THE PRICE OF STABILITY FOR NETWORK DESIGN WITH FAIR COST ALLOCATION*

ELLIOT ANSHELEVICH†, ANIRBAN DASGUPTA‡, JON KLEINBERG §, ÉVA TARDOS¶, TOM WEXLER‖, AND TIM ROUGHGARDEN**

Abstract. Network design is a fundamental problem for which it is important to understand the effects of strategic behavior. Given a collection of self-interested agents who want to form a network connecting certain endpoints, the set of *stable* solutions — the Nash equilibria — may look quite different from the centrally enforced optimum. We study the quality of the *best* Nash equilibrium, and refer to the ratio of its cost to the optimum network cost as the *price of stability.* The best Nash equilibrium solution has a natural meaning of stability in this context — it is the optimal solution that can be proposed from which no user will defect.

We consider the price of stability for network design with respect to one of the most widely-studied protocols for network cost allocation, in which the cost of each edge is divided equally between users whose connections make use of it; this fair-division scheme can be derived from the Shapley value, and has a number of basic economic motivations. We show that the price of stability for network design with respect to this fair cost allocation is $O(\log k)$, where k is the number of users, and that a good Nash equilibrium can be achieved via *best-response dynamics* in which users iteratively defect from a starting solution. This establishes that the fair cost allocation protocol is in fact a useful mechanism for inducing strategic behavior to form near-optimal equilibria. We discuss connections to the class of *potential games* defined by Monderer and Shapley, and extend our results to cases in which users are seeking to balance network design costs with latencies in the constructed network, with stronger results when the network has only delays and no construction costs. We also present bounds on the convergence time of best-response dynamics, and discuss extensions to a weighted game.

Key words. network design, price of stability, Shapley cost-sharing

AMS subject classifications. 68Q99, 90B18, 91A43

1. Introduction.

In many network settings, the system behavior arises from the actions of a large number of independent agents, each motivated by self-interest and optimizing an individual objective function. As a result, the global performance of the system may not be as good as in a case where a central authority can simply dictate a solution; rather, we need to understand the quality of solutions that are consistent with self-interested behavior. Recent theoretical work has framed this type of question in the following general form: how much worse is the solution quality of a Nash

*A preliminary version of this paper appeared in *Proc. 45th Annual Symposium on Foundations of Computer Science, 2004.*

†Rensselaer Polytechnic Institute, Department of Computer Science, Troy, NY. Research supported by ITR grant 0311333. Email: eanshel@cs.rpi.edu.

‡Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853. Supported by the Department of Computer Science. Email: adg@cs.cornell.edu.

§Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853. Email: kleinber@cs.cornell.edu. Supported in part by a David and Lucile Packard Foundation Fellowship and NSF grants 0081334 and 0311333.

¶Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853. Supported in part by NSF grant CCR-032553, ITR grant 0311333, and ONR grant N00014-98-1-0589. Email: eva@cs.cornell.edu.

‖Cornell University, Department of Computer Science, Upson Hall, Ithaca, NY 14853. Supported by ITR grant 0311333. Email: wexler@cs.cornell.edu.

**Stanford University, Computer Science Department, Gates Building, Stanford, CA 94305. Supported in part by ONR grant N00014-04-1-0725, an NSF CAREER Award, and an Alfred P. Sloan Fellowship. Email: tim@cs.stanford.edu.

equilibrium[1], compared to the quality of a centrally enforced optimum? Questions of this genre have received considerable attention in recent years, for problems including routing [37, 39, 13], load balancing [14, 15, 27, 38], and facility location [41]; see [34, Chapters 17–21] for an overview of this literature.

An important issue to explore in this area is the middle ground between centrally enforced solutions and completely unregulated anarchy. In most networking applications, it is not the case that agents are completely unrestricted; rather, they interact with an underlying protocol that essentially proposes a collective solution to all participants, each of which can either accept it or defect from it. As a result, it is in the interest of the protocol designer to seek the *best* Nash equilibrium; this can naturally be viewed as the optimum subject to the constraint that the solution is *stable*, with no agent having an incentive to unilaterally defect from it once it is offered. Hence, one can view the ratio of the solution quality at the best Nash equilibrium relative to the global optimum as a *price of stability*, since it captures the problem of optimization subject to this constraint. Some recent work [3, 13] has considered this definition (termed the "optimistic price of anarchy" in [3]); it stands in contrast to the larger line of work in algorithmic game theory on the *price of anarchy* [35] — the ratio of the *worst* Nash equilibrium to the optimum — which is more suited to worst-case analysis of situations with essentially no protocol mediating interactions among the agents. Indeed, one can view the activity of a protocol designer seeking a good Nash equilibrium as being aligned with the general goals of mechanism design [33] — producing a game that yields good outcomes when players act in their own self-interest.

Network Design Games. Network design is a natural area in which to explore the price of stability, given the large body of work in the networking literature on methods for sharing the cost of a designed network — often a virtual overlay, multicast tree, or other sub-network of the Internet — among a collection of participants. (See e.g. [19, 22] for overviews of work in this area).

A cost-sharing mechanism can be viewed as the underlying protocol that determines how much a network serving several participants will cost to each of them. Specifically, say that each user i has a pair of nodes (s_i, t_i) that it wishes to connect; it chooses an s_i-t_i path S_i; and the cost-sharing mechanism then charges user i a cost of $C_i(S_1, \dots, S_k)$. (Note that this cost can depend on the choices of the other users as well.) Although there are in principle many possible cost-sharing mechanisms, research in this area has converged on a few mechanisms with good theoretical and empirical behavior; here we focus on the following particularly natural one: the cost of each edge is shared equally by the set of all users whose paths contain it, so that

$$C_i(S_1, S_2, \dots, S_k) = \sum_{e \in S_i} \frac{c_e}{|\{j : e \in S_j\}|}.$$

This equal-division mechanism has a number of basic economic motivations; it can be derived from the Shapley value [32], and it can be shown to be the unique cost-sharing scheme satisfying a number of different sets of axioms [19, 22, 32]. For the former reason, we will refer to it as the *Shapley cost-sharing mechanism.* Note that the total edge cost of the designed network is equal to the sum of the costs in the union of all S_i, and the costs allocated to users in the Shapley mechanism completely pay for this total edge cost: $\sum_{i=1}^{k} C_i(S_1, S_2, \dots, S_k) = \sum_{e \in \cup_i S_i} c_e$.

[1]Recall that a Nash equilibrium is a state of the system in which no agent has an interest in unilaterally changing its own behavior.

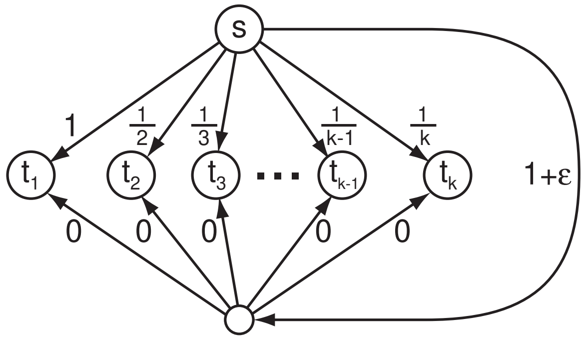

FIG. 1.1. *An instance in which the price of stability converges to $H(k) = \Theta(\log k)$ as $\varepsilon \to 0$.*

Now, the general question is to determine how this basic cost-sharing mechanism serves to influence the strategic behavior of the users, and what effect this has on the structure and overall cost of the network one obtains. Given a solution to the network design problem consisting of a vector of paths $(S_1, \ldots, S_k)$ for the k users, user i would be interested in deviating from this solution if there were an alternate s_i-t_i path S_i' such that changing to S_i' would lower its cost under the resulting allocation: $C_i(S_1, \ldots, S_{i-1}, S_i', S_{i+1}, \ldots, S_k) < C_i(S_1, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_k)$. We say that a set of paths is a *Nash equilibrium* if no user has an interest in deviating. As we will see below, there exists a set of paths in Nash equilibrium for every instance of this network design game. (In this paper, we will only be concerned with *pure* Nash equilibria; i.e., with equilibria where each user deterministically chooses a single path.)

The goal of a network design protocol is to suggest for each user i a path S_i so that the resulting set of paths is in Nash equilibrium and its total cost exceeds that of an optimal set of paths by as small factor as possible; this factor is the *price of stability* of the instance. It is useful at this point to consider a simple example that illustrates how the price of stability can grow to a super-constant value (with k). Suppose k players wish to connect from the common source s to their respective terminals t_i, and assume player i has its own path of cost $1/i$, and all players can share a common path of cost $1 + \varepsilon$ for some small $\varepsilon > 0$ (see Figure 1.1). The optimal solution would connect all agents through the common path for a total cost of $1 + \varepsilon$. However, if this solution were offered to the users, they would defect from it one by one to their alternative paths. The unique Nash equilibrium has a cost of $\sum_{i=1}^{k} 1/i = H(k)$.

While the price of stability in this instance grows with k, it only does so logarithmically. It is thus natural to ask how large the price of stability can be for this network design problem. If we think about the example in Figure 1.1 further, it is also interesting to note that a Nash equilibrium is reached by players taking turns updating their paths (in other words, best-response dynamics) starting from an optimal solution; it is natural to ask to what extent this holds in general.

Our Results. Our first main result is that in every instance of the network design problem with Shapley cost-sharing, there always exists a Nash equilibrium of total cost at most $H(k)$ times optimal. In other words, the simple example in Figure 1.1 is in fact the worst possible case.

We prove this result using a *potential function* method due to Rosenthal [36] (based on [6]) and later generalized by Monderer and Shapley [30]: one defines a potential function Φ on possible solutions and shows that every improving move of one of the users (to lower its own cost) reduces the value of Φ. Since the set of possible solutions is finite, it follows that every sequence of improving moves leads to a Nash equilibrium. The goal of Monderer's and Shapley's and Rosenthal's work was to prove existence statements of this sort; for our purposes, we make further use of the potential function to prove a bound on the price of stability. Specifically, we give bounds relating the value of the potential for a given solution to the overall cost of that solution; if we then iterate using best-response dynamics starting from an optimal solution, the potential does not increase, and hence we can bound the cost of any solution that we reach. Thus, for this network design game, best-response dynamics starting from the optimum do in fact always lead to a good Nash equilibrium.

We can extend our basic result to a number of more general settings. To begin with, the $H(k)$ bound on the price of stability extends directly to the case in which users are selecting arbitrary subsets of a ground set (with elements' costs shared according to the Shapley value), rather than paths in a graph; it also extends to the case in which the cost of each edge is a non-decreasing concave function of the number of users on it. In addition, our results also hold if we introduce capacities into our model; each edge e may be used by at most u_e players, where u_e is the capacity of e.

We arrive at a more technically involved set of extensions if we wish to add latencies to the network design problem. Here each edge has a concave *construction cost* $c_e(x)$ when there are x users on the edge, and a *latency cost* $d_e(x)$; the cost experienced by a user is the full latency plus a fair share of the construction cost, $d_e(x) + c_e(x)/x$. We give general conditions on the latency functions that allow us to bound the price of stability in this case by $d \cdot H(k)$, where d depends on the delay functions used. Moreover, we obtain stronger bounds in the case where users experience only delays, not construction costs; this includes a result that relates the cost of a best Nash equilibrium to that of an optimum with twice as many players, and a result that improves the potential-based bound on the price of stability for the single-source delay-only case.

Since a number of our proofs are obtained by following the results of best-response dynamics via a potential function, it is natural to investigate the speed of convergence of best-response dynamics for this game. We show that with k players, it can run for a time exponential in k. Whether there is a way to schedule players' moves to make best-response dynamics converge in a polynomial number of steps for this game in general is an interesting open question.

Finally, we consider a natural generalization of the cost-sharing model that carries us beyond the potential-function framework. Specifically, suppose each user has a *weight* (perhaps corresponding to the amount of traffic it plans to send), and we change the cost-allocation so that user i's payment for edge e is equal to the ratio of its weight to the total weight of all users on e. In addition to being intuitively natural, this definition is analogous to certain natural generalizations of the Shapley value [29]. The weighted model, however, is significantly more complicated: there is no longer a potential function whose value tracks improvements in users' costs when they greedily update their solutions. We also show, using a construction involving user weights that grow exponentially in k, that the price of stability can be as high as $\Omega(k)$. We have obtained some initial positive results here, including the convergence of best-response dynamics when all users seek to construct a path from a node s to a node t (the price of stability here is 1), and in the general model of users selecting sets from a ground set, where each element appears in the sets of at most two users.

Related Work. Network design games under a different model were considered by a subset of the authors in [3]; there, the setting was much more "unregulated" in that users could offer to pay for an arbitrary fraction of any edge in the network. This model resulted in instances where no pure Nash equilibrium existed; and in many cases in [3] when pure Nash equilibria did exist, certain users were able to act as "free riders," paying very little or nothing at all. The present model, on the other hand, ensures that there is always a pure Nash equilibrium within a logarithmic factor of optimal, in which users pay for a fair portion of the resources they use. Network creation games of a fairly different flavor — in which users correspond to nodes, and can build subsets of the edges incident to them — have been considered in [2, 12, 5, 16, 21, 31]. The model in this paper associates users instead with connection requests, and allows them to contribute to the cost of any edge that helps them to form a path that they need.

The bulk of the work on cost-sharing (see e.g. [19, 22] and the references there) tends to assume a fixed underlying set of edges. Jain and Vazirani [23] and Kent and Skorin-Kapov [26] consider cost-sharing for a single source network design game. Cost-sharing games assume that there is a central authority that designs and maintains the network, and decides appropriate cost-shares for each agent, depending on the graph and all other agents, via a complex algorithm. The agents' only role is to report their utility for being included in the network.

Here, on the other hand, we consider a simple cost-sharing mechanism, the Shapley-value, and ask what the strategic implications of a given cost-sharing mechanism are for the way in which a network will be designed. This question explores the feedback between the protocol that governs network construction and the behavior of self-interested agents that interact with this protocol. An approach of a similar style, though in a different setting, was pursued by Johari and Tsitsiklis [24]; there, they assumed a network protocol that priced traffic according to a scheme due to Kelly [25], and asked how this protocol would affect the strategic decisions of self-interested agents routing connections in the network.

The special case of our game with only delays is closely related to the congestion games of [39, 37]. They consider a game where the amount of flow carried by an individual user is infinitesimally small (a *non-atomic game*), while in this paper we assume that each user has a unit of flow, which it needs to route on a single path. In the non-atomic game of [39, 37] the Nash equilibrium is essentially unique (hence there is no distinction between the price of anarchy and stability), while in our atomic game there can be many equilibria. Fabrikant, Papadimitriou, and Talwar [17] consider our atomic game with delays only. They give a polynomial time algorithm to minimize the potential function Φ in the case that all users share a common source, and show that finding any equilibrium solution is PLS-complete for multiple source-sink pairs. Our results extend the price of anarchy results of [39, 37] about non-atomic games to results on the price of stability for the case of single source atomic games. Subsequent to our work, further results on the price of anarchy and stability in atomic games with delays were obtained in [4, 7, 11, 10, 40]. For games without delays, Agarwal and Charikar [1] give improved bounds on the price of stability in single-source undirected networks, and Fiat et al. [20] give bounds with the additional assumption that each vertex is the destination of some player. Other aspects of these and closely related games were recently explored in [9, 18].

A weighted game similar to ours is presented by Libman and Orda [28], with a different mechanism for distributing costs among users. They do not consider the price of stability, and instead focus on convergence in parallel networks. Recently, Chen and Roughgarden [8] proved general results on the price of stability and the existence of approximate Nash equilibria in the weighted version of our game.

2. Nash Equilibria of Network Design with Shapley Cost-Sharing. In this section we consider the *Fair Connection Game* for k players as defined in the Introduction. Let a directed graph $G = (V, E)$ be given, with each edge having a nonnegative cost c_e. Each player i has a set of terminal nodes T_i that he wants to connect. A strategy of player i is a set of edges $S_i \subset E$ such that S_i connects all nodes in T_i. We assume that we use the Shapley value to share the cost of the edges, i.e. all players using an edge split up the cost of the edge equally. Given a vector of players' strategies $S = (S_1, \ldots, S_k)$, let x_e be the number of agents whose strategy contains edge e. Then the cost to agent i is $C_i(S) = \sum_{e \in S_i}(c_e/x_e)$, and the goal of each agent is to connect its terminals with minimum total cost.

In the worst case, Nash equilibria can be very expensive in this game, so that the price of anarchy becomes as large as k. To see this, consider k players with common source s and sink t, and two parallel edges of cost 1 and k. The worst equilibrium has all players selecting the more expensive edge, thereby paying k times the cost of the optimal network. However, we can bound the price of stability by $H(k)$, which is the harmonic sum $1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$, as follows.

THEOREM 2.1. *The price of stability of the fair connection game is at most $H(k)$.*

Proof: The fair connection game that we have defined falls into the class of congestion games as defined by Rosenthal [36], as the cost of edge e to a user i is $f_e(x) = c_e/x$, which depends only on edge e and the number of users x whose strategy contains e. Rosenthal [36] shows that all congestion games have deterministic Nash equilibria. He proves this using a potential function Φ, defined as follows.

$$\Phi(S) = \sum_{e \in E} \sum_{x=1}^{x_e} f_e(x) \tag{2.1}$$

Rosenthal [36] shows that for any strategy $S = (S_1, \ldots, S_k)$ if a single player i deviates to strategy S'_i, then the change in the potential value $\Phi(S) - \Phi(S')$ of the new strategy set $S' = (S_1, \ldots, S'_i, \ldots, S_k)$ is exactly the change in the cost to player i. Note that the change of player i's strategy affects the cost of many other players $j \neq i$, but the value of Φ is not affected by the change in the cost of these players, it simply tracks the cost of the player who changes its strategy. Monderer and Shapley [30] call a game in which such a function Φ exists a *potential game.* To show that such a potential game has a Nash equilibrium, start from any state $S = (S_1, \ldots, S_k)$ and consider a sequence of selfish moves (allowing players to change strategies to improve their costs). In a congestion game any sequence of such improving moves leads to a Nash equilibrium as each move decreases the potential function Φ, and hence must lead to a stable state.

Neither Rosenthal nor Monderer and Shapley say anything about the quality of Nash equilibria with respect to the centralized optimum, but we can use their potential function to establish our bound. Let x_e be defined as above with respect to S. Now the potential function of Equation 2.1 in our case is $\Phi(S) = \sum_{e \in E} c_e H(x_e)$. Consider the strategy $S^* = (S^*_1, \ldots, S^*_k)$ defining the optimal centralized solution. Let $OPT = \sum_{e \in S^*} c_e$ be the cost of this solution. Then, $\Phi(S^*) \leq \sum_{e \in S^*}(c_e \cdot H(k))$, which is exactly $H(k) \cdot OPT$. Now we start from strategy S^* and follow a sequence of improving self-interested moves. We know that this will result in a Nash equilibrium S with $\Phi(S) \le \Phi(S^*)$.

Note that the potential value of any solution S is at least the total cost: $\Phi(S) \ge \sum_{e \in S} c_e = cost(S)$. Therefore, there exists a Nash equilibrium with cost at most $H(k) \cdot OPT$, as desired. ∎

Recall from the example in Figure 1.1 that the upper bound of Theorem 2.1 is tight.

Unfortunately, even though Theorem 2.1 says that cheap Nash equilibria exist, finding them is NP-complete.

THEOREM 2.2. *Given an instance of a fair connection game and a value C, it is NP-hard to determine if the game has a Nash equilibrium of cost at most C.*

Proof: The reduction is from 3D-Matching. Given an instance of 3D-Matching with node sets X, Y, Z, form a graph $G = (V, E)$ as follows. Form a node for each node in X, Y, and Z, and also a node $v_{i,j,k}$ for each 3D edge (x_i, y_j, z_k). Also add an additional node t. Form a directed edge from each node $v_{i,j,k}$ to t with cost function $c_e = 3$. Form a directed edge from each node v in X, Y, Z to all nodes representing 3D edges that contain v. Make these edges have a cost $c_e = 0$. Let $C = |X| + |Y| + |Z|$, and form a player for each node v in $X \cup Y \cup Z$. This player has two terminals: v and t.

If there exists a 3D Matching in the 3D-Matching instance, then there exists a Nash equilibrium in the above fair connection game of cost C: Take the 3D Matching M, and let S_i for the player whose terminals are v and t be the edge from v to the unique node $v_{i,j,k}$ corresponding to the 3D edge in M, and the edge from this node to t. Since M is a matching, the cost of S is exactly $3C/3 = C$. S is a Nash equilibrium, since any deviation for a player involves paying for some edge of cost 3 by himself, while the current amount he is paying is 1.

If no 3D Matching exists, then any solution to the fair connection game must cost more than C. Therefore, no Nash equilibrium can exist of cost at most C. This finishes the proof. ∎

Notice that the same proof shows that determining if there exists a Nash equilibrium that costs as little as OPT is NP-complete.

We can extend the results of Theorem 2.1 to concave cost functions. Consider the extended fair connection game where instead of a constant cost c_e, each edge has a cost which depends on the number of players using that edge, $c_e(x)$. We assume that $c_e(x)$ is a nondecreasing, concave function, modeling the buy-at-bulk economies of scale of buying edges that can be used by more players. Notice that the cost of an edge $c_e(x)$ might increase with the number of players using it, but the cost per player $f_e(x) = c_e(x)/x$ decreases if $c_e(x)$ is concave.

THEOREM 2.3. *Take a fair connection game with each edge having a nondecreasing concave cost function $c_e(x)$, where x is the number of players using edge e. Then the price of stability is at most $H(k)$.*

Proof: The proof is analogous to the proof of Theorem 2.1. We use the potential function $\Phi(S)$ defined by (2.1). As before, the change in potential if a player i deviates equals exactly the change of that player's payments. We start with the strategy S^* with minimum total cost, and perform a series of improving deviations until we reach a Nash equilibrium S with $\Phi(S) \le \Phi(S^*)$. To finish the proof all we need to show is that $cost(S) \le \Phi(S) \le H(k) \cdot cost(S)$ for all strategies S. The second inequality follows since $c_e(x)$ is nondecreasing and therefore $\sum_{x=1}^{x_e} (c_e(x)/x) \le H(x_e) \cdot c_e(x_e)$. To see that $cost(S) \le \Phi(S)$ notice that since $c_e(x)$ is concave, the cost per player must decrease with x, i.e. $c_e(x)/x$ is a nonincreasing function. Therefore, $cost(S) = \sum_{e\in S} c_e(x_e) = \sum_{e\in S} x_e \cdot (c_e(x_e)/x_e) \leq \Phi(S)$, which finishes the proof. ∎

Notice that the requirement of cost functions being concave is general enough to encompass the utility function of a player being a combination of the cost he has to pay for his edges and the distance between his terminals in the network of bought edges. If c_e is the cost function of an edge, we simply set $c'_e(x) = c_e(x) + x$. The payment of each player i now becomes $|S_i| + \sum_{e\in S_i}(c_e(x_e)/x_e)$, and c'_e is still concave if c_e is concave.

Extensions. The proof of Theorem 2.3 extends to a general congestion game, where players attempt to share a set of resources R that they need. Instead of having an underlying graph structure, we now think of each $s \in R$ as a resource with a concave cost function $c_s(x)$ of the number of users selecting sets containing s. The possible strategies of each player i is a set $\mathcal{S}_i$ of subsets of R. Each player seeks to select a set $S_i \in \mathcal{S}_i$ so as to minimize his cost. Since the proofs above did not rely on the graph structure, they translate directly to this extension.

We can further extend the results to the case where the cost of a player is a combination of the cost $c_e(x)/x$, and a function of the selected set, such as the distance between terminals in the network design case. More precisely, the price of stability is still at most $H(k)$ if each player is trying to minimize the cost $\sum_{e\in S_i}(c_e(x_e)/x_e) + d_i(S_i)$ where c_e is monotone increasing and concave, and d_i is an arbitrary function specific to player i (e.g. a distance function, or diameter of S_i, etc.). The proof is analogous to Theorem 2.3, except with a new potential $\Phi(S) = \sum_i d_i(S_i) + \sum_{e\in S}\sum_{x=1}^{x=x_e}(c_e(x)/x)$. Notice that this is technically not a congestion game on the given graph G. Finally we note that all these results (as well as those subsequent) hold in the presence of capacities. Adding capacities u_e to each edge e and disallowing more than u_e players to use e at any time does not substantially alter any of our proofs.

The Case of Undirected Graphs. While the bound of $H(k)$ is tight for general directed graphs, it is not tight for undirected graphs. Finding the correct bound is an interesting open problem; see [20] for some recent progress. In the case of two players, our bound on the price of stability is $H(2) = 3/2$. In Section 4 we show that this bound can be improved to $4/3$ in the case of two players and a single source. We also give an example to show that this bound is tight.

3. Dealing with Delays. In most of the previous section, we assumed that the utility of a player depends only on the cost of the edges he uses. What changes if we introduce latency into the picture? We have extended this to the case when the players' cost is a combination of "design" cost and the length of the path selected. More generally, delay on an edge does not have to be simply the "hop-count", but can also depend on congestion, i.e., on the number of players using the edge. In this section we will consider such a model.

Assume that each edge has both a cost function $c_e(x)$ and a latency function $d_e(x)$, where $c_e(x)$ is the cost of building the edge e for x users and the users will share this cost equally, while $d_e(x)$ is the delay suffered by users on edge e if x users are sharing the edge. The goal of each user will be to minimize the sum of his cost and his latency. If we assume that both the cost and latency for each edge depend only on the number of players using that edge, then this fits directly into our model of a congestion game above: the total cost felt by each user on the edge is $f_e(x) = c_e(x)/x + d_e(x)$. If the function $xf_e(x)$ is concave then Theorem 2.3 applies. But while concave functions are natural for modeling cost, latency tends to be convex.

3.1. Combining Costs and Delays. First, we extend the argument in the proof of Theorem 2.3 to general functions f_e. The most general version of this argument is expressed in the following theorem.

THEOREM 3.1. *Consider a fair connection game with arbitrary edge-cost functions f_e. Suppose that $\Phi(S)$ is as in Equation 2.1, with $cost(S) \leq A \cdot \Phi(S)$, and $\Phi(S) \leq B \cdot cost(S)$ for all S. Then, the price of stability is at most $A \cdot B$.*

Proof: Let S^* be a strategy such that S_i^* is the set of edges i used in the centralized optimal solution. We know from above that if we perform a series of improving deviations on it, we must converge to a Nash equilibrium S' with potential value at most $\Phi(S^*)$. By our assumptions, $cost(S') \leq A \cdot \Phi(S') \leq A \cdot \Phi(S^*) \leq AB \cdot cost(S^*) = AB \cdot OPT$. ■

Our main interest in this section are functions $f_e(x)$ that are the sums of the fair share of a cost and a delay, i.e., $f_e(x) = c_e(x)/x + d_e(x)$. We will assume that $d_e(x)$ is monotone increasing, while $c_e(x)$ is monotone increasing and concave.

COROLLARY 3.2. *If $c_e(x)$ is concave and nondecreasing, $d_e(x)$ is nondecreasing for all e, and $x_e d_e(x_e) \leq A \sum_{x=1}^{x_e} d_e(x)$ for all e and x_e, then the price of stability is at most $A \cdot H(k)$. In particular, if $d_e(x)$ is a polynomial with degree at most l and nonnegative coefficients, then the price of stability is at most $(l+1) \cdot H(k)$.*

Proof: For functions $f_e(x) = c_e(x)/x + d_e(x)$, both the cost and potential of a solution come in two parts corresponding to cost c and delay d.

For the part corresponding to the cost, the potential over-estimates the cost by at most a factor of $H(k)$ as proved in Theorem 2.3. If on the delay, the potential underestimates the cost by at most a factor of A, then we get the bound of $A \cdot H(k)$ for the price of stability by Theorem 3.1. ■

Therefore, for reasonable delay functions, the price of stability cannot be too large. In particular, if the utility function of each player depends on a concave cost and delay that is independent of the number of users on the edge, then we get that the price of stability is at most $H(k)$ as we have shown at the end of the previous section. If the delay grows linearly with the number of users, then the price of stability is at most $2H(k)$.

3.2. Games with Only Delays. In this subsection we consider games with only delay. We assume that the cost of a player for using an edge e used by x players is $f_e(x) = d_e(x)$, and d_e is a monotone increasing function of x. This cost function models delays that are increasing with congestion.

We will mostly consider the special case when there is a common source s. Each player i has one additional terminal t_i, and the player wants to connect s to t_i via a directed path. Fabrikant, Papadimitriou, and Talwar [17] showed that in this case, one can compute the Nash equilibrium minimizing the potential function Φ via a minimum cost flow computation. For each edge e they introduce many parallel copies, each with capacity 1, and cost $d_e(x)$ for integers $x > 0$. We will use properties of a minimum cost flow for establishing our results.

3.2.1. A Bicriteria Result. First we show a bicriteria bound, and compare the cost of the cheapest Nash equilibrium to that of the optimum solution with twice as many players.

THEOREM 3.3. *Consider the single source case of a congestion game with only delays. Let S be the minimum cost Nash equilibrium and S^* be the minimum cost solution for the problem where each player i is replaced by two players. Then $cost(S) \leq cost(S^*)$.*

Proof: Consider the Nash equilibrium obtained by Fabrikant et al [17] via a minimum cost flow computation. Assume that x_e is the number of users using edge e at this equilibrium. By assumption, all users share a common source s. Let $D(v)$ denote the cost of the minimum cost path in the residual graph from s to v. The length of the path of user i is at most $D(t_i)$ (as otherwise the residual graph would have a negative cycle) and hence we get that $cost(S) \leq \sum_i D(t_i)$.

Now consider modified delay function $\hat{d}_e$ for each edge $e = (u, v)$. Define $\hat{d}_e(x) = d_e(x)$ if $x > x_e$, and $\hat{d}_e(x) = D(v) - D(u)$ if $x \leq x_e$. Note that for any edge e we have $D(v) - D(u) \leq d_e(x_e + 1)$ as edge $e = (u, v)$ is in the residual graph with cost $d_e(x_e+1)$. This implies that the modified delay $\hat{d}$ is monotone. For edges with $x_e \neq 0$ we also have that $d_e(x_e) \leq D(v) - D(u)$ as the reverse edge (v, u) is in the residual graph with cost $-d_e(x_e)$, so the delay of an edge is not decreased.

Now observe that, subject to new delay $\hat{d}$, the shortest path from s to t_i has length $D(t_i)$. The minimum possible cost of two paths from s to t_i for the two users corresponding to user i is then at least $2D(t_i)$ for each player i. Therefore the minimum cost of a solution with delays $\hat{d}$ is at least $2\sum_i D(t_i)$.

To bound $cost(S^*)$ we need to bound the difference in cost of a solution when measured with delays $\hat{d}$ and d. Note that for any edge $e = (u, v)$ and any number x we have that $x\hat{d}_e(x) - xd_e(x) \leq x_e(D(v) - D(u))$, and hence the difference in total cost is at most $\sum_{e=(u,v)} x_e(D(v) - D(u)) = \sum_i D(t_i)$. Using this, we get that $cost(S^*) \geq \sum_i D(t_i) \geq cost(S)$. ∎

Note that a similar bound is not possible for a model with both costs and delays, when additional users compensate to some extent for the price of stability. Consider a problem with two parallel links e and e' and k users. Assume on link e the cost function is $c_e(x) = 1 + \varepsilon$ for a small $\varepsilon > 0$, and the latency function is $d_e(x) = 0$. On the other link e' the cost is $c_{e'}(x) = 0$, and the delay with x users is $d_{e'}(x) = 1/(k - x + 1)$. The optimum solution is to use the first edge e, and it costs $1 + \varepsilon$. Note that the optimum with any number of extra users costs the same. On the other hand, the only Nash equilibrium is to have all users on e', incurring delay 1, for a total cost of k.

3.2.2. Bounding the Price of Stability with Only Delays. Note that the $H(k)$ term in Corollary 3.2 comes from the concave cost c, and so the bound obtained there improves by an $H(k)$ factor when the cost consists only of the delay. The results from Corollary 3.2 already tell us that if the delay functions are such that $x_e d_e(x_e) \leq A \sum_{x=1}^{x_e} d_e(x)$, the the price of stability is at most A. Specifically, we know that if the delays are polynomial of degree l, then the price of stability is at most $l + 1$, and therefore with linear delays the price of stability is at most 2.

Roughgarden [37] showed a tighter bound for non-atomic games. He assumed that the delay is monotone increasing, and the total cost of an edge $xd_e(x)$ is a convex function of traffic x. He showed that for any class of such functions $\mathcal{D}$ containing all constant functions, the price of anarchy is always obtained on a two node, two link network. Let us call $\alpha(\mathcal{D})$ the price of anarchy for non-atomic games with delays from the class $\mathcal{D}$ (which is also the price of stability, since the Nash equilibrium is unique in that context). For example, Roughgarden [37] showed that for polynomials of degree at most l this bound is $O(l/\log l)$, and for linear delays it is $4/3$. Here we extend this result to a single source atomic game, and thereby show tighter bounds than in Corollary 3.2 for the single source case.

Theorem 3.4. *If in a single source fair connection game all costs are delays, and all delays are from a set $\mathcal{D}$ satisfying the above condition, then the price of stability* *is at most* $\alpha(D)$.

Proof: As in the proof of Theorem 3.3 consider the Nash equilibrium obtained via a minimum cost flow computation, and let $D(v)$ be the length of the shortest path from s to v in the residual graph. As before we have that $cost(S) \le \sum_i D(t_i)$. Further, for each edge $e = (u, v)$ we have that $D(v) - D(u) \le d_e(x_e + 1)$, and for edges with $x_e \ne 0$, we also have that $d_e(x_e) \le D(v) - D(u)$.

Give each edge $e = (u, v)$ a capacity of x_e, and augment our network by adding a parallel edge e' with constant delay $D(v) - D(u)$. Let $\hat{G}$ denote the resulting network flow problem. Note that the new capacity and the added links do not affect the equilibrium, as $d_e(x_e) \le D(v) - D(u)$. For each edge e, the two parallel copies: edge e with new capacity x_e and edge e', can carry any number of paths at least as cheaply as the original edge e could since $D(v) - D(u) \le d_e(x_e + 1)$, hence this change in the network can only improve the minimum possible cost. We will prove the bound in this new network by comparing the cost of the Nash equilibrium with the minimum possible cost of a (possibly fractional) flow carrying one unit of flow from s to each of the terminals t_i.

The nice property of $\hat{G}$ is that the optimum fractional flow $\hat{x}$ in $\hat{G}$ is easy to determine. Consider an edge $e = (u, v)$ that is used by $x_e \ne 0$ paths in the equilibrium. We will obtain a fractional flow $\hat{x}_e$ by splitting the corresponding x_e amount of flow between the two edges e and e'. For an edge e let $\ell_e(x) = d_e(x) + x d'_e(x)$. By assumption, $d_e(x) \le \ell_e(x)$ for all x. For an edge e such that $\ell_e(x_e) \le D(v) - D(u)$, we set $\hat{x}_e = x_e$, and $\hat{x}_{e'} = 0$. Otherwise, let $\hat{x}_e$ be such that $\ell_e(\hat{x}_e) = D(v) - D(u)$, and let $\hat{x}_{e'} = x_e - \hat{x}_e$.

First, we claim that $\hat{x}$ is the minimum cost fractional solution in $\hat{G}$. For all edges $e = (u, v)$ such that $\hat{x}_e \ne x_e$, we have that $\ell_e(\hat{x}_e) = D(v) - D(u)$. When $\hat{x}_e = x_e$, then we have that flow $\hat{x}_e$ is equal to the capacity of the edge, and $\ell_e(\hat{x}_e) \le D(v) - D(u)$. Therefore, if there is a negative cycle in the residual graph of $\hat{x}_e$ with constant edge costs $\ell_e(x_e)$ for e and costs $D(v) - D(u)$ for e', then this is also a negative cost cycle in G with constant edge costs $D(v) - D(u)$. This is impossible, however, since x_e is a min-cost flow with those costs. We can now use Lemma 3.5 to see that $\hat{x}_e$ is also a min-cost flow for edge costs $x d_e(x)$.

The theorem then follows, as on each original edge $e \in E$ the cost $x_e d_e(x_e)$ is at most $\alpha(\mathcal{D})$ times the cost of the corresponding two edges e and e' in $\hat{G}$ by Lemma 3.6. ■

To finish the proof of the Theorem, we require the following lemmas.

LEMMA 3.5. *Let G be a network, and x_e be a fractional flow sending one unit of flow from the source s to each sink t_i. Let ℓ denote the gradient of the total cost $x d_e(x)$, that is, let $\ell_e(x) = d_e(x) + x d'_e(x)$ for each edge e. The flow x_e is minimum cost subject to the cost $\sum_e x d_e(x)$ if and only if it is a minimum cost flow subject to the constant cost function $c_e = \ell_e(x_e)$.*

Proof: If the flow x_e is not of minimum cost subject to costs c_e, then the residual graph has a negative cycle, and moving a small amount of flow along the cycle decreases the cost $\sum_e x d_e(x)$, as the cost c_e is exactly the gradient of this objective function. To see the other direction, we use the fact that the cost function is convex by assumption, and hence all local optima are also global optima. ■

Next, it is useful to recall from [37] what is $\alpha(\mathcal{D})$. Consider edge e, with delay $d(x)$ from class $\mathcal{D}$. Now consider a graph with two parallel links: edge e, which has delay $d(x)$, that will carry some r units of flow, and a parallel link e' with constant delay $d(r)$ independent of the traffic. Now the unique Nash equilibrium is to route all r units of flow on e, while we get the optimum by setting x such that the gradient $c(x) = d(x) + xd'(x)$ is equal to $d(r)$, and sending x units of flow along e, and the remainder $r - x$ along edge e'. This is because of the following Lemma from [37].

LEMMA 3.6 ([37]). *If a set $\mathcal{D}$ of delay functions satisfies the above condition, then the price of stability is at most $\alpha(D) = \max_{r,x,d\in\mathcal{D}} rd(r)/(xd(x) + (r - x)d(r))$, and the maximum is achieved by setting x such that $d(x) + xd'(x) = d(r)$.*

4. The Undirected Case. While the bound of $H(k)$ on the price of stability is tight for general directed graphs with costs, it is not tight for undirected graphs. Finding the correct bound is an interesting open problem. In the case of two players, our bound on the price of stability is $H(2) = 3/2$. We now show that that this bound can be improved to $4/3$ in the case of two players and a single source.

Here is an example of an undirected two-player game with the price of stability approaching $4/3$. Let G have 3 nodes: s, t_1, and t_2. Player 1 wants to connect t_1 with s, and player 2 wants to connect t_2 with s. There are edges (s, t_1) and (s, t_2) with cost 2. There is an edge (t_1, t_2) with cost $1 + \varepsilon$. The optimal centralized solution has cost $3 + \varepsilon$. However, the cheapest Nash has cost 4. This example implies that the following claim is tight.

CLAIM 4.1. *The price of stability is at most* $4/3$ *in a fair connection game with two players in an undirected graph, each having two terminals with one terminal in common.*

Proof: Let s be the common terminal, and let t_1 and t_2 be the personal terminals. Consider the optimal centralized solution (S_1, S_2). Let $X_1 = S_1 \backslash S_2$ be the edges only being used by player 1, $X_2 = S_2 \backslash S_1$ be the edges only used by player 2, and $X_3 = S_1 \cap S_2$ be the edges shared by the two players. Let (S_1', S_2') be a Nash equilibrium that a series of improving responses converges to starting with (S_1, S_2). Similarly, let $Y_1 = S_1' \backslash S_2'$, $Y_2 = S_2' \backslash S_1'$, and $Y_3 = S_1' \cap S_2'$. Finally, set $x_i = cost(X_i)$ and $y_i = cost(Y_i)$ for $1 \leq i \leq 3$. By the properties of $\Phi(S_1, S_2)$ from the proof of Theorem 2.1, we know that $\Phi(S_1', S_2') \leq \Phi(S_1, S_2)$. Substituting in the definition of Φ, we obtain that

$$y_1 + y_2 + \frac{3}{2}y_3 \leq x_1 + x_2 + \frac{3}{2}x_3. \tag{4.1}$$

Look at S_1' and S_2' as paths instead of sets of edges (there will be no cycles since then this would not be a Nash). We now show that in (S_1', S_2'), as in any Nash equilibrium, once the paths of the two players merge, they do not separate again. Suppose to the contrary that this happens. Let v be the first node that S_1' and S_2' have in common, and set P_1 and P_2 be the subpaths of S_1' and S_2' after v, respectively. We know that $cost(P_1 \backslash P_2) = cost(P_2 \backslash P_1)$, since if they were not equal, say $cost(P_1 \backslash P_2) > cost(P_2 \backslash P_1)$, then player 1 could deviate to P_2 instead and pay strictly less. However, even if they are equal, player 1 could deviate to use P_2 instead of P_1, and pay strictly less, since he will pay the same as before on edges in $P_1 \cap P_2$, and pay only $cost(P_1 \backslash P_2)/2$ in total on the other edges. Therefore, the only way this could be a Nash equilibrium is if $P_1 \cap P_2 = P_1 = P_2$, as desired.

Consider a deviation from (S_1', S_2') that player 1 could make. He could decide to use $X_1 \cup X_2 \cup Y_2 \cup Y_3$ instead of $S_1' = Y_1 \cup Y_3$. This is a valid deviation because player 1 still connects his terminals by following X_1 until X_1 meets with X_2, then following X_2 back to t_2, and then following S_2' to s. Since (S_1', S_2') is a Nash equilibrium, this deviation must cost more to player 1 than his current payments, and so $x_1 + x_2 + y_2/2 + y_3/2 \geq y_1 + y_3/2$. By symmetric reasoning, $x_1 + x_2 + y_1/2 + y_3/2 \geq y_2 + y_3/2$.

If we add these inequalities together, we obtain that

$$y_1/2 + y_2/2 \leq 2x_1 + 2x_2. \tag{4.2}$$

To show that the price of stability is at most 4/3, it is enough to show that $cost(S'_1, S'_2) \leq \frac{4}{3}cost(S_1, S_2)$. Using the above notation, this is the same as showing $3y_1 + 3y_2 + 3y_3 \leq 4x_1 + 4x_2 + 4x_3$. We do this by using Inequalities 4.1 and 4.2 as follows:

$$\begin{aligned} 3y_1 + 3y_2 + 3y_3 &\leq 3y_1 + 3y_2 + 4y_3 \\ &= \frac{1}{3}(y_1 + y_2) + \frac{8}{3}(y_1 + y_2 + \frac{3}{2}y_3) \\ &\leq \frac{4}{3}(x_1 + x_2) + \frac{8}{3}(x_1 + x_2 + \frac{3}{2}x_3) \\ &= 4x_1 + 4x_2 + 4x_3 \end{aligned}$$

■

5. Convergence of Best Response. In this section, we show that, in general, best response dynamics in our game can take a long time to converge to an equilibrium. Specifically, we construct a sequence of best responses that takes exponential time to converge to a Nash equilibrium.

THEOREM 5.1. *Best response dynamics for k players may run in time exponential in k.*

To prove this, we now construct an example (shown in Figure 5.1) in which by appropriate ordering of the best response of players, we can simulate a $\Omega(k)$-bit binary counter. The idea is that we have a set of players corresponding to each bit of the counter. Then we describe a sequence of best response moves that lead to the counter incrementing from the "all zeros" state onwards. Since with n bits, we can implement a counter that counts upto 2^n, then as long as we can show that each increment of the counter corresponds to a sequence of best response moves from the current configuration, we would have shown an exponentially long best response sequence. In what follows, we first describe the set of gadgets to construct the counter, and then show the set of inequalities that result from the increments of the counter.

The graph has $3n$ players: n "bit" players, each denoted by bit(i), and for each bit player we also have two "auxiliary" players. The auxiliary players of the bit(i) player are denoted by $A(i)$ and $B(i)$. We construct the graph as follows. For each player we form a gadget as shown in Figure 5.1. The gadget for each bit player and each auxiliary player has only two path options, we call these the 0 path and the 1 path. Figure 5.1 shows the set of edges that belong to each gadget. The gadget for the i^{th} bit player, for instance, consists of edges $\alpha(i)$, $\beta(i)$, a set of edges $t(j,i)$, one for each $j > i$, and a few unnamed edges as shown in the figure. To construct the graph, we simply take the union of these gadgets. The labeled edges are shared with the other gadgets, whereas the unlabeled edges are not. Furthermore, the label Δ actually denotes the cost of the corresponding edges, these edges too are individual to each gadget.

Thus, each of the $\alpha(i)$ edges are in the gadgets of players $A(i)$ and bit(i), and so appear in the 1 path of both $A(i)$ and bit(i). Each edge $\beta(i)$ is in the 1 path of two players $B(i)$ and bit(i), and each edge $e(i)$ is in the 0 path of $A(i)$ and the 1 path of $B(i)$. Each edge $f(i,j)$ and $t(i,j)$ is actually shared by a number of gadgets. For a specific i^* and j^*, the same edge $f(j^*, i^*)$, for instance, is present in the 0-path of

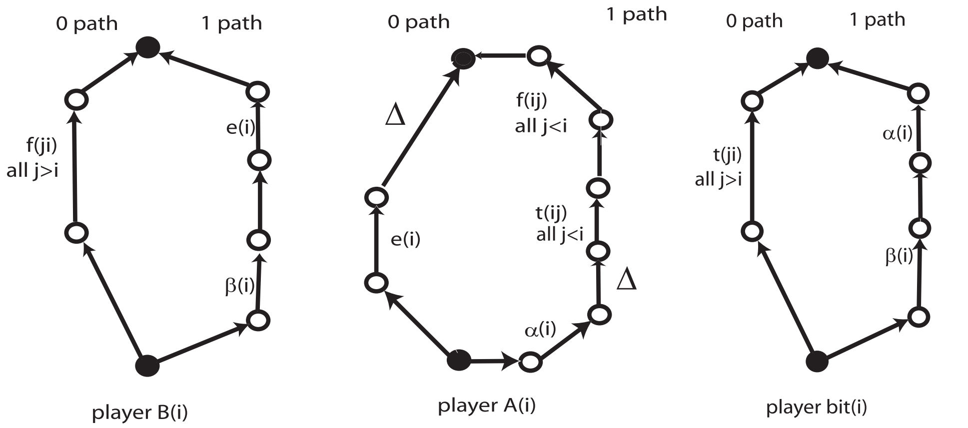

FIG. 5.1. *The construction of an exponential Best-Response run. The filled-in nodes are the sources and sinks of the players.*

bit_1	bit_0	B_1	A_1	B_0	A_0
0	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	0	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	0	1	0	0
1	0	0	1	0	0
1	0	1	1	0	0
1	0	1	0	0	0
1	1	1	0	0	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	0	1	0

TABLE 5.1

Table showing the first 11 *steps of the scheduled best response run for 2 players. The number "1" represents that this player is taking the 1-path in the current configuration, and "0" represents that it takes the 0-path. The first, fifth, eleventh, and last configuration correspond to the counter values of 00, 01, 10, and 11 respectively.*

all $B(i)$ players such that $i < j^*$ and also in the 1 path of all the $A(j)$ players such that $j > i^*$. In addition to the unnamed edges drawn in the gadget, assume there are unnamed edges before and after every edge type of $f(i,j)$ or $t(i,j)$ and these unnamed edges are directed and specific to a particular gadget. In the above Figure, we have not specified the order in which the edges of, for example, $f(j,i)$ for all $j > i$ in the gadget of $bit(i)$ proceed. In all of these cases, there is one index that is fixed, and one that is variable, so we assume that these edges appear in increasing order of the variable index.

We will later prove that *no player will ever choose a path in this graph other than the* 0 *or the* 1 *path corresponding to its gadget.* This significantly simplifies the strategy sets of the players, so now we can simply think of each player as choosing either strategy 0 or 1. Below we describe an exponential-length best-response run that simulates a binary counter. A sample run of this type is shown in Table 5.1. The first, fifth, and eleventh step shown correspond to the values 00, 01, and 10 of the counter, and indeed these are the values of the strategies of the bit(i) players. The rest of the steps pictured are intermediate steps, and only exist to get the bit(i) players into the correct configuration.

In what follows, we abuse notation and represent the cost of an edge by the label itself, e.g. $\alpha(i), f(i,j)$, $t(i,j)$ will also be used to denote the cost of the corresponding edges. All the unnamed edges are zero cost, except for the two sets of edges, one in each path of $A(i)$ that each have cost Δ, which is large, say $3n$ times bigger than the sum of the cost of all the named edges. Thus player $A(i)$ always pays at least Δ in order to get to the sink, but never should agree to pay more than $\Delta + \Delta/3n$.

We also refer to the player going on the "one" path as the player being set and going on the "zero" path as the player being reset. Each player has one source and one sink and the paths of each player are as shown in the gadgets above. The costs of the paths of i^{th} bit player are referred to as $x_i^{(0)}$ and $x_i^{(1)}$, and those of the player $A(i)$ and $B(i)$ as $a_i^{(0)}$, $a_i^{(1)}$ and $b_i^{(0)}$, $b_i^{(1)}$ respectively. Note that these denote the actual costs of these paths, not the cost shares in any configuration. Now we describe the sequence of best-response moves that lead to the counter incrementing. For the moves that we will describe to actually be best-response moves, particular inequalities will have to hold on the edge costs. Below we give a sequence of best-response moves that is exponential length, together with the inequalities that must hold for these to be valid best-response moves. We will later show that all of these inequalities can be satisfied.

Start Step : All the players are reset.

General Step : The bits from 1 to $\ell-1$ are all set. The bits from $\ell+1$ to n maybe at 0 or 1. The ℓ^{th} bit is currently at 0 and has to be set at 1. Also, all the $A(j)$ players are reset. The $B(j)$ players are set if and only if the j^{th} players are set.

Notice that the first, fifth, and last step in Table 5.1 are of this form. We now show a sequence of best responses which will set the ℓ'th bit, set all the bits 1 to $\ell-1$ to 0 (which is what should happen in a counter when the ℓ'th bit is set), and return all the auxiliary players to the form described in the "General Step" above.

- First, the ℓ^{th} bit sets. At this point, there are no other players using either of bit ℓ^{th} player's paths, so for this to happen, we need only the following to be true.

$$x_\ell^{(1)} < x_\ell^{(0)}$$
$$\text{implying, } \alpha(\ell) + \beta(\ell) < \sum_{j>\ell} t(j,\ell)$$

The setting of the bit(ℓ) player will now first trigger $A(\ell)$ and then $B(\ell)$, as follows.

- The cost of 1 path of $A(\ell)$ has now decreased by $\alpha(\ell)/2$ because of the player bit(ℓ) using it. $A(\ell)$ is thus triggered and is allowed to set. Since both the

0 and 1 path have the Δ cost edges that are to be paid by $A(i)$ alone, they do not matter in the best response calculations. Thus, for this to be a valid best response move, it is enough that,

$$a_\ell^{(1)} - \alpha(\ell)/2 < a_\ell^{(0)}$$
$$\text{implying that, } \alpha(\ell)/2 + \sum_{j<\ell} f(\ell,j) + \sum_{j<\ell} t(\ell,j) < e(\ell)$$

- The setting of $A(\ell)$ triggers all the $B(i)$ for $i < \ell$ to be reset. Recall that the corresponding $A(i)$ are already reset. We allow these $B(i)$ to reset. Due to $A(\ell)$ being on its 1 path, the cost of 0 path has changed by $f(\ell,i)/2$. Since only the player bit(i) is using the edge $\beta(i)$, then for $B(i)$ to want to reset, we need that,

$$b_i^{(0)} - f(\ell,i)/2 < b_i^{(1)} - e(i)/2 - \beta(i)/2$$
$$\text{that is, } \sum_{j>i} f(j,i) - f(\ell,i)/2 < e(i)/2 + \beta(i)/2$$

- $A(\ell)$ now also triggers all the bit players bit(i) for all $i < l$ to reset by reducing the cost of the 0 path of bit(i) by $t(\ell,i)/2$. For each such $i < \ell$, the bit $B(i)$ has also just been reset. Thus for this to be a best response move, it should be the case that,

$$x_i^{(0)} - t(\ell,i)/2 < x_i^{(1)}$$
$$\sum_{j>i} t(j,i) - t(\ell,i)/2 < \alpha(i) + \beta(i)$$

- Now because of the setting of the bit(ℓ), the 1 path of $B(\ell)$ became cheaper by $\beta(\ell)/2$. $B(\ell)$ wants to set and is allowed to. For this, we need that

$$b_\ell^{(1)} - \beta(\ell)/2 < b_\ell^{(0)}$$
$$e(\ell) + \beta(\ell)/2 < \sum_{j>\ell} f(j,\ell)$$

- Lastly, we now want $A(\ell)$ to reset. As a result of the setting of $B(\ell)$, the 0 path of $A(\ell)$ became cheaper by $e(\ell)/2$. Edges on the 1-path of $A(\ell)$ are shared by players bit(x) and $B(x)$ for $x < i$, and by the player bit(ℓ). So we need that the 0-path of $A(\ell)$ be less expensive to it even if all these players pay off their corresponding shares on the 1-path. Again since the Δ-cost edges are present on both sides, they do not matter in the computation of the best response. That is, we need that

$$a_\ell^{(0)} - e(\ell)/2 < a_\ell^{(1)} - \sum_{j<\ell} (f(\ell,j)/2 + t(\ell,j)/2) - \alpha(\ell)/2$$
$$e(\ell)/2 < \alpha(\ell)/2 + \sum_{j<\ell} (f(\ell,j)/2 + t(\ell,j)/2)$$

- Now we have the subgame of bit and auxiliary players from 1 to $\ell - 1$ being completely reset, and no other player corresponding to the higher-numbered bits influencing any of their paths. This corresponds to the first $\ell - 1$ bits of the counter becoming 0 because the ℓ'th bit just became 1. We can now use the best response run corresponding to incrementing the first $\ell - 1$ bits of the counter from all 0's to all 1's again, without any interference from the players corresponding to the higher-numbered bits. This gets us back to the configuration in the start of the recursion (the "General Step" above), except now we need to deal with the $(\ell+1)^{st}$ bit.

Proof of Theorem 5.1: We now prove that the above game has an exponential best response run under the above best response scheduling.

All we need to show is that the moves described in the scheduling are best responses. We first argue that each player has only two cheap paths available to him, which we have described as the zero path and the one path. To complete the construction we next need to come up with a set of values for the links that satisfy the set of best response inequalities above. Taken together, it follows that the moves are all best responses and simulate a $\Omega(k)$-size counter.

First note that the 0 path and 1 path of any one particular player are vertex disjoint. Recall that the unnamed edges, both in the figure and the ones in between the $f(i,j)$ edges and the $t(i,j)$ edges are all exclusive to each gadget. By our construction, the unnamed edges also impose the following property on the 0/1-paths of all players: the 0/1-path of any one player must either have at least one edge in common or must be vertex disjoint from the 0/1-path of any other player. We do not have to worry about the case of just having vertices in common.

We need to argue that each player has only two cheap paths available to him. Intuitively the argument is as follows. If one player deviates out of his gadget (i.e, takes a path other than its 0 or 1 path), it will never be able to come back to his own sink, or would have to pay an exorbitant amount (Δ) in order to reach its sink. Thus, the only available cheap strategies to a player are the 0 and 1 paths.

Before going into the details, we first define a function to make the notation simpler. For any edge e, define the function $sink(e)$ to be the set of all possible sinks (terminal nodes of players) that paths from this edge can lead to. Note that the graph only has edges $f(x,y)$ and $t(x,y)$ where $x > y$, there are no corresponding edges for $x \leq y$. For notational simplicity, for a general tuple (x,y), we define $sink(f(x,y))$ as follows. Note that $f(x,y)$ can lead to the sink of $A(x)$ and $B(y)$ if $x > y$, as well as to $sink(f(x+1,y))$ and $sink(f(x,y+1))$. Thus, inductively, if $x > y$

$$sink(f(x,y)) = \{A(x), \dots, A(n), B(y), \dots, B(n)\}$$

and $\emptyset$ otherwise (since the corresponding $f(x,y)$ edges do not appear in the gadget at all).

Similarly, since for $x \leq y$, the edges $t(x,y)$ are not present in the construction, define $sink(t(x,y)) = \emptyset$ for $x \leq y$. In order to compute $sink(t(x,y))$ for $x > y$, note that $t(x,y)$ can lead to $A(x)$, $bit(y)$, as well as $sink(t(x+1,y))$, $sink(t(x,y+1))$, and $sink(f(x,1))$ (since in the 1 path of $A(x)$, edge $t(x,y)$ can be followed by $f(x,1)$). Thus, if $x > y$

$$sink(t(x,y)) = \{bit(y), \dots, bit(n), A(x), \dots, A(n), B(1), \dots, B(n)\}$$

and $\emptyset$ else.

We now flesh out the argument for each player for each of the two cases, the set-case and the reset-case. Consider player $B(i)$. If a strategy follows the first unnamed edge of 1-path, then it reaches $\beta(i)$. From the end vertex of $\beta(i)$ it has the option of choosing the next unnamed edge in 1-path of gadget $B(i)$, or in the 1-path of $bit(i)$. In the first case the strategy has to go through $e(i)$. But after it crosses $e(i)$ there only is a zero-cost edge left to be covered to the sink of $B(i)$, and so other deviations are useless. If the path instead had chosen to enter the gadget of $bit(i)$ after the edges $\beta(i)$, then it would have to travel through $\alpha(i)$ and then either get stuck at the sink of $bit(i)$ or travel through one or more edges of the form $\{t(i,y), y < i\}$. But in order to enter an edge of the form $t(i,y)$ this strategy would have to pay at least $\Delta/3n$ to share the edge priced Δ, which is more than the total cost of $B(i)$. So any strategy starting on the 1-path of $B(i)$ does not enter the $A(i)$ gadget.

Next consider a strategy starting out on the 0-path of $B(i)$ through the first unnamed edge. The edges then appearing on this path are of the form $\{f(x,i), x > i\}$. These edges also appear in the gadgets $A(x)$. If this strategy chooses to deviate to exit the $B(i)$ gadget after the edge $f(x,i)$ and enter the $A(x)$ one, the labeled edge that it meets next is $f(x, i+1)$. But no path from $f(x, i+1)$ leads to the sink of $B(i)$, as is verified from the sink-function above. So this deviation cannot happen. We have now shown that the best strategies of $B(i)$ must be the 0 and 1 paths only.

Now, take the $A(i)$ player and a strategy which starts out on the first unnamed edge of the 1-path. This strategy then has to go through $\alpha(i)$. From the endpoint of $\alpha(i)$ the only edges are to the sink of $bit(i)$, or continue along the $A(i)$ gadget. The edges that appear next are of the form $\{t(i,x), x < i\}$. If the path does not continue along the gadget of $A(i)$, and instead switches to the 0-path of $bit(x)$, then the next edge it encounters is $t(i+1, x)$. But according to the sink function, there is no way to reach the sink of $A(i)$ from $t(i+1,x)$, so this is not a valid deviation. Next, the path cannot deviate out of any the $\{f(i,x), x < i\}$ edges since $A(i)$ does not lie in any $sink(f(i+1,x))$. Thus strategies of $A(i)$ starting out on the 1-path of $A(i)$ stay on this path.

Next consider the strategy of $A(i)$ that starts out through the unnamed edge in the 0-path of $A(i)$. This path cannot deviate after $e(i)$ because it would just get stuck in the sink of $B(i)$. Thus, we have shown that the best strategies of $A(i)$ must be the 0 and 1 paths only.

Lastly consider the i-th bit player. Along 0-path of bit-i, the only shared nodes and edges of this player are the $t(x,i)$ edges that are each shared with $A(x)$ for all $x > i$. Suppose this bit player follows the $t(x,i)$ edge to the gadget of $A(x)$ with $x > i$. The next edge is then $t(x, i+1)$ which does not lead to $bit(i)$-sink. Thus the 0-path strategies do not allow any deviations. A path starting with the first edge of the 1-path of this bit-player, can deviate after $\beta(i)$ and enter edge $e(i)$ of the $B(i)$ gadget. But from there all paths get stuck at sinks. Thus the 1-path deviations cannot happen after $\beta(i)$. But the deviations cannot happen after $\alpha(i)$ either, as the remaining edge is zero cost.

This concludes the proof that the best strategies of the players always correspond to the 0 and 1 paths in the gadgets.

For the last part of the construction, we show that it is possible to come up with a set of values for the links such that the best response inequalities are satisfied. Let the edge costs be as follows. In all the remaining formulae let c be any constant greater than 10. Let $\alpha(i) = 1$, and $\beta(i) = 2c^i(n-i) - c^i/2 - 3e(i)/2$. Also, $e(i) = \sum_{j<i} f(i,j) + \sum_{j<i} t(i,j) + 3/4$ for all i. Finally, for all pairs i, j, $f(i,j) = c^j$ and $t(i,j) = 2c^j$. Given these values, we can check that the inequalities above are satisfied for all $i < n$, and thereby we can have a run of best responses of length exponential in the number of players. ∎

6. Weighted Players. So far we have assumed that players sharing an edge e pay equal fractions of e's cost. We now consider a game with fixed edge costs where players have weights $w_i \geq 1$, and players' payments are proportional to their weight. More precisely, given a strategy $S = (S_1, \ldots, S_k)$, define W to be the total weight of all players, and let W_e be the sum of the weights of players using e. Then player i's payment for edge e will be $\frac{w_i}{W_e} c_e$.

Note that the potential function $\Phi(S)$ used for the unweighted version of the game is not a potential function once weights are added. In particular, in a weighted game, improving moves can increase the value of $\Phi(S)$, as this is no longer a congestion game. The following theorem uses a new potential function for a special class of weighted games.

THEOREM 6.1. *In a weighted game where each edge e is in the strategy spaces of at most two players, there exists a potential function for this game, and hence a Nash equilibrium exists.*

Proof: Consider the following potential function. For each edge e used by players i and j, define

$$\Phi_e(S) = \begin{cases} c_e w_i & \text{if player } i \text{ uses } e \text{ in } S \\ c_e w_j & \text{if player } j \text{ uses } e \text{ in } S \\ c_e \theta_{ij} & \text{if both players } i \text{ and } j \text{ use } e \text{ in } S \\ 0 & \text{otherwise} \end{cases}$$

where $\theta_{ij} = (w_i + w_j - \frac{w_i w_j}{w_i + w_j})$. For any edge e with only one player i, simply set $\Phi_e(S) = w_i c_e$ if i uses e and 0 otherwise. Define $\Phi(S) = \sum_e \Phi_e(S)$. We now simply need to argue that if a player makes an improving move, then $\Phi(S)$ decreases. Consider a player i and an edge e that player i joins. If the edge already supported another player j, then i's cost for using e is $c_e \frac{w_i}{w_i + w_j}$, while the change in $\Phi_e(S)$ is

$$c_e(w_i - \frac{w_i w_j}{w_i + w_j}) = c_e \frac{{w_i}^2}{w_i + w_j}.$$

Thus the change in potential when i joins e equals the cost i incurs, scaled up by a factor of w_i. In fact, it is easy to show the more general fact that when player i moves, the change in $\Phi(S)$ is equal to the change in player i's payments scaled up by w_i. This means that improving moves always decrease $\Phi(S)$, thus proving the theorem. ■

Note that this applies not only to paths, but also to the generalized model in which players select subsets from some ground set. The analogous condition is that no ground element appears in the strategy spaces of more than two players.

COROLLARY 6.2. *Any two-player weighted game has a Nash equilibrium.*

While the above potential function also implies a bound on the price of stability, even with only two players this bound is very weak. However, if there are only two players with weights 1 and $w \geq 1$, then we can show that the price of stability is at most $1 + \frac{1}{1+w}$, and this is tight for all w.

The following result shows the existence of Nash equilibria in weighted single commodity games.

THEOREM 6.3. *For any weighted game in which all players have the same source s and sink t, best response dynamics converges to a Nash equilibrium, and hence Nash equilibria exist.*

Proof: Start with any initial set of strategies S. For every $s - t$ path P define the marginal cost of P to be $c(P) = \sum_{e \in P} \frac{c_e}{W_e}$, where W_e is the sum of the weights of players using e in the state S. Observe that if player i currently uses path P, then i's payment is $w_i c(P)$. Define $P(S)$ to be a tuple of the values $c(P)$ over all paths P, sorted in increasing order. We want to show that the cheapest improving deviation of any player causes $P(S)$ to strictly decrease lexicographically.

Suppose that one of the best moves for player i is to switch paths from P_1 to P_2. Let $\mathcal{P}$ denote the set of paths that intersect $P_1 \cup P_2$. For any pair of paths P and Q, let $c_P(Q)$ denote the new value of $c(Q)$ after player i has switched to path P. To show that $P(S)$ strictly decreases lexicographically, it suffices to show that

$$\min_{P \in \mathcal{P}} c_{P_2}(P) < \min_{P \in \mathcal{P}} c(P). \tag{6.1}$$

Define $P' = \arg\min_{P \in \mathcal{P}} c(P)$. Since P_2 was i's best response, $c_{P_2}(P_2) \leq c_P(P)$ for all paths P. In particular, $c_{P_2}(P_2) \leq c_{P'}(P')$. We also know that $c_{P'}(P') \leq c(P')$, since in deviating to P', player i adds itself to some edges of P'. In fact, $c_{P'}(P') < c(P')$ unless $P' = P_1$. Assuming $P' \neq P_1$, we now have that $c_{P_2}(P_2) < c(P')$, which proves inequality 6.1. If $P' = P_1$, then since player i decided to deviate, $c_{P_2}(P_2) < c(P_1)$. Therefore, we once again have that $c_{P_2}(P_2) < c(P')$, as desired. ■

In the case where the graph consists of only 2 nodes s and t joined by parallel links, we can similarly show that any sequence of improving responses converge to a Nash equilibrium.

Weighted games with three or more players need not have a pure Nash equilibrium [8]. The following claim shows that, even when Nash equilibria do exist, the price of stability bounds from the unweighted case do not carry over.

THEOREM 6.4. *There are weighted games for which the price of stability is* $\Theta(\log W)$ *and* $\Theta(k)$.

An example exhibiting this is a modified version of the graph in Figure 1.1. Change the edge with cost $1 + \varepsilon$ to cost 1, and for all other edges with positive cost, set the new cost to be $\frac{1}{2}$. For $1 \leq i \leq k$ let player i have weight $w_i = 2^{i-1}$. Since each player has a greater weight than all smaller weight players combined, the only Nash equilibrium has cost $\frac{k}{2} = \Theta(\log W)$, while the optimal solution has cost 1.

7. Conclusions and Open Questions. For the Fair Connection Game, we showed that the price of stability is always at most $H(k)$, and that this is tight for directed graphs. However, the case for undirected graphs remains largely unresolved. The results of Section 4 show that, at least for two players, the price of stability for undirected graphs can be strictly better than that for directed. The largest lower bound that we know of for the price of stability in undirected networks is 12/7. The worst-case price of stability in undirected games could be constant, showing that fair sharing works very well in such games. See [1] for improved price of stability bounds in the undirected case, as well as Fiat et al. [20] for recent progress on a special case of this question.

Another open question is whether or not a good Nash equilibrium can be computed in polynomial time for a large number of players. One approach to solving this problem is to first compute an approximation to the centralized optimum, and then simulate best-response dynamics to reach a Nash equilibrium with cost at most $O(\log k)$ times that of the initial state. Unfortunately, Theorem 5.1 shows that arbitrary best-response dynamics can require exponential time to converge. On the other hand, it is possible that best responses can always be scheduled in a way that guarantees convergence in polynomial time. A weaker goal is to show how to compute a $(1+\varepsilon)$-approximate Nash equilibrium, where no player can decrease its cost by more than a $1 + \varepsilon$ factor, in polynomial time.

For the version of the Fair Connection Game with latencies instead of edge costs, the main challenge is to provide price of stability results for the general scenario, rather that just the single-source special case. See [7, 10] for recent results for the special case of linear latency functions.

Finally, for weighted games (Section 6) we have given only preliminary results. Recently, Chen and Roughgarden [8] proved that pure Nash equilibria need not exist in such games. They also showed that every weighted game has an $O(\log w_{\max})$-approximate Nash equilibrium that costs at most $O(\log W)$ times the centralized optimum (where $w_{\max}$ is the maximum weight and W is the sum of the weights, assuming that the minimum weight is 1). These bounds are nearly tight. But the following natural question is still open: what is the price of stability of approximate Nash equilibria in Weighted Fair Connection Games if all players share a common terminal?

REFERENCES

[1] A. Agarwal, M. Charikar. On the price of stability in undirected networks. Unpublished manuscript.

[2] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On Nash equilibria for a network creation game. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 89–98, 2006.

[3] E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 511–520, 2003.

[4] B. Awerbuch, Y. Azar, and L. Epstein. The price of routing unsplittable flow. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 57–66, 2005.

[5] V. Bala and S. Goyal. A non-cooperative model of network formation. *Econometrica*, 68(5):1181–1229, 2000.

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

[7] I. Caragiannis, M. Flammini, C. Kaklamanis, P. Kanellopoulos, and L. Moscardelli. Tight bounds for selfish and greedy load balancing. In *Proceedings of the 33rd Annual International Colloquium in Automata, Languages, and Programming (ICALP)*, volume 4051 of *Lecture Notes in Computer Science*, pages 311–322, 2006.

[8] H. Chen and T. Roughgarden. Network design with weighted players. In *Proceedings of the 18th ACM Symposium on Parallelism in Algorithms and Architextures (SPAA)*, pages 28–37, 2006.

[9] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, pages 72–81, 2006.

[10] G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proceedings of the 13th Annual Eurpoean Symposium on Algorithms (ESA)*, pages 59–70, 2005.

[11] G. Christodoulou and E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–73, 2005.

[12] J. Corbo and D. C. Parkes. The price of selfish behavior in bilateral network formation games. In *Proceedings of the 24th ACM Symposium on Principles of Distributed Computing (PODC)*, pages 99–107, 2005.

[13] J. R. Correa, A. S. Schulz, and N. E. Stier Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29(4):961–976, 2004.

[14] A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 287–296, 2002.

[15] A. Czumaj and B. Vöcking. Tight bounds for worst-case equilibria. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 413–420, 2002.

[16] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. J. Shenker. On a network creation game. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed*

Computing (PODC), pages 347–351, 2003.

[17] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[18] A. Fanelli, M. Flammini, G. Melideo, and L. Moscardelli. Multicast transmissions in non-cooperative networks with a limited number of selfish moves. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 4162 of *Lecture Notes in Computer Science*, pages 363–374, 2006.

[19] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[20] A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, and R. Shabo. On the price of stability for designing undirected networks with fair cost allocations. In *Proceedings of the 33rd Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 4051 of *Lecture Notes in Computer Science*, pages 608–618, 2006.

[21] H. Heller and S. Sarangi. Nash Networks with Heterogeneous Agents. Working Paper Series 2001, E-2001-1, Virginia Tech.

[22] Shai Herzog, Scott Shenker, Deborah Estrin. Sharing the "Cost" of Multicast Trees: An Axiomatic Analysis. IEEE/ACM Transactions on Networking, Dec. 1997.

[23] K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 364–372, 2001.

[24] R. Johari and J. N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[25] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.

[26] K. Kent and D. Skorin-Kapov. Population monoton cost allocation on mst's. In Operations Research Proceedings KOI, 43–48, 1996.

[27] E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *Lecture Notes in Computer Science*, pages 404–413, 1999.

[28] L. Libman and A. Orda. Atomic resource sharing in noncooperative networks. *Telecommunication Systems*, 17(4):385–409, 2001.

[29] D. Monderer, D. Samet. Variations of the Shapley Value. In Handbook of Game Theory Vol. III (Ed. R.J. Aumann and S. Hart), Elsevier Science, 2002.

[30] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[31] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *Electronic Proceedings of the 5th International Workshop on Peer-to-Peer Systems (IPTPS)*, 2006.

[32] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.

[33] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1/2):166–196, 2001.

[34] N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge, 2007.

[35] C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 749–753, 2001.

[36] R. W. Rosenthal. The network equilibrium problem in integers. *Networks*, 3(1):53–59, 1973.

[37] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal of Computer and System Sciences*, 67(2):341–364, 2003.

[38] T. Roughgarden. Stackelberg scheduling strategies. *SIAM Journal on Computing*, 33(2):332–350, 2004.

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

[40] S. Suri, C. Tóth, and Y. Zhou. Selfish load balancing and atomic congestion games. *Algorithmica*, 2007. To appear.

[41] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 416–425, 2002.