

Exact and Approximate Equilibria for Optimal Group Network Formation

Elliot Anshelevich and Bugra Caskurlu

Computer Science Department, RPI, 110 8th Street, Troy, NY 12180
{eanshel,caskub}@cs.rpi.edu

Abstract. We consider a process called Group Network Formation Game, which represents the scenario when strategic agents are building a network together. In our game, agents can have extremely varied connectivity requirements, and attempt to satisfy those requirements by purchasing links in the network. We show a variety of results about equilibrium properties in such games, including the fact that the price of stability is 1 when all nodes in the network are owned by players, and that doubling the number of players creates an equilibrium as good as the optimum centralized solution. For the most general case, we show the existence of a 2-approximate Nash equilibrium that is as good as the centralized optimum solution, as well as how to compute good approximate equilibria in polynomial time. Our results essentially imply that for a variety of connectivity requirements, giving agents more freedom can paradoxically result in more efficient outcomes.

1 Introduction

Many modern computer networks, including the Internet itself, are constructed and maintained by self-interested agents. This makes network design a fundamental problem for which it is important to understand the effects of strategic behavior. Modeling and understanding of the evolution of nonphysical networks created by many heterogenous agents (like social networks, viral networks, etc.) as well as physical networks (like computer networks, transportation networks, etc.) has been studied extensively in the last several years. In networks constructed by several self-interested agents, the global performance of the system may not be as good as in the 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. Much research in the theoretical computer science community has focused on this performance gap and specifically on the notions of the *price of anarchy* and the *price of stability* — the ratios between the costs of the worst and best Nash equilibrium¹, respectively, and that of the globally optimal solution.

In this paper, we study a network design game that we call the *Group Network Formation Game*, which captures the essence of strategic agents building a network together in many scenarios. In this game players correspond to nodes of a graph (although not all nodes need to correspond to players), and the players can have extremely varied connectivity requirements. For example, there might be several different “types” of nodes in the graph, and a player desires to connect to at least one of every type (so that this player’s connected component forms a Group Steiner Tree [11]). Or instead, a player might want to connect

¹ Recall that a (pure-strategy) Nash equilibrium is a solution where no single player can switch her strategy and become better off, given that the other players keep their strategies fixed.

to at least k other player nodes. The first example above is useful for many applications where a set of players attempt to form groups with “complementary” qualities. The second example corresponds to a network of servers where each server want to be connected to at least k other servers so that it can have a backup of its data; or in the context of IP networks, a set of ISPs that want to increase the reliability of the Internet connection for their customers, and so decide to form multi-homing connections through k other ISPs [22]. Many other types of connectivity requirements fit into our framework, and so the results we give in this paper will be relevant to many different types of network problems.

We now formally define the *Group Network Formation Game* as follows. Let an undirected graph $G = (V, E)$ be given, with each edge e having a nonnegative cost $c(e)$. This graph represents the possible edges that can be built. Each player i corresponds to a single node in this graph (that we call a *player* or *terminal* node), which we will also denote by i . Similarly to [2], a strategy of a player is a payment vector p_i of size $|E|$, where $p_i(e)$ is how much player i is offering to contribute to the cost of edge e . We say that an edge e is *bought*, i.e., it is included in the network, if the sum of payments of all the players for e is at least as much as the cost of e ($\sum_i p_i(e) \geq c(e)$). Let G_p denote the subgraph of bought edges corresponding to the strategy vector $p = (p_1, \dots, p_N)$. G_p is the outcome of this game, since it is the network which is purchased by the players.

To define the utilities/costs of the players, we must consider their connectivity requirements. Group Network Formation Game considers the class of problems where the players’ connectivity requirements can be compactly represented with a function $F : 2^U \rightarrow \{0, 1\}$, where $U \subseteq V$ is the set of player nodes, similar to [12]. This function F has the following meaning. If S is a set of terminals, then $F(S) = 1$ iff the connectivity requirements of all players in S would be satisfied if S formed a connected component in G_p . For the example above, where each player wants to connect to at least one player from each “type”, the function $F(S)$ would evaluate to 1 exactly when S contains at least one player of each type. Similarly, for the “data backup” example above, the function $F(S)$ would evaluate to 1 exactly when S contains at least $k + 1$ players. In general, we will assume that the connectivity requirements of the players are represented by a monotone “happiness” function F . The monotonicity of F means that if the connectivity requirements of a player are satisfied in a graph G_p , then they are still satisfied when a player is connected to strictly more nodes. We will call a set of player nodes S a “happy” group if $F(S) = 1$. While not all connectivity requirements can be represented as such a function, it is a reasonably general class that includes the examples given above. Therefore an instance of our game consists of a graph $G = (V, E)$, player nodes $U \subseteq V$, and a function F that states the connectivity requirements of the players. We will say that player i ’s connectivity requirements are *satisfied* in G_p if and only if $F(S_i(G_p)) = 1$ for $S_i(G_p)$ being the terminals in i ’s connected component of G_p . While required to connect to a set of terminal nodes satisfying its connectivity requirements, each player also tries to minimize her total payments, $\sum_{e \in E} p_i(e)$ (which we will denote by $|p_i|$). We conclude the definition of our game by defining the cost function for each player i as:

$$\begin{aligned} - \text{cost}(i) &= \infty && \text{if } F(S_i(G_p)) = 0 \\ - \text{cost}(i) &= \sum_{e \in E} p_i(e) && \text{otherwise.} \end{aligned}$$

In our game, all players want to be a part of a happy group which can correspond to many connectivity requirements, some of which are mentioned above. The socially optimal solution (which we denote by OPT) for this game is the cheapest possible network where every connected component is a happy group, since this is the solution maximizing social

welfare². For our first example above, OPT corresponds to the cheapest forest where every component is a Group Steiner Tree, for the second to the Terminal Backup problem [3], and in general it can correspond to a variety of constrained forest problems [12]. Our goals include understanding the quality of exact and approximate Nash equilibria by comparing them to OPT, and thereby understanding the efficiency gap that results because of the players’ self-interest. By studying the price of stability, we also seek to reduce this gap, as the best Nash equilibrium can be thought of as the best outcome possible if we were able to suggest a solution to all the players simultaneously.

In the Group Network Formation Game, we don’t assume the existence of a central authority that designs and maintains the network, and decides on appropriate cost-shares for each player. Instead we use a cost-sharing scheme which is sometimes referred to as “arbitrary cost sharing” [2, 8] that permits the players to specify the actual amount of payment for each edge. This cost-sharing mechanism is necessary in scenarios where very little control over the players is available, and gives more freedom to players in specifying their strategies, i.e., has a much larger strategy space. The main advantage of such a model is that the players have more freedom in their choices, and less control is required over them. A disadvantage of such a system, however, is that it does not guarantee the existence of Nash equilibria (unlike more constrained systems such as fair sharing [1]). Studying the existence of Nash equilibria under arbitrary cost sharing has been an interesting research problem and researchers have proven existence for many important games [2, 8, 13, 14]. Interestingly, in many of these problems it has been shown that the equilibrium is indeed cheap, i.e., costs as much as the socially optimal network. As we show in this paper, this tells us that in the network design contexts we consider, *arbitrary sharing produces more efficient outcomes while giving the players more freedom.*

Related Work Over the last few years, there have been several new papers using arbitrary cost-sharing, e.g., [8, 14, 15]. Recently, Hoefer [13] proved some interesting results for a generalization of the game in [2], and considered arbitrary sharing in variants of Facility Location.

Unquestionably one of the most important decisions when modeling network design involving strategic agents is to determine how the total cost of the solution is going to be split among the players. Among various alternatives [6], the “fair sharing” mechanism is the most relevant to ours [1, 4, 5, 10]. In this cost sharing mechanism, the cost of each edge of the network is shared equally by the players using that edge. This model has received much attention, mostly because of the following three reasons. Firstly, it nicely quantifies what people mean by “fair” and has an excellent economic motivation since it is strongly related to the concept of Shapley value[1]. Secondly, fair sharing naturally models the congestion effects of network routing games, and so network design games with fair sharing fall into the well-studied class of “congestion games” [4, 7, 16, 21]. Thirdly, this model has many attractive mathematical properties including guarantees on the existence of Nash equilibrium that can be obtained by natural game playing [1].

Despite all of the advantages of congestion games mentioned above, there are extremely important disadvantages as well. Firstly, although congestion games are guaranteed to have Nash equilibria, these equilibria may be very expensive. Anshelevich et al. [1] showed that the cheapest Nash equilibrium solution can be $\Omega(\log n)$ times more expensive than OPT, and that this bound is tight. As we prove in this paper, arbitrary cost-sharing will often

² The solution that maximizes the social welfare is the one that minimizes the total cost of all the players.

guarantee the existence of Nash equilibria that are as cheap as the optimal solution. Secondly, fair sharing inherently assumes the existence of a central authority that regulates the agent interactions or determines the cost shares of the agents, which may not be realistic in many network design scenarios. Arbitrary cost sharing allows the agents to pick their own cost shares, without any requirements by the central authority. Thirdly, although the players are trying to minimize their payments in fair cost sharing, they are not permitted to adjust their payments freely, i.e., a player cannot directly specify her payments on each edge, but is rather asked to specify which edges she wants to use. In the network design contexts that we consider here, we prove that giving players more freedom can often result in better outcomes.

The research on non-cooperative network design and formation games is too much to survey here, see [17, 19, 21] and the references therein.

Our Results Our main results are about the existence and computation of cheap approximate equilibria. By an α -approximate Nash equilibrium, we mean that no player in such a solution has a deviation that will improve their cost by a factor of more than α . While our techniques are inspired by [2], our problem and connectivity requirements are much more general, and so require the development of much more general arguments and payment schemes.

- In Section 3, we show that in the case where all nodes are player nodes, there exists a Nash equilibrium as good as OPT, i.e., the price of stability is 1.
- In Section 4, we show that in the general case where some nodes may not be player nodes, there exists a 2-approximate Nash equilibrium as good as OPT.
- We show that if every player is replaced by two players (or if every player node has at least two players associated with it), then the price of stability is 1. This is in the spirit of similar results from selfish routing [1, 21], where increasing the total amount of players reduces the price of anarchy.
- Starting with a β -approximation to OPT, we provide poly-time algorithms for computing an $(1 + \epsilon)$ -approximate equilibrium with cost no more than β times OPT, for the case where all nodes are player nodes. The same holds for the general case with the factor being $(2 + \epsilon)$ instead.

Since for monotone happiness functions F , OPT corresponds to a constrained forest problem [12], then the last result gives us a poly-time algorithm with $\beta = 2$. Notice that we assumed that the function F is monotone, i.e., that the addition of more terminals to a component does not hurt. This assumption is necessary, since as we prove in Section 5, if F is not monotone there may not exist *any* approximate Nash equilibria. We also show that the results above are only possible in our model with arbitrary cost-sharing, and not with fair sharing.

Because of its applications to multi-homing [3, 22], we are especially interested in the behavior of Terminal Backup connectivity requirements, i.e., when a player node desires to connect to at least k other player nodes. For this special case, we prove a variety of results, such as price of anarchy bounds and the extension of fair sharing results from [1] to this new problem. The lower bounds for Terminal Backup also hold for the general Group Network Formation Game, showing that while the price of stability may be low, the price of anarchy can be as high as the number of players.

2 Properties of the Socially Optimal Network

In this section, we will show some useful properties of the socially optimal network for the Group Network Formation Game, which we refer to as OPT. For notational convenience, we will extend the definition of the happiness function to subgraphs and use $F(S)$ to denote the value of the happiness function for the set of terminal nodes in a network S .

The cost of a network for player i in which her connectivity requirements are not satisfied is ∞ . Therefore, OPT should be the minimum cost network that satisfies all the players. Furthermore, since the satisfaction of the players only depends on the terminal nodes they are connected to, then OPT is acyclic, since otherwise one can obtain a cheaper network that satisfies all the players simply by deleting any one of the edges of a cycle included in OPT.

Observation 1 *The socially optimal network for the Group Network Formation Game is the minimum cost forest that satisfies all the players.*

Let $e = (i, j)$ be an arbitrary edge of a tree T of OPT. Removal of e will divide T into 2 subtrees, namely T_i and T_j (let T_i be the tree containing node i). After removal of e , connection requirements of some of the players in T will be dissatisfied, i.e., either $F(T_i) = 0$ or $F(T_j) = 0$, since otherwise $OPT - e$ would be a network that is cheaper than OPT and satisfies all the players. Therefore, once e is deleted from OPT, all the players in T_i or T_j or both will be dissatisfied. The players that are dissatisfied upon removal of e are said to *witness* e . If e is witnessed by only the players in T_i or only the players in T_j then e is said to be an edge *witnessed from 1-side*. Analogously, we say e is *witnessed from 2-sides* if it is witnessed by all the players in T .

In general, some of the edges of a tree T may be witnessed from 1-side whereas some others are witnessed from 2-sides. We show that the edges of T witnessed from 2-sides form a connected component in T .

Proposition 1. *Let $e = (i, j)$ be an edge of T that is witnessed from 1-side, w.l.o.g. from the side of i . Then all the edges in T_i are also witnessed from 1-side.*

Proof. Let $f = (u, v)$ be an arbitrary edge in T_i and let v be the node closer to i in T_i . If f is removed from T then T would be divided into 2 trees, namely T_u and T_v where $T_u \subset T_i$ and $T_v \supset T_j$. Since e is witnessed only by the players in T_i , $F(T_i) = 0$ and $F(T_j) = 1$. Because F is a monotonic function $F(T_u) = 0$ and $F(T_v) = 1$ and therefore f is witnessed from 1-side, from the side of u . ■

Corollary 1. *The edges of T witnessed from 2-sides form a connected component in T .*

Proof. Let r be a node of T that is incident to an arbitrary edge f witnessed from 2-sides, and root the tree T at r . Let e be an arbitrary edge in T that is witnessed from 2-sides. Observe that all the edges between e and the root r are witnessed from 2-sides, since if an edge of this path were witnessed from 1-side, then by Proposition 1 so would either e or f . Therefore, the set of edges of T that are witnessed from 2-sides form a connected component in T that contains r . ■

3 When all Nodes are Terminals

For the *Group Network Formation Game*, we don't know whether there exists an exact Nash equilibrium for all possible instances of the problem. However, for the special case where each node of G is a terminal node, we prove that Nash equilibrium is guaranteed to exist. Specifically, there exists a Nash equilibrium whose cost is as much as OPT, and therefore price of stability is 1. In this section, we will prove this result by explicitly forming the stable payments on the edges of OPT by giving a payment algorithm. The payment algorithm, which will be formally defined below, loops through all the players and decides the payments of them for all their incident edges. The algorithm never asks a player i to pay for the cost of an edge e that is not incident to i .

Since we are trying to form a Nash equilibrium, no player should have an incentive of unilateral deviation when the algorithm terminates. To have an easier analysis we want our algorithm to have a stronger property: we not only want it to ensure stability at termination but also at each intermediate step. To ensure this stronger property, whenever a player i is assigned to make a payment for an edge e during the execution of the algorithm, it should compute $\chi_i(p_i)$, the cheapest deviation of player i from p_i in $G - e$ that satisfies her (assuming the rest of the payments to buy OPT are made by other players), and should ensure that the cost of p_i never exceeds the cost of $\chi_i(p_i)$ at each iteration. The payment for all the edges of OPT will be decided when the algorithm terminates and we will conclude that the resulting strategy profile is a Nash equilibrium since the cost of the strategy p_i of each player i will be at most her cheapest deviation $\chi_i(p_i)$ with respect to p_i .

Let p^* be a strategy vector that buys all the edges of $OPT - e$, i.e., the entry of $p^*(f) = c(f)$ if f is in $OPT - e$ and $p^*(f) = 0$ otherwise. The deviation $\chi_i(p_i)$ is the cheapest strategy of player i that satisfies her connectivity requirements assuming $\sum_{j \neq i} p_j = p^* - p_i$. Observe that all edges of OPT such that i is not contributing any payment to them can be used by i freely in $\chi_i(p_i)$. Therefore, when computing $\chi_i(p_i)$, the algorithm should not use the actual cost of the edges in $G - e$, but instead for each edge f it should use the cost i would face if she is to use f . We call this the *modified cost of f for i* , and denote it by $c'_i(f)$. Specifically, for f not in OPT , $c'_i(f) = c(f)$, the actual cost of f . For the edges f of $OPT - e$ that i has not contributed anything to (i.e., $p_i(f) = 0$), we have that $c'_i(f) = 0$, since from i 's perspective, she can use these edges for free because other players have paid for them. For all the other edges f that i is paying $p_i(f)$ for, $c'_i(f) = p_i(f)$, since that is how much it costs for i to use f in her deviation from the payment strategy p_i . We use the notation $\chi_i(p_i)$ for both the deviation itself and also the cost of it; in what follows the meaning will be clear from the context.

Observe that the algorithm asks the players to pay for their incident edges only. Therefore, each edge is considered for payment twice. For each edge $e = (u, v)$ where u is the parent of v , first v is asked to pay for e at the maximum amount that will not create an incentive for unilateral deviation for her. At the later iterations of the algorithm, when u is processed, the algorithm asks u to pay for the remaining cost of e . Recall that whenever the algorithm asks a player to contribute to the cost of an edge it also computes her cheapest deviation and ensures that no player makes a payment that will create an incentive of unilateral deviation. Therefore, if the payment algorithm does not break at any of the intermediate stages, then it finds a Nash equilibrium whose cost is as much as OPT. To prove our result all we need to do is prove that the algorithm never breaks at an intermediate stage. We will prove this by constructing a network cheaper than OPT which satisfies all the players whenever the algorithm breaks, thus forming a contradiction.

Input: The socially optimal network OPT
Output: The payment scheme for OPT
Initialize $p_i(e) = 0$ for all players i and edges e ;
Root each tree T of OPT by an arbitrary node incident to an edge witnessed from 2-sides;
Loop through all trees T of OPT;
 Loop through all nodes i of T in reverse BFS order;
 Loop through all edges of T_i incident to i ;
 Let $d(e) = c(e) - \sum_{j \neq i} p_j(e)$;
 If $\chi_i(p_i) - \sum_f p_i(f) \geq d(e)$
 Set $p_i(e) = d(e)$;
 Else break;
 Define g to be the parent edge of node i ;
 Set $p_i(g) = \min\{\chi_i(p_i) - \sum_f p_i(f), c(g)\}$;

Algorithm 1: Algorithm that generates payments on the edges of OPT

Specifically, we will consider networks formed by players' deviations. Define $X_i(p_i)$ to be the graph formed by removing the edges paid for by p_i from OPT, and then adding the edges paid for in $\chi_i(p_i)$. In other words, $X_i(p_i)$ is the network formed if player i deviates from her current strategy p_i to her best response $\chi_i(p_i)$, while assuming that the rest of OPT is paid for by other players. Since the cost of p_i is always at most the cost of $\chi_i(p_i)$, we know that graphs $X_i(p_i)$ are always at most the cost of OPT. They also have the following nice property.

Lemma 1. *Let C be the connected component containing i in $X_i(p_i)$. Then, C contains either all players of T or another tree T' of OPT.*

Proof. By Lemma 2 given below, all the subtrees T_u linked to i by a 1-sided edge (u, i) will be in C always, since the edge (u, i) will be entirely paid for by u . Let T_v be a subtree linked to i by a 2-sided edge (v, i) , and suppose to the contrary that T_v is not in C . Since (v, i) is a 2-sided edge, then $T - T_v$ is not a happy component, and so for i to be satisfied, it must connect to some tree T' of OPT, as desired. ■

Lemma 2. *Let $e = (i, j)$ be an edge of T that is witnessed from 1-side, w.l.o.g. from the side of i . Then when the algorithm asks the player in i to make payment for e she will pay for the whole cost of e .*

Proof. We will prove the lemma by induction on the number of edges in the subtree T_i . First consider the case where i is a leaf-node of T , i.e., the number of edges in T_i is 0, as the base case of induction. If the player in i does not pay an amount $c(e)$ for the edge e then she has a deviation χ_i whose cost is less than $c(e)$. Then we will ask player i to play χ_i as her strategy. Consider the graph obtained by deleting e from OPT and adding the edges bought by the strategy χ_i . This new graph is clearly cheaper than OPT since the total cost of the edges bought by χ_i is less than $c(e)$. Since none of the players except the one sitting in i was witnessing e in OPT the connection requirement of all of them will be satisfied. The connection requirement of the player sitting in i is trivially satisfied since she is the only deviating player. Since the new graph satisfies all the players and is cheaper than OPT, the player in i will pay $c(e)$ for e .

Now consider an arbitrary edge $e = (i, j)$ that is witnessed from 1-side. Note that all the edges in T_i are witnessed from 1-side by Proposition 1 and therefore are bought by the players sitting at lower level terminal nodes by the inductive assumption. So player sitting at i is not asked to pay for any edge but e by the algorithm. Assume player in i does not pay an amount $c(e)$ for the edge e when she is asked to make payment for e . Then she has a deviation χ_i whose cost is less than $c(e)$. Then we will ask player i to play χ_i as her strategy. In this case, $X_i(p_i)$ is clearly cheaper than OPT as before since the total cost of the edges bought by χ_i is less than $c(e)$. Since none of the players except the ones in T_i were witnessing e in OPT and e is the only edge of OPT that does not exist in $X_i(p_i)$, then all the players except the ones in T_i are satisfied in $X_i(p_i)$. However, since all the edges of T_i are part of $X_i(p_i)$, then all the players in T_i are in the same connected component of $X_i(p_i)$. Since $X_i(p_i)$ satisfies i , it also satisfies all the players in T_i as well. Since $X_i(p_i)$ satisfies all the players and cheaper than OPT, we have a contradiction. Therefore, the player located at i must pay $c(e)$ for e . ■

For the purpose of contradiction, assume our algorithm broke while deciding the payment strategy of the player located at u . Observe that the algorithm can only break while a player u is making payment for one of the incident edges in T_u . Let $e = (u, v)$ be that edge. Note that e is an edge witnessed from 2 sides since otherwise it would be fully bought by v by Lemma 2 and thus player u would not be asked to pay for it. Since the algorithm broke, then u has a deviation $\chi_u(p_u)$ in $G - e$ such that the modified cost of $\chi_u(p_u)$ is strictly less than what u has been asked to pay for so far. The key observation here given by Lemma 1 is that $\chi_u(p_u)$ is either connecting u to all the players of T_u , or to a different tree of OPT. χ_u cannot be connecting u to all the subtrees of T_u since then $X_u(p_u)$ connects all the players in T and is cheaper than OPT.

Therefore, χ_u is connecting u to a different tree T' of OPT. In the graph $X_u(p_u)$, all the players of T' will be satisfied since F is monotone. The set of players in the subtrees of T_u that are connected to u in $X_u(p_u)$ and all the players in $T - T_u$ are satisfied in $X_u(p_u)$ since they are connected to all the players of T' . Therefore, we formed a new subgraph $X_u(p_u)$ that is cheaper than OPT and satisfies all the players except the ones that are in the subtrees of T_u that are not in the same connected component with u in $X_u(p_u)$. To complete the proof all we need is to show that we can connect the players in these subtrees simultaneously to happy connected components without increasing the cost, and we prove this in the following lemma.

Lemma 3. *Let v be a terminal node such that the player sitting at v could not buy $e = (v, u)$, the higher level incident edge of v , after paying $d(f)$ for each incident edge f in T_v . Then by replacing the strategies of a subset of players in T_v by their respective deviations, we can connect all the players in T_v to the trees of OPT other than T , without using any edges of $T - T_v$.*

Proof. We will show that by replacing the strategies of a subset of players in T_v by their respective deviations, we can connect all the players in T_v to the trees of OPT other than T by induction on the size of T_v . Consider the case where v is a leaf node as the base case. If the player located at v cannot buy e , then she has a deviation χ_v which is strictly cheaper than $c(e)$. Observe that the set of edges bought by χ_v does not connect v to $T - v$ since e is the cheapest path between v and $T - v$. Therefore, v will be connected to a different tree of OPT if the player sitting at v plays χ_v as her strategy.

Now consider an arbitrary terminal node v . If the player located at v cannot buy e then she has a deviation $\chi_v(p_v)$ which is cheaper than what she has paid so far plus the cost of e . Therefore, $X_v(p_v)$ is strictly cheaper than OPT. By Lemma 1, we know that in $X_v(p_v)$, either v is connected to a different tree of OPT, or v is connected to all the players in T . However, if v is connected to all players of T then $X_v(p_v)$ satisfies all the players and is cheaper than OPT, which is a contradiction. Therefore, v is connected to a different tree T' of OPT in $X_v(p_v)$. The set of players in the subtrees of T_v that are still connected to v in $X_v(p_v)$ are connected T' as well, and so they are satisfied. Let v_1, v_2, \dots, v_k be the children of v such that they are not connected to v in $X_v(p_v)$. By the inductive hypothesis, for every subtree T_{v_i} , there is a set of players S_i in this subtree such that by letting this set of players deviate, we create a solution where all players in T_{v_i} are satisfied, and their deviations do not use the edges of $T - T_{v_i}$.

Consider the graph G' that results from letting the set of players $\cup_i S_i \cup \{v\}$ deviate. First, notice that all the players in each subtree T_{v_i} are still connected to a tree T' of OPT, since their deviations do not use edges of $T - T_{v_i}$, and so the fact that other players deviate and possibly remove their payments from some edges of $T - T_{v_i}$ does not affect them. The players connected to v in $X_v(p_v)$ are still connected to a tree T' , since all the edges used by them in $X_v(p_v)$ to connect to T' are still there in G' . Therefore, all the players of T_v are connected to trees of OPT other than T .

To finish the proof, we must show that none of the deviations used to obtain G' are using edges of $T - T_v$. This is true for the deviations in subtrees T_{v_i} by the inductive hypothesis. Now suppose to the contrary that $\chi_v(p_v)$ uses $T - T_v$, i.e., that v and $T - T_v$ are in the same connected component of $X_v(p_v)$. In this case, the above graph G' satisfies all the players in T (not just the ones in T_v), and so G' is a feasible solution that is cheaper than OPT, a contradiction. ■

4 Good Equilibria in the General Game

In Section 3, we saw that a good equilibrium always exists when all nodes are terminals. In this section, we consider the general Group Network Formation Game, and show that there always exists a 2-approximate Nash equilibrium that is as cheap as the centralized optimum. By a 2-approximate Nash equilibrium, we mean a strategy profile $p = (p_1, p_2, \dots, p_n)$ such that no player i can reduce her cost by more than a factor of 2 by unilaterally deviating from p_i to p'_i , i.e., $|p'_i| > |p_i/2|$ for any unilateral deviation p'_i of i . To prove this, we first look at an important special case that we call the *Group Network Formation of Couples Game* or *GNFCG*. This game is exactly the same as the Group Network Formation Game, except that every terminal node is guaranteed to have at least two players located at that node (although not all nodes need to be player nodes).

Theorem 1. *If the price of stability for the GNFCG is 1 then there exists a 2-approximate Nash Equilibrium for the Group Network Formation Game that costs as much as OPT.*

Proof. Assume we are given an instance $\mathfrak{S}_1 = (N_1, G, T, F)$ of a Group Network Formation Game, i.e., we are given a set of players $N_1 = (1, 2, \dots, n)$, a graph $G = (V, E)$ such that each edge $e \in E$ is associated with a nonnegative cost $c(e)$, a set of terminal nodes $T \subseteq V$ such that each player $i \in N_1$ is located at a terminal node $u \in T$ and a monotone happiness function $F : 2^T \rightarrow \{0, 1\}$. All we need to show is that \mathfrak{S}_1 has a 2-approximate Nash

equilibrium as cheap as OPT assuming price of stability for the Group Network Formation of Couples Game is 1.

We will first define an instance $\mathfrak{S}_2 = (N_2, G, T, F)$ of the Group Network Formation of Couples Game. Observe that the graph G , the set of terminal nodes T and the monotone happiness function F for both \mathfrak{S}_1 and \mathfrak{S}_2 are the same; however, the number of players of \mathfrak{S}_2 is twice as much as the number of players of \mathfrak{S}_1 , i.e., $N_2 = (1, 2, \dots, 2n)$. For each player $i \in N_1$ of \mathfrak{S}_1 located at $u \in T$ there are 2 corresponding players $i, (n+i) \in N_2$ of \mathfrak{S}_2 located at u .

First observe that the socially optimal network for both games \mathfrak{S}_1 and \mathfrak{S}_2 is the same since any network satisfying the players of \mathfrak{S}_1 also satisfies the players of \mathfrak{S}_2 and vice versa. Let OPT denote the socially optimal network of both of the games. Since we have assumed price of stability for Group Network Formation of Couples Game is 1, we know that there exists a stable strategy profile $p = (p_1, p_2, \dots, p_{2n})$ for the players N_2 of \mathfrak{S}_2 that buys OPT. To illustrate, for each player $i \in N_2$, the cost of their cheapest deviation $\chi_i(p_i)$ is at most the cost of p_i . Furthermore, since player i and player $(n+i)$ sit at the same terminal node for $i \leq n$, the strategy p_i is also a stable strategy for player $(n+i)$ and vice versa.

To complete the proof, all we need to do is to give a strategy profile $p' = (p'_1, p'_2, \dots, p'_n)$ for the players of \mathfrak{S}_1 such that for each $i \in N_1$, the cost of p'_i is at most 2 times more than the cost of $\chi_i(p'_i)$. We define p' as follows. For each player $i \in N_1$, $p'_i = p_i + p_{(n+i)}$. Observe that both p_i and $p_{(n+i)}$ are stable strategies for player $i \in N_1$ since she is located at u and therefore the cost of $\chi_i(p'_i)$ is greater than (or equal to) the costs of both p_i and $p_{(n+i)}$ and therefore, the cost of $\chi_i(p'_i)$ is at least half of the cost of p'_i . ■

Because of Theorem 1, we will focus on the GNFCG in the rest of the section and prove the existence of a Nash equilibrium as cheap as OPT. This result is interesting in its own right, since it states that to form an equilibrium that is as good as the optimum solution, it is enough to double the number of players. Such results are already known for many variants of congestion games and selfish routing [1, 21], but as Theorem 2 shows, we can also prove such results for games with arbitrary sharing.

Given a set of bought edges T ; a strategy profile p such that for all players i , p_i is the cheapest strategy satisfying i , assuming rest of the payments to buy all the edges of T are made by other players, is a Nash equilibrium. To prove that price of stability is 1 for GNFCG, we give an algorithm that forms such a strategy profile on the edges of OPT.

Recall that the payment strategies of all the players have to be stable when the algorithm terminates. As in Section 3, to have an easier analysis we not only want our algorithm to ensure stability at termination but also at each intermediate step. To ensure this stronger property, whenever a player i is assigned to make a payment for an edge e during the execution of the algorithm, it should compute $\chi_i(p_i)$, the cheapest deviation of player i from p_i that satisfies her, and should ensure that the cost of p_i never exceeds the cost of $\chi_i(p_i)$ at each iteration by using the modified costs of the edges as in Section 3. In the rest of the section we prove our main theorem for the GNFCG.

Theorem 2. *For GNFCG, there exists a Nash equilibrium as cheap as the socially optimal network, i.e., the price of stability is 1.*

For ease of explanation, we will first consider the case where all the edges of OPT are witnessed from two sides and later illustrate how our algorithm can be modified for the case where some of the edges are witnessed from one side only. We start by rooting each connected component of OPT arbitrarily by a high degree non-player node. Throughout the paper,

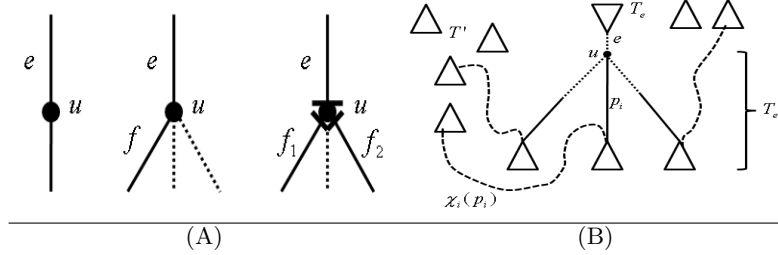


Fig. 1. (A) Illustrates the assignment of the player to pay for the cost of e . (B) Shows how to construct a cheap network that satisfies all the players in T_e by using the deviations of a subset S of them.

the term *high degree node* refers to the nodes with degree 3 or more. On each connected component T of OPT, we run a 2-phase algorithm. In the first phase of the algorithm, we assign players to make payments to the edges of T in a bottom-up manner, i.e., we start from a lowest level edge e of T and pick a player i to make some payment for e and continue with the next edge in the reverse BFS order. In the first phase of the algorithm, we ask a player i to contribute only for the cost of edges on the unique path between her and the root and furthermore, the payment for each edge is made by only one player.

Algorithm (Phase 1) For an arbitrary edge $e = (u, v)$ where u is the lower level incident node of e , the assignment of the player to pay for e is as follows. If u is a terminal node, we ask a player i located at node u to make maximum amount of payment on e that will not make p_i unstable, i.e., we set $p_i(e) = \min\{\chi_i(p_i) - |p_i|, c(e)\}$. If u is a degree 2 nonterminal node then we ask the player who has completely bought the other incident edge of u , i.e., made a payment equal to $c(e)$, to make maximum amount of payment on e that will not make her strategy unstable as shown on the left of Figure 1(A). Note that it may be the case that no player has bought the other incident edge of u in which case we don't ask any player to pay for e and the payment for e will be postponed to the second phase of the algorithm. If u is a high degree nonterminal then the selection of the player to pay for e is based on the number of lower level incident edges of u that are bought in the previous iterations of the algorithm. If none of the lower level incident edges of u are bought then we postpone the payment on e to the second phase of the algorithm. If exactly one of the lower level incident edges of u , namely f , is bought then we ask the player who bought f to make maximum amount of payment on e that will not make her strategy unstable as shown in the middle of Figure 1(A). If 2 or more of the lower level incident edges of u are already bought, namely f_1, f_2, \dots, f_l , then we fix the strategies of the players i_1, i_2, \dots, i_l that bought those edges, i.e., the players i_1, i_2, \dots, i_l are not going to pay any more and therefore the strategies of those players that will be returned at the end of the algorithm are already determined. Since there are two players located at every terminal, pick an arbitrary player located at the same terminal as one of i_1, i_2, \dots, i_l that has not made any payments yet, and assign her to make maximum amount of payment for e that will not make her strategy unstable as shown on the right of Figure 1(A). We later prove that such a player always exists, i.e., not all of i_1, i_2, \dots, i_l are the last players to make payment at their respective terminal nodes.

We here present the analysis of the first phase of the algorithm. When we are talking about a player i , let T denote the connected component of OPT containing i and let T' denote the set of other connected components of OPT. For an arbitrary edge e of T , we use

T_e in order to refer to the subtree of T below e and T_u to refer to the subtree below a node u . To prove the existence of a Nash equilibrium as cheap as OPT, we show that whenever our algorithm cannot form stable payments on the edges of OPT we can find a subgraph of G that is cheaper than OPT and satisfies all the players. Since OPT is the cheapest network satisfying all the players, we will end up with a contradiction.

We next give a series of lemmas that successively proves the following. For every edge e that could not be bought in the first phase of the algorithm by the assigned player to make payment for it, we can connect all the terminal nodes in T_e to the connected components of T' *without using any of the edges of $T - T_e$* by simply setting $p_i = \chi_i(p_i)$ for a subset S of players in T_e . The deviations of the subset S of the players are depicted in Figure 1(B). The condition that no edges of $T - T_e$ are used by the deviations is crucial, since that is what allows us to have a set of players all deviate at once and still be satisfied afterwards. The fact that such a “re-wiring” exists allows us to argue in our proofs that at least one of the incident edges of the root of T will be bought during the first phase of the algorithm.

Notation We now define some helpful notation in order to prove some lemmas about the first phase of the algorithm which is fully specified above. When we are talking about a player i , let T denote the connected component of OPT containing i and let T' denote the set of other connected components of OPT. The strategy of a player is denoted by p_i which is a vector of length m , the total number of edges in G , where each entry of p_i indicates the payment player i is making for the corresponding edge. We use p^* for the strategy vector that buys OPT, i.e., $p^*(e) = c(e)$ if e is an edge of OPT and $p^*(e) = 0$ otherwise. Observe that when the algorithm terminates it should be that $\sum_i p_i = p^*$. We use the notation $G(p)$ for the subgraph of bought edges by strategy p , i.e., the subgraph composed of the edges for which $p(e) = c(e)$. For instance, $G(p^*)$ denotes OPT, $G(p^* - p_i)$ denotes the subgraph composed of edges bought by players other than i and $G(p^* - p_i + \chi_i(q_i))$ denotes the subgraph of bought edges if player i deviates from her strategy p_i to $\chi_i(q_i)$. Finally, for an arbitrary edge e of T , we use T_e in order to refer to the subtree of T below e and $\overline{T_e}$ to refer to the rest of the tree $T - T_e$. The notation for the subtree of T rooted at a node u is analogously T_u .

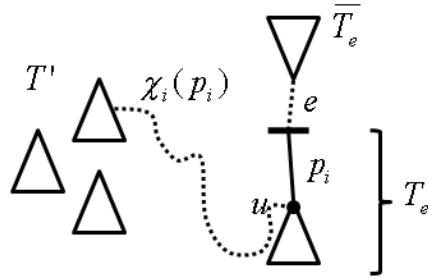


Fig. 2. Shows the deviation $\chi_i(p_i)$ of a player i that is located at terminal u and could not buy the edge e .

Lemma 4. *Let u be an arbitrary terminal node and i be the first player to make payments that is located at u . Let e be the first edge between u and the earliest ancestor of u which is either a terminal or a high degree nonterminal node such that player i did not buy e . Then all the players in T_u will be satisfied in the subgraph $G(p^* - p_i + \chi_i(p_i)) - \overline{T_e}$.*

Proof. Since any edge f between u and e is bought by player i , i.e., $p^*(f) = p_i(f) = c(f)$ and player i did not pay for any other edge in G , then the subgraph $G(p^* - p_i)$ consists of the connected components T' , T_u and $\overline{T_e}$. Observe that the players in T_u , including i , are not satisfied in $G(p^* - p_i)$ since according to our assumption, any edge of OPT is witnessed from two sides, and so any subset of T_e cannot be a happy component. Since a player i is always satisfied if she unilaterally deviates from her strategy p_i to $\chi_i(p_i)$, then player i is satisfied in $G(p^* - p_i + \chi_i(p_i))$, the subgraph of the bought edges when player i unilaterally deviates to her best deviation $\chi_i(p_i)$ from her strategy p_i . Since player i is satisfied in $G(p^* - p_i + \chi_i(p_i))$ and not satisfied in $G(p^* - p_i)$, then the subgraph $G(p^* - p_i + \chi_i(p_i)) - G(p^* - p_i)$ must include a path P_u between T_u and one of the connected components in T' or $\overline{T_e}$.

For the purpose of contradiction, assume P_u is a path between T_u and $\overline{T_e}$. Since $T - T_u - \overline{T_e}$ is a path of degree 2 nonterminal nodes between T_u and $\overline{T_e}$, then all the players of T will be in the same connected component of $G(p^* - p_i + \chi_i(p_i))$ and therefore the subgraph $G(p^* - p_i + \chi_i(p_i))$ satisfies all the players. This is because i must be satisfied in $G(p^* - p_i + \chi_i(p_i))$, and so it is in a happy component. Observe that $G(p^* - p_i)$ has the same set of edges as OPT except e and the edges bought by player i . Since the cost of $\chi_i(p_i)$ is equal to the cost of p_i , then $G(p^* - p_i + \chi_i(p_i))$ is a subgraph satisfying all the players and cheaper than OPT. More precisely, the cost of $G(p^* - p_i + \chi_i(p_i))$ is less than the cost of OPT by an amount $c(e) - p_i(e)$. Since OPT is the cheapest network satisfying all the players, this is a contradiction and therefore there cannot exist a path between T_u and $\overline{T_e}$ in $G(p^* - p_i + \chi_i(p_i))$. Since the players in T_u are in a connected component that is disjoint from all the nodes and the edges of $\overline{T_e}$ in $G(p^* - p_i + \chi_i(p_i))$, then they are all satisfied in $G(p^* - p_i + \chi_i(p_i)) - \overline{T_e}$ as shown in Figure 2. ■

Lemma 5. *Let i be player such that i did not buy e , i.e., $p_i(e) < c(e)$, even though the algorithm assigned i to make payment for e . Then there exists a subset of players $S = (s_1, \dots, s_k)$ in T_e with deviations $\chi_1(p_1), \dots, \chi_k(p_k)$ from their respective strategies p_1, \dots, p_k such that all the players in T_e will be satisfied in the subgraph $G(p^* - \sum_{i \in S} p_i + \sum_{i \in S} \chi_i(p_i)) - \overline{T_e}$.*

Proof. We will prove this result by induction on the number of nodes in T_e . If T_e has only 1 node u , then the lemma holds by Lemma 4. Below we will use the notation $R(S)$ to denote $G(p^* - \sum_{i \in S} p_i + \sum_{i \in S} \chi_i(p_i))$ and $R(S, e)$ to denote $R(S) - \overline{T_e}$.

Let us assume that the lemma holds for all instances such that the number of nodes in T_e is at most k and let's prove that the lemma also holds when the number of nodes is $k + 1$. Let Γ be the set composed of connected components of $G(\sum_{i \in T_e} p_i)$ that involves at least one terminal node and let C be the connected component involving i . Observe that C is the highest level connected component, i.e., the one that is adjacent to e .

Let u be the highest level terminal or high degree nonterminal node in C . If u is a terminal node then the result directly follows from Lemma 4, with $S = \{i\}$, since the terminals in T_u are the same as in T_e . Therefore, assume u is a high-degree nonterminal node. Observe that at least one of the lower level incident edges of u is a bought edge since otherwise C would not be a connected component of $G(\sum_{i \in T_e} p_i)$ that involves at least one terminal node. Recall that if at least 2 of the lower level incident edges of u are bought, i is only

assigned to make payment for the edges above u , i.e., player i has not made any payment on the edges below u . Therefore, player i can be viewed as located at u and all we need to do is to exactly repeat the proof of Lemma 4, one again giving us the desired result with $S = \{i\}$. Let us now consider the final case, which is depicted in Figure 1(B), where u is a high-degree nonterminal node such that exactly one lower level incident edge of u is bought.

In this case, let C_1, C_2, \dots, C_k be the elements of Γ that are one level lower than C and let e_1, e_2, \dots, e_k be the immediate higher level unpaid edges of C_1, C_2, \dots, C_k respectively. By the inductive hypothesis, each of these edges e_j already has a desired set of players S_j in T_{e_j} .

The connected component containing i in the subgraph $G(p^* - p_i + \chi_i(p_i))$ may also contain a player j in T_{e_j} . Since player i did not make any payment for the edges in T_{e_j} , all the edges of T_{e_j} are also part of $G(p^* - p_i + \chi_i(p_i))$ and therefore all the players in T_{e_j} are in the same connected component with i in $G(p^* - p_i + \chi_i(p_i))$. Let Λ be the edges of e_1, e_2, \dots, e_k such that T_{e_j} is *not* in the same connected component of $G(p^* - p_i + \chi_i(p_i))$ as i . Then, we set the set S to be $\cup_{e_j \in \Lambda} S_j \cup \{i\}$.

We must now prove that all the players in T_e are satisfied in the subgraph $R(S, e) = G(p^* - \sum_{i \in S} p_i + \sum_{i \in S} \chi_i(p_i)) - \overline{T_e}$. First, we prove this for the players in T_{e_j} for $e_j \in \Lambda$. By the inductive hypothesis, they are all satisfied in the subgraph $R(S_j, e_j) = G(p^* - \sum_{i \in S_j} p_i + \sum_{i \in S_j} \chi_i(p_i)) - \overline{T_{e_j}}$. Let f be an arbitrary edge in a connected component of a player in T_{e_j} in the subgraph $R(S_j, e_j)$. This edge cannot be in $\overline{T_e}$, since $\overline{T_e} \subseteq \overline{T_{e_j}}$. This edge cannot be paid for by a player outside T_{e_j} , since those players only pay for edges in $\overline{T_{e_j}}$. Therefore, f must still be present in $R(S, e)$, and so all players in T_{e_j} are still satisfied in $R(S, e)$.

Now consider the players in C and in T_{e_j} for $e_j \notin \Lambda$. They are satisfied in $G(p^* - p_i + \chi(p_i))$ since i is satisfied and they are in the same connected component. Let f be an arbitrary edge in the connected component of $G(p^* - p_i + \chi(p_i))$ containing player i . These players are satisfied in $R(S) = G(p^* - \sum_{i \in S} p_i + \sum_{i \in S} \chi_i(p_i))$, since f is still present in $R(S)$. This is because if f were being paid for by a player j , then it would be part of some subtree T_{e_j} with $e_j \notin \Lambda$, and so $j \notin S$, and those payments on f would remain unchanged. We need to prove that players in C and in T_{e_j} for $e_j \notin \Lambda$ are satisfied in $R(S, e)$, and so it is enough to show that $f \notin \overline{T_e}$. If this were not the case, then $\overline{T_e}$ is in the same connected component of $R(S)$ as i . Since i is satisfied in $R(S)$, then so are all the players in $\overline{T_e}$. We already proved that all players in all subtrees T_{e_j} are satisfied in $R(S)$, and so $R(S)$ is a feasible solution (all the players in the graph are satisfied). Notice, however, that the solution $R(S)$ is cheaper than OPT, by the same argument as in Lemma 4, and so this is not possible.

Therefore, all the players in T_e are satisfied in $R(S, e)$, as desired. ■

Lemma 6. *Let C_1, C_2, \dots, C_k be the highest level connected components of the subgraph $G(\sum_i p_i)$ that include at least one terminal node and let e_1, e_2, \dots, e_l be the higher level incident edges of C_1, C_2, \dots, C_k respectively. Let $\overline{T_{e_1, e_2, \dots, e_l}} = T - T_{e_1} - T_{e_2} - \dots - T_{e_k}$. Then for each C_j there exists a corresponding set S_j of players in T_{e_j} such that all players in T are satisfied in $G(p^* - \sum_{l=1}^k (\sum_{j \in S_l} p_j - \sum_{j \in S_l} \chi_j(p_j))) - \overline{T_{e_1, e_2, \dots, e_k}}$.*

Proof. Let i be an arbitrary player in T . Without loss of generality assume it is in T_{e_j} . There is a set S_j such that player i is satisfied in $G(p^* - \sum_{j \in S_j} p_j - \sum_{j \in S_j} \chi_j(p_j)) - \overline{T_{e_j}}$ due to Lemma 5. To prove the lemma, all we need to show is that every edge of

$G\left(p^* - \sum_{j \in S_j} p_j + \sum_{j \in S_j} \chi_j(p_j)\right) - \overline{T_{e_j}}$ is also an edge of $G\left(p^* - \sum_{l=1}^k \left(\sum_{j \in S_l} p_j - \sum_{j \in S_l} \chi_j(q_j)\right)\right) - \overline{T_{e_1, e_2, \dots, e_k}}$.

First let us consider the edges of T . Observe that none of the edges in $\overline{T_{e_j}}$ are in $G(p^* - \sum_{j \in S_j} p_j + \sum_{j \in S_j} \chi_j(q_j)) - \overline{T_{e_j}}$. Therefore all we need is to check the edges in T_{e_j} . Since the algorithm never assigns a player i to pay for the edges that are not on the unique path between the terminal i and the root of T , none of the the players in $T - T_{e_j}$ made payment for any edge in T_{e_j} . Since the players in $\bigcup_{l \neq j} S_l$ did not make payment on the edges of T_{e_j} , then an edge of T_{e_j} that is in $G\left(p^* - \sum_{j \in S_j} p_j + \sum_{j \in S_j} \chi_j(p_j)\right) - \overline{T_{e_j}}$ is also an edge of $G\left(p^* - \sum_{l=1}^k \left(\sum_{j \in S_l} p_j - \sum_{j \in S_l} \chi_j(p_j)\right)\right) - \overline{T_{e_1, e_2, \dots, e_k}}$.

Now let us consider the edges outside of T . Since the algorithm never asks the players to pay for the edges outside T , an edge not in T is in $G\left(p^* - \sum_{j \in S_j} p_j + \sum_{j \in S_j} \chi_j(p_j)\right) - \overline{T_{e_j}}$ if and only if it is also in $G\left(\sum_{j \in S_j} \chi_j(p_j)\right)$. Therefore, any edge of \overline{T} that is in $G\left(p^* - \sum_{j \in S_j} p_j + \sum_{j \in S_j} \chi_j(p_j)\right) - \overline{T_{e_j}}$ is in $G\left(p^* - \sum_{k=1}^l \left(\sum_{j \in S_k} p_j - \sum_{j \in S_k} \chi_j(p_j)\right)\right) - \overline{T_{e_1, e_2, \dots, e_l}}$ as well. ■

Lemma 7. *At least one of the incident edges of the root of T will be bought during the first phase of the algorithm.*

Proof. For the purpose of contradiction assume none of the incident edges of the root of T is bought at the first phase of the algorithm and let's obtain a contradiction by constructing a subgraph that is cheaper than OPT and satisfies all the players. According to Lemma 6, the graph $R = G\left(p^* - \sum_{l=1}^k \left(\sum_{j \in S_l} p_j - \sum_{j \in S_l} \chi_j(p_j)\right)\right) - \overline{T_{e_1, e_2, \dots, e_k}}$ satisfies all the players. Therefore, all we need to show is that the cost of R is less than the cost of OPT. Since the cost of $\chi_i(p_i)$ is equal to the cost of p_i , then we know by the same argument as in the proof of Lemma 4 that R is strictly cheaper than OPT. ■

Algorithm (Phase 2) In the second phase of the algorithm, we ask the players that have not made any payments yet to make stable payments for the remaining edges and buy them. Let Γ be the set composed of connected components of $G\left(\sum_{i \in T_e} p_i\right) - T'$ that include at least one terminal node. In other words, Γ consists of connected components of the edges in T purchased so far by the algorithm (a single terminal node with no adjacent bought edges would also be a component in Γ). We call a connected component $C_1 \in \Gamma$ *immediately below* a connected component $C \in \Gamma$ if after contracting the components in Γ , C is above C_1 in the resulting tree and there are no other components of Γ between them. In the second phase of the algorithm, we form payments on the edges in a top-down manner as we explain next. We start from the connected component $C \in \Gamma$ that includes the root of T (this must be true by Lemma 7) and assign a player i in C that has not made any payments yet to buy *all* the edges between C and the connected components that are immediately below C . The set of edges i should buy are shown in Figure 3. We prove that such a player i always exists in Lemma 8. Observe that once i buys all the edges between C and the connected components C_1, C_2, \dots, C_k that are immediately below C , all these $k + 1$ connected components form a single connected C that contains the root. We repeat this procedure, i.e., pick a player i in the top-most connected component C that has not made a payment yet to buy all the edges between C and the connected components that are immediately below C , until all the players in T are in the same connected component and all of T is paid for.

To show that our algorithm forms an equilibrium payment, we need to prove that no player has a deviation from the payment assigned to her. This is true for players making payments during the first phase by construction. To finish the proof, we need to show that a strategy p_i that buys all the edges between a connected component C and the connected components C_1, C_2, \dots, C_k that are immediately below C is a stable strategy for any player i in C , which we show in Lemma 9.

This concludes the proof of Theorem 2. Recall that for ease of explanation, we only considered the case where all edges of OPT are witnessed from two sides until now. In Lemma 10, we modify this algorithm to return a Nash equilibrium that purchases OPT even if some of the edges of OPT are witnessed from one side.

Lemma 8. *At any stage of the algorithm, each connected component of Γ has a player i such that $p_i(e) = 0$ for all edges e .*

Proof. The statement is trivially true at the start of the algorithm. In the first phase, consider a time when we ask a player i , which is not the first to pay among the players with the same terminal node of i , to make payment for some edges. This only occurs when at least 2 of the lower level incident edges of a high-degree nonterminal u are bought. But then 2 or more connected components of Γ merge. Since each of these connected components had at least one player who hasn't made any payment yet, and only one of them is being asked to pay at this moment, then every component of Γ still has at least one player that has not made any payments.

In the second phase of the algorithm, one player i buys all the edges between a connected component C and the connected components C_1, C_2, \dots, C_k that are immediately below C as shown in Figure 3. Similar to the above case, after the player i makes her payment, at least 2 connected components of Γ merge. Therefore, the lemma holds at the end of the second phase of the algorithm. ■

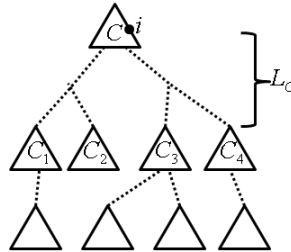


Fig. 3. Illustrates the set of edges to be bought by a player i located at the connected component that contains the root.

Lemma 9. *Let $C \in \Gamma$ be a connected component of bought edges containing the root and let $C_1, C_2, \dots, C_k \in \Gamma$ be the connected components of bought edges that are immediately below C . Then for any player i in C the strategy p_i that buys all the edges between C and C_1, C_2, \dots, C_k is stable.*

Proof. Let L_C denote the set of edges between C and C_1, C_2, \dots, C_k . Observe that even though all the edges in L_C are not bought, a player j may have made a payment $p_j(e) < c(e)$ for some edge $e \in L_C$ in the first phase of the algorithm. Therefore, the cost of the strategy p_i of player $i \in C$ that buys all the edges of L_C , which we denote by l_i , may be less than $\sum_{e \in L_C} c(e)$. More precisely, $l_i = \sum_{e \in L_C} (c(e) - \sum_j p_j(e))$.

We claim that a strategy p_i of a player $i \in C$ that buys all the edges in L_C is stable. For the purpose of contradiction, assume p_i is not a stable strategy. Then, player i has a deviation $\chi_i(p_i)$ from p_i such that the cost of $\chi_i(p_i)$ is less than l_i and therefore the subgraph $G(p^* - p_i + \chi_i(p_i))$ is cheaper than OPT. Since player i did not make any payment for the edges in C , all the players in C are in the same connected component of $G(p^* - p_i + \chi_i(p_i))$ and therefore are satisfied. If the players in C, C_1, C_2, \dots, C_k are also in the same connected component with i in $G(p^* - p_i + \chi_i(p_i))$ then $G(p^* - p_i + \chi_i(p_i))$ satisfies all the players and is cheaper than OPT. Therefore, the players in some of the connected components C_1, C_2, \dots, C_k are not in the same connected component with i in $G(p^* - p_i + \chi_i(p_i))$.

Let D be the subset of connected components C_1, C_2, \dots, C_k the players of which are not in the same connected component with i in $G(p^* - p_i + \chi_i(p_i))$. Let e_1, \dots, e_d be the edges that were unpaid for in the first phase of the algorithm directly above the components in D . Then by Lemma 6, there is a set of players S such that all the players in T_{e_1}, \dots, T_{e_d} are satisfied in the graph $G(p^* - \sum_{j \in S} p_j + \sum_{j \in S} \chi_j(p_j)) - \overline{T_{e_1, e_2, \dots, e_d}}$. We claim that the subgraph $R = G(p^* - \sum_{j \in S \cup i} p_j + \sum_{j \in S \cup i} \chi_j(p_j))$ satisfies all the players in T and is cheaper than OPT. The latter is clear by definition of $\chi_i(p_i)$. All players in C and in the subtrees below $C_j \notin D$ are satisfied in R , since they are satisfied in $G(p^* - p_i + \chi_i(p_i))$, and by construction, the payments p_j for $j \in T_{e_1}, \dots, T_{e_d}$ do not contain edges of $\overline{T_{e_1, e_2, \dots, e_d}}$. All players in components of D are satisfied in R as well, since the only edges missing from R that were in $G(p^* - \sum_{j \in S} p_j + \sum_{j \in S} \chi_j(p_j)) - \overline{T_{e_1, e_2, \dots, e_d}}$ are edges of p_i , which are all edges of $\overline{T_{e_1, e_2, \dots, e_d}}$. Therefore, all the edges are still there that are needed to make the players in D (and the subtrees below them) be satisfied. Since we constructed a subgraph that satisfies all players and is cheaper than OPT, we have a contradiction. ■

Lemma 10. *Price of stability is 1 for the Group Network Formation of Couples Game even if some of the edges of OPT are witnessed from one side.*

Proof. We will show the result by slightly modifying the first phase of the algorithm. Recall that by Corollary 1, the edges witnessed from 2-sides form a connected component of T , which we will refer to as D , and so we root the tree T at a node in D . Observe that there exists a subset S of nodes of D such that all the edges that are witnessed from 1-side are subtrees of T rooted at some the nodes of S . If OPT has edges witnessed from 1-side, i.e., not all nodes of S are leaf nodes, we ask the players to buy the edges witnessed from 1-side first. Specifically, for each $i \in S$, we ask the players in the subtree of 1-sided edges rooted at i to buy all the edges in their subtree, using the algorithm from [2] for the Single Source Connection Game. However, we ask only one player per terminal node to form payments in this algorithm. The proof that this set of players can indeed buy all the edges of this subtree using stable payments is exactly the same as the proof of the Single Source Connection Game payment algorithm so we will not repeat it here.

Once we have formed the payment on the edges witnessed from 1-side, we use our payment algorithm. If i is a terminal node, one of the players in i pays for the higher level incident edge of i in D . Let us now consider the case where i is a high degree nonterminal node. If the subtree composed of edges that are witnessed from 1-side has more than one

terminal nodes, we contract the subtree and ask a second player j in one of these terminal nodes to pay for the higher level incident edge of i . The key observation here is that there are at least 2 players in the subtree (one per each terminal node) that did not make any payment yet, and so Lemma 8 still holds right after player j makes her payments. If the subtree has only one terminal node then the payment for the higher level incident edge depends on the number of lower level incident edges of i that are bought. If only one of the lower level incident edges of i is bought, the one that belongs to the subtree of edges witnessed from 1-side, then we ask this player that bought all the edges of the subtree to pay for the higher level incident edge of i . If at least 2 of the lower level incident edges of i is bought then 2 connected components of bought edges merge at i and therefore we ask a player that has not made payment yet to pay for the higher level incident edge of i . ■

The proof of our 2-approximate Nash equilibrium result suggests an algorithm which forms a cheaper network whenever a 2-approximate Nash equilibrium cannot be found. Using techniques similar to [2], this allows us to form efficient algorithms to compute approximate equilibria:

Theorem 3. *Suppose we have an α -approximate socially optimal graph G_α for an instance of the Group Network Formation Game. Then for any $\epsilon > 0$, there is a polynomial time algorithm which returns a $2(1+\epsilon)$ -approximate Nash equilibrium on a feasible graph G' , where $\text{cost}(G') \leq \text{cost}(G_\alpha)$. Furthermore, if all the terminal nodes have an associated player or each terminal node is associated with at least 2 players, there is a polynomial time algorithm which returns a $(1+\epsilon)$ -approximate Nash equilibrium on a feasible graph G' .*

Proof. We will first prove that given an α -approximate socially optimal graph G_α for an instance of the Group Network Formation of Couples Game (or Group Network Formation Game where all terminal nodes have an associated player) and any $\epsilon > 0$, there is a polynomial time algorithm which returns a $(1+\epsilon)$ -approximate Nash equilibrium on a feasible graph G' , where $\text{cost}(G') \leq \text{cost}(G_\alpha)$.

To define our algorithm, recall that the proof in Section 4 (and Section 3) followed by contradiction since the network at hand was optimal. The proof for obtaining a $(1+\epsilon)$ -approximate Nash equilibrium in polynomial time on a given α -approximate socially optimal network G_α is based on following these suggested algorithms to obtain a cheaper network whenever a Nash equilibrium cannot be found. However, the improvements we consider should be substantial enough to ensure the time-bound, while they should be small enough to ensure the approximation ratio.

To find a $(1+\epsilon)$ -approximate Nash equilibrium, i.e., a solution where no player can reduce its cost by more than a factor of $(1+\epsilon)$ by taking any deviation, we start by defining $\gamma = \frac{c(G_\alpha)\epsilon}{\alpha(1+\epsilon)m}$, where m is the total number of edges of the graph G . We now use our payment algorithms to pay for all but γ of each edge in G_α . Since G_α is not optimal, it is possible that even with the γ reduction in price, the algorithm may not buy one of the incident edges of the root in the first phase or a player in a connected component of bought edges C may not be able to pay for all the edges between C and lower level components (a player may not buy a higher level incident edge witnessed from 1-side or all of her lower level incident edges when all nodes are terminals). However, the proofs of Lemma 7 and Lemma 9 (the proofs of Lemma 1 and Lemma 2 when all nodes are terminals) indicate how we can rearrange G_α to reduce its cost. If we modify G_α in this manner, it is easy to show that we have reduced the cost by at least γ .

Each call to our payment algorithm can be made to run in polynomial time. Since each call which fails to form the payments reduces the cost by γ , we can have at most $\frac{\alpha(1+\epsilon)m}{\epsilon}$ calls. Therefore, in time polynomial in m and ϵ^{-1} , we obtained a network G' with $c(G') \leq c(G_\alpha)$ such that we have a 2-approximate Nash equilibrium on G' if the cost of its edges were decreased by γ .

For all payment strategies p_i and for each edge e in G' , we now increase $p_i(e)$ in proportion to p_i so that e is now fully paid for. Now clearly G' is fully paid for. To show that we obtained a $(1 + \epsilon)$ -approximate Nash equilibrium, all we need to show is that each stable strategy p_i became a $(1 + \epsilon)$ -approximately stable after they are proportionally increased.

Observe that the payment player i makes is increased to $\frac{c(G')p_i}{c(G') - m'\gamma}$, where m' denotes the number of edges in G' . To see that this is $(1 + \epsilon)$ -approximate Nash equilibrium, note that p_i was a stable payment before it was increased and therefore $\chi_i(p_i)$, deviation of player i with respect to p_i , was as expensive as p_i . Therefore, by deviating with respect to p_i , player i can gain at most a factor of

$$\frac{c(G')}{c(G') - m'\gamma} \leq \frac{c(G')}{c(G') - \frac{m'c(G_\alpha)\epsilon}{\alpha(1+\epsilon)m}} \leq \frac{c(G')}{c(G') - \frac{c(G')\epsilon}{(1+\epsilon)}} = (1 + \epsilon).$$

We have given a polynomial-time algorithm which returns a $(1 + \epsilon)$ -approximate Nash equilibrium on a feasible graph G' , where $cost(G') \leq cost(G_\alpha)$ for the Group Network Formation Game where all nodes of terminals or each terminal node has at least 2 players associated with.

Given an instance $\mathfrak{S}_1 = (N_1, G, T, F)$ of Group Network Formation Game, we can obtain an instance $\mathfrak{S}_2 = (N_2, G, T, F)$ of Group Network Formation of Couples Game that has twice as many players on the same graph G , with the same set of terminals T and the same happiness function F such that each player $i \in N_1$ has 2 corresponding players $j, k \in N_2$ as illustrated in the proof of Theorem 1. The strategy $p_i = p_j + p_k$ is $2(1 + \epsilon)$ -approximately stable for player i since the costs of both p_j and p_k are within a factor of $(1 + \epsilon)$ of the cost of $\chi_i(p_i)$. ■

Since for all monotone functions F , finding OPT is a constrained forest problem [12], then Theorem 3 gives us a poly-time algorithm for $\alpha = 2$.

5 Inapproximability Results and Terminal Backup

Recall that in this paper, we consider games where the happiness functions are monotone. Theorem 4 shows that this property of happiness functions is critical for even approximate stability.

Theorem 4. *For the Group Network Formation Game where the happiness functions may not be monotone, there is no α -approximate Nash equilibrium for any α .*

Proof. To prove that there is no approximate Nash equilibrium for the Group Network Formation Game we give such an instance of the problem as shown in Figure 4(A). All nodes are player nodes. We define a component to be happy if all the players in it are happy according to the following connectivity requirements: The player on the left is happy if it is connected to at most one other terminal. The 2 players on the right are happy if and only if they are connected to exactly 1 other terminal. The cost of each edge is as given in Figure

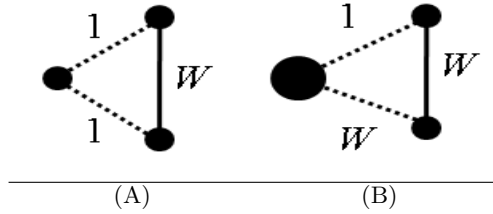


Fig. 4. (A) An instance of a game where there is no approximate Nash equilibria if the happiness function is not monotone. (B) An instance of a Group Network Formation Game where there is no approximate Nash equilibrium on OPT if the fair sharing cost-sharing mechanism is used.

4(A), where W is very large. Observe that there is only one feasible solution of the game, i.e., all the players are happy and where only the edge whose cost is W is purchased. The player on the left cannot contribute to the cost of the edge in any approximate Nash equilibrium since it is already happy and therefore has a deviation of cost 0. Since the other 2 players have to be sharing the cost of W , at least one of them should be paying at least $W/2$. Since that player has a deviation of cost 1, at least one of the players can reduce its cost by a factor of $W/2$ by deviating and therefore there is no α -approximate Nash equilibrium for this game where $\alpha < W/2$. Since W can be arbitrarily large and it is independent of the number of players, there is no approximate Nash equilibrium for that instance of the general Group Network Formation Game. ■

Recall that congestion games, including our game with fair sharing, are guaranteed to have Nash equilibria, although they may be expensive. The following theorem studies the quality (cost) of approximate Nash equilibrium and shows that there may not be any approximately stable solution that is as cheap as the socially optimal network.

Theorem 5. *For the Group Network Formation Game, there may not be any approximate Nash equilibrium whose cost is as much as OPT if the fair cost-sharing mechanism is used.*

Proof. In Figure 4(B), all nodes are player nodes. We define a component to be happy if all the players in it are happy according to the following connectivity requirements: The large node on the left is happy always, and the 2 nodes on the right want to connect to at least one other terminal. In OPT, the 2 players on the right would have to buy the edge between them which has a cost of W . When the fair sharing cost scheme is used, both of the nodes have to make a payment of $W/2$ on that edge even though the top player has a deviation of cost 1. Since the top player can reduce its cost by a factor of $W/2$ by deviating, there is no α -approximate Nash equilibrium for this game where $\alpha < W/2$. Since W can be arbitrarily large and it is independent of the number of players, there is no approximate Nash equilibrium on OPT. ■

Because of its applications to multi-homing [3, 22], we are especially interested in the behavior of Terminal Backup connectivity requirements, i.e., when a player node desires to connect to at least $k - 1$ other player nodes.

Theorem 6. *For the Group Network Formation Game and the Terminal Backup problem, the Price of Anarchy is n and $2k - 2$ respectively. Furthermore, these bounds are tight.*

Proof. The *price of anarchy* refers to the ratio of the worst (most expensive) Nash equilibrium and the optimal centralized solution. In the Group Network Formation Game, the price of anarchy is at most N , the number of players. This is simply because if the worst Nash equilibrium p costs more than N times OPT, the cost of the optimal solution, then there must be a player whose payments in p are strictly more than OPT, so he could deviate by purchasing the entire optimal solution by himself, and form a connected components that makes her happy with smaller payments than before. More importantly, there are cases when the price of anarchy actually equals N . This is demonstrated with the example in Figure 5(A).

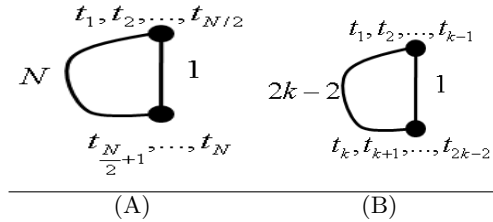


Fig. 5. (A) An instance of a Group Network Formation Game that has a Nash equilibrium whose cost is N times more than the socially optimal network. (B) An instance of Terminal Backup problem that has a Nash equilibrium whose cost is $2k - 2$ times more than the socially optimal network.

Suppose there are N players, and G consists of 2 nodes which are joined by 2 disjoint paths, one of cost 1 and and one of cost N . Half of the players have their terminal at one node and the other half of the players have their terminal at the other node. The only happy components must include both nodes. This corresponds to Terminal backup requirements where every player wants to be connected to at least $N/2 + 1$ terminals. Then, the worst Nash equilibrium has each player contributing 1 to the long path, and has a cost of N . The optimal solution here has a cost of only 1, so the price of anarchy is N . Therefore, the price of anarchy could be very high in the Group Network Formation Game.

We are also interested in the price of anarchy of the Terminal Backup problem which is a special case of Group Formation Games. In Figure 5(B), we give an example where price of anarchy is $2k - 2$ and we prove that this bound is tight below. In this case, k is the connectivity requirement: components are happy if and only if they contain at least k terminal nodes.

To prove the result all we need to do is to show that no equilibrium will cost more than $2k - 2$ times OPT since Figure 5(B) shows an instance of a Terminal Backup problem with an equilibrium whose cost is exactly $2k - 2$ times the cost of OPT.

For the purpose of contradiction, assume there is an instance of a Terminal Backup problem that has an equilibrium whose cost is more than $2k - 2$ times the cost of OPT. Let EEQ denote this expensive equilibrium solution. Observe that there must exist a connected component T of OPT such that the total payments of the players of T in the expensive equilibrium solution EEQ is more than $2k - 2$ times the cost of T . Since EEQ is a Nash equilibrium, no player in T pays more than the cost of C . Therefore, T includes more than $2k - 2$ terminals. However, as we have shown in the proof of Theorem 7, there exists a set

of edges C_i for every player i such that $C_i \subseteq T$, C_i contains at least k terminals including i , and every edge of T is contained in no more than $2k - 2$ sets C_i .

We define α_i be the cost of C_i , which is at least the cost of connecting a player i in T to at least $k - 1$ other players in T . Since no edge of T is used by more than $2k - 2$ sets C_i , we know that $\sum_{i \in T} \alpha_i \leq (2k - 2) c(T)$ where $c(T)$ denotes the cost of the component T .

However, in EEQ, the total payment of all the players in T is more than $2k - 2$ times the cost of T , i.e., $\sum_{i \in T} p_i > (2k - 2) c(T)$. Therefore, $\sum_{i \in T} p_i > \sum_{i \in T} \alpha_i$ which implies that there exists a player i in T such that $p_i > \alpha_i$. Therefore, EEQ cannot be a Nash equilibrium since player i can reduce its cost to α_i by deviating.

The lower bounds for Terminal Backup also hold for the general Group Network Formation Game, showing that while the price of stability may be low, the price of anarchy can be as high as the number of players. ■

Theorem 7. *For the Terminal Backup problem, in the Shapley cost-sharing model, the price of stability is at most $H(2k - 2)$.*

Proof. The Terminal Backup problem, under the Shapley cost-sharing model (or fair sharing), falls into a class of games called the congestion games (or potential games). In this model, the strategy of a player i is just a set of edges S_i that contains at least k terminals including i , and the edges that are built are $\cup_i S_i$. The players split the cost of every edge evenly, i.e., a player i using edge $e \in S_i$ must pay $c(e)/x_e$, where $x_e = |\{S_i | e \in S_i\}|$. [1] showed that in a network design game with the Shapley-cost sharing mechanism, the cost of the Nash equilibrium obtained by the best-response dynamics starting from OPT is at most $H(n)$ times more expensive than OPT, where $H(n)$ is the n^{th} harmonic number and n is the maximum number of players using a single edge on OPT for their connectivity requirements. Therefore, to obtain a price of stability bound for the Terminal Backup problem, all we need to do is to select sets S_i for the players in OPT while ensuring that no edge appears in too many sets S_i .

To prove the result, we need to show that we can select sets S_i for the players on OPT such that no edge is used more than $2k - 2$ players. We first start by replacing all the edges of OPT by 2 directed edges. We can now form an Euler tour in each connected component of OPT since every vertex has an equal number of incoming and outgoing edges incident to it. Then, we ask each player i to follow the Euler tour in the clockwise direction until it encounters $k - 1$ other distinct terminals. We set S_i to be this set of edges. Observe that each directed edge appears in at most $k - 1$ sets S_i belonging to distinct closest players in the counterclockwise direction. Since each undirected edge of OPT had been replaced with 2 directed edges, each edge of OPT is used by at most $2k - 2$ sets S_i . ■

References

1. E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, T. Roughgarden. The Price of Stability for Network Design with Fair Cost Allocation. *SIAM Journal on Computing*, Volume 38, Issue 4 (November 2008), pp. 1602-1623.
2. E. Anshelevich, A. Dasgupta, É. Tardos, T. Wexler. Near-Optimal Network Design with Selfish Agents. In *Theory of Computing*, Volume 4 (2008), pp. 77-109.
3. E. Anshelevich and A. Karagiozova. Terminal Backup, 3D Matching, and Covering Cubic Graphs. In *Proc. 39th ACM Symposium on Theory of Computing (STOC 2007)*.
4. C. Chekuri, J. Chuzhoy, L. Lewin-Eytan, J. Naor, and A. Orda. Non-cooperative multicast and facility location games. Proceedings of the 7th ACM Conference on Electronic Commerce (EC), Ann Arbor, Michigan (2006), pp. 72-81.

5. H. Chen, T. Roughgarden. Network Design with Weighted Players. In *SPAA 2006*.
6. H. Chen, T. Roughgarden, and G. Valiant. Designing Networks with Good Equilibria. *SODA 2008*.
7. G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. *ESA, 2005*.
8. A. Epstein, M. Feldman, and Y. Mansour. Strong Equilibrium in Cost-Sharing Connection Games. *EC 2007*.
9. A. Fabrikant, C. Papadimitriou, K. Talwar. The complexity of pure Nash equilibria. In *STOC, 2004*.
10. A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, R. Shabo. On the Price of Stability for Designing Undirected Networks with Fair Cost Allocations. *Proceedings of ICALP 2006*, pp. 608–618.
11. N. Garg, G. Konjevod, R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *SODA 2000*.
12. M. Goemans, and D. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM Journal on Computing*, 24:296–317, 1995.
13. M. Hoefer. Non-cooperative Facility Location and Covering Games. In *ISAAC 2006*.
14. M. Hoefer. Non-cooperative Tree Creation. In *MFCS 2006*.
15. M. Hoefer, P. Krysta. Geometric Network Design with Selfish Agents. *COCOON 2005*.
16. R. Holzman, N. Law-Yone. Strong Equilibrium in congestion games. *Games and Economic Behavior*, 21, 1997.
17. M. Jackson, A survey of models of network formation: stability and efficiency. *Group Formation in Economics: Networks, Clubs and Coalitions*, eds. G. Demange and M. Wooders, Cambridge Univ. Press.
18. D. Monderer, L. Shapley. Potential Games. *Games and Economic Behavior* 14(1996), 124–143.
19. N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani (eds.), *Algorithmic Game Theory*, Cambridge University Press.
20. R. W. Rosenthal. The network equilibrium problem in integers. *Networks*, 3:53-59, 1973.
21. T. Roughgarden. *Selfish Routing and the Price of Anarchy*, MIT Press.
22. D. Xu, E. Anshelevich, and M. Chiang. On Survivable Access Network Design: Complexity and Algorithms. *INFOCOM 2008*.