Enforcing Efficient Equilibria in Network Design Games via Subsidies*

John Augustine
Department of Computer Science and Engineering
Indian Institute of Technology Madras, Chennai, India

Ioannis Caragiannis
Department of Computer Engineering and Informatics
University of Patras & CTI, Greece

Angelo Fanelli
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore

Christos Kalaitzis
Department of Computer Engineering and Informatics
University of Patras & CTI, Greece

ABSTRACT

The efficient design of networks has been an important engineering task that involves challenging combinatorial optimization problems. Typically, a network designer has to select among several alternatives which links to establish so that the resulting network satisfies a given set of connectivity requirements and the cost of establishing the network links is as low as possible. The MINIMUM SPANNING TREE problem, which is well-understood, is a nice example.

In this paper, we consider the natural scenario in which the connectivity requirements are posed by selfish users who have agreed to share the cost of the network to be established according to a well-defined rule. The design proposed by the network designer should now be consistent not only with the connectivity requirements but also with the selfishness of the users. Essentially, the users are players in a so-called network design game and the network designer has to propose a design that is an equilibrium for this game. As it is usually the case when selfishness comes into play, such equilibria may be suboptimal. In this paper, we consider the following question: can the network designer enforce particular designs as equilibria or guarantee that efficient designs are consistent with users' selfishness by appropriately subsidizing some of the network links? In an attempt to understand this question, we formulate corresponding optimization problems and present positive and negative results.

*This work was partially supported by the grant NRF-RF2009-08 "Algorithmic aspects of coalitional games" and by the EC-funded STREP project EULER.

SPAA'12, June 25–27, 2012, Pittsburgh, Pennsylvania, USA.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Network design games, equilibria, subsidies

1. INTRODUCTION

Network design is a rich class of combinatorial optimization problems that model important engineering questions arising in modern networks. In an ideal scenario, a network designer that acts on behalf of a central authority is given an edge-weighted graph representing the potential links between nodes and their operation cost, and connectivity requirements between the nodes. The objective of the network designer is to compute a subgraph (the network to be established) of minimum cost that satisfies all connectivity requirements. Depending on the structure of the connectivity requirements, this definition leads to many optimization problems ranging from problems that are well-understood and efficiently solvable such as the MINIMUM SPANNING TREE to problems whose optimal solutions are even hard to approximate.

In this paper, we consider the scenario in which users are selfish and have agreed to a well-defined rule according to which they will share the cost of the network to be established. The connectivity requirements are now posed by the users; each user wishes to connect two specific nodes. A design should satisfy each connectivity requirement through a path connecting these two nodes in the established network. According to the particular cost sharing rule we consider, the corresponding user will then share the cost of each link in her path with the other users that use this link. Even though the network designer can still resort to the rich toolset of network design algorithms in order to propose a network of reasonable cost, this approach neglects the selfish behavior of the users. A user may not be satisfied with the current design since a different path that satisfies her connectivity requirement may cost her less. Then, she could unilaterally propose an alternative path that possibly includes links that were not in the proposal of the network designer. Other users could also act similarly and these negotiations compute the network to be established in a chaotic manner. The role of the network designer is almost canceled and, furthermore, it is not clear when the selfish users will reach an agreement (if they ever do) and, even if they do, whether this agreement will be really beneficial for the users as a whole, i.e., whether the total (or social) cost of the established network will be reasonable. So, the goal of the network designer is to propose a design (i.e., a network and, subsequently, a path to each user and an associated cost) that not only meets the connectivity requirements of the users but is also consistent with their selfish nature. Furthermore, the design should be efficient, i.e., the network to be established should have reasonable social cost. Essentially, the users are engaged as players in a non-cooperative strategic game, called a *network design game*, and the role of the network designer is to propose an efficient design that is an equilibrium of this game.

Typically, efficiency is not an easy goal when selfishness comes into play. This leads to the following question which falls within one of the main lines of research in *Algorithmic Game Theory*: how is the social cost affected by selfish behavior? The notion of the *price of anarchy* [13, 17] can quantify this relation. Expressed in the context of a network design game, it would be defined as the ratio of the social cost of the worst possible Nash equilibrium over the social cost of an optimal design. Hence, it is pessimistic in nature and (as its name suggests) provides a worst-case guarantee for conditions of total anarchy. Instead, the notion of the *price of stability* that was introduced by Anshelevich *et al.* [2] is optimistic in nature and quantifies how easy the job of the network designer is. It is defined as the ratio of the social cost of the best equilibrium over the cost of the optimal design and essentially answers the following question: what is the best one can hope from a design given that the players are selfish?

Unfortunately, the price of stability can be large which would mean that every design that is consistent with selfishness has high social cost. The central authority could then intervene in order to mitigate the impact of selfishness. One solution that seems natural would be to contribute to the social cost of the network to be established by partially subsidizing some of the links. According to this scenario, the network designer has to compute a design and decide which links in the established network should be subsidized by the central authority. The users will then share the unsubsidized portion of the cost of the network links they use. Essentially, they will be involved in a new network design game and the goal of the network designer should be to guarantee that the design and the subsidies computed induce an equilibrium for this new game. The problem becomes non-trivial when the central authority runs on a limited budget. What is the best design the network designer can guarantee given this budget? Alternatively, what is the minimum amount of subsidies sufficient in order to achieve a given social cost? Can optimality be achieved? Can the corresponding designs be computed efficiently?

Problem statement. In an attempt to understand these questions, we introduce and study two related optimization problems. In Stable Network Enforcement (SNE), we are given a network design game on a graph together with a particular target network T, and we wish to compute the minimum amount of subsidies that have to be put on the links of T so that the design is acceptable to the users. In Stable Network Design (SND), we are given a particular budget together with the input game, and we wish to compute a network T that satisfies the connectivity requirements and to assign an amount of subsidies to the links of T within the stipulated budget so that the design is acceptable to the users. The objective is to minimize the social cost of T. Besides the standard version of both problems, we also consider their *all-or-nothing* version in which a link can either be fully subsidized or not subsidized at all; this version captures restrictions that may arise in practice, e.g., when partially subsidizing network links is logistically infeasible.

Even though some of our results apply to general network design games, our emphasis is on a special class of network design games, called *broadcast games*. In such a game, there is a special node in the input graph called the *root*. There is one player associated with each distinct non-root node and her connectivity requirement is a path from her associated node to the root. A nice property of such games is that an optimal design is the solution of the Minimum Spanning Tree problem on the input graph and can be computed efficiently. Even in this seemingly simple case, as we will see, selfish behavior imposes challenging restrictions. Hence, broadcast games showcase the difficulties in solving SNE and SND that are due to selfishness.

Related work. Strategic games that arise from network design scenarios have received much attention in the *Algorithmic Game Theory* literature. The first related paper is probably [3]. The particular network design games that we consider in the current paper were introduced by Anshelevich *et al.* in [2]. An important observation made there is that network design games admit a potential function that was proposed by Rosenthal [18] for a broader class of games called congestion games. Using Rosenthal's potential and a simple but elegant argument, Anshelevich *et al.* [2] proved that the price of stability is at most $\mathcal{H}_n$, the n-th harmonic number, where n is the number of players. The $\mathcal{H}_n$ bound is known to be tight for directed networks only. For undirected networks, better bounds on the price of stability of $O(\log \log n)$ for broadcast games and $O(\log n / \log \log n)$ for generalizations known as multicast games are presented in [14] and [12], respectively. Still, the best lower bounds are only constant (e.g., see [5]). The papers [8] and [7] provide bounds on the quality of equilibria reached when players enter a multicast game one by one and play their best response and then (when all players have arrived) they concurrently play until an equilibrium is reached.

Another intriguing question is related to the complexity of computing equilibria in such games. In general, the problem was recently proved to be PLS-hard [20]. The corresponding hardness reduction does not apply to multicast or broadcast games. Unfortunately, the classical approach of minimizing the potential function that has been proved useful in the case of network congestion games [11] cannot be applied to multicast games; the authors of [8] prove that minimizing Rosenthal's potential function is NP-hard. Furthermore, in multicast games, computing an equilibrium of minimum social cost is NP-hard [20]. Approximate equilibria is the subject of [1].

Monetary incentives in strategic games have been considered in many different contexts. Most of the work in *Mechanism Design* (see [16] for an introduction) uses such incentives to motivate players to act truthfully. The (non-exhaustive) list also includes their use in *Cooperative Game Theory* in order to encourage coalitions of players reach stability [4] and as a means to stabilize normal form games [15]. However, the particular use of monetary incentives in the current paper is substantially different and aims at improving the performance of the system the game represents. In this direction, other tools such as taxes [10], Stackelberg strategies [19], and coordination mechanisms [9] have also been considered recently. An approach that is closer to ours has been followed in [6] where subsidies are used in multicast games; unlike our approach, the subsidies are collected as taxes from the players in order to guarantee efficient worst-case equilibria.

Overview of results and roadmap. In this paper we present the following results. First, we observe that SNE can be solved in polynomial time using linear programming; this observation applies to general instances of SNE. For instances of SNE with broadcast games, we present a much simpler LP in which the number of variables and constraints is linear and quadratic in the number of players, respectively. On the other hand, SND is proved to be NP-hard even for broadcast instances. In particular, detecting whether a minimum spanning tree can be enforced as an equilibrium without using any subsidies is NP-hard. This result implies that detecting whether the price of stability of a given broadcast game is 1 or not is NP-hard. In this direction, we have a stronger result: approximating the price of stability of a broadcast game is APX-hard. The last two statements significantly extend the NP-hardness result of [20] and indicate that, besides the rough estimates provided by the known bounds on the price of stability which hold for a broad class of games, the estimate the network designer can make about the most efficient designs of a particular broadcast game will also be rough. These results are presented in Section 3.

Next, we consider broadcast instances of SNE and the question of how much subsidies are sufficient and necessary in order to enforce a given minimum spanning tree as an equilibrium. We show that this can be done using a percentage of 37% of the weight of the minimum spanning tree as subsidies. The proof has two main components. First, we show how to prove this upper bound by decomposing the game into subgames with a significantly simpler structure than the original one. Second, in order to compute the subsidies in each subgame, we use a virtual approximation of the cost experienced by the players on the links of the network. We also demonstrate that our upper bound is tight: an amount of 37% of the minimum spanning tree weight as subsidies may be necessary for some simple instances. These results are presented in Section 4.

Surprisingly, in contrast to the standard case, we prove that the all-or-nothing version of SNE is hard to approximate within any factor even when restricted to instances with broadcast games. The corresponding proof is long and technically involved and indicates that the only approximation guarantee should bound the amount of subsidies as a constant fraction of the weight of the minimum spanning tree. Interestingly, we prove that significantly more subsidies may be necessary compared to the standard version of SNE. In particular, there are broadcast instances which require a percentage of 61% of the weight of the minimum spanning tree as subsidies in order to enforce it as an equilibrium. These results are presented in Section 5.

We begin with preliminary definitions and notation in Section 2 and conclude with interesting open problems in Section 6. Due to lack of space, some proofs have been omitted.

2. DEFINITIONS AND NOTATION

A *network design game* consists of an edge-weighted undirected graph $G = (V, E, w)$, a set N of n players, and a source-destination pair of nodes (s_i, t_i) for each player i. Each player wishes to connect her source to her destination and, in order to do this, she can select as a strategy any path T_i connecting s_i to t_i in G. The tuple $T = (T_1, T_2, ..., T_n)$ that consists of the strategies of the players (with one strategy per player) is called a *state*. We say that player i uses edge a in T if her strategy T_i contains a. With some abuse in notation, we also denote by T the set of edges included in strategies $T_1, T_2, ..., T_n$ as well as the subgraph of G induced by these edges. We say that an edge $a \in E$ is *established* if at least one player uses edge a. Consider such an edge a and let $n_a(T)$ be the number of players whose strategies in T contain a. Throughout the paper, we also use the notation $n^i_a(T)$ to denote whether player i uses edge a ($n^i_a(T) = 1$) or not ($n^i_a(T) = 0$). Each player i in N experiences a cost of $\texttt{cost}_i(T) = \sum_{a \in T_i} \frac{w_a}{n_a(T)}$, i.e., the weight of each established edge is shared as cost among the players using it.

The state T is called a (*pure Nash*) *equilibrium* if no player has an incentive to unilaterally deviate from T in order to decrease her cost, i.e., for each player i and possible strategy T'_i that connects the source-destination pair (s_i, t_i) in G, it holds that $\texttt{cost}_i(T) \leq \texttt{cost}_i(T_{-i}, T'_i)$. The notation T_{-i}, T'_i denotes the state in which player i uses strategy T'_i and the remaining players use their strategies in T. Throughout the paper, we denote by $\texttt{wgt}(A)$ the total weight of the set of edges A in G, i.e., $\texttt{wgt}(A) = \sum_{a \in A} w_a$. The quality of a state is measured by the total weight of the established edges. Since the weight of each established edge is shared as cost among the players that use it, the quality of a state coincides with the total cost experienced by all players, i.e., $\texttt{wgt}(T) = \sum_i \texttt{cost}_i(T)$. The *price of stability* of a network design game is simply the ratio of the weight of the edges established in the best equilibrium over the optimal cost among all states of the game.

Given an edge-weighted graph $G = (V, E, w)$, a *subsidy assignment* b is a function that assigns a subsidy $b_a \in [0, w_a]$ to each edge $a \in E$. The cost of a subsidy assignment is simply the sum of the subsidies on all edges of G, i.e., $\sum_{a \in E} b_a$. We use the term *all-or-nothing* to refer to subsidies that are constrained so that $b_a \in \{0, w_a\}$ for each edge $a \in E$. Given a set of edges A in G, we use the notation $b(A)$ in order to refer to the total amount of subsidies assigned to the edges of A in the subsidy assignment b, i.e., $b(A) = \sum_{a \in A} b_a$. We refer to $b(E)$ as the cost of the subsidy assignment b. Given a network design game on a graph G and a subsidy assignment b on the edges of G, we use the term *extension* of the original game with subsidies b in order to refer to the network design game on graph G (with the same players and strategy sets as in the original game) with the only difference being that the cost of a player at a state T is now $\texttt{cost}_i(T; b) = \sum_{a \in T_i} \frac{w_a - b_a}{n_a(T)}$. When a particular state T is an equilibrium of the extension of the original game with subsidies b, we say that the subsidy assignment b *enforces* T as an equilibrium in the extension of the original game.

An instance of the Stable Network Design problem (SND) consists of a network design game on a graph G, a budget B, and a positive number K. The question is whether there exists a subsidy assignment b of cost at most B on the edges of G so that a subgraph of G of total weight at most K is an equilibrium for the extension of the original game with subsidies b. An instance of the Stable Network Enforcement problem (SNE) consists of a network design game on a graph G, a budget B, and a state T. The question is whether there exists a subsidy assignment of cost at most B on the edges of G so that b enforces T as an equilibrium on the extension of the original game with subsidies b. Note that the subsidy assignment does not need to put any subsidies to edges not in T. In the integral versions of SNE and SND, the subsidy assignment in question is all-or-nothing. Of course, optimization versions of the above problems are natural. For example, in an optimization version of SNE, we are given the network design game on a graph G and a state T, and we require the subsidy assignment in question to be of minimum cost.

Broadcast games are special cases of network design games. In a broadcast game, the graph G has exactly $n+1$ nodes; all players have the same destination node, which is called the *root* and is denoted by r, and distinct non-root nodes as sources. In such games, we refer to a player with a source node u as the player associated with node u (and use u to identify the player). Clearly, any state T in such a game spans all nodes of G and a minimum spanning tree is a state that minimizes the total cost experienced by the players. Given any spanning tree T and a non-root node u, we denote by T_u the path from u to r in T. In broadcast games, we mostly consider equilibria that are spanning trees.

3. THE COMPLEXITY OF SNE AND SND

We begin the presentation of our results with the following observation.

Theorem 1. Stable Network Enforcement *is in P.*

This theorem applies to general instances of SNE and follows by expressing the problem using linear programming; the corresponding LPs have a large number of variables or constraints. A more detailed discussion will appear in the final version. In the following, we present a much simpler LP that applies to broadcast instances.

Given a broadcast game on an edge-weighted graph $G = (V, E, w)$ (with a root node r and n non-root nodes) and a spanning tree T of G, our LP solves the optimization version of the problem by computing a subsidy assignment b of minimum cost so that T is an equilibrium in the extension of the original game with subsidies b. The subsidies in question are the variables of the LP. We remark that, even though we only need to use variables for the subsidies on the edges of T, we assume that b_a is defined for each edge a of E in order to simplify the presentation; it should be clear that, in any optimal solution of the linear program below, $b_a = 0$ for each edge $a \in E \setminus T$. The variables are constrained so that $b_a \in [0, w_a]$. There are extra constraints that require that no player associated with a node u has an incentive to change her strategy in T and use an edge (u, v) that does not belong to T and the path from v to r in T. The corresponding LP is:

$$\begin{aligned} \min \quad & \sum_{a \in E} b_a \qquad (1) \\ \text{s.t.} \quad & \forall u \in V \setminus \{r\}, v \in V \text{ such that } (u,v) \in E \setminus T, \\ & \sum_{a \in T_u} \frac{w_a - b_a}{n_a(T)} \le w_{(u,v)} + \sum_{a \in T_v} \frac{w_a - b_a}{n_a(T) + 1 - n_a^u(T)} \\ & \forall a \in E, 0 \le b_a \le w_a \end{aligned}$$

The above LP has n variables and $O(|E|)$ constraints. Observe that the first set of constraints just requires that the players have no incentive to deviate to very specific paths. Hence, the correctness of the LP (i.e., its equivalence with the optimization version of SNE) is not obvious and is given by the following lemma; the proof is omitted.

Lemma 2. *Consider an instance of* Stable Network Enforcement *consisting of a broadcast game on a graph G and a state T. A subsidy assignment b enforces T as an equilibrium in the extension of the broadcast game in G if and only if the constraints of LP (1) are satisfied.*

Next, we prove that the restriction of SND to broadcast instances is NP-hard. The hardness proof uses instances of SND with budget equal to zero with target equilibrium weight equal to the weight of the minimum spanning tree. In our reduction, there are many different minimum spanning trees but it is hard to detect whether there is one that is an equilibrium in the corresponding broadcast game.

Theorem 3. *Given an instance of* Stable Network Design *consisting of a broadcast game on a graph G, budget B, and a positive number K, it is NP-hard to decide whether there exists a subsidy assignment b of cost at most B so that the extension of the game with subsidies b has a tree of weight at most K as an equilibrium. Moreover, it is NP-hard even when B is set to zero.*

We will first describe a gadget that is used in the proof of Theorem 3; we call it the `Bypass` gadget of capacity κ. The gadget is shown in Figure 1. Let ℓ be the minimum positive integer such that $\mathcal{H}_{\kappa+\ell} - \mathcal{H}_\kappa > 1$. The `Bypass` gadget consists of a root node r connected to one end of a path of ℓ nodes formed with edges of unit weight. We call this the *basic path* of the `Bypass` gadget. The node c on the far end of the path from r is called the *connector node*. There is an edge from c to r of weight $\mathcal{H}_{\kappa+\ell} - \mathcal{H}_\kappa$, which we call the *bypass edge*.

Suppose this gadget is connected to a subgraph S of β nodes as shown in Figure 1. For the moment, we are not concerned with how the nodes in S are connected to each other. Consider the instance of SNE consisting of the broadcast game on the graph G of Figure 1, budget $B = 0$, and let T be a minimum spanning tree of G. Note that T does not include the bypass edge; it includes all edges in the basic path from c to r instead.

Lemma 4. *If $\beta < \kappa$, then the player associated with node c has an incentive to deviate from her strategy in T and use the bypass edge. Otherwise, no player associated with a node in the basic path has any incentive to deviate from T.*

Proof. Regardless of how the players associated with nodes in the subgraph S are routed, since S is connected to the `Bypass` gadget through node c, there are $\beta+1$ players

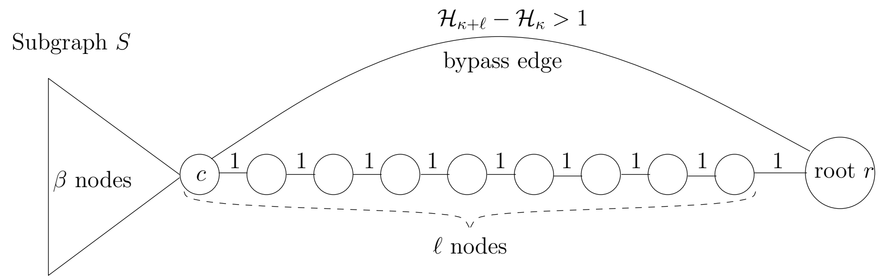

Figure 1: The `Bypass` gadget with capacity κ.

that need to use a path from c to r. Let us focus on the player associated with node c. If she and all the β players from S take the basic path, then her cost will be $\sum_{i=1}^{\ell} \frac{1}{\beta+i} = \mathcal{H}_{\beta+\ell} - \mathcal{H}_{\beta}$. If $\beta < \kappa$, then $\mathcal{H}_{\kappa+\ell} - \mathcal{H}_{\kappa} < \mathcal{H}_{\beta+\ell} - \mathcal{H}_{\beta}$, and therefore, the player associated with node c has an incentive to deviate to the bypass edge. On the other hand, if $\beta \geq \kappa$, then $\mathcal{H}_{\kappa+\ell} - \mathcal{H}_{\kappa} \geq \mathcal{H}_{\beta+\ell} - \mathcal{H}_{\beta}$ and any player associated with a node in the basic path experiences a cost of at most $\mathcal{H}_{\beta+\ell} - \mathcal{H}_{\beta}$ in T. Hence, no such player has an incentive to deviate from T. □

Proof of Theorem 3. We show that the problem is NP-hard even when we consider the special case where $B = 0$ and K equals the weight of the minimum spanning tree of the input graph G. In other words, given a broadcast game on a graph G with root node r, we ask: does this game have a minimum spanning tree of G as an equilibrium? We use a reduction from Bin Packing.

We use a stricter form of Bin Packing defined as follows. We are given a set of n items indexed by $i \in \{1, 2, \ldots, n\}$. The size of each item i is a positive even integer denoted by s_i. Since bin packing is strongly NP-hard, we assume that s_i is bounded by a polynomial in n. We are also given a set of k bins indexed by $j \in \{1, 2, \ldots, k\}$, each of even integer capacity C, which we assume to be at least as large as $\max_i s_i$. We furthermore assume that $\sum_{i=1}^{n} s_i = kC$. We ask whether each item can be allocated to one of the k bins so that the total size of items in each bin is exactly C. Our definition of Bin Packing is somewhat stricter than the conventional definition in which the capacity of bins and the size of the items is not restricted to be even and bins are not required to be filled to the brim. However, we note that it is quite straightforward to see that this restricted version of the problem can be reduced from the conventional version by first adding a suitable number of unit-sized items and then doubling the size of all items and the capacity of all bins. The number of additional items is upper-bounded by the total capacity of the bins. Therefore, our restriction of Bin Packing is also strongly NP-hard.

Given a restricted instance of Bin Packing, we now construct an instance of SNE as follows. For each item i of size s_i, we create a star graph with one center node which we denote by x_i and $s_i - 1$ leaves. The edges connecting the leaves to the center node of the star have zero weight. Let X be the set of center nodes. For each bin j, construct a `Bypass` gadget with capacity $\kappa = C$. Again, let ℓ be the number of unit-weight edges in the basic path of each `Bypass` gadget. Recall that ℓ is the minimum positive integer such that $\mathcal{H}_{C+\ell} - \mathcal{H}_C > 1$; this implies that ℓ is linear in C. We denote the connector node in the gadget corresponding to bin j by c_j. Let χ be the set of all connector nodes. We connect sets χ and X by a complete bipartite edge set with edges having weight $2(\mathcal{H}_{C+\ell} - \mathcal{H}_C)$. Observe that any minimum spanning tree of G consists of the $k\ell$ unit-weight edges in the `Bypass` gadgets, the zero-weight edges connecting the leaves to their star center, and n edges that connect nodes of χ to nodes of X so that each node of X is connected to exactly one node of χ. We set K to be the weight of the minimum spanning tree, i.e., $K = k\ell + 2n(\mathcal{H}_{C+\ell} - \mathcal{H}_C)$.

We claim that a minimum spanning tree T_{ne} of G is an equilibrium for the broadcast game on G if and only if the Bin Packing instance has a solution. We prove this claim in both directions.

Let T_{ne} be a minimum spanning tree that is an equilibrium. Let $\beta_j + 1$ be the number of nodes in the subtree of T_{ne} rooted at c_j. From Lemma 4, we know that since T_{ne} is an equilibrium, for all j, it holds that $\beta_j \geq C$. However, we also know from the properties of the Bin Packing instance and the construction of graph G that $\sum_{j=1}^{k} \beta_j = \sum_{i=1}^{n} s_i = kC$. Clearly, it follows that for all j, we have $\beta_j = C$. Therefore, the allocation of item i to bin j whenever x_i is connected to c_j will lead to a solution for the Bin Packing instance since the total size of these items is exactly $\beta_j = C$.

To show the other direction, let us suppose that we have a solution to the Bin Packing instance. We construct a minimum spanning tree T_{ne} as follows. T_{ne} contains the edges from the leaves to the corresponding star center, the basic paths from the connector nodes to the root node r, and the edge (x_i, c_j) for each item i that is allocated to bin j. Note that, for $j = 1, ..., k$, the number of nodes in the subtree of T_{ne} rooted at c_j is exactly C. So, any player associated with a node in a basic path experiences a cost of at most $\mathcal{H}_{C+\ell} - \mathcal{H}_C$ in T_{ne}. Furthermore, observe that each edge of T_{ne} between nodes of χ and X is used by at least two players in T_{ne}. So, any player associated with a node in a star experiences a cost of at least $2(\mathcal{H}_{C+\ell} - \mathcal{H}_C)$. Hence, no player has an incentive to deviate to a path that includes a node of χ that she does not use in T_{ne}. Any such path would include an edge of weight $2(\mathcal{H}_{C+\ell} - \mathcal{H}_C)$ between a node in χ and a node in X that is used only by that player. So, for each j, the C players associated with nodes in the subtree of c_j in T_{ne} have no incentive to deviate to a path that does not use node c_j. By Lemma 4, player c_j (and, consequently, all players that have node c_j in their path to r in T_{ne}) has no incentive to deviate to the bypass edge connecting c_j to r. This holds for any other node in the basic path of a `Bypass` gadget as well. Therefore, it follows that T_{ne} is an equilibrium. □

Note that the proof of Theorem 3 essentially implies that deciding whether the price of stability of a given broadcast

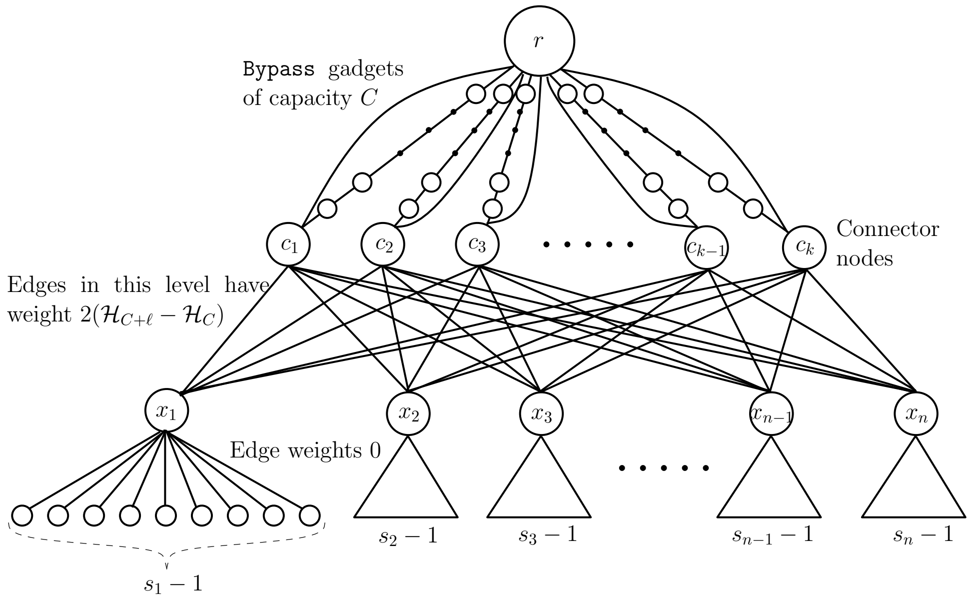

Figure 2: The graph G constructed from an instance of Bin Packing**.**

game is 1 or not is NP-hard. The next statement (proof omitted) provides an even stronger negative result. It implies that given instances of SND consisting of a broadcast game on a graph G and a budget B, it is NP-hard to approximate within a factor better than 571/570 the minimum weight among all equilibria in any extension of the original game with subsidies of cost at most B.

Theorem 5. *Approximating the price of stability of a broadcast game within a factor better than* 571/570 *is NP-hard.*

4. BOUNDS ON THE AMOUNT OF SUBSIDIES

In this section, we provide tight bounds on the amount of subsidies sufficient in order to enforce a minimum spanning tree as an equilibrium in the extension of the original broadcast game. The result is expressed as a constant fraction of the weight of the minimum spanning tree. We first prove our upper bound which is more involved. The proof uses two key ideas: first, the input SNE instance is appropriately decomposed into subinstances of SNE which have a significantly simpler structure. Our decomposition is such that the desired bound has to be proved for the subinstances; in order to do so, we use a second idea and exploit a virtual cost function that upper-bounds the actual cost experienced by the players in the extension of the game (in the subinstances) with subsidies. The main property of this virtual cost function that simplifies the analysis considerably is that the total amount of subsidies necessary depends only on the weight of the tree (and not on its structure).

Theorem 6. *Given an instance of* Stable Network Enforcement *consisting of a broadcast game on a graph G and a minimum spanning tree T of G, there is a subsidy assignment b of cost at most $\mathtt{wgt}(T)/e$ that enforces T as an equilibrium of the extension of the game with subsidies b, where e is the basis of the natural logarithm.*

Proof. We decompose the graph G into copies G^1, G^2, ..., G^k so that the following properties hold:

- G^j has the same set of nodes and set of edges with G.
- The edge weights in G^j belong to $\{0, c_j\}$ for some $c_j > 0$.
- If the weight of an edge a in G^j is non-zero, then the weight of a is non-zero in each of the copies $G^1, ..., G^{j-1}$ of G.
- The weight of each edge in G is equal to the sum of its weights in the copies of G.

The decomposition proceeds as follows. Let c_1 be the minimum non-zero weight among the edges of G. We construct a copy G^1 of G (i.e., with the same set of nodes and set of edges) and with edge weights equal to zero if the corresponding edge of G has zero weight and equal to c_1 otherwise. Then, we decrease each non-zero edge weight by c_1 in G and proceed in the same way with the definition of the edge weights in the copy G^2, and so on. We denote by k the number of copies of G that have some edge of non-zero weight. Note that c_k may be infinite if G contains edges of infinite weight, but k is upper bounded by the number of edges in G. Clearly, the weight of an edge in the original graph is the sum of its weights in the copies of G.

We denote by T^j the spanning tree of G^j that has the same set of edges with T. We first observe that T^j is a minimum spanning tree of G^j. Assume that this is not the case; then, there must be an edge a_1 with zero weight in G^j that does not belong to T^j such that some edge a_2 of the edges of T^j with which a_1 forms a cycle has non-zero weight c_j. By the definition of our decomposition phase, this implies that a_2 has higher weight than a_1 in G. This means that we could remove a_2 from T and include a_1 in order to obtain a spanning tree with strictly smaller weight, i.e., T would not be a minimum spanning tree.

Now, in order to compute the desired subsidy assignment that enforces T as an equilibrium in the extension of the broadcast game in G, we will exploit appropriate subsidy assignments for the broadcast games in each copy of G. We have the following lemma; its proof exploits a virtual cost function in order to compute the subsidy assignment b^j.

LEMMA 7. *Let $c_j > 0$. Consider a broadcast game on a graph G^j whose edges have weights in $\{0, c_j\}$ and let T^j be a minimum spanning tree of G^j. Then, there is a subsidy assignment b^j of cost at most $\texttt{wgt}(T^j)/e$ that enforces T^j as an equilibrium in the extension of the game with subsidies b^j.*

PROOF. We call edges of weight 0 and c_j *light* and *heavy* edges, respectively. We also call a player associated with a node v a *light* player if the weight of the edge connecting v to its parent in T^j is zero; otherwise, we call v a *heavy* player. We denote by m_a the number of heavy players which use edge a. Clearly, $m_a \leq n_a(T^j)$.

We will introduce a *virtual cost* associated with each edge of T^j in order to upper-bound the contribution of the edge to the real cost experienced by each player that uses the edge in T^j in the extension of the game with subsidies. In particular, given subsidies y_a assigned to the heavy edge a with $y_a \in [0, c_j]$, we define the virtual cost of edge a as $\texttt{vc}(a, y_a) = c_j \ln \frac{m_a}{m_a - 1 + y_a/c_j}$. The virtual cost of a light edge is always zero. The next claim follows by the definition of the virtual cost.

CLAIM 8. *For any heavy edge a with subsidies y_a, it holds that $\texttt{vc}(a, y_a) \geq \frac{c_j - y_a}{n_a(T^j)}$.*

DEFINITION 9. *Consider a path q in T^j and a subsidy assignment y on the edges of T^j. We say that y is such that subsidies are packed on the least crowded heavy edges of q if $y_a < c_j$ for a heavy edge a implies that $y_{a'} = 0$ for every heavy edge a' of q with $m_{a'} > m_a$.*

We extend the notation of virtual cost so that $\texttt{vc}(q, b^j)$ denotes the sum of the virtual cost of the edges of a path q in T^j under the subsidy assignment b^j. The following claim follows by the definitions and will be very useful later.

CLAIM 10. *Consider a path q and denote by q' the set of heavy edges of q and a subsidy assignment y. If $\cup_{a\in q'}\{m_a\}$ consists of the $|q'|$ consecutive integers $t - |q'| + 1, t - |q'| + 2, ..., t$, then the virtual cost of path q when subsidies are packed on its least crowded heavy edges is $\texttt{vc}(q, y) = c_j \ln \frac{t}{t - |q'| + y(q)/c_j}$.*

PROOF. Recall that the only edges that contribute to the virtual cost of q are the heavy edges in q'. If $y(q) = 0$ (i.e., no subsidies are put on the edges of q'), the virtual cost is

$$\begin{aligned}\texttt{vc}(q, y) &= \sum_{a\in q'} \texttt{vc}(a, y_a)\\ &= \sum_{a\in q'} c_j \ln \frac{m_a}{m_a - 1 + y_a/c_j}\\ &= \sum_{i=t-|q'|+1}^{t} c_j \ln \frac{i}{i-1}\\ &= c_j \ln \frac{t}{t - |q'| + y(q)/c_j}.\end{aligned}$$

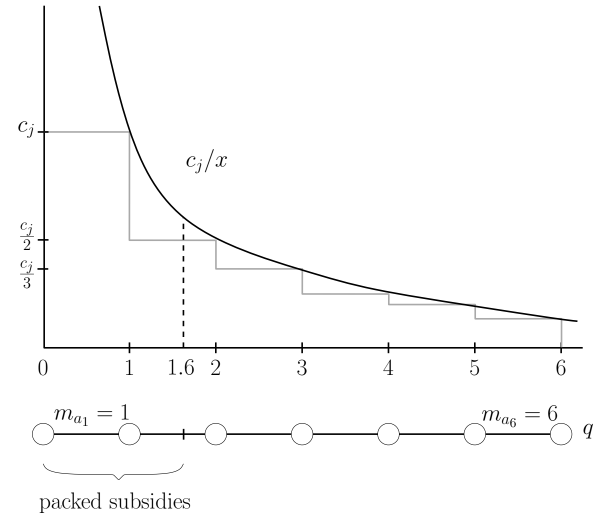

Figure 3: A visualization of the virtual cost in a path q, with 6 heavy edges and $\cup_{a\in q}\{m_a\} = \{1, 2, ..., 6\}$, when subsidies are packed on its less crowded edges. The leftmost edge and a fraction of 60% of the second leftmost one have been subsidized. The virtual cost $\ln \frac{6}{1.6}$ (see Claim 10) is the area to the right of the dashed line that is below the black line. The real cost experienced by the player associated with the far left node is the area to the right of the dashed line that is below the grey line.

The first two equalities follow by the definition of the virtual cost, the third one follows since $\cup_{a\in q'}\{m_a\} = \{t - |q'| + 1, t - |q'| + 2, ..., t\}$ and $y_a = 0$, and the last one is obvious.

We now consider the case $y(q) > 0$. Since subsidies are packed on the least crowded heavy edges of q, there must be a heavy edge $a \in q'$ such that $y_a > 0$ so that $y_{a'} = 0$ for each heavy edge a' with $m_{a'} > m_a$ and $y_{a''} = c_j$ for each heavy edge a'' with $m_{a''} < m_a$. Let $q'_1 = \{a' \in q' : y_{a'} = 0\}$ and $q'_2 = q' \setminus (q'_1 \cup \{a\})$. Observe that the edges of q'_2 and the light edges of q do not contribute to the virtual cost of q. Hence,

$$\begin{aligned}\texttt{vc}(q, y) &= \sum_{a'\in q'_1} \texttt{vc}(a', y_{a'}) + \texttt{vc}(a, y_a)\\ &= \sum_{a'\in q'_1} c_j \ln \frac{m_{a'}}{m_{a'} - 1} + c_j \ln \frac{m_a}{m_a - 1 + y_a/c_j}\\ &= c_j \sum_{i=t-|q'_1|+1}^{t} \ln \frac{i}{i-1} + c_j \ln \frac{t - |q'_1|}{t - |q'_1| - 1 + y_a/c_j}\\ &= c_j \ln \frac{t}{t - |q'_1| - 1 + y_a/c_j}\\ &= c_j \ln \frac{t}{t - |q'| + y(q)/c_j}.\end{aligned}$$

The first two equalities follow by the definition of the virtual cost, the third one follows since the definition of q'_1 implies that $\cup_{a'\in q'_1}\{m_{a'}\} = \{t - |q'_1| + 1, t - |q'_1| + 2, ..., t\}$ and $m_a = t - |q'_1|$, the fourth equality is obvious, and the last one follows since $y(q) = y_a + |q'_2|c_j$ and $|q'| = |q'_1| + |q'_2| + 1$. □

Figure 3 provides a visualization of the virtual cost in a path when subsidies are packed on its less crowded heavy edges.

Now, we compute the subsidy assignment b^j that assigns no subsidies to the light edges and subsidies to the heavy edges of T^j as follows. Denote by L the set of leaf-nodes of T^j such that the path T^j_u connecting such a leaf-node u to the root node r in T^j contains at least one heavy edge. For each leaf-node u of L, we pack subsidies to the least crowded heavy edges of T^j_u so that the virtual cost on the path T^j_u is exactly c_j. In particular, let S be the set of edges of T^j defined as follows. Denote by $p(v)$ the parent of node v in T^j. A heavy edge $(v, p(v))$ belongs to S if $\mathtt{vc}(T^j_{p(v)}, 0) < c_j$ and $\mathtt{vc}(T^j_v, 0) \geq c_j$. Observe that the set S disconnects the leaves of L from the root node. Indeed, if this was not the case, there would be a heavy edge that is used by exactly one heavy player and is not assigned any subsidies; by the definition of the virtual cost, its virtual cost would be infinite. All heavy edges that are on the side of the partition together with the root node are assigned zero subsidies; the heavy edges on the other side of the partition are assigned subsidies of c_j and do not contribute to the virtual cost of the paths they belong to. An edge $a = (v, p(v))$ of S is assigned subsidies b^j_a with

$$b^j_a = c_j \left(1 - m_a \left(1 - \exp\left(\frac{\mathtt{vc}(T^j_{p(v)}, 0)}{c_j} - 1\right)\right)\right).$$

This definition implies that $\mathtt{vc}(T^j_{p(v)}, 0) + \mathtt{vc}(a, b^j_a) = c_j$. In this way, we guarantee that the virtual cost of any path to the root node r is at most c_j if it contains at least one heavy edge and zero otherwise.

We will now show that, given the subsidies we have assigned to the edges of T^j, no player has an incentive to deviate from her path to the root in T^j. Consider the player associated with a node u and recall that the definition of subsidies and Claim 8 guarantee that the cost experienced by the player when she uses T^j_u is at most c_j. Let q_u be a path from u to r in G^j that is different than T^j_u and consider the edges of q_u that do not belong to T^j (since $q_u \neq T^j_u$ there is at least one such edge). If any such edge has weight c_j, this means that, by deviating to q_u, the player associated with node u would experience a cost of at least c_j and, hence, has no incentive to do so. So, in the following we assume that the edges of q_u that do not belong to T^j have zero weight. Now, consider the subgraph H of G^j induced by the edges in the paths T^j_u and q_u. Let C be a cycle of H. It consists of edges of T^j and edges not belonging to T^j that have zero weight. This implies that all edges in C have zero weight, otherwise we could replace an edge of C that belongs to T^j (and has non-zero weight) with an edge of C that does not belong to T (which has zero weight) and obtain a spanning tree of G^j with strictly smaller weight than T^j; this would contradict the assumption that T^j is a minimum spanning tree of G^j. So, all edges of T^j_u that are contained in a cycle of H have zero weight. The remaining edges of T^j_u are also used by q_u. We conclude that the total cost experienced by the player associated with node u is the same no matter whether she uses path T^j_u or q_u and, hence, she has no incentive to deviate from path T^j_u to q_u.

We will now show that the total amount of subsidies put on the edges of T^j in this way is exactly $\mathtt{wgt}(T^j)/e$. In order to show this, we will show that the total amount of subsidies put on the edges of T^j equals the total amount of subsidies put by the same procedure on the edges of another tree $\bar{T}$ that consists of a single path from the root that spans all the nodes and has the same number of heavy edges as the original one. As an intermediate step, consider two edges $g_1 = (v_1, p(v_1))$ and $g_2 = (v_2, p(v_2))$ of S such that the least common ancestor u of nodes v_1 and v_2 in T^j has the largest depth. We denote by h_1 and h_2 the number of heavy edges in the subtrees of v_1 and v_2, respectively, and by q_1 and q_2 the paths connecting u to v_1 and v_2 in T^j, respectively. Also, denote by q'_1 (resp. q'_2) the subset of q_1 (resp. q_2) consisting of heavy edges. Assume that the virtual cost of the path in T^j from r to u is ℓ for some $\ell \in [0, c_j)$; then, the virtual cost of the paths q_1 and q_2 is exactly $c_j - \ell$. Denote by $b^j_{g_1}$ and $b^j_{g_2}$ the subsidies assigned to edges g_1 and g_2 by the above procedure, respectively. Since the edges g_1 and g_2 are selected so that their least common ancestor u has the largest depth, the edges in the path q_1 are not used by any heavy player different than those in the subtree of T^j rooted at v_1. Similarly, the edges in the path q_2 are not used by any heavy player other than those in the subtree of T^j rooted at v_2. Hence, both paths q_1 and q_2 satisfy the condition of Claim 10 above in the sense that $\cup_{a \in q'_1}\{m_a\} = \{h_1+1, h_1+2, ..., h_1+|q'_1|\}$ and $\cup_{a \in q'_2}\{m_a\} = \{h_2+1, h_2+2, ..., h_2+|q'_2|\}$, respectively. Hence, we can express the virtual cost of paths q_1 and q_2 as

$$\mathtt{vc}(q_1, b^j) = c_j \ln \frac{h_1 + |q'_1|}{h_1 + b^j_{g_1}/c_j} = c_j - \ell$$

and

$$\mathtt{vc}(q_2, b^j) = c_j \ln \frac{h_2 + |q'_2|}{h_2 + b^j_{g_2}/c_j} = c_j - \ell,$$

respectively. Equivalently, we have $h_1 + |q'_1| = \exp(1 - \ell/c_j)(h_1 + b^j_{q_1}/c_j)$ and $h_2 + |q'_2| = \exp(1 - \ell/c_j)(h_2 + b^j_{g_2}/c_j)$. By summing these last two equalities, we obtain that $h_1 + h_2 + |q'_1| + |q'_2| = \exp(1 - \ell/c_j)(h_1 + h_2 + (b^j_{g_1} + b^j_{g_2})/c_j)$ which implies

$$c_j \ln \frac{h_1 + h_2 + |q'_1| + |q'_2|}{h_1 + h_2 + (b^j_{g_1} + b^j_{g_2})/c_j} \quad = \quad c_j - \ell. \qquad (2)$$

Now, consider the following transformation of T^j to another tree T'. The only change is performed in the paths q_1, q_2 and all subtrees of their nodes besides node u. We replace all these edges in T^j with a path q originating from u and spanning all the nodes in q_1 and q_2 and their subtrees so that exactly $h_1 + h_2 + |q'_1| + |q'_2|$ heavy edges are used. Let q' be the set of heavy edges in q. We pack a total amount $c_j(h_1 + h_2) + b^j_{g_1} + b^j_{g_2}$ of subsidies (i.e., the same total amount of subsidies used in the heavy edges of q_1, q_2 and in the subtrees of nodes v_1 and v_2 in T^j) on the least crowded heavy edges of path q while the assignment of subsidies on the other heavy edges of T' is the same as in the corresponding edges in T^j. Now, the path q satisfies the condition of the Claim 10 above in the sense that $\cup_{a \in q'}\{m_a\} = \{1, ..., h_1 + h_2 + |q'_1| + |q'_2|\}$. Hence, the virtual cost of path q when a total amount $c_j(h_1 + h_2) + b^j_{g_1} + b^j_{g_2}$ of subsidies is packed on its least crowded heavy edges is the one at the left hand side of equality (2) and is exactly $c_j - \ell$ while the virtual cost of the path from the root to u in T' is not affected by our transformation (the number of heavy players in the subtree of node u stays the same after the transformation) and is equal to ℓ. Hence, we have transformed T^j to T' so that the same total amount of subsidies is used and guarantees that any path from the root to a node has virtual cost at most c_j. By executing the same transformation in T' repeatedly, we end up with a tree $\bar{T}$ which consists of a path $\bar{q}$ spanning all the nodes and has the same number of heavy edges as the original tree T^j (and, obviously, the same total weight). Let $\bar{q}'$ be the set of heavy edges in $\bar{q}$. The transformation guarantees that by packing the original total amount of subsidies on the least crowded heavy edges of $\bar{q}$, we have that its virtual cost is exactly c_j. Also, note that $\cup_{a\in\bar{q}'}\{m_a\} = \{1, 2, ..., |\bar{q}'|\}$ and, by Claim 10, the virtual cost of path $\bar{q}$ when a total amount $b(T^j)$ of subsidies is packed on its least crowded heavy edges is $c_j \ln \frac{|\bar{q}'|c_j}{b(T^j)} = c_j$. This implies that the total amount of subsidies in the original tree is $b(T^j) = |\bar{q}'|c_j/e = \mathtt{wgt}(T^j)/e$. □

Now, for each copy G^j of G, we use the procedure in the proof of Lemma 7 to compute a subsidy assignment b^j so that the tree T^j is an equilibrium for the extension of the broadcast game on the graph G^j with subsidies b^j. For the original game on the graph G, we assign an amount of $b_a = \sum_{j=1}^k b_a^j$ as subsidies to edge a (i.e., equal to the total amount of subsidies assigned to a for each copy of G). By the properties of our decomposition and Lemma 7, the total amount of subsidies is $\sum_{j=1}^k \mathtt{wgt}(T^j)/e = \mathtt{wgt}(T)/e$.

It remains to show that T is an equilibrium for the original broadcast game. Consider a node u of G and let q_u be any path connecting u with r in G. The cost experienced by the player associated with node u in T is

$$\begin{aligned}
\mathtt{cost}_u(T;b) &= \sum_{a\in T_u} \frac{w_a - b_a}{n_a(T)} = \sum_{a\in T_u}\sum_{j=1}^k \frac{w_a^j - b_a^j}{n_a(T^j)} \\
&= \sum_{j=1}^k \sum_{a\in T_u} \frac{w_a^j - b_a^j}{n_a(T^j)} \\
&\le \sum_{j=1}^k \sum_{a\in q_u} \frac{w_a^j - b_a^j}{n_a(T^j) + 1 - n_a^u(T^j)} \\
&= \sum_{a\in q_u} \frac{w_a - b_a}{n_a(T_{-u}, q_u)} \\
&= \mathtt{cost}_u(T_{-u}, q_u; b),
\end{aligned}$$

i.e., not larger than the cost she would experience by deviating to path q_u; this implies that T is indeed enforced as an equilibrium in the extension of the broadcast game on G with the particular subsidies. The equalities follow by the definition of the cost experienced by the player associated with node u, or the definition of our decomposition, or due to the exchange of sums. The inequality follows since, by Lemma 7, T^j is enforced as an equilibrium in the extension of the broadcast game in G^j with subsidies b^j. □

We now present our lower bound.

THEOREM 11. *For every $\epsilon > 0$, there exist a broadcast game on a graph G and a minimum spanning tree T of G such that the cost of any subsidy assignment that enforces T as equilibrium of the extension of the broadcast game with subsidies is at least $(1/e - \epsilon)\mathtt{wgt}(T)$.*

PROOF. Consider the graph G which consists of a cycle with $n+1$ edges of unit weight that span the root node r and the n nodes which are associated with the players. Let $a = (r, u)$ be an edge incident to the root node r and let T be the path that contains all edges of G besides a. Clearly, T is a minimum spanning tree of G. Now, in order to satisfy that the player associated with node u has no incentive to deviate from her strategy in T and use edge a instead, we have to put subsidies on some of the edges of the path T. The maximum decrease in the cost of the player associated with node u is obtained when subsidies are packed on the least crowded edges of T (i.e., on the edges of T that are further from the root); equivalently, the minimum amount of subsidies necessary in order to decrease the cost of this player to 1 is obtained when subsidies are packed on the least crowded edges of T. Let k be the number of edges that are subsidized. Since the player associated with node u has no incentive to deviate, the cost of $\mathcal{H}_n - \mathcal{H}_k$ she experiences at the $n-k$ edges on which we do not put subsidies is at most 1 while the total amount of subsidies is at least $k-1$. Using the inequality $x \ge \ln(1+x)$ for $x \ge 0$, we obtain $1 \ge \mathcal{H}_n - \mathcal{H}_k = \sum_{t=k+1}^n \frac{1}{t} \ge \sum_{t=k+1}^n \ln\frac{t+1}{t} = \ln\frac{n+1}{k+1}$ which implies that that the total amount of subsidies is at least $k - 1 \ge \frac{n+1}{e} - 2$. The weight of T is n and the bound follows by selecting n to be sufficiently large. □

5. ALL-OR-NOTHING SUBSIDIES

In this section, we consider the all-or-nothing version of SNE. Interestingly, in contrast to the standard version, we prove that its optimization version is hard to approximate within any factor.

THEOREM 12. *Given an instance of all-or-nothing* STABLE NETWORK ENFORCEMENT *consisting of a broadcast game on a graph G and a minimum spanning tree T of G, approximating (within any factor) the minimum cost over all-or-nothing subsidy assignments that enforce T as an equilibrium in the extension of the broadcast game is NP-hard.*

The proof of Theorem 12 is omitted. Theorem 12 probably indicates that the only approximation guarantee we should hope for all-or-nothing SNE is to bound the amount of subsidies as a constant fraction of the weight of an optimal design. The next statement implies that significantly more subsidies may be necessary compared to the standard version of SNE in order to enforce a minimum spanning tree as an equilibrium.

THEOREM 13. *For every $\epsilon > 0$, there exist a broadcast game on a graph G and a minimum spanning tree T of G such that the cost of any all-or-nothing subsidy assignment that enforces T as an equilibrium in the extension of the broadcast game is at least $\left(\frac{e}{2e-1} - \epsilon\right)\mathtt{wgt}(T)$.*

PROOF. We define a graph G with $n+1$ nodes r, v_1, ..., v_n which has the path $\langle r, v_1, v_2, ..., v_n\rangle$ as minimum spanning tree. Let $x = (n - n/e + 1)^{-1}$. Edges (r, v_1) and (v_i, v_{i+1}) for $i = 1, ..., n-2$ have weight x. Edge (v_{n-1}, v_n) has weight 1. The graph contains two additional edges: edge (r, v_{n-1}) has weight x and edge (r, v_n) has weight 1. If we do not put subsidies on the edge (v_{n-1}, v_n), then we have to put subsidies on each of the remaining edges in the path in order to guarantee that the player associated with node v_n has no incentive to use the direct edge (v_n, r), i.e., a total amount of $(n-1)x$ as subsidies. If we put subsidies on the edge (v_{n-1}, v_n), we still have to guarantee that the player associated with node v_{n-1} has no incentive to deviate to the direct edge (v_{n-1}, r). Using the same reasoning as in the proof of Theorem 11, we will need an amount of at least $(n/e-2)x$ as subsidies on the edges of the path $\langle r, v_1, v_2, ..., v_{n-1}\rangle$, for a total of $1+(n/e-2)x$. By the definition of x, we have that the amount of subsidies is at least $\frac{n-1}{n-n/e+1}$ in both cases while the total weight of T is $\frac{2n-n/e}{n-n/e+1}$. The bound follows by selecting n to be sufficiently large. □

6. OPEN PROBLEMS

Our work has revealed several open questions. Concerning the particular results obtained, it is interesting to design a combinatorial algorithm for SNE which, on input a graph G and a minimum spanning tree T on G, enforces T as an equilibrium on the corresponding broadcast game using minimum subsidies. Lemma 2 may be helpful in this direction. For the integral version of SNE, we have left open the question whether it is always possible to enforce a given minimum spanning tree as an equilibrium in a broadcast game using all-or-nothing subsidies of cost strictly smaller than the weight of a minimum spanning tree. Given our negative result in Theorem 12, this is probably the only approximation that makes sense. It is tempting to conjecture that our lower bound is tight, i.e., there is an algorithm that always uses a fraction of at most $\frac{e}{2e-1} \approx 61\%$ of the weight of the minimum spanning tree as subsidies in order to do so.

Approximating SND would also be interesting. Given the known hardness statements (e.g., [20]) or the lack of positive results concerning the complexity of computing equilibria, this is a far more challenging goal. A concrete question for SND instances consisting of broadcast games could be the following: can we compute in polynomial time an equilibrium tree using subsidies of cost at most an α fraction of the weight of the minimum spanning tree? Our results (Theorems 1 and 6) indicate that the answer is clearly positive if $\alpha \geq 1/e$. Is this also possible if α is an arbitrarily small constant? Definitely, more general instances of SND (e.g., involving multicast games) are challenging as well. Finally, variations of SNE and SND that consider deviations of coalitions of players (as opposed to unilateral deviations), players with different demands, or different cost sharing protocols deserve investigation.

Acknowledgments. We thank Edith Elkind, Ning Chen, Nick Gravin, and Alex Skopalik for helpful discussions.

7. REFERENCES

[1] S. Albers and P. Lenzner. On approximate Nash equilibria in network design. In *Proceedings of the 6th International Workshop on Internet and Network Economics (WINE)*, LNCS 6484, pp. 14–25, 2010.

[2] E. Anshelevich, A. Dasgupta, J. M. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4): 1602–1623, 2008.

[3] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-optimal network design with selfish agents. *Theory of Computing*, 4(1): 77–109, 2008.

[4] Y. Bachrach, E. Elkind, R. Meir, D. V. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability in coalitional games. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*, LNCS 5814, pp. 122–134, 2009.

[5] V. Bilò, I. Caragiannis, A. Fanelli, and G. Monaco. Improved lower bounds on the price of stability of undirected network design games. In *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT)*, LNCS 6386, pp. 90–101, 2010.

[6] N. Buchbinder, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative cost sharing games via subsidies. *Theory of Computing Systems*, 47(1): 15–37, 2010.

[7] M. Charikar, H. J. Karloff, C. Mathieu, J. Naor, and M. E. Saks. Online multicast with egalitarian cost sharing. In *Proceedings of the 20th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pp. 70–76, 2008.

[8] C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. *IEEE Journal on Selected Areas in Communications*, 25(6): 1193–1206, 2007.

[9] G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. *Theoretical Computer Science*, 410(36): 3327–3336, 2009.

[10] R. Cole, Y. Dodis, and T. Roughgarden. How much can taxes help selfish routing? *Journal of Computer and System Sciences*, 72(3): 444–467, 2006.

[11] 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)*, pp. 604–612, 2004.

[12] 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 International Colloquium on Automata, Languages and Programming (ICALP)*, LNCS 4051, Part I, pp. 608–618, 2006.

[13] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS)*, LNCS 1563, pp. 404–413, 1999.

[14] J. Li. An $O(\log n/\log\log n)$ upper bound on the price of stability for undirected Shapley network design games. *Information Processing Letters*, 109(15): 876–878, 2009.

[15] D. Monderer and M. Tennenholtz. K-Implementation. *Journal of Artificial Intelligence Research*, 21: 37–62, 2004.

[16] N. Nisan. Introduction to mechanism design (for computer scientists). Chapter 9 in *Algorithmic Game Theory*, Cambridge University Press, pp. 209–241, 2007.

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

[18] R. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2: 65–67, 1973.

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

[20] V. Syrgkanis. The complexity of equilibria in cost sharing games. In *Proceedings of the 6th International Workshop on Internet and Network Economics (WINE)*, LNCS 6484, pp. 366–377, 2010.