NEAR-OPTIMAL NETWORK DESIGN WITH SELFISH AGENTS*

ELLIOT ANSHELEVICH†, ANIRBAN DASGUPTA‡, ÉVA TARDOS §, AND TOM WEXLER ¶

Abstract. We introduce a simple network design game that models how independent selfish agents can build or maintain a large network. In our game every agent has a specific connectivity requirement, i.e. each agent has a set of terminals and wants to build a network in which his terminals are connected. Possible edges in the network have costs and each agent's goal is to pay as little as possible. Determining whether or not a Nash equilibrium exists in this game is NP-complete. However, when the goal of each player is to connect a terminal to a common source, we prove that there is a Nash equilibrium as cheap as the optimal network, and give a polynomial time algorithm to find a $(1+\varepsilon)$-approximate Nash equilibrium that does not cost much more. For the general connection game we prove that there is a 3-approximate Nash equilibrium that is as cheap as the optimal network, and give an algorithm to find a $(4.65+\varepsilon)$-approximate Nash equilibrium that does not cost much more.

Key words. Game Theory, Network Design, Nash Equilibrium, Connection Game, Price of Stability

AMS subject classifications.

1. Introduction. Many networks, including the Internet, are developed, built, and maintained by a large number of agents (Autonomous Systems), all of whom act selfishly and have relatively limited goals. This naturally suggests a game-theoretic approach for studying both the behavior of these independent agents and the structure of the networks they generate. The stable outcomes of the interactions of non-cooperative selfish agents correspond to Nash equilibria. Typically, considering the Nash equilibria of games modeling classical networking problems gives rise to a number of new issues. In particular, Nash equilibria in network games can be much more expensive than the best centralized design. Papadimitriou [31] uses the term *price of anarchy* to refer to this increase in cost caused by selfish behavior. The price of anarchy has been studied in a number of games dealing with various networking issues, such as load balancing [14, 15, 28, 35], routing [34, 36, 37], facility location [39], and flow control [2, 16, 38]. In some cases [34, 36] the Nash equilibrium is unique, while in others [28] the best Nash equilibrium coincides with the optimum solution and the authors study the quality of the worst equilibrium. However, in some games the quality of even the best possible equilibria can be far from optimal (e.g. in the prisoner's dilemma). The best Nash equilibrium can be viewed as the best solution that selfish agents can agree upon, i.e. once the solution is agreed upon, the agents do not find it in their interest to deviate. Papadimitriou [31] defines the price of anarchy to study the question of how *bad* an equilibrium can be. We study the complementary question of *how good an equilibrium can be* in the context of a network design game.

*A preliminary version of this paper appeared in *Proc. 35th Annual ACM Symposium on Theory of Computing*, 2003.

†Department of Computer Science, Rennselaer Polytechnic Institute, Troy, NY 12180. Email: `eanshel@cs.rpi.edu` Research supported in part by an NSF graduate fellowship.

‡Yahoo Research. Email: `anirban@yahoo-inc.com`. Research supported by the Computer Science Department of Cornell University.

§Department of Computer Science, Cornell University, Ithaca, NY 14853. Email: `eva@cs.cornell.edu`. Research supported in part by ONR grant N00014-98-1-0589.

¶Department of Computer Science, Cornell University, Ithaca, NY 14853. Email: `wexler@cs.cornell.edu`. Research supported in part by an NSF graduate fellowship.

Schultz and Stier [37] study the ratio of the best equilibrium to the optimum, in the context of a capacitated routing game. We call this ratio the *price of stability*, a term introduced in [6].[1]

In this paper we consider a simple network design game where every agent has a specific connectivity requirement, i.e. each agent has a set of terminals and wants to build a network in which his terminals are connected. Possible edges in the network have costs and each agent's goal is to pay as little as possible. This game can be viewed as a simple model of network creation. Alternatively, by studying the best Nash equilibria, our game provides a framework for understanding those networks that a central authority could persuade selfish agents to purchase and maintain, by specifying to which parts of the network each agent contributes. An interesting feature of our game is that selfish agents will find it in their individual interests to *share* the costs of edges, and so effectively cooperate.

More precisely, we study the following network game for N players, which we call the *connection game*. For each game instance, we are given an undirected graph G with non-negative edge costs. Players form a network by purchasing some subgraph of G. Each player has a set of specified terminal nodes that he would like to see connected in the purchased network. With this as their goal, players offer payments indicating how much they will contribute towards the purchase of each edge in G. If the players' payments for a particular edge e sum to at least the cost of e, then the edge is considered *bought*, which means that e is added to our network and can now be used by any player. Each player would like to minimize his total payments, but insists on connecting all of his terminals. We allow the cost of any edge to be shared by multiple players. Furthermore, once an edge is purchased, any player can use it to satisfy his connectivity requirement, even if that player contributed nothing to the cost of this edge. Finding the centralized optimum of the connection game, i.e. the network of bought edges which minimizes the sum of the players' contributions, is the classic network design problem of the generalized Steiner tree [1, 22]. We are most interested in deterministic Nash equilibria of the connection game, and in the price of stability, as the price of anarchy in our game can be quite bad. In a game theoretic context it might seem natural to also consider *mixed* Nash equilibria when agents can randomly choose between different strategies. However, since we are modeling the construction of large-scale networks, randomizing over strategies is not a realistic option for players.

Our Results. We study deterministic Nash equilibria of the connection game, and prove bounds on the price of stability. We also explore the notion of an *approximate equilibrium*, and study the question of how far from a true equilibrium one has to get to be able to use the optimum solution, i.e. how unhappy would the agents have to be if they were forced to pay for the socially optimal design. We view this as a two parameter optimization problem: we would like to have a solution with cost close to the minimum possible cost, and where users would not have large incentives to deviate. Finally, we examine how difficult it is to find equilibria at all.

Our results include the following.

- In Section 3 we consider the special case when the goal of each player is to connect a single terminal to a common source. We prove that in this case, there is a Nash equilibrium, the cost of which is equal to the cost of the optimal network. In other words, with a single source and one terminal per player, the price of stability is 1.

[1] In the conference version of our paper we used the term *optimistic price of anarchy* instead.

Furthermore, given an $\varepsilon > 0$ and an α-approximate solution to the optimal network, we show how to construct in polynomial time an $(1+\varepsilon)$-approximate Nash equilibrium (players only benefit by a factor of $(1+\varepsilon)$ in deviating) whose total cost is within a factor of α to the optimal network.

We generalize these results in two ways. First, we can extend the results to the case when the graph is directed and players seek to establish a directed path from their terminal to the common source. Note that problems in directed graphs are often significantly more complicated than their undirected counterparts [10, 19]. Second, players do not have to insist on connecting their terminals at all cost, but rather each player i may have a maximum cost $\max(i)$ that he is willing to pay, and would rather stay unconnected if his cost exceeds $\max(i)$.

• In Section 4 we consider the general case, when players may want to connect more than 2 terminals, and they do not necessarily share a single source node. In this case, there may not exist a deterministic Nash equilibrium. When deterministic Nash equilibria do exist, the costs of different equilibria may differ by as much as a factor of N, the number of players, and even the price of stability may be nearly N. However, in Section 4 we prove that there is always a 3-approximate equilibrium that pays for the optimal network. Furthermore, we show how to construct in polynomial time a $(4.65+\varepsilon)$-approximate Nash equilibrium whose total cost is within a factor of 2 to the optimal network.

• Finally, in Section 5 we show that determining whether or not a Nash equilibrium exists is NP-complete when the number of players is part of the input. In addition, we give a lower bound on the approximability of a Nash on the centralized optimum in our game.

Related Work. We view our game as a simple model of how different service providers build and maintain the Internet topology. We use a game theoretic version of network design problems considered in approximation algorithms [22]. Fabrikant et al [18] study a different network creation game. Network games similar to that of [18] have also been studied for modeling the creation and maintenance of social networks [9,23]. In the network game considered in [3,9,18,23] each agent corresponds to a single node of the network, and agents can only buy edges adjacent to their nodes. This model of network creation seems extremely well suited for modeling the creation of social networks. However, in the context of communication networks like the Internet, agents are not directly associated with individual nodes, and can build or be responsible for more complex networks. There are many situations where agents will find it in their interest to *share* the costs of certain expensive edges. An interesting feature of our model which does not appear in [9, 18, 23] is that we allow agents to share costs in this manner. To keep our model simple, we assume that each agent's goal is to keep his terminals connected, and agents are not sensitive to the length of the connecting path.

Since the proceedings version of this paper [7], there have been several new papers about the connection game, e.g., [5, 8, 13, 17, 24–26]. Probably the most relevant such model to our research is presented in [6] (and further addressed in [11, 12, 21]). In [6], extra restrictions of "fair sharing" are added to the Connection Game, making it a congestion game [33] and thereby guaranteeing some nice properties, like the existence of Nash equilibria and a bounded price of stability. While the connection game is not a congestion game, and is not guaranteed to have a Nash equilibrium, it actually behaves much better than [6] when all the agents are trying to connect to a single common node. Specifically, the price of stability in that case is 1, while the best known bound for the model in [6] is $\frac{\log n}{\log \log n}$ [4]. Moreover, all such models (including cost-sharing models described below) restrict the interactions of the agents to improve the quality of the outcomes, by forcing them to share the costs of edges in a particular way. This does not address the contexts when we are not allowed to place such restrictions on the agents, as would be the case when the agents are building the network together without some overseeing authority. However, as we show in this paper, is is still possible to nudge the agents into an extremely good outcome without restricting their behavior in any way.

Jain and Vazirani [27] study a different cost-sharing game related to Steiner trees. They assume that each player i has a utility u_i for belonging to the Steiner tree. Their goal is to give a truthful mechanism to build a Steiner tree, and decide on cost-shares for each agent (where the cost charged to an agent may not exceed his utility). They design a mechanism where truth-telling is a dominant strategy for the agents, i.e. selfish agents do not find it in their interest to misreport their utility (in hopes of being included in the Steiner tree for smaller costs). Jain and Vazirani give a truthful mechanism to share the cost of the minimum spanning tree, which is a 2-approximation for the Steiner tree problem. This game is quite analogous to our single source network creation game considered in Section 3. We can view the maximum payment $\max(i)$ of agent i as his utility u_i. However, in our game there is no central authority designing the Steiner tree or cost shares. Rather, we study Nash equilibria of our game. Also, in our game, agents must offer payments for each edge of the tree (modeling the cooperation of selfish agents), while in a mechanism design framework, agents pay the mechanism for the service, and do not care what edge they contribute to.

Finally, [7] is the conference version of this paper. It differs in several respects, most notably in the proofs of Section 4.

2. Model and Basic Results.

The Connection Game. We now formally define the connection game for N players. Let an undirected graph $G = (V, E)$ be given, with each edge e having a nonnegative cost $c(e)$. Each player i has a set of terminal nodes that he must connect. The terminals of different players do not have to be distinct. A strategy of a player is a payment function p_i, where $p_i(e)$ is how much player i is offering to contribute to the cost of edge e. Any edge e such that $\sum_i p_i(e) \geq c(e)$ is considered *bought*, and G_p denotes the graph of bought edges with the players offering payments $p = (p_1, \ldots, p_N)$. Since each player must connect his terminals, all of the player's terminals must be connected in G_p. However, each player tries to minimize his total payments, $\sum_{e \in E} p_i(e)$.

A Nash equilibrium of the connection game is a payment function p such that, if players offer payments p, no player has an incentive to deviate from his payments. This is equivalent to saying that if p_j for all $j \neq i$ are fixed, then p_i minimizes the payments of player i. A $(1+\varepsilon)$-approximate Nash equilibrium is a function p such that no player i could decrease his payments by more than a factor of $1 + \varepsilon$ by deviating, i.e. by using a different payment function $p_i{}'$.

Some Properties of Nash Equilibria. Here we present several useful properties of Nash equilibria in the Connection Game. Suppose we have a Nash equilibrium p, and let T^i be the smallest tree in G_p connecting all terminals of player i. It follows from the definitions that (1) G_p is a forest, (2) each player i only contributes to costs of edges on T^i, and (3) each edge is either paid for fully or not at all.

Property 1 holds because if there was a cycle in G_p, any player i paying for any edge of the cycle could stop paying for that edge and decrease his payments while his terminals would still remain connected in the new graph of bought edges. Similarly, Property 2 holds since if player i contributed to an edge e which is not in T^i, then he could take away his payment for e and decrease his total costs while all his terminals would still remain connected. Property 3 is true because if i was paying something for e such that $\sum_i p_i(e) > c_e$ or $c_e > \sum_i p_i(e) > 0$, then i could take away part of his payment for e and not change the graph of bought edges at all.

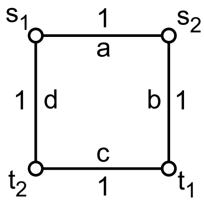

FIG. 2.1. *A game with no Nash equilibria.*

Nash Equilibria May Not Exist. It is not always the case that selfish agents can agree to pay for a network. There are instances of the connection game which have no deterministic Nash equilibria. In Figure 2.1, there are 2 players, one wishing to connect node s_1 to node t_1, and the other s_2 to t_2. Now suppose that there exists a Nash equilibrium p. By Property 1 above, in a Nash equilibrium G_p must be a forest, so assume without loss of generality it consists of the edges a, b, and c. By Property 2, player 1 only contributes to edges a and b, and player 2 only contributes to edges b and c. This means that edges a and c must be bought fully by players 1 and 2, respectively. At least one of the two players must contribute a positive amount to edge b. However, neither player can do that in a Nash equilibrium, since then he would have an incentive to switch to the strategy of only buying edge d and nothing else, which would connect his terminals with the player's total payments being only 1. Therefore, no Nash equilibria exist in this example.

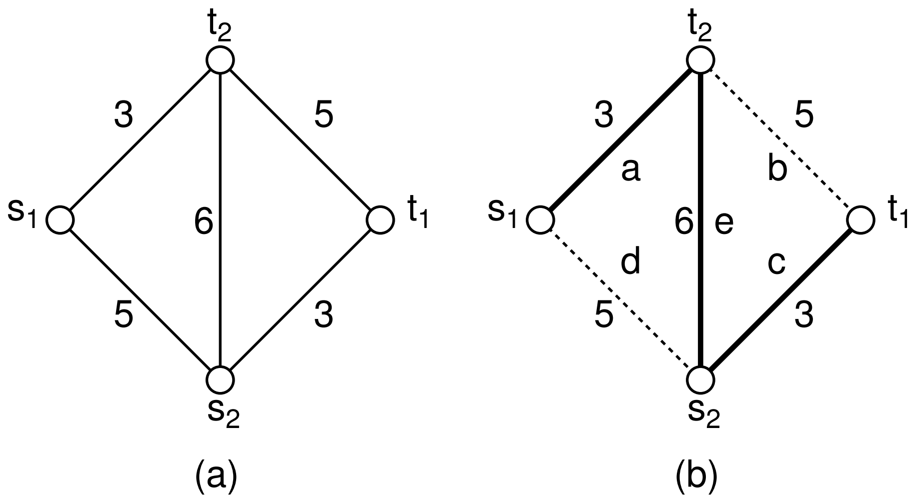

FIG. 2.2. *A game with only fractional Nash equilibria*

Fractional Nash Equilibria. When looking at the connection game, we might be tempted to assume that giving players the opportunity to share costs of edges is an unnecessary complication. However, sometimes players must share costs of edges for all players to agree on a network. There are game instances where the only Nash equilibria in existence require that players split the cost of an edge. We will call such Nash equilibria *fractional* and we will call Nash equilibria that do not involve players sharing costs of edges *non-fractional*.

Figure 2.2(a) is an example of a connection game instance where the only Nash equilibria are fractional ones. Once again, player 1 would like to connect s_1 and t_1, and player 2 would like to connect s_2 and t_2. First, note that there is a fractional Nash equilibrium, as shown in Figure 2.2(b). In this case we have that player 2 contributes 5 to edge e and player 1 contributes 1 to e and 3 to both of a and c. It is easy to confirm that neither player has an incentive to deviate.

Now we must show that there are no non-fractional Nash equilibria in this example. Observe that if edge e is not bought, then we have a graph which is effectively equivalent to the graph in which we showed there to be no Nash equilibria at all. Therefore any non-fractional Nash equilibria must buy edge e. Given that edge e must be bought, it is clear that player 2 will only contribute to edge e. For a Nash equilibrium p to be non-fractional, this would mean that player 2 either buys edge e fully or buys nothing at all. Suppose player 2 buys e. The only response for which player 1 would not want to deviate would be to buy a and c. But then player 2 has an incentive to switch to either edge b or d. Now suppose player 2 does not buy e. Then the only response for which player 1 would not want to deviate would be to either buy a and b or buy c and d. Either way, player 2 does not succeed in joining his source to his sink, and thus has an incentive to buy an edge. Hence, there are no non-fractional Nash equilibria in this graph.

The Price of Anarchy. We have now shown that Nash equilibria do not have to exist. However, when they exist, how bad can these Nash equilibria be? As mentioned above, the price of anarchy often refers to the ratio of the worst (most expensive) Nash equilibrium and the optimal centralized solution. In the connection 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 connect his terminals with smaller payments than before. More importantly, there are cases when the price of anarchy actually equals N, so the above bound is tight. This is demonstrated with the example in Figure 2.3. Suppose there are N players, and G consists of nodes s and t which are joined by 2 disjoint paths, one of length 1 and and one of length N. Each player has a terminal at s and t. 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 connection game. However, notice that in this example the *best* Nash equilibrium (which is each player buying $\frac{1}{N}$ of the short path) has the same cost as the optimal centralized solution. We have now shown that the price of anarchy can be very large in the connection game, but the price of stability remains worth considering, since the above example shows that it can differ from the (conventional) price of anarchy by as much as a factor of N.

All the results in this section also hold if G is directed or if each player i has a maximum cost $\max(i)$ beyond which he would rather pay nothing and not connect

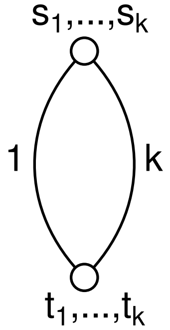

FIG. 2.3. *A game with price of anarchy of N*

his terminals.

3. Single Source Games. As we show in Section 5, determining whether or not Nash equilibria exist in a general instance of the connection game is NP-Hard. Furthermore, even when equilibria exist, they may be significantly more expensive than the centrally optimal network. In this section we define a class of games in which there is always a Nash equilibrium, and the price of stability is 1. Furthermore, we show how we can use an approximation to the centrally optimal network to construct a $(1+\epsilon)$-approximate Nash equilibrium in poly-time, for any $\epsilon > 0$.

DEFINITION 3.1. *A* single source game *is a game in which all players share a common terminal s, and in addition, each player i has exactly one other terminal t_i.*

We will now show that the price of stability is 1 in single source games. To do this, we must argue that there is a Nash equilibrium that purchases T^*, the minimum cost Steiner tree on the players' terminal nodes. There are a number of standard cost-sharing methods for sharing the cost of a tree among the terminals. The two most commonly studied methods are the Shapley value and the Marginal Cost mechanisms [20]. The Marginal Cost (or VCG) mechanisms are very far from being budget balanced, i.e. the agents do not pay for even a constant fraction of the tree built. The Shapley value mechanism is budget balanced: the cost of each edge is evenly shared by the terminals that use the edge for their connection (i.e., the terminals in the subtree below the edge e). However, the mechanism does not lead to a Nash equilibrium in our game: some players can have cheaper alternate paths, and hence benefit by deviating. Jain and Vazirani [27] give a truthful budget balanced cost-sharing mechanism to pay for the minimum spanning tree, which is a 2-approximate budget balanced mechanism for the Steiner tree problem. However, it is only a 2-approximation, and the cost-shares are not associated with edges that the agents use. Here we will show that while the traditional Steiner tree cost-sharing methods do not lead to a Nash equilibrium, such a solution can be obtained.

THEOREM 3.2. *In any single source game, there is a Nash equilibrium which purchases T^*, a minimum cost Steiner tree on all players' terminal nodes.*

Proof. Given T^*, we present an algorithm to construct payment strategies p. We will view T^* as being rooted at s. Let T_e be the subtree of T^* disconnected from s when e is removed. We will determine payments to edges by considering edges in reverse breadth first search order. We determine payments to the subtree T_e before we consider edge e. In selecting the payment of agent i to edge e we consider c', the cost that player i faces if he deviates in the final solution: edges f in the subtree T_e are considered to cost $p_i(f)$, edges f not in T^* cost $c(f)$, while all other edges cost 0. We never allow i to contribute so much to e that his total payments exceed his cost of connecting t_i to s.

ALGORITHM 3.1.

```
Initialize p_i(e) = 0 for all players i and edges e.
Loop through all edges e in T* in reverse BFS order.
  Loop through all players i with t_i ∈ T_e until e paid for.
    If e is a cut in G set p_i(e) = c(e).
    Otherwise
      Define c'(f) = p_i(f) for all f ∈ T* and
             c'(f) = c(f)   for all f ∉ T*.
      Define χ_i to be the cost of the cheapest path from s to
        t_i in G \ {e} under modified costs c'.
      Define p_i(T*) = Σ_{f∈T*} p_i(f).
      Define p(e) = Σ_j p_j(e).
      Set p_i(e) = min{χ_i − p_i(T*), c(e) − p(e)}.
    end
  end
end
```

We first claim that if this algorithm terminates, the resulting payment forms a Nash equilibrium. Consider the algorithm at some stage where we are determining i's payment to e. The cost function c' is defined to reflect the costs player i faces if he deviates in the final solution. We never allow i to contribute so much to e that his total payments exceed his cost of connecting t_i to s. Therefore it is never in player i's interest to deviate. Since this is true for all players, p is a Nash equilibrium.

We will now prove that this algorithm succeeds in paying for T^*. In particular, we need to show that for any edge e, the players with terminals in T_e will be willing to pay for e. Assume the players are unwilling to buy an edge e. Then each player has some path which explains why it can't contribute more to e. We can use a carefully selected subset of these paths to modify T^*, forming a cheaper tree that spans all terminals and doesn't contain e. This would clearly contradict our assumption that T^* had minimum cost.

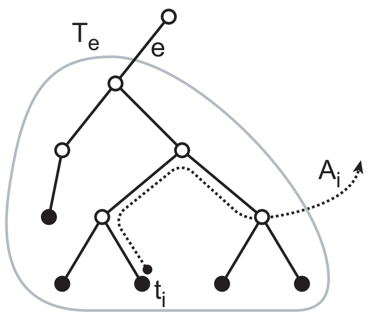

FIG. 3.1. *Alternate paths in single source games.*

Define player i's *alternate path* A_i to be the path of cost χ_i found in Algorithm 3.1, as shown in Figure 3.1. If there is more than one such path, choose A_i to be the path which includes as many ancestors of t_i in T_e as possible before including edges outside of T^*. To show that all edges in T^* are paid for, we need the following technical lemma concerning the structure of alternate paths.

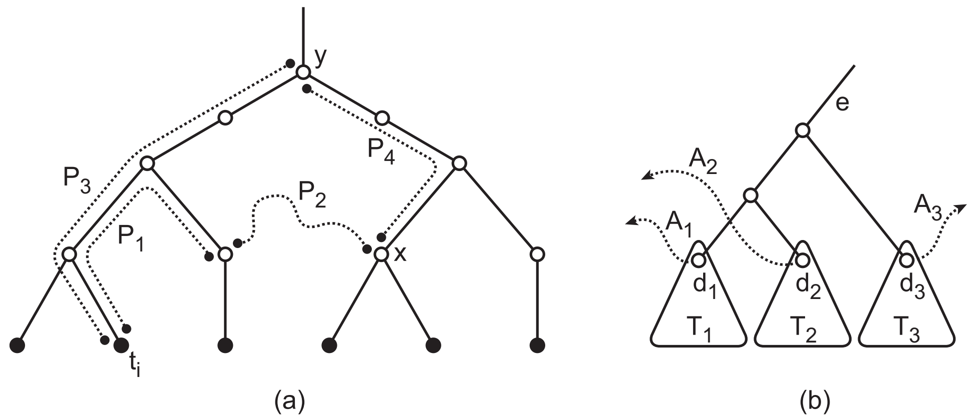

FIG. 3.2. *Alternate path structure in the proof of Theorem 3.2.*

LEMMA 3.3. *Suppose A_i is i's alternate path at some stage of the algorithm. Then there are two nodes v and w on A_i, such that all edges on A_i from t_i to v are in T_e, all edges between v and w are in $E \setminus T^*$, and all edges between w and s are in $T^* \setminus T_e$.*

Proof. Once A_i reaches a node w in $T^* \setminus T_e$, all subsequent nodes of A_i will be in $T^* \setminus T_e$, as all edges f in $T^* \setminus T_e$ have cost $c'(f) = 0$ and the source s is in $T^* \setminus T_e$. Thus, suppose A_i begins with a path P_1 in T_e, followed by a path P_2 containing only edges not in T^*, before reaching a node x in T_e, as shown in Figure 3.2(a). Let y be the lowest common ancestor of x and t_i in T_e. Observe that P_1 is strictly below y. Define P_3 to be the path from t_i to y in T_e, and define P_4 to be the path from y to x in T_e. We now show that under the modified cost function c', $P_3 \cup P_4$ is at least as cheap as $P_1 \cup P_2$. Since $P_1 \cup P_2$ includes a higher ancestor of t_i than A_i (namely y), this contradicts our choice of A_i.

Consider the iterations of the algorithm during which player i could have contributed to edges in P_3. At each of these steps the algorithm computes a cheapest path from t_i to s. At any time, player i's payments are upper bounded by the modified cost of his alternate path, which is in turn upper bounded by the modified cost of any path, in particular A_i. Furthermore, at each of these steps the modified costs of all edges in A_i above x are 0. Therefore i's contribution to P_3 is always at most the modified cost of $P_1 \cup P_2$. The modified cost of P_4 is always 0, as none of the edges in P_4 are on player i's path from t_i to s in T^*. Together these imply that $c'(P_3 \cup P_4) = c'(P_3) \leq c'(P_1 \cup P_2)$. □

Thus, players' alternate paths may initially use some edges in T_e, but subsequently will exclusively use edges outside of T_e. We use this fact in the following lemma.

LEMMA 3.4. *Algorithm 3.1 fully pays for every edge in T^*.*

Proof. Suppose that for some edge e, after all players have contributed to e, $p(e) < c(e)$.

For each player i, consider the longest subpath of A_i containing t_i and only edges in T_e. Call the highest ancestor of t_i on this subpath i's *deviation point*, denoted d_i. Note that it is possible that $d_i = t_i$. Let D be a minimum set of deviation points such that every terminal in T_e has an ancestor in D.

Suppose every player i with a terminal t_i in D deviates to A_i, as shown in Figure 3.2(b), paying his modified costs to each edge. Any player i deviating in this manner does not increase his total expenditure, as player i raised $p_i(e)$ until p_i matched the modified cost of A_i. The remaining players leave their payments unchanged.

We claim that now the edges bought by players with terminals in T_e connect all these players to $T^* \setminus T_e$. To see this, first consider any edge f below a deviation point d_i in D. By Lemma 3.3, player i is the only deviating player who could have been contributing to f. If i did contribute to f, then f must be on the unique path from t_i to d_i in T_e. But by the definition of d_i, this means that f is in A_i. Thus player i will not change his payment to f.

Define T_i to be the subtree of T_e rooted at d_i. We have shown that all edges in T_i have been bought. By Lemma 3.3, we know that A_i consists of edges in T_i followed by edges in $E \setminus T$ followed by edges in $T^* \setminus T_e$. By the definition of c', the modified cost of those edges in $E \setminus T^*$ is their actual cost. Thus i pays fully for a path connecting T_i to $T^* \setminus T_e$.

We have assumed that the payments generated by the algorithm for players with terminals in T_e were not sufficient to pay for those terminals to connect to $T^* \setminus T_e$. However, without increasing any players' payments, we have managed to buy a subset of edges which connects all terminals in T_e to $T^* \setminus T_e$. This contradicts the optimality of T^*. Thus the algorithm runs to completion. □

Since we have also shown that the algorithm always produces a Nash equilibrium, this concludes the proof of the theorem. □

We will now argue that Algorithm 3.1 works even if the graph is directed. It is still the case that if the algorithm does succeed in assigning payments to all edges, then we are done. Hence, to prove correctness, we will again need only show that failure to pay for an edge implies the existence of a cheaper tree, thus yielding a contradiction. The problem is that Lemma 3.3 no longer holds; it is possible that some of the players attempting to purchase an edge e have an alternate paths which repeatedly moves in and out of the subtree T_e. Thus, the argument is slightly more complex.

LEMMA 3.5. *Algorithm 3.1 fully pays for every edge in T^* for directed graphs.*

Proof. Suppose the algorithm fails to pay for some edge e. At this point, every player i with a terminal in T_e has an alternate path A_i, as defined earlier. Define D to be the set of vertices contained in both T_e and at least one alternate path. Note that D contains all terminals that appear in T_e. We now create $D' \subseteq D$ by selecting the *highest* elements of D; we select the set of nodes from D that do not have ancestors with respect to T_e in D. Every terminal in T_e has a unique ancestor in D' with respect to T_e, and every node in D' can be associated with at least one alternate path.

For any node $v_j \in D'$, let A_j be the alternate path containing v_j. If more than one such path exists, simply select one of them. Define A'_j to be the portion of this path from v_j to the first node in $T \setminus T_e$. We can now form T' as the union of edges from $T \setminus T_e$, all paths A'_j, and every subtree of T_e rooted at a node in D'. T' might not be a tree, but breaking any cycles yields a tree which is only cheaper.

It is clear that all terminals are connected to the root in T', since every terminal in T_e is connected to some node in D', which in turn is connected to $T \setminus T_e$. Now we just need to prove that the cost of our new tree is less than the cost of the original. To do so, we will show that the total cost of the subtrees below nodes in D', together with the cost of adding any additional edges needed by the paths A'_j, is no greater than the total payments assigned by the algorithm to the players in T_e thus far. Hence it will be helpful if we continue to view the new tree as being paid for by the players. In particular, we will assume that all players maintain their original payments for all edges below nodes in D', and the additional cost of building any path A'_j is covered by the player for which A_j was an alternate path. It now suffices to show that no player increases their payment.

For the case of those players who are not associated with a node from D', this trivially holds, since their new payments are just a subset of their original payments. Now consider a player i who must pay for any unbought edges in the path A'_j, which starts from node $v_j \in D'$. Note that player i's terminal might not be contained within the subtree rooted at v_j. If it is, then we are done, since in this case, player i's new cost is at most the cost of A_j, which is exactly i's current payment.

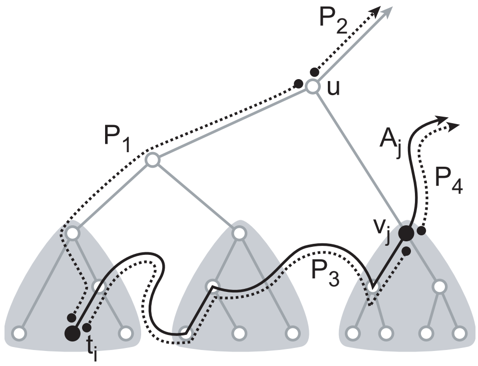

FIG. 3.3. *Alternate path structure in the proof of Lemma 3.5.*

Thus suppose instead that player i's terminal lies in a subtree rooted at a node $v_k \in D'$ (this case is shown in Figure 3.3). Define u to be the least common ancestor of v_k and v_j in T_e. Observe that u can not be either v_k or v_j, as this would contradict the minimality of the set D'. Define P_1 to be the current payments made by player i from its terminal to u, and let P_2 be the current payments made by player i from u to e (inclusive). Define P_3 as the cost of $A_j \setminus A'_j$ and let P_4 be the cost of A'_j. By the definition of alternate path,

$$P_1 + P_2 = P_3 + P_4.$$

Furthermore, since we have already successfully paid for a connection to u, we know that

$$P_3 \geq P_1,$$

since otherwise, when we were paying for the edges between v_k and u, player i would have had an incentive to deviate by purchasing P_3 and then using the path from v_j to u in T_e, which would have been free for i. Hence $P_4 \leq P_2$.

Therefore we can bound player i's contribution to edges below D' by P_1 (since u lies above v_k), and we can bound player i's contribution to A'_j by P_2. Taken together, we have that player i's new cost has not increased. Thus no player has increased their payments, and yet the total cost of the tree has decreased, which is a contradiction. □

We have shown that the price of stability in a single source game is 1. However, the algorithm for finding an optimal Nash equilibrium requires us to have a minimum cost Steiner tree on hand. Since this is often computationally infeasible, we present the following result.

THEOREM 3.6. *Suppose we have a single source game and an α-approximate minimum cost Steiner tree T. Then for any $\varepsilon > 0$, there is a poly-time algorithm which returns a $(1+\varepsilon)$-approximate Nash equilibrium on a Steiner tree T', where $c(T') \le c(T)$.*

Proof. To find a $(1+\varepsilon)$-approximate Nash equilibrium, we start by defining $\gamma = \frac{\varepsilon c(T)}{(1+\varepsilon)n\alpha}$. We now use Algorithm 3.1 to attempt to pay for all but γ of each edge in T. Since T is not optimal, it is possible that even with the γ reduction in price, there will be some edge e that the players are unwilling to pay for. If this happens, the proof of Theorem 3.2 indicates how we can rearrange T to decrease its cost. If we modify T in this manner, it is easy to show that we have decreased its cost by at least γ. At this point we simply start over with the new tree and attempt to pay for that.

Each call to Algorithm 3.1 can be made to run in polynomial time. Furthermore, since each call which fails to pay for the tree decreases the cost of the tree by γ, we can have at most $\frac{(1+\varepsilon)\alpha n}{\varepsilon}$ calls. Therefore in time polynomial in n, α an ε^{-1}, we have formed a tree T' with $c(T') \le c(T)$ such that the players are willing to buy T' if the edges in T' have their costs decreased by γ.

For all players and for each edge e in T', we now increase $p_i(e)$ in proportion to p_i so that e is fully paid for. Now T' is clearly paid for. To see that this is a $(1+\varepsilon)$-approximate Nash equilibrium, note that player i did not want to deviate before his payments were increased. If we let m' be the number of edges in T', then i's payments were increased by

$$\gamma \frac{p_i(T')}{c(T') - m'\gamma} m' = \frac{\varepsilon c(T) p_i(T') m'}{(1+\varepsilon)n\alpha(c(T') - m'\gamma)} \le \frac{\varepsilon c(T) p_i(T')}{\alpha(1+\varepsilon)(1-\varepsilon)c(T')} \le \varepsilon p_i(T').$$

Thus any deviation yields at most an ε factor improvement. □

Extensions. Both theorems 3.2 and 3.6 can be proven for the case where our graph G is directed, and players wish to purchase paths from t_i to s. Once we have shown that our theorems apply in the directed case, we can extend our model and give each player i a maximum cost $\max(i)$ beyond which he would rather pay nothing and not connect his terminals. It suffices to make a new terminal t'_i for each player i, with a directed edge of cost 0 to t_i and a directed edge of cost $\max(i)$ to s.

4. General Connection Games. In this section we deal with the general case of players that can have different numbers of terminals and do not necessarily share the same source terminal. As stated before, in this case the price of anarchy can be as large as N, the number of players. However, even the price of stability may be quite large in this general case.

Consider the graph illustrated in Figure 4.1, where each player i owns terminals s_i and t_i. The optimal centralized solution has cost $1+3\varepsilon$. If the path of length 1 were bought, each player $i > 2$ will not want to pay for any ε edges, and therefore the situation of players 1 and 2 reduces to the example in Section 2 of a game with no Nash equilibria. Therefore, any Nash equilibrium must involve the purchase of the path of length $N-2$. In fact, if each player $i > 2$ buys $\frac{1}{N-2}$ of this path, then we have a Nash equilibrium. Therefore, for any $N > 2$, there exists a game with the price of stability being nearly $N-2$.

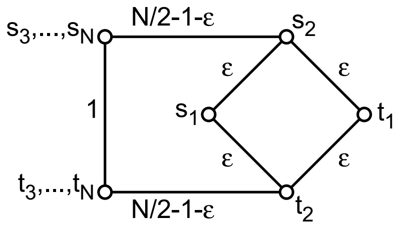

FIG. 4.1. *A game with high price of stability.*

Because the price of stability can be as large as $\Theta(N)$, and sometimes pure Nash equilibria may not exist at all, we cannot hope to be able to provide cheap Nash equilibria for the multi-source case. Therefore, we consider how cheap α-approximate Nash equilibria with small α can be, and obtain the following result, which tells us that there always exists a 3-approximate Nash equilibrium as cheap as the optimal centralized solution.

THEOREM 4.1. *For any optimal centralized solution T^*, there exists a 3-approximate Nash equilibrium such that the purchased edges are exactly T^*.*

We prove this theorem in Subsection 4.3 using the sufficient conditions for an approximate Nash equilibrium of Theorem 4.2. In Subsection 4.2 we address the key special case where the underlying graph is a path, which is then extended to the general case via a simple induction. In Subsection 4.4 we give lower bounds and a polynomial time algorithm for finding an approximate Nash equilibrium.

4.1. Connection Sets and Sufficient Conditions for Approximate Nash Equilibria. Given a set of bought edges T, denote by a *stable payment* p_i for player i a payment such that player i has no better deviation than p_i, assuming that the rest of T is bought by the other players. A Nash equilibrium must consist of stable payments for all players. However, what if in some solution, a player's payment p_i is not stable, but is a union of a small number of stable payments? This implies that each player's best deviation is not much less than its current payment. Specifically, we have the following general theorem.

THEOREM 4.2. *Suppose we are given a payment scheme $p = (p_1, \ldots, p_N)$, with the set of bought edges T. If for all i, the payment p_i is a union of at most α stable payments (with respect to T), then p is an α-approximate Nash equilibrium.*

Proof. Let p'_i be the best deviation of player i given p, and let $p_i^1, p_i^2, \ldots, p_i^\alpha$ be the stable payments which together form p_i. The fact that p'_i is a valid deviation for i means that the set of bought edges T with p_i taken out and p'_i added still connects the terminals of i. p_i^j being a stable payment means that if i only pays for p_i^j and the rest of T is bought by other players, then the best deviation of i is al least as expensive than p_i^j. In this case, p'_i is still a possible deviation, since if taking out p_i and adding p'_i connects the terminals of i, then so does taking out p_i^j and adding p'_i. Therefore, we know that the cost of p'_i is no smaller than the cost of any p_i^j, and $\alpha \cdot cost(p'_i) \geq cost(p_i)$, as desired. □

Notice that the converse of this theorem is not true. Consider an example where player i is contributing to an edge which it does not use to connect its terminals. If this edge is cheap, this would still form an approximate Nash equilibrium. However, this edge would not be contained in any stable payment of player i, so p_i would not be a union of stable payments.

To prove Theorem 4.1, we will construct a payment scheme on the optimal centralized solution such that each player's payment is a union of 3 stable payments. The stable payments we use for this purpose involve each edge being paid for by a single player, and have special structure. We call these payments *connection sets*. Since there is no sharing of edge costs by multiple players in connection sets, we will often use sets of edges and sets of payments interchangeably. T^* below denotes an optimal centralized solution, which we know is a forest.

DEFINITION 4.3. *A connection set S of player i is a subset of edges of T^* such that for each connected component C of the graph $T^* \setminus S$, we have that either*
(1) any player that has terminals in C has all of its terminals in C, or
(2) player i has a terminal in C.

Intuitively, a connection set S is a set such that if we removed it from T^* and then somehow connected all the terminals of i, then all the terminals of all players are still connected in the resulting graph. We now have the following lemma, the proof of which follows directly from the definition of a connection set.

LEMMA 4.4. *A connection set S of player i is a stable payment of i with respect to T^*.*

Proof. Suppose that player i only pays exactly for the edges of S, and the other players buy the edges in $T^*\backslash S$. Let Q be a best deviation of i in this case. In other words, let Q be a cheapest set of edges such that the set $(T^*\backslash S) \cup Q$ connects all the terminals of i. To prove that S is a stable payment for i, we need to show that $cost(S) \leq cost(Q)$.

Consider two arbitrary terminals of any player. If these terminals are in different components of $T^*\backslash S$, then by definition of connection set, each of these components must have a terminal of i. Therefore, all terminals of all players are connected in $(T^*\backslash S) \cup Q$, since $(T^*\backslash S) \cup Q$ connects all terminals of i. Since T^* is optimal, we know that $cost(T^*) \leq cost((T^*\backslash S) \cup Q)$. Since $S \subseteq T^*$ and Q is disjoint from $T^*\backslash S$, then $cost((T^*\backslash S) \cup Q) = cost(T^*) - cost(S) + cost(Q)$, and so $cost(S) \leq cost(Q)$. □

As a first example of a connection set consider the edges S^i of T^* that are used exclusively by player i. More formally, let T^i be the unique smallest subtree of T^* containing all terminals of player i, and let S^i be the set of edges belonging only to T^i and no other tree T^j.

LEMMA 4.5. *The set S^i defined above is a connection set for player i.*

Proof. Suppose to the contrary that there are at least two components of $T^*\backslash S^i$ that contain terminals t_1 and t_2 of some player $j \neq i$. Since T^* is a tree that connects all terminals, this means that the path in T^* between t_1 and t_2 must also be contained in T^j. But this implies that the edges of S^i whose removal disconnected this path also belong to T^j, which contradicts the definition of S^i. □

Each player i will pay for this connection set, the set of edges used only by player i. We want each player to pay for at most 2 additional connection sets. Without loss of generality we can contract the edges now paid for, forming a new tree T^* which the players must pay for. For the remainder of this section we will assume that each edge belongs to at least two different T^i's, and will have players pay for at most two connection sets.

4.2. Approximate Nash Equilibrium in Paths. In this subsection we consider the key special case when the tree T^* is a path P. In the next section we use induction to extend the proof to the general case.

We will use v_k, $k = 1 \dots n$ to denote the nodes on the path P in the order $v_1, \dots, v_n$, and will refer to the terminals in this order, for example, the "first" terminal of player i will mean the one closest to v_1. Denote the set of all terminals located at v_k by U_k, and assume that each edge is in at least two different T^i's as mentioned above.

Roughly speaking, the idea is that for each player i we associate an edge of the path with each terminal that belongs to player i, and have player i pay for these edges. For a terminal u we define the set of edges $Q(u)$ as the possible edges that can be associated with u. For every terminal $u \in U_k$ owned by a player i, with $k \neq n$, define a subpath $Q(u)$ as follows (illustrated on Figure 4.2(a)). If i owns another terminal in U_ℓ with $\ell > k$, then set $Q(u)$ to be the subpath of P from v_k to the first such node v_ℓ. If there is no such node (because u is i's last terminal in P), set $Q(u)$ to be the subpath of P starting at the first terminal of i, and ending at v_k. Notice that $Q(u)$ is not defined for $u \in U_n$, so if i has a terminal in U_n, the $Q(u)$ paths for terminals of i will look like Figure 4.2(b).

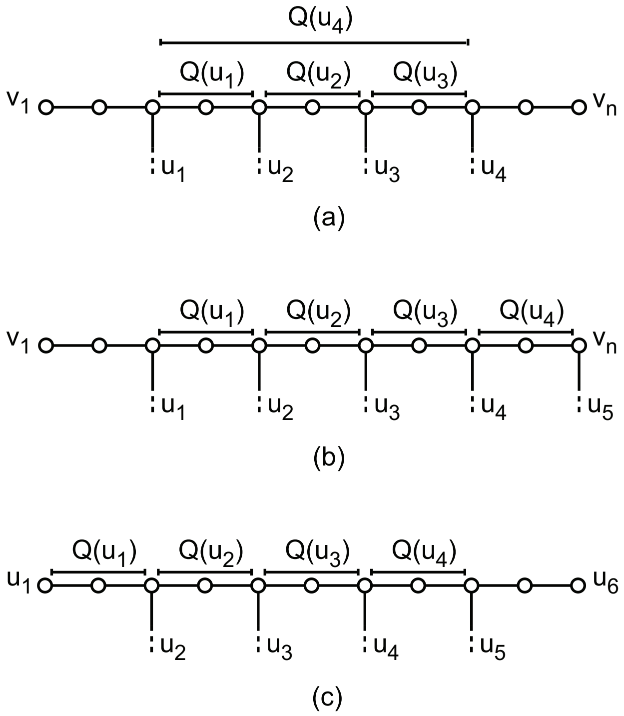

FIG. 4.2. *The paths $Q(u)$ for a single player i. (a) i has no terminal in U_n (b) i has a terminal in U_n (c) i is the special player of Lemma 4.11 that has terminals both in v_1 and v_n.*

A key observation about the Q sets is that if a player i pays for one edge in each $Q(u)$ (excluding the one belonging to the last terminal) the resulting set forms a connection set.

LEMMA 4.6. *Consider a payment S^i by player i that contains at most one edge from each path $Q(u)$, where u are terminals of i excluding the last terminal of i. Then, S^i forms a connection set.*

Proof. Every component of $P \backslash S^i$ contains a terminal of i, since there is a terminal of i between every two $Q(u)$'s for u belonging to i, as well as before the first such $Q(u)$, and after the last one. This means that S^i is a single connection set. $\square$

Unfortunately, we cannot assign each edge to a different terminal, as shown by

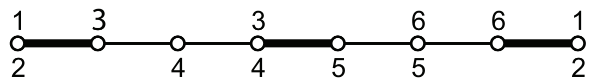

FIG. 4.3. *The bold edges along the path form a single connection set connecting* two *neighboring terminals of players 1 and 2.*

the example of Figure 4.3. The bold edges in this example are used only by players 1 and 2, and belong to the Q paths of the first terminals of players 1 and 2. This leaves us with three edges and only two terminals to assign them to. However, note that the set of bold edges is a single connection set by itself, even though it contains more than one edge in every Q path. We say that a set L of edges along the path is coupled if all the edges $e \in L$ belong to the exact same sets $Q(u)$ for $u \in \cup U_k$. We need to extend our ideas so far to allow us to assign such coupled sets of edges to a terminal, rather than just assigning a single edge.

DEFINITION 4.7. *A max-coupled-set L is a maximal set of edges of P such that for every edge $e \in L$, the set of paths $Q(u)$ that contain e is exactly the same, for $u \in \cup U_k$.*

The key property of max-coupled-sets is they form a connection set between two consecutive terminals of one player.

LEMMA 4.8. *Consider a max-coupled-set L, and order all components C of $P \backslash L$ along the path. For all components except the two end components, any player that has terminals in C has all of its terminals in C.*

Proof. Consider a component C of $P \backslash L$ that is neither the first nor the last component, such that a player j has a terminal t in C. Consider the edges of L directly adjacent to C. If the earlier such edge belongs to some path $Q(u)$ with u a terminal of j, then this gives us a contradiction, since the paths $Q(u)$ for terminals of j change at t, and never become the same. This contradicts the fact that L is a coupled set, since both edges of L must be in exactly the same Q paths. On the other hand, if the earlier edge of L adjacent to C does not belong to any path $Q(u)$ with u a terminal of j, then for the edges of L on the other side of C to belong to the same Q paths, it must be that all terminals of j are inside C, as desired. □

This implies the following extension of Lemma 4.6

LEMMA 4.9. *Consider a payment S^i by player i that contains at most one max-coupled-set from each path $Q(u)$, where u are terminals of i excluding the last terminal of i along the path. Then, S^i forms a connection set.*

Proof. We must prove that every component of $P \backslash S^i$ obeys one of the two properties from the definition of a connection set. Consider a component of $P \backslash S^i$ that does not contain a terminal of i. By the argument in Lemma 4.6, this component must be bordered by edges of the same max-coupled-set, and by Lemma 4.8, this component satisfies the first property in the definition of a connection set. □

Now we are ready to prove our main result for paths. To help with the induction proof appearing in Subsection 4.3 for the case of trees, we need to prove a somewhat stronger statement for paths.

THEOREM 4.10. *Assume the optimal tree T^* is a path P, and each edge of P is used by at least two players. There exists a payment scheme fully paying for path P such that each player i pays for at most 2 connection sets. Moreover, players with terminals in U_n pay for at most 1 connection set.*

Proof. In our payment, we will assign max-coupled-sets of edges to terminals u. By Lemma 4.9 the edges S^i assigned to the terminals of player i, excluding the last terminal of i, form a single connection set. For players that do not have a terminal in U_n the max-coupled-set assigned to the last terminal forms a second connection set. Since a max-coupled-set is a connection set by itself, this would meet the conditions of the Theorem.

To form this payment, we form a bipartite matching problem as follows. Let Y have a node for each max-coupled-set of edges in P, and let Z be the nodes of $v_1, \ldots, v_{n-1}$ of P. Form an edge between a node $v_k \in Z$ and node $L \in Y$ if there exists some terminal $u \in U_k$ such that $L \subseteq Q(u)$. This edge signifies that some player owning $u \in U_k$ could pay for L. In addition, if $u \in U_k$ is the last terminal of a player i, but $k \neq n$, then we also form an edge between $v_k \in Z$ and $L \in Y$ if $L \subseteq T^i$. These edges signify the "additional" max-coupled-set that this player might pay for since it owns no terminals in U_n.

We claim that this graph has a matching that matches all nodes in Y, and we will use such a matching to assign the max-coupled-sets to terminals according to the edges in this matching. To prove that such a matching exists, we use Hall's Matching Theorem. For $X \subseteq Y$, define $\partial(X)$ to be the set of nodes in Z which X has edges to. According to Hall's Matching Theorem, there exists a matching in this bipartite graph with all nodes of Y incident to an edge of the matching if for each set $X \subseteq Y$, $|\partial(X)| \geq |X|$. To prove that this condition is satisfied, arrange the edges $E(X)$ in the max-coupled-sets X in the order they appear in P. We want to show that between every two max-coupled-sets of X, there is a node belonging to $\partial(X)$. This will yield $|X|-1$ nodes in $\partial(X)$. Then we show that there is an additional node in $\partial(X)$ before all the edges $E(X)$.

Consider some edge e of $E(X)$ that belongs to a max-coupled-set L, and suppose a previous edge e' in $E(X)$ belongs to a different a max-coupled-set L'. Since these are different and maximal coupled sets, there must be some path $Q(u)$ that contains exactly one of e, e'. The player corresponding to this path $Q(u)$ must have a terminal between e and e' that is in the set $\partial(X)$.

We need to prove that there is an additional node in $\partial(X)$ before the set $E(X)$. Let L be the first max-coupled-set of X that appears in P. The player corresponding to a path $Q(u)$ containing L must have a terminal in $\partial(X)$ before L.

Therefore, $|X| \leq |\partial(X)|$ for all $X \subset Y$, and hence there always exists a matching that covers the max-connection-sets Y. □

This finishes our proof that if T^* is a path, then there exists a 3-approximate Nash equilibrium that purchases exactly T^* (2-approximate when all edges in T^* are used by at least two players). To prove the general case, however, we need the following strengthening.

LEMMA 4.11. *Suppose there exists a player i with a terminal $s \in U_1$ and a terminal in U_n. Then there exists a payment scheme as in Theorem 4.10 and moreover i has at least 2 terminals in the component of $P \backslash S^i$ containing v_n.*

Proof. We change the definition of $Q(u)$ for the terminals of i, as shown on Figure 4.2(b). We let $Q(u)$ be the path immediately to the left of u, until it reaches the next terminal of i.

We show that the proof of Theorem 4.10 goes through in this case with minor changes. First note that the max-coupled-sets are exactly the same sets as before. Note that the max-coupled-sets assigned to player i now will form a single connection set, and further the last terminal of i before U_n would be in the component of $P \backslash S^i$ containing v_n, as we desired.

We must now verify that the bipartite graph formed in the proof of Theorem 4.10

actually has a matching that covers all of the max-coupled sets. To do this, we need to prove that $|X| \leq |\partial(X)|$ for a set $X \subset Y$, which we do once again by showing that between every pair of max-coupled-sets in X there exists a node of $\partial(X)$, and there is a further node of $\partial(X)$ in front of the set $E(X)$.

As before, if we have two edges e and e' that belong two two different max-coupled-sets, then any player j that has a set $Q(u)$ containing exactly one of e and e' most have a terminal in $\partial(X)$ between e and e'. To see that we have a node in $\partial(X)$ before $E(X)$ let L be the first max-coupled-set of X that appears in P, the let j be the player corresponding to a path $Q(u)$ containing L. If $j \neq i$ then j has a terminal in $\partial(X)$ before L. Recall that each edge is used by at least two players so we can select a $Q(u)$ set containing L that belongs to a player $j \neq i$.

We can now continue with the process given in the proof of Theorem 4.10 to form the desired payment scheme. □

We will need the following further observation about the proof: in the proof at most one terminal is assigned any set of edges among the terminals in each set U_k, for any node v_k of the path.

LEMMA 4.12. *There exists a set $A = \{u_1, u_2, \ldots u_{n-1}\}$ with $u_k \in U_k$ such that only the terminals $u_1, \ldots, u_{n-1}$ are assigned max-coupled-sets in the payment formed in the proof of Lemma 4.11.*

4.3. Proof of Theorem 4.1 (Existence of 3-Approx Nash Equilibrium). In this subsection, we prove that for any optimal centralized solution T^*, there exists a 3-approximate Nash equilibrium such that the purchased edges are exactly T^*. For simplicity of the proof, we assume that T^* is a tree, since otherwise we can apply this proof to each component of T^*.

Recall that we used T^i to denote the unique smallest subtree of T^* which connects all terminals of player i. We formed the first connection set using Lemma 4.5 by the edges that belong to a single T^i. Contracting these edges, we can assume that all edges are used by at least two players, and we will construct a payment scheme in which each player is paying for at most 2 connection sets.

Intuition and Proof Outline. The idea of the proof is to select two terminals of a player i, let P be the path connecting them in T^*, and let $t_1 = v_1, \ldots, v_n = t_n$ denote the nodes along the path P. We apply the special case for paths, Lemma 4.11, to the path P with all players j with sets $T^j \cap P$ nonempty. Then we apply the induction hypothesis for each subtree rooted at the nodes v_k of path P, where we use the one player j (by Lemma 4.12) that has a max-coupled-set assigned to node v_k as a "special" player, whose two terminals we select to form a path as above.

To make the induction go through we need a stronger version of Theorem 4.1 analogous to the stronger version of the path lemma (Lemma 4.11).

THEOREM 4.13. *Assume each edge of the optimal tree T^* is used by at least two players, let t be a terminal, and i a player with terminal t. Then there exists a way to pay for T^* by assigning at most two connection sets to each player, so that the following hold:*
(1) each player j that has t as a terminal has at most one connection set assigned,
(2) for the connection set S^i assigned to player i the set $T^* \backslash S^i$ has an additional terminal of i in the component containing terminal t.

Proof. Let s be another terminal of player i, and let P be the path connecting s and t in T^*. Let $s = v_1, \ldots, v_n = t$ be the sequence of nodes along P, and let T^*_k be the subtree of $T^* \backslash P$ rooted at node v_k.

Now we define a problem on path P, and subproblems for each of the subtrees

T_k^*. First we define the problem for path P. A player j will have a terminal at node v_k if player j has a terminal in the subtree T_k^*. With this definition, each edge of P is used by at least two players. We apply Lemma 4.11 with i as the special player (by choice of the path i has both v_n and v_1 as terminals in the induced problem on the path). We assign each player connection sets.

Next we will define the problems on the trees T_k^*. For this subproblem we say that v_k is a terminal for any player that has a terminal outside of the subtree T_k^*. We use the induction hypothesis to assign connection sets to players in T_k^*. Recall that by Lemma 4.12 at most one player, say player i_k, is assigned a max-coupled-set to a terminal in v_k on the path problem. We use v_k as the terminal t in the recursive call, and i_k as the special player.

To finish the proof we need to argue that the assignment satisfies the desired properties of our theorem. We will need to have a few cases to consider.

Consider a player j that has $T^j \cap P = \emptyset$. This player j has all its terminals in a subtree T_k^*, and hence by the induction hypothesis, it has at most two connection sets assigned in subtree T_k^*.

Now consider a player j that has t as a terminal, but $j \neq i$. For each subtree that has terminals of player j the recursive call has assigned at most one connection set to player j, and we may have also assigned a connection set at path P. Note that in the path P the player j owns terminal t, so its last terminal is t, and has no set $Q(t)$. We claim that combining all the sets j is paying for into one set S^j forms a single connection set. To see why consider the connected components of $T^j \backslash S^j$. Connected components contained in a subtree T_k^* satisfy one of the connection set properties by the induction hypothesis. If a terminal u at a node v_k of the path problem was assigned a max-coupled-set along the path P then in the recursive call we guaranteed that player j has a terminal connected to the root v_k, so the component containing v_k has a terminal in $T^j \backslash S^j$. Finally, the last component along the path contains the terminal t.

A similar argument applies for the special player i: the union of all connection sets assigned to i for the path and for the recursive calls combines to a single connection set S^i that satisfies the extra requirement that set $T^* \backslash S^i$ has an additional terminal in the component containing terminal t.

Finally, consider a player j where t is not a terminal of j (though it may be included in T^j). As before for each subtree that has terminals of player j the recursive call has assigned at most one connection set to player j, and we may have also assigned a connection set at path P. This case differs from the previous ones in two points. First, if $t \notin T^j$ then the last node v_{k_j} of T^j along the path may have an extra connection set assigned to it; second, the node t is not a terminal for player j. As a result of these differences, combining all the sets assigned to player j to a single set S^j may not form a single connection set. Consider the connected components of $T^j \backslash S^j$. Connected components fully contained in a subtree T_k^* satisfy one of the connection set properties by the induction hypothesis. Most components that intersect the path P must also have a terminal of player j: if a terminal u in the path problem at a node v_k was assigned a connection set along the path P then in the recursive call we guaranteed that player j has a terminal connected to the root v_k, hence the component containing node v_k has a terminal in $T^j \backslash S^j$.

However, there can be components of $T^j \backslash S^j$ intersecting the path P that do not satisfy either of the connection set properties. If v_{k_j} is the last node along the path P in T^j, then if v_{k_j} has a max-coupled-set assigned to player j (e.g., if $v_{k_j} \neq t$) then

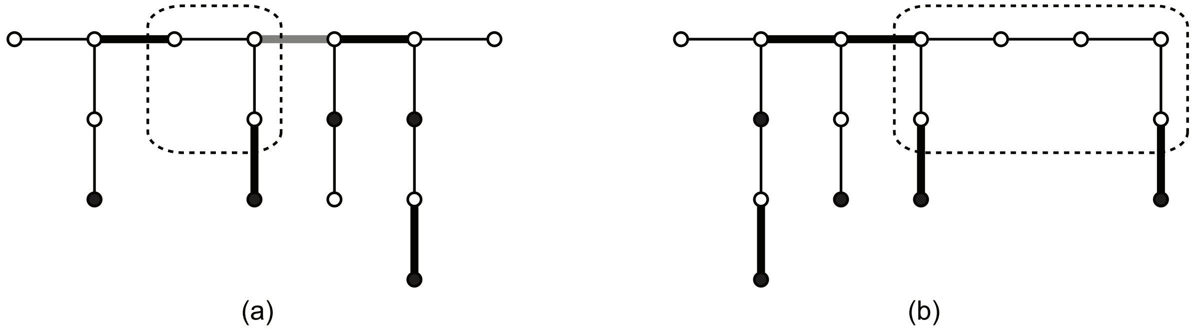

FIG. 4.4. *Sets assigned to a player that do not form one connection set. The nodes in dark are terminals for one player j, and dark edges are assigned to this player with the grey edge assigned to the last terminal of j along the path. (a): the case when $t \notin T^j$, and a set (the grey edge) is assigned to the last terminal of j along P. (b): the case when no connection set is assigned to the last terminal of j along P.*

the component(s) of $T^j \backslash S^j$ adjacent to this max-coupled-set may not satisfy either of the connection set properties. Otherwise, the last component along the path P may not satisfy these properties. See Figure 4.4 for examples for each possibility. In Figure 4.4(a), the grey edge is the max-coupled-set assigned to v_{k_j}, which results in the highlighted component not having any terminals of j. In Figure 4.4(b), nothing is assigned to v_{k_j}, and this also results in the highlighted component not having any terminals of j. Notice, however, that all other components obey the connection set properties since they each have a terminal of j. In either case (whether v_{k_j} has a max-coupled-set assigned to it or not), removing one of the max-coupled-sets L_j (the one assigned to v_{k_j}, or one bordering the final component along P with no terminals) results in a connection set $\bar{S}^j = S^j \backslash L_j$, and the max-coupled-set L_j alone forms a second connection set. □

4.4. Extensions and Lower Bounds. We have now shown that in any game, we can find a 3-approximate Nash equilibrium purchasing the optimal network. We proved this by constructing a payment scheme so that each player pays for at most 3 connection sets. This is in fact a tight bound. In the example shown in Figure 4.5, there must be players that pay for at least 3 connection sets. There are N players, with only two terminals (s_i and t_i) for each player i. Each player must pay for edges not used by anyone else, which is a single connection set. There are $2N - 3$ other edges, and if a player i pays for any 2 of them, they are 2 separate connection sets, since the component between these 2 edges would be uncoupled and would not contain any terminals of i. Therefore, there must be at least one player that is paying for 3 connection sets.

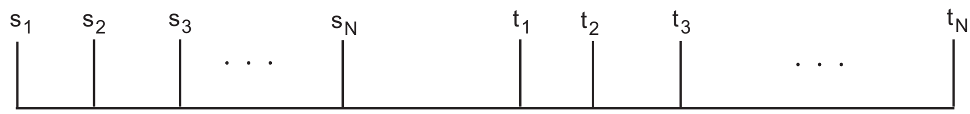

FIG. 4.5. *A graph where players must pay for at least 3 connection sets.*

This example does not address the question of whether we can lower the approximation factor of our Nash equilibrium to something other than 3 by using a method other than connection sets. As a lower bound, in Section 5 we give a simple sequence of games such that in the limit, any Nash equilibrium purchasing the optimal network must be at least $(\frac{3}{2})$-approximate.

Polynomial-time algorithm. Since the proof of Theorem 4.1 is constructive, it actually contains a polynomial-time algorithm for generating a 3-approximate Nash equilibrium on T^*. We can use the ideas from Theorem 3.6 to create an algorithm which, given an α-approximate Steiner forest T, finds a $(3+\varepsilon)$-approximate Nash equilibrium which pays for a Steiner forest T' with $c(T') \leq c(T)$, as follows. However, this algorithm requires a polynomial-time optimal Steiner tree finder as a subroutine. We can forego this requirement at the expense of a higher approximation factor.

We start by defining $\gamma = \frac{\varepsilon c(T)}{(1+\varepsilon)n\alpha}$, for ε small enough so that γ is smaller than the minimum edge cost. The algorithm of Theorem 4.1 generates at most 3 connection sets for each player i, even if the forest of bought edges is not optimal. We use this algorithm to pay for all but γ of each edge in T. We can check if each connection set is actually cheaper than the cheapest deviation of player i, which is found by the cheapest Steiner tree algorithm. If it is not, we can replace this connection set with the cheapest deviation tree and run this algorithm over again. The fact that we are replacing a connection set means that all the terminals are still connected in the new tree. If we modify T in this manner, it is easy to see that we have decreased its cost by at least γ.

We can now use the arguments from Theorem 3.6 to prove that this algorithm produces a $(3+\varepsilon)$-approximate Nash equilibrium, and runs in time polynomial in n, α, and ε^{-1}. It requires a poly-time Steiner tree subroutine, however. If each player only has two terminals, finding the cheapest Steiner tree is the same as finding the cheapest path, so this is possible, and we can indeed find a cheap $(3+\varepsilon)$-approximate Nash equilibrium in polynomial time.

For the case where players may have more than two terminals, we can easily modify the above algorithm to use polynomial time approximations for the optimal Steiner tree, at the expense of a higher approximation factor. If we use a 2-approximate Steiner forest T, and an optimal Steiner tree 1.55-approximation algorithm from [32] as our subroutine, then the above algorithm actually gives a $(4.65+\varepsilon)$-approximate Nash equilibrium on T' with $c(T') \leq 2 \cdot OPT$, in time polynomial in n and ε^{-1}.

5. Lower bounds and NP-Hardness.

Lower bounds for approximate Nash on the optimal network.

CLAIM 5.1. *For any $\epsilon > 0$, there is a game such that any equilibrium which purchases the optimal network is at least a $(\frac{3}{2} - \epsilon)$-approximate Nash equilibrium.*

Proof. Construct the graph H_N on $2N$ vertices as follows. Begin with a cycle on $2N$ vertices, and number the vertices 1 through $2N$ in a clockwise fashion. For vertex i, add an edge to vertices $i+N-1 \mod (2N)$ and $i+N+1 \mod (2N)$. Let all edges have cost 1. Finally, we will add N players with 2 terminals, s_i and t_i, for each player i. At node j, add the label s_j if $j \leq N$ and t_{j-N} otherwise. Figure 5.1(a) shows such a game with $N = 5$.

Consider the optimal network T^* consisting of all edges in the outer cycle except (s_1, t_N). We would like to show that any Nash which purchases this solution must be at least $(\frac{6N-21}{4N-11})$-approximate. This clearly would prove our claim.

First we show that players 1 and N are not willing to contribute too much to any solution that is better than $(\frac{3}{2})$-approximate. Suppose we have such a solution. Define x to be player 1's contribution to his connecting path in T^*, and define y to

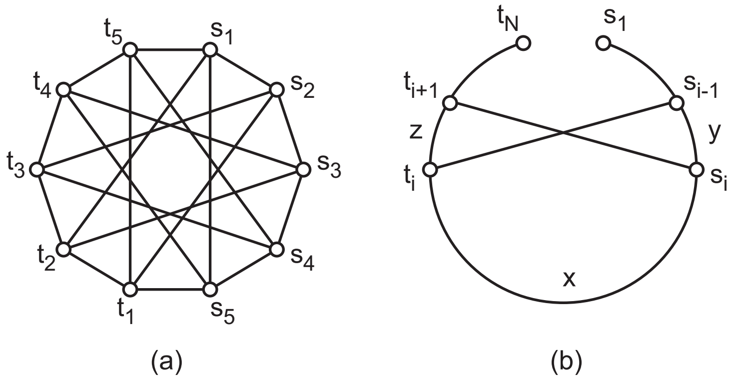

FIG. 5.1. *A game with best Nash equilibrium on OPT tending to at least a $\frac{3}{2}$-approximation.*

be his contribution to the remainder of T^*. Thus player 1 has a total payment of $x + y$. Player 1 can deviate to only pay for x. Furthermore, player 1 could deviate to purchase only y and the edge (s_1, t_N). If we have a solution that is at most $(\frac{3}{2})$-approximate, then we have that $\frac{x}{x+y} \geq \frac{2}{3}$ and similarly $\frac{y+1}{x+y} \geq \frac{2}{3}$. Taken together this implies that $\frac{1}{x+y} \geq \frac{1}{3}$, or $x + y \leq 3$. A symmetric argument shows that player N is also unwilling to contribute more than 3.

Thus we have that the remaining $N - 2$ players must together contribute at least $2N - 7$. Therefore there must be some player other than 1 or N who contributes $\frac{2N-7}{N-2}$. Suppose player i is such a player. Let x be the amount that player i contributes to his connecting path in T^*. Let y be his contribution to (s_{i-1}, s_i) and let z be his contribution to (t_i, t_{i+1}). See Figure 5.1(b).

Now consider three possible deviations available to player i. He could choose to contribute only x. He could contribute y and purchase edge (s_{i-1}, t_i) for an additional cost of 1. Or he could contribute z and purchase edge (s_i, t_{i+1}), also for an additional cost of 1. We will only consider these possible deviations, although of course there are others. Note that if i was contributing to any other portion of T^*, then we could remove those contributions and increase x, y, and z, thereby strictly decreasing i's incentive to deviate. Thus we can safely assume that these are i's only payments, and hence

$$x + y + z \geq \frac{2N - 7}{N - 2}.$$

Since i is currently paying at least $x+y+z$, we know that his incentive to deviate is at least $\max(\frac{x+y+z}{x}, \frac{x+y+z}{y+1}, \frac{x+y+z}{z+1})$. This function is minimized when $x = y+1 = z+1$. Solving for x we find that

$$x \geq \frac{4N - 11}{3N - 6}.$$

Thus player i's incentive to deviate is at least

$$\frac{x + y + z}{x} \geq \frac{3x - 2}{x} = 3 - \frac{2}{x} \geq 3 - 2\frac{3N - 6}{4N - 11} = \frac{6N - 21}{4N - 11}.$$

Therefore as N grows, this lower bound on player i's incentive to deviate tends towards $\frac{3}{2}$. Note that in this proof, we only considered one optimal network, namely T^*. If we modify G by increasing the costs of all edges not in T^* by some small $\varepsilon > 0$, then T^* is the only optimal network. Repeating the above analysis under these new costs still yields a lower bound of $\frac{3}{2}$ for the best approximate Nash on T^* in the limit as N grows and ε tends to 0. □

NP Completeness. In this section, we present a brief proof that determining the existence of Nash equilibria in a given graph is NP-complete if the number of players is $O(n)$ (where n is the number of nodes in the graph). We present a reduction from 3-SAT to show that the problem is NP-hard. The graph constructed will have unit cost edges.

Consider an arbitrary instance of 3-SAT with clauses C_j and variables x_i. We will have a player for each variable x_i, and two players for each clause C_j. For each variable x_i construct the gadget shown in Figure 5.2a. The source and sink of the player x_i are the vertices s_i and t_i respectively. When player x_i buys the left path or right path, this corresponds to x_i being set to be true or false, respectively. For clarity, we will refer to this player as being the i^{th} variable player.

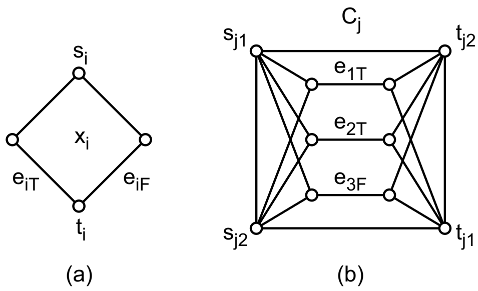

FIG. 5.2. *Gadgets for the NP-completeness reduction.*

Next, we construct a gadget for each clause C_j. The construction is best explained through an example clause $C_j = (x_1 \vee x_2 \vee \overline{x}_3)$ whose gadget is given in Figure 5.2b. The two players for C_j have their source sink pairs as (s_{j1}, t_{j1}) and (s_{j2}, t_{j2}) respectively. We will call both players on this gadget clause players. The final graph is constructed by gluing these gadgets together at the appropriate labeled edges. Specifically, the edges in clause gadget C_j labeled e_{1T}, e_{2T}, and e_{3F} are the same edges that appear in the corresponding variable gadgets. In other words, among all clauses and variable gadgets, there is only one edge labeled e_{iT} and only one labeled e_{iF}, and all the interior nodes in the gadget for each clause C_j are nodes in variable gadgets.

Suppose that there is a satisfying assignment A in our 3-SAT instance. Consider the strategy in which variable player i fully buys the left path if x_i is true in A and fully buys the right path otherwise. Since this is a satisfying assignment, by our construction each clause gadget has at least one interior edge fully paid for by a variable player. For each clause C_j, let e be one such edge, and let both players on this gadget buy the unique path of length 3 that connects their terminals which uses edge e. It is easy to see that the clause players are satisfied as the cost of this path to each clause player is 2, the minimum that he has to pay on any path from source to sink under the current payment scheme. The cheapest deviation for each variable player also costs 2, and therefore they do not have any incentive to move either. Thus, this forms a Nash equilibrium.

Suppose now that there is a Nash equilibrium. We will argue that this Nash equilibrium has to have a specific set of edges paid for. First, note that the contribution of each player is not more than 2, as the length of the shortest path is exactly 2.

Now suppose some perimeter edge of clause C_j is being bought. We know from the example in Figure 2.1 that perimeter edges cannot be bought by the clause players in C_j alone, for that would not constitute a Nash strategy. Therefore there must be some other player, variable or clause, contributing to the perimeter edge of C_j. Also, since this is a Nash strategy, any perimeter edge on which there is a positive contribution by any player must be fully bought. And once any perimeter edge of C_j has a positive contribution from a non-C_j player the payments of both the clause players of C_j will be strictly less than 2 in a Nash strategy.

Suppose one of the clauses, C_j, has some perimeter edge bought. Since at Nash equilibrium, the set of edges bought must form a Steiner forest, we look at the component of the Steiner forest that has the the clause C_j. We will show that the number of edges in this component is more than twice the number of players involved. Then, there must be a player who is paying more than 2, and hence this cannot be a Nash equilibrium.

Suppose there are x clause players and y variable players in the component of the forest containing C_j. We know from the example in Figure 2.1 that $x + y > 2$. Then, the total number of nodes in the Nash component containing C_j is $2x + 3y$ as we have to count the two source-sink nodes for each clause player and the three nodes on the path of each variable player. Since this is a connected tree, the total number of edges in this component is $2x + 3y - 1$. The average payment per player is then given by $\frac{2x+3y-1}{x+y} = 2 + \frac{y-1}{x+y}$. Now if $y > 1$, then the average payment per player is more than 2. Thus there must be some player who is paying more than 2, which is infeasible in a Nash. If $y = 1$, then the average payment per player is exactly 2. But again, since we know that the clause players of C_j pay strictly less than 2 each, there must be some player who pays strictly more than 2, which is again impossible. Lastly, we cannot have $y = 0$ as then whenever a clause player participates in paying for another clause, he must use a node in the path of a variable player, and thereby include this variable player in the component of C_j.

This implies that variable players only select paths within their gadget. Furthermore, it implies that variable players must pay fully for their entire path. Suppose i is a variable player who has selected the left (true) path, but has not paid fully for the second edge in that path. The remainder of this cost must be paid for by some clause player or players. But for such a clause player to use this edge, he must also buy two other edges, which are not used by any other player. Hence such a clause player must pay strictly more than 2. But there is always a path he could use to connect of cost exactly 2, so this can not happen in a Nash equilibrium. Thus we have established that variable players pay fully for their own paths.

Now consider any clause gadget. Since we have a Nash equilibrium, we know that only internal edges are used. But since each clause player can connect his terminals using perimeter edges for a cost of exactly 2, one of the interior variable edges must be bought by a variable player in each clause gadget. If we consider a truth assignment A in which x_i is true if and only if player i selects the left (true) path, then this obviously satisfies our 3-SAT instance, as every clause has at least one variable forcing it to evaluate to true.

Therefore, this game has a Nash equilibrium if and only if the corresponding formula is satisfiable, and since this problem is clearly in NP, determining whether a

Nash equilibrium exists is NP-Complete.

	Single-Source	Multi-Source
Result	$\exists$ Nash equilibrium with cost equal to OPT	$\exists$ 3-approx Nash equilibrium with cost equal to OPT
Can handle directed	Yes	No
Players can have more than 2 terminals	No	Yes
Players can have maximum amount they are willing to pay, $\max(i)$	Yes	No
Polynomial time alg	Finds $(1+\varepsilon)$-approx Nash equilibrium that costs at most $1.55 \cdot OPT$	Finds $(4.65+\varepsilon)$-approx Nash equilibrium that costs at most $2 \cdot OPT$

TABLE 5.1
Extensions for our main results in the Connection Game (OPT is the cost of the centralized optimum)

	Single-Source	Multi-Source
Exists Nash	(1,1)	(3,1)
Can find Nash in poly-time	$(1+\varepsilon, 1.55)$	$(4.65+\varepsilon, 2)$
Lower Bounds on Existence	(1,1)	(1.5,1)

TABLE 5.2
Bicriteria approximations, written as (β, α), meaning there exists (or it is possible to find) a β-approximate Nash equilibrium that is only a factor of α more expensive than the centralized optimum.

6. Conclusion and Summary of Results. A summary of the major results in this paper can be found in Tables 5.1 and 5.2. The first table summarizes our results for the single-source and the general case, and the extensions for which these results hold. Table 5.2 summarizes our results in terms of bicriteria approximations, where the goal is to find an approximate Nash equilibrium that is approximately optimal in cost. Notice that while in the general multi-source case we have shown the existence of a cheap 3-approximate equilibrium, and a lower bound of 1.5 for this approximation, most of the bicriteria-approximation space remains unexplored. For example, it is still extremely possible that there exists a $(1+\varepsilon)$-approximate Nash equilibrium that costs $(1+\varepsilon)$ times the optimal centralized solution. Moreover, it is also very possible that there exists a $(1+\varepsilon)$-approximate Nash equilibrium that costs $2 \cdot OPT$ and *can be found in polynomial time.* Such a result would be extremely interesting, since when considering (β, α) approximate solutions as in Table 5.2, it is often much more important to ensure that β is small instead of α.

Acknowledgements. This project arose in a class taught by Eric Friedman. It began as joint work with Ranjith Rajagopalan. We would like to thank him for many useful insights and contributions.

REFERENCES

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. SIAM Journal on Computing, 24(3) 445-456 (1995).

[2] A. Akella, R. Karp, C. Papadimitrou, S. Seshan, and S. Shenker. Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP. In Proceedings of SIGCOMM, 2002

[3] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, L. Roditty. On nash equilibria for a network creation game. In *Proceedings of the 17th annual ACM-SIAM Symposium on Discrete Algorithms*, pp 89–98, 2006.

[4] A. Agarwal, M. Charikar. Unpublished manuscript.

[5] E. Anshelevich, B. Caskurlu. Price of stability in survivable network design. Submitted to SODA 2008.

[6] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, T. Roughgarden. The price of stability for network design with fair cost allocation. FOCS, '04.

[7] E. Anshelevich, A. Dasgupta, É. Tardos, T. Wexler. Near-optimal network design with selfish agents. STOC, '03.

[8] B. Awerbuch, Y. Azar, A. Epstein. The price of routing unsplittable flow. newblock Proc. of 37th STOC (2005), 57-66.

[9] V. Bala and S. Goyal. A Non-Cooperative Model of Network Formation. In *Econometrica* 68(2000), pages 1181–1229.

[10] M. Charikar, C. Chekuri, T. Cheung, Z. Dai, A. Goel, S. Guha. Approximation Algorithms for Directed Steiner Problems. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

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

[12] H. Chen, T. Roughgarden. Network design with weighted players. In *Proceedings of the 18th annual ACM Symposium on Parallelism in Algorithms and Architectures*, 2006.

[13] G. Christodoulou, E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the 37th annual ACM Symposium on Theory of Computing*, 2005.

[14] A. Czumaj, P. Krysta, and B. Vöcking. Selfish Traffic Allocation for Server Farms. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, pages 287–296, 2002.

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

[16] D. Dutta A. Goel, and J. Heidemann. Oblivious AQM and Nash Equilibrium. In *Proceeding of the IEEE Infocom*, 2003.

[17] A. Epstein, M. Feldman, Y. Mansour. Strong equilibrium in cost sharing connection games. In *Proceedings of the 8th Annual ACM Conference on Electronic Commerce*, 2007.

[18] A. Fabrikant, A. Luthra, E. Maneva, S. Papadimitriou, and S. Shenker. On a Network Creation Game. *ACM Symposium on Principles of Distributed Computing (PODC)*, 2003.

[19] U. Feige. A treshold of $\lg n$ for approximating set-cover. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 314-318, 1996.

[20] J. Feigenbaum, C. H. Papadimitriou, S. Shenker. Sharing the Cost of Multicast Transmissions. JCSS 63 (1) pages 21–41, 2001.

[21] A. Fiat, H. Kaplan, M. Levy, S. Olonetsky, R. Shabo. On the price of stability for designing undirected networks with fair cost allocations. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

[22] M. Goemans, and D. Williamson. A General Approximation Technique for Constrained Forest Problems. SIAM Journal on Computing, 24:296-317, 1995

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

[24] M. Hoefer. Non-cooperative tree creation. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, 2006.

[25] M. Hoefer. Non-cooperative facility location and covering games. In *Proceedings of the 17th International Symposium on Algorithms and Computation (ISAAC 2006)*, 2006.

[26] M. Hoefer, P. Krysta. Geometric network design with selfish agents. In *Proceedings of the 11th Internation Conference on Computing and Combinatorics (COCOON)*, 2005.

[27] K. Jain and V. Vazirani. Applications of Approximation Algorithms to Cooperative Games. In *Proceedings in the Annual ACM Symposium on Theory of Computing*, 2001.

[28] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413, 1999.

[29] M. Mavronicolas and P. Spirakis. The price of selfish routing. In *Proceedings of the 33rd Annual*

ACM Symposium on the Theory of Computing, pages 510–519, 2001.

[30] D. Monderer and L. S. Shapley. Potential Games. *Games and and Economic Behavior* 14, pages 124-143, 1996.

[31] C. Papadimitriou. Algorithms, Games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, pages 749–753, 2001.

[32] G. Robins and A. Zelikovsky Improved Steiner Tree Approximation in Graphs. In *Proc. 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 770–779, 2000.

[33] R.W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *Int. J. Game Theory* 2, 65–67, 1973.

[34] T. Roughgarden. The price of anarchy is independent of the network topology. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing*, 2002.

[35] T. Roughgarden. Stackelberg scheduling strategies. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing*, pages 104–113, 2001.

[36] T. Roughgarden and É. Tardos. How bad is selfish routing? In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 93–102, 2000. Full version to appear in *Journal of the ACM.*

[37] A. Schulz and N. Stier Moses. On the Performance of User Equilibria in Traffic Networks. In *ACM/SIAM Symposium on Discrete Algorithms*, 2003.

[38] S. Shenker. Making Greed Work in Networks: A Game-Theoretic Analysis of Switch Service Disciplines. In *IEEE/ACM Transactions on Networking*, pages 819–831, 1995.

[39] A. Vetta. Nash equilibria in competitive societies with applications to facility location, traffic routing and auctions. In *Proceedings of the Annual IEEE Symposium on the Foundations of Computer Science*, 2002.