of other edges. But this can only be the case one time for e'.
   (c) e' may want to reduce its current price to regain some flow again:
      – Effect on edge e with multiple potential users (sharing one user with e'): This may increase the costs from latencies for the users on e triggering e to lower its price in a later round (to profit from users that are not the potential user of e'). This again could imply increased costs for the user on e' and potentially loosing flow bringing it back to Case 1, 2. At some point, this will reach a point where e' will not be able to reduce its price profitably any further.
      – Effect on other edges than e with multiple potential users (sharing one user with e'): Either other affected edges also gain flow ending in Case 1, 1. or they loose flow bringing us back to Case 1. 2. until it is not profitable anymore to lower the price.

Case 2: Current e has only one potential user. In this case, edge e aims at facilitating as much of the demand of its potential user as possible. As e has only one potential user, the possible outcomes after e has optimized its price are:

1. e has increased its price or left its price as it is without affecting the flow and any other edge.
2. e has decreased its price to regain flow, bringing us to Case 1, 2. (c) with e' replaced by e.

$\square$

Proof (Proof of Thm 1). Algorithm 1 terminates if there exists no edge that can increase its profit. The response to the computed prices is a users' subgame equilibrium which is guaranteed by the constraints in problem (6a)-(7c), resp. (7a)-(7d). The solution corresponds hence to an SE by definition. $\square$

Our Algorithm 1 provable produces an SE for every game instance given Assumption 1 and Assumption 2, we have hence proven existence of an equilibrium under those conditions and furthermore means to compute an SE.

4 Conclusion

In this article, we presented positive results on the existence and computation of equilibria for a specific class of selfish network pricing games when users route atomic splittable flow. We consider networks where several paths may have an edge in common, extending known results for parallel and parallel-series networks. An algorithm that provably converges to an equilibrium when the latency functions have certain convexity properties is presented. With these results, we closed a research gap and extended known results for this problem for non-atomic network users with insides on atomic splittable routing.

Acknowledgment

The authors thank Daniel Schmand and Lutz Welpelo for the numerous valuable discussions on the topic. This work was partially supported by the *Swiss National Science Foundation* (SNSF) - CRSII2-154458 *SwissSenseSynergy* project and an University of Geneva Digital Fellows 2017 grant.

References

1. Daron Acemoglu and Asuman Ozdaglar. Price competition in communication networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2006.
2. Daron Acemoglu and Asuman Ozdaglar. Competition and efficiency in congested markets. *Mathematics of Operations Research*, 32(1):1–31, 2007.
3. Elliot Anshelevich and Shreyas Sekar. Price competition in networked markets: How do monopolies impact social welfare? In *International Conference on Web and Internet Economics (WINE)*, 2015.
4. Umang Bhaskar, Lisa Fleischer, Darrell Hoy, and Chien-Chung Huang. Equilibria of atomic flow games are not unique. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 748–757. SIAM, 2009.
5. Vittorio Bilò and Cosimo Vinci. On stackelberg strategies in affine congestion games. In *International Conference on Web and Internet Economics*, pages 132–145. Springer, 2015.
6. Vittorio Bilò and Cosimo Vinci. Dynamic taxes for polynomial congestion games. In *ACM Conference on Economics and Computation (EC)*, 2016.
7. Kostas Bimpikis and Asuman Ozdaglar. Competition with atomic users. In *Signals, Systems and Computers, 2007. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on*, pages 1445–1449. IEEE, 2007.
8. Patrick Briest, Martin Hoefer, and Piotr Krysta. Stackelberg network pricing games. *Algorithmica*, 62(3):733–753, 2012.
9. Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Improving the efficiency of load balancing games through taxes. In *International Workshop on Internet and Network Economics (WINE)*, 2008.
10. Shuchi Chawla and Feng Niu. The price of anarchy in bertrand games. In *ACM Conference on Electronic Commerce (EC)*, 2009.
11. Shuchi Chawla and Tim Roughgarden. Bertrand competition in networks. In *International Symposium on Algorithmic Game Theory (SAGT)*, 2008.
12. Richard Cole, Yevgeniy Dodis, and Tim Roughgarden. Pricing network edges for heterogeneous selfish users. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 521–530. ACM, 2003.
13. Roberto Cominetti, José R. Correa, and Nicolás E. Stier-Moses. The impact of oligopolistic competition in networks. *Operations Research*, 57(6):1421–1437, 2009.
14. Drew Fudenberg and Jean Tirole. *Game Theory.* MIT Press, 1991.
15. Ara Hayrapetyan, Éva Tardos, and Tom Wexler. A network pricing game for selfish traffic. *Distributed Computing*, 19(4):255–266, 2007.
16. Chien-Chung Huang. Collusion in atomic splittable routing games. *Theory of Computing Systems*, 52(4):763–801, 2013.
17. Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani (Eds.). *Algorithmic Game Theory.* Cambridge University Press, 2007.
18. Tim Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.
19. Tim Roughgarden. *Selfish routing and the price of anarchy.* MIT press Cambridge, 2005.
20. Chaitanya Swamy. The effectiveness of stackelberg strategies and tolls for network congestion games. *ACM Transactions on Algorithms (TALG)*, 8(4):36, 2012.