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 heterogonous 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 equilibrium[1], respectively, and that of the globally optimal solution.

[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.

A. Fiat and P. Sanders (Eds.): ESA 2009, LNCS 5757, pp. 239–250, 2009.

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 [10]). Or instead, a player might want to connect 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 [21]. 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 $p = (p_1, \ldots, p_N)$. 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 [11]. 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:

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

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 welfare[2]. 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 [11]. 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 [2,8,12,13]. 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.*

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

Related Work. Over the last few years, there have been several new papers using arbitrary cost-sharing, e.g., [8,13,14]. Recently, Hoefer [12] 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 [1,4,5,9]. 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" [4,7,15,20]. 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 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 [16,18,20] 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 [1,20], 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 [11], 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 [3,21], 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 subgraph S.

Observation 1. *Since the satisfaction of the players only depends on the terminal nodes they are connected to, OPT is acyclic and therefore, OPT 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. In the full version of the paper, we show that the edges of T witnessed from 2-sides form a connected component in T. Due to limited space, all our proofs are omitted but the full version of the paper is available online at www.cs.rpi.edu/∼eanshel.

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 $c'(f)$. 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

```
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) - Σ_{j≠i} p_j(e);
            If χ_i(p_i) - Σ_f p_i(f) ≥ 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{χ_i(p_i) - Σ_f p_i(f), c(g)};
```

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

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.

Recall 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 prove this by constructing a network cheaper than OPT which satisfies all the players whenever the algorithm breaks, thus forming a contradiction in the full version of the paper.

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 $p = (p_1, p_2, \ldots, p_n)$ 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.*

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 [1,20], 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, 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

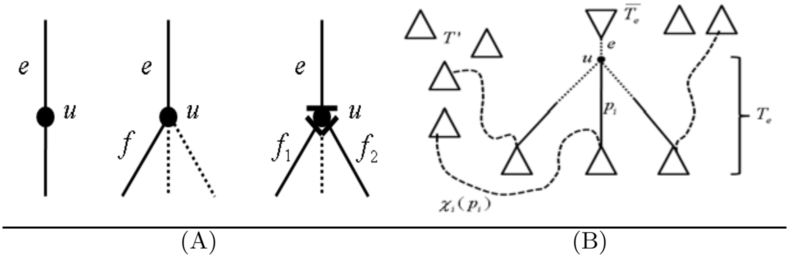

Fig. 1. (A) Illustrates the assignment of the player to pay for the cost of e. (B) Shows how to construct a cheap network that satisfies all the players in T_e by using the deviations of a subset S of them.

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 $f_1, f_2, \ldots, f_l$, then we fix the strategies of the players $i_1, i_2, \ldots, i_l$ that bought those edges, i.e., the players $i_1, i_2, \ldots, i_l$ 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 $i_1, i_2, \ldots, i_l$ 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 prove in the full version that such a player always exists, i.e., not all of $i_1, i_2, \ldots, i_l$ are the last players to make payment at their respective terminal nodes.

We here present an outline of our analysis of this 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 give a series of lemmas in the full version 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.

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 $G_p - T'$ 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 connected 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 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. We prove that such a player i always exists in the full version of the paper. Observe that once i buys all the edges between C and the connected components $C_1, C_2, \ldots, C_k$ that are immediately below C, all these $k+1$ connected components form a single connected component 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 $C_1, C_2, \ldots, C_k$ that are immediately below C is a stable strategy for any player in C, which we show in the full version of the paper.

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 the full version of the paper, 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.

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 $cost\,(G') \leq cost\,(G_\alpha)$. 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'.*

Since for all monotone functions F, finding OPT is a constrained forest problem [11], 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.*

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.*

Because of its applications to multi-homing [3,21], 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. For the Terminal Backup problem, in the Shapley cost-sharing model, the price of stability is at most $H(2k-2)$.*

References

1. Anshelevich, E., Dasgupta, A., Kleinberg, J., Tardos, É., Wexler, T., Roughgarden, T.: The Price of Stability for Network Design with Fair Cost Allocation. SIAM Journal on Computing 38(4), 1602–1623 (2008)
2. Anshelevich, E., Dasgupta, A., Tardos, É., Wexler, T.: Near-Optimal Network Design with Selfish Agents. Theory of Computing 4, 77–109 (2008)
3. Anshelevich, E., Karagiozova, A.: Terminal Backup, 3D Matching, and Covering Cubic Graphs. In: Proc. 39th ACM Symposium on Theory of Computing (2007)
4. Chekuri, C., Chuzhoy, J., Lewin-Eytan, L., Naor, J., Orda, A.: Non-cooperative multicast and facility location games. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC), Ann Arbor, Michigan, pp. 72–81 (2006)
5. Chen, H., Roughgarden, T.: Network Design with Weighted Players. In: SPAA 2006 (2006)
6. Chen, H., Roughgarden, T., Valiant, G.: Designing Networks with Good Equilibria. In: SODA 2008 (2008)
7. Christodoulou, G., Koutsoupias, E.: On the price of anarchy and stability of correlated equilibria of linear congestion games. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 59–70. Springer, Heidelberg (2005)
8. Epstein, A., Feldman, M., Mansour, Y.: Strong Equilibrium in Cost-Sharing Connection Games. In: EC 2007 (2007)
9. Fiat, A., Kaplan, H., Levy, M., Olonetsky, S., Shabo, R.: On the Price of Stability for Designing Undirected Networks with Fair Cost Allocations. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 608–618. Springer, Heidelberg (2006)
10. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner tree problem. In: SODA 2000 (2000)
11. Goemans, M., Williamson, D.: A General Approximation Technique for Constrained Forest Problems. SIAM Journal on Computing 24, 296–317 (1995)
12. Hoefer, M.: Non-cooperative Facility Location and Covering Games. In: Asano, T. (ed.) ISAAC 2006. LNCS, vol. 4288, pp. 369–378. Springer, Heidelberg (2006)
13. Hoefer, M.: Non-cooperative Tree Creation. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 517–527. Springer, Heidelberg (2006)
14. Hoefer, M., Krysta, P.: Geometric Network Design with Selfish Agents. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 167–178. Springer, Heidelberg (2005)
15. Holzman, R., Law-Yone, N.: Strong Equilibrium in congestion games. Games and Economic Behavior 21 (1997)
16. Jackson, M.: A survey of models of network formation: stability and efficiency. In: Demange, G., Wooders, M. (eds.) Group Formation in Economics: Networks, Clubs and Coalitions. Cambridge Univ. Press, Cambridge
17. Monderer, D., Shapley, L.: Potential Games. Games and Economic Behavior 14, 124–143 (1996)
18. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V. (eds.): Algorithmic Game Theory. Cambridge University Press, Cambridge
19. Rosenthal, R.W.: The network equilibrium problem in integers. Networks 3, 53–59 (1973)
20. Roughgarden, T.: Selfish Routing and the Price of Anarchy. MIT Press, Cambridge
21. Xu, D., Anshelevich, E., Chiang, M.: On Survivable Access Network Design: Complexity and Algorithms. In: INFOCOM 2008 (2008)