Non-cooperative Tree Creation
(Extended Abstract)

Martin Hoefer *

Department of Computer & Information Science,
Konstanz University, Box D 67, 78457 Konstanz, Germany
hoefer@inf.uni-konstanz.de

Abstract. In this paper we consider the *connection game*, a simple network design game with independent selfish agents that was introduced by Anshelevich et al [4]. In addition we present a generalization called *backbone game* to model hierarchical network and backbone link creation between existing network structures. In contrast to the connection game each player considers a number of groups of terminals and wants to connect at least one terminal from each group into a network. In both games we focus on an important subclass of *tree games*, in which every feasible network is guaranteed to be connected.

For tree connection games, in which every player holds 2 terminals, we show that there is a Nash equilibrium as cheap as the optimum network. We give a polynomial time algorithm to find a cheap $(2+\epsilon)$-approximate Nash equilibrium, which can be generalized to a cheap $(3.1+\epsilon)$-approximate Nash equilibrium for the case of any number of terminals per player. This improves the guarantee of the only previous algorithm for the problem [4], which returns a $(4.65+\epsilon)$-approximate Nash equilibrium. Tightness results for the analysis of all algorithms are derived.

For single source backbone games, in which each player wants to connect one group to a common source, there is a Nash equilibrium as cheap as the optimum network and a polynomial time algorithm to find a cheap $(1+\epsilon)$-approximate Nash equilibrium.

1 Introduction

Analyzing networks like the Internet, which is created and maintained by independent selfish agents with relatively limited goals, has become a research area in game theory and computer science attracting a lot of interest. In the considered models it is important to explore the boundary between stable and socially efficient solutions. Hence, it is of interest to characterize the price of stability [3], which is the ratio of the cost of the best Nash equilibrium over the cost of a socially optimum solution. This captures *how good stability can get*, and has been studied in routing and network creation games [3, 4, 10, 16]. The more prominent measure is the price of anarchy [12] describing the cost of the worst instead of the best Nash equilibrium. It has received attention in networking problems, for instance routing [15], facility location [17] and load balancing [6, 12]. In

* Supported by DFG Research Training Group 1042 "Explorative Analysis and Visualization of Large Information Spaces".

R. Královic and P. Urzyczyn (Eds.): MFCS 2006, LNCS 4162, pp. 517–527, 2006.

this paper we consider these measures for the *connection game*, a game-theoretic model for network topology creation introduced by Anshelevich et al [4]. In a connection game each of the k selfish agents has a connectivity requirement, i.e. she holds a number of terminals at some nodes in a given graph and wants to connect these nodes into a component. Possible edges have costs, and agents offer money to purchase them. Once the sum of all agents offers for an edge exceeds its cost, it is considered *bought*. Bought edges can be used by *all* agents to establish their connection, no matter whether they contribute to the cost. Each agent tries to fulfill her connection requirement at the least possible cost.

In the connection game it might be optimal for the agents to create disconnected local subnetworks. The Internet, however, receives its power as a platform for information sharing and electronic trade from the fact that it is *globally* connected. Hence, it is reasonable to assume that agents to some extent have an interest in being connected to the network of other agents. We incorporate this idea by focusing on *tree connection games* - connection games, in which every feasible solution is connected. Furthermore we study the interest in globally connected networks in hierarchical networks with an extended model, which we call *backbone game*. We assume a scenario with existing, globally unconnected subnetworks of small capacity. Each agent wants to connect a set of subnetworks with a connected network of high performance backbone links. Backbone links can start and end at any terminal in the subnetworks, so we can consider subnetworks as groups of terminals in the graph and adjust the connectivity requirements to be present between certain groups. Each player must connect at least one terminal of each of her groups into a connected network at the least cost. Purchasing and using edges works similar to the connection game.

Related Work. The connection game was introduced and studied in [4], where a variety of results were presented. Both prices of anarchy and stability are $\Theta(k)$. It is NP-complete to determine, whether a given game has a Nash equilibrium at all. There is a polynomial time algorithm that finds a $(4.65 + \epsilon)$-approximate Nash equilibrium on a 2-approximate network. For the single-source case, in which each player needs to connect a single terminal to a common source, a polynomial time algorithm finds a $(1 + \epsilon)$-approximate Nash equilibrium on a 1.55-approximate solution. We denote these algorithms by ADTW-SS for the single source and ADTW for the general case. In [3] adjusted connection games were used to study the performance of the Shapley value cost sharing protocol. Each edge is bought in equal shares by each player using it to connect its terminals. The price of stability in this game is $O(\log k)$. Furthermore extended results were presented, e.g. on delays, weighted games and best-response dynamics. Recently, connection games have been studied in a geometric setting. In [10] bounds were shown on the price of anarchy and the minimum incentives to deviate from an assignment purchasing the socially optimum network. The case of 2 players and 2 terminals per player was characterized in terms of prices of anarchy and stability, approximate equilibria and best-response dynamics.

A network creation game of different type was considered in [7,5,2]. Here each agent corresponds to a node and can only create edges that are incident to her node. Similar settings are recently receiving increased attention in the area of social network analysis. See [11] for a recent overview over developments in the area of social network design games. In the context of large-scale computational networks, however, a lot of these models lack properties like arbitrary cost sharing of edges and complex connectivity requirements.

Our Results. In this paper we will consider tree connection games (TCG) and single source backbone games (SBG). The games exhibit connection requirements such that every feasible solution network is connected. We analyze them with respect to strict and approximate pure-strategy Nash equilibria. We are especially interested in deterministic polynomial time algorithms for a two-parameter optimization problem: Try to assign payments to the players such that the purchased feasible network is cheap and the incentives to deviate are low. In Section 2 we show that for any tree connection game with two terminals per player the price of stability is 1. We outline an algorithm that allocates edge costs of a centralized optimum solution to players such that no player has an incentive to deviate. As this algorithm is not efficient, we show in Section 3 how to find a 2-approximate Nash equilibrium purchasing the optimum network for TCGs with any number of terminals per player. It can be translated into a polynomial time algorithm for $(3.1 + \epsilon)$-approximate Nash equilibria purchasing a network of cost at most 1.55 times the optimum network cost. This improves over ADTW that provides $(4.65 + \epsilon)$-approximate Nash equilibria. In addition we derive a tightness argument for the design technique of our algorithm and ADTW. Both algorithms consider only the optimum network and use a bounding argument for deriving approximate Nash equilibria. We show that both are optimal with respect to the class of deterministic algorithms working only on the optimum network. Thus, methods with better performance guarantees can be found, however, they must explicitly employ cost and structure of possible deviations. This significantly complicates their design and analysis.

In Section 4 we introduce the backbone game. Some results from the connection game translate directly: (1) both prices of anarchy and stability are in $\Theta(k)$, (2) it is NP-complete to determine, whether a given game has a Nash equilibrium and (3) there is a lower bound of $\left(\frac{3}{2} - \epsilon\right)$ on approximate Nash equilibria purchasing the optimum network. Here we show that for SBGs the price of stability is 1. A $(1 + \epsilon)$-approximate Nash equilibrium can be found in polynomial time. The procedure delivers the same results for three different generalizations: (1) games with a single source group, (2) games with a directed graph, in which players need a direct connection to the single source and (3) games, in which each player i has a threshold max(i) and would like to stay unconnected if the assigned cost exceeds max(i).

All proofs sketched or missing in this extended abstract will be given completely in the full version of this paper.

2 The Price of Stability

Connection Games. The connection game for k players is defined as follows. For each game there is an undirected graph $G = (V, E)$, and a nonnegative cost $c(e)$ associated with each edge $e \in E$. Each player owns a set of terminals located at nodes of the graph that she wants to connect. A strategy for a player i is a function p_i, which specifies for each edge the amount $p_i(e)$ that i offers for the purchase of e. If the sum of the offers of all players to an edge e exceeds $c(e)$, the edge is bought. Bought edges can be used by all players to connect their terminals, no matter whether they contribute to the edge costs or not. An (a-approximate) Nash equilibrium is a payment scheme such that no player can reduce $\sum_{e \in E} p_i(e)$ (by more than a factor of a) by unilaterally choosing a different strategy. Note that each player insists on connecting her terminals, and hence considers only such strategies as alternatives.

The problem of finding an optimum centralized network for all players and an optimum strategy for a single player are the classic network design problems of the Steiner forest [1, 9] and the Steiner tree [14], respectively. For the rest of this paper we will denote an optimum centralized network by T^*. The subtree of T^* that player i uses to connect her terminals is denoted by T^i.

Tree Connection Games. We will deal with an interesting class of connection games, the tree connection games (TCG), which are games with tree connection requirements.

Definition 1. *In a connection game there are tree connection requirements if for any two nodes v_1 and v_{l+1} carrying terminals, there is a sequence of players $i_1, \ldots, i_l$ and nodes $v_2, \ldots v_l$ such that player i_j has terminals at nodes v_j and v_{j+1} for $j = 1, \ldots, l$.*

A TCG can be thought of as a splitting of a single global player into k players, which preserves the overall connection requirements. For the subclass of TCG with 2 terminals per player we will use the term path tree connection game (PTCG). A first observation is that the price of anarchy of the PTCG is k. This is straightforward with an instance consisting of two nodes and two parallel edges, where each node holds a terminal of each player. One edge e_1 has cost k, the other edge e_2 a cost of 1. If each player is assigned to purchase a share of 1 of e_1, the solution forms a Nash equilibrium.

Our algorithmic framework for deriving exact and approximate Nash equilibria is as follows. Until there is no player left, in each iteration it picks a player, assigns payments, removes the player and reduces the edge costs by the amount she paid. As candidates for this it considers leaf players.

Definition 2. *A player owns a* lonely terminal *t, if t is located at a node, where no terminal of another player is located. A player i in a TCG is a* leaf player*, if she owns a lonely terminal, and there is at most one node with a non-lonely terminal of i.*

Algorithmic Framework

1. $c^1(e) = c(e)$ for all $e \in E$.
2. For $iter \leftarrow 1$ to k
3. $\quad$ i is a leaf player if possible; otherwise an arbitrary player
4. $\quad$ Determine p_i using c^{iter}
5. $\quad$ Set $c^{iter+1}(e) \leftarrow c^{iter}(e) - p_i(e)$ for all $e \in E$
6. $\quad$ Remove i, contract edges of cost 0

Theorem 1. *The price of stability in the PTCG is 1.*

Proof. To prove our theorem we need the following technical lemma. Consider a game in which T^* contains all nodes from the graph G. There are two players h and i and a single source terminal at a node s shared by both players. Player i holds exactly one additional terminal.

Lemma 1. *In the described game there is a Nash equilibrium as cheap as T^*.*

Proof. (sketch) Algorithm 1 is used to construct a Nash equilibrium purchasing T^* for a game as described in the lemma. It considers edges in reverse BFS order from s with adjusted edge costs. Let T_e denote the part of T^* below an edge e and T_u the part below a node u, where $e \notin T_e$ but $u \in T_u$. When assigning the cost of e we use a cost function c' with $c'(e') = 0$ for $e' \in T^* \backslash T_e$ and $c'(e') = c(e')$ otherwise. A_i and A_h are the cheapest feasible deviation trees excluding e for players i and h, resp. We first focus on the question, whether p_h allows a cheaper deviation for h. In opposite to player i it is not trivial for player h to assume that all edges outside of T_e have cost 0. Consider a node u where multiple subtrees join. We know for each edge $e_1, e_2, e_3, \ldots$ below u that the tree $T_{e_j} + e_j$ is the optimum forest to connect the terminals of T_{e_j} to $T^* \backslash T_{e_j}$. But player h owns terminals in possibly *all* subtrees T_{e_j}. Is there a cheaper forest for h to connect her terminals in T_u to $T^* \backslash T_u$ than her calculated contribution?

Algorithm 1

1. For each edge e in reverse BFS order
2. Find cheapest deviations A_h and A_i for players h and i under c' and given p_h and p_i on T_e.
3. Assign $p_i(e) = \min(c(e),\ c'(A_i) - p_i(T_e))$.
4. Assign $p_h(e) = \min(c(e) - p_i(e),\ c'(A_h) - p_h(T_e))$.

Lemma 2. *The payment function p_h constructed by Algorithm 1 allows no cheaper deviation for player h.*

Proof. Let the edges $e_1, \ldots, e_l$ be the edges directly below a node u in T_u. Assume the algorithm was able to assign payments that cover the costs of each $T_{e_j} + e_j$, and that u is the first node, at which $p_h(T_u)$ is not optimal for h.

We create a new cost function c'_h with $c'_h(e') = 0$ for $e' \in T^* \backslash T_u$ and $c'_h(e') = c(e') - p_i(e')$ otherwise. We will see that T^* is the optimum network under c'_h. Suppose the cheapest deviation tree A_h is cheaper than T^* under c'_h (i.e. the contribution of h to T^*). W.l.o.g. A_h includes all edges of cost 0, especially all edges purchased completely by i and all edges of T^* outside T_u. Let T_{e_j} be a tree that is not completely part of A_h. Consider for each terminal t of h located in T_{e_j} the path from t to u in A_h. We denote this set of paths by P_{e_j}. Let P'_{e_j} be the set of subpaths from P_{e_j} containing for every $P \in P_{e_j}$ the first part between the terminal of h and the first node $w \notin T_{e_j}$. This node always exists because $u \notin T_{e_j}$, and it is in T^*, because T^* covers all nodes in G. The network $A_{e_j} = \bigcup_{P \in P'_{e_j}} P$ was considered as a feasible deviation when constructing the payments for $T_{e_j} + e_j$, as it connects every terminal in T_{e_j} to a node of $T^* \backslash T_{e_j}$. Furthermore, the payments of i were the same, hence the cost of A_{e_j} was the same. Using the assumption that u is the first node, for which T_u is not optimal, we know that $c(A_{e_j}) \geq c(T_{e_j} + e_j)$. So after substituting A_{e_j} by T_{e_j} and e_j in A_h, the new network is at least as cheap as $T_{e_j} + e_j$. To show that this new network is also feasible, suppose we iteratively remove a path $P \in P'_{e_j}$. Now there might other terminals, whose connections to u use parts of P. The last node w of P is the first node of P outside of T_{e_j}, and it stays connected to u as P is the first part of a path to u. All other nodes of P are in T_{e_j} and will be connected by T_{e_j} and e_j. Hence, all terminals affected by the

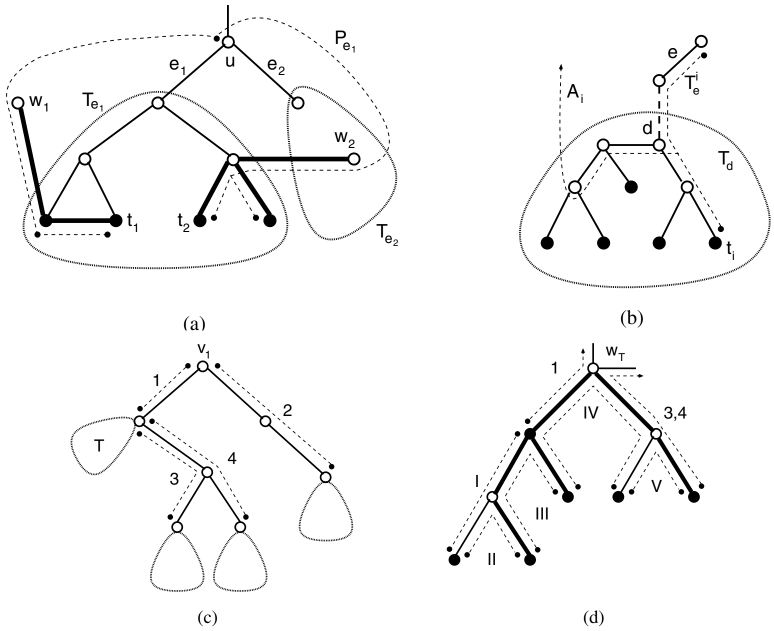

Fig. 1. (a), (b) Alternate trees and paths in PTCGs; (c) Distribution of hierarchical players for a parent player i, (d) distribution of personal players in the component T needed only by i

removal of P will finally be reconnected to u. In this way A_h can be transformed into T^* without cost increase. This proves that T^* is optimal under c'_h, so Algorithm 1 finds Nash equilibria. □

Figure 1a depicts the argument. Paths from the set P_{e_1} are indicated by dashed lines. The subgraph A_{e_1} of A_h is drawn bold. A node w can be either completely outside T_u (like w_1 for t_1) or in another T_{e_j} (like w_2 for t_2). Replacing A_{e_1} by T_{e_1} yields a feasible network that is not more expensive.

Finally, it is also possible to show that the payments calculated by Algorithm 1 purchase T^*. This proves Lemma 1. □

Now we outline how to use Lemma 1 to prove Theorem 1. Suppose we are given a PTCG with k players. At first we simplify the graph by constructing an equivalent *metric-closure game* with the same players, terminals and a complete graph G' on the nodes of T^*. Edge costs are equal to the cost of the shortest path in G. Then we use the framework with an induction on the number of players. Assume that the theorem holds for any PTCG with $k-1$ players. Now consider step 4 for a game with k players. If there is no leaf player, we can feasibly pick any player i and let $p_i = 0$. Otherwise, if i is a leaf player, we assign her to pay as much as possible on T^i such that she has no incentive to deviate. If the reduced network after the framework iteration is optimal under the reduced cost for the remaining $k-1$ players, the theorem follows with the induction hypothesis. Here we use Lemma 1 and Algorithm 1 to make the argument. Introduce a global player h, who accumulates all players except i. Players h and i share a single source at a node s, and an equilibrium assignment for h represents an optimal solution for the remaining players after removal of i. Hence, our inductive step is proven by Lemma 1. $\square$

3 Approximation of Nash Equilibria

In this section we present an algorithm to calculate cheap approximate Nash equilibria in polynomial time. The algorithm sketched in the proof of Theorem 1 is not efficient. Either we must provide the socially optimum network as input or we must construct the optimum deviation for the collective player h to improve the solution network. In any case this requires to solve an instance of the Steiner tree problem. Instead, in this section we use *connection sets* to construct polynomial time approximation algorithms.

Definition 3. *[4] A connection set S of player i is a subset of edges of T^i, such that for each connected component C in $T^* \setminus S$ either (1) there is a terminal of i in C, or (2) any player that has a terminal in C has all of its terminals in C.*

For any node without a terminal and degree 2 in T^* the incident edges belong to the same connection sets, so for convenience we assume that T^* has no such nodes. If every player purchases at most α connection sets, the payments will form an α-approximate Nash equilibrium. Note that a subset of a connection set also is a connection set.

An algorithm for PTCGs. In connection games with 2 terminals per player the edges of T^* can be partitioned into equivalence classes S_J, where e and f belong to the same class iff $J = \{j : e \in T^j\} = \{j : f \in T^j\}$. Each S_J forms a connection set for all players $j \in J$, which is maximal under the subset relation. We will say that connection set S_J is *needed* by J. In the following PTCGs we will only consider maximal connection sets and not explicitly mention a player. This information is given by the subtrees, in which the set is located. Furthermore, when tree connection requirements are present, connection sets are contiguous.

Our algorithm uses the framework. In step 4 it assigns a leaf player i to purchase 2 connection sets. If i is no leaf player, $p_i = 0$. If a leaf player i is removed in step 5, distinct connection sets might join and subsequently be wrongly regarded as a single connection set. This happens when they are needed by player sets differing only by i. We will use the notion of endangered sets to refer to these problematic sets.

Definition 4. *A connection set is called* endangered set *for player i if it is needed by the set of players $J \cup \{i\}$, and there is another connection set (called* forcing set*) needed by the set J, with $i \notin J$.*

Lemma 3. *For any leaf player in a PTCG there are at most 2 endangered sets.*

We will denote the endangered set with empty forcing set as the *personal set*, the endangered set with nonempty forcing set as the *community set*. In step 4 we simply assign a leaf player to purchase these sets. It requires an easy inductive argument to show that the algorithm then works correctly. This yields the following theorem.

Theorem 2. *For any optimum centralized solution T^* in a PTCG, there exists a 2-approximate Nash equilibrium such that the purchased edges are exactly T^*.*

For PTCGs we can use a 1.55-approximation algorithm for the Steiner tree problem [14] to get an initial approximation T. Furthermore, we use shortest-path algorithms to find deviations and connection sets of optimum cost. Similar to [4] we can iteratively improve T by exchanging connection sets with better paths. Polynomial running time is ensured by substantial cost reduction in each exchange step, which can be achieved by an appropriate adjustment of the edge costs. Thus, for PTCGs there is an algorithm constructing $(2+\epsilon)$-approximate Nash equilibria on 1.55-approximate networks in time polynomial in n and ϵ^{-1}, for any $\epsilon > 0$.

An algorithm for TCGs. Next we adjust our algorithm to deliver 2-approximate Nash equilibria puchasing T^* for TCGs with any number of terminals per player. While this adjustment is possible using connection sets, it remains an open problem, whether Theorem 1 also holds for TCGs with more terminals per player.

Each player (denoted as parent player) is divided into a set of child players with 2 terminals per player. Terminals of the child players are located at the same nodes as the ones of the parent player. In addition terminals of child players are distributed such that they create a PTCG. Hence, the set of all child players can purchase T^* with 2 connection sets per player.

The algorithm again uses the framework, and in step 4 a special procedure to assign the cost of T^* to the parent player i. First player i is divided into child players. Then child players of i are iteratively assigned to purchase endangered sets and removed. In the end i has to purchase all edges assigned to her child players. To identify personal and community sets for the child players of i, one might first create a PTCG by splitting all other parent players. However, the next lemma ensures that the assignment of personal and community sets for a leaf parent player i does not depend on the splitting of the other parent players. In step 4 we thus assign edges without explicitly splitting other players than i.

Lemma 4. *The endangered sets of child players of a leaf parent player* i *are independent of the division of other parent players.*

In the remainder of the section we show how to divide a player i into *hierarchical*, *personal* and *superfluous* child players such that the union of connection sets purchased by her child players forms 2 connection sets. At first we disregard all non-lonely terminals but one. Denote the node carrying the last remaining non-lonely terminal t by v_1. If the player has only lonely terminals, we pick t arbitrarily. Then consider T^* rooted at v_1. Once we arrive at an edge e needed only by i, the tree connection requirements guarantee that the whole subtree below e is also needed only by i. Here we insert child players in a hierarchical fashion. We contract all edges that are needed only by i. Let this adjusted tree be denoted T' and consider it in BFS-order rooted at v_1. For each node v carrying a terminal of i, we introduce a new child player. She has a terminal at v and the nearest ancestor of v in the tree carrying a terminal of i (see Figure 1c). These child players will be termed *hierarchical players*.

Second, we consider the portions of the tree that were contracted to form T'. For each maximal connected subtree $T \subset T^i$ that is needed only by i, let w_T be the root node that T shares with T'. Let player j be the first hierarchical child player, whose terminal t_j was placed at w_T. This player connects upwards in T'. Now we consider

T in DFS-order and locate t_j at the first node carrying a terminal of i. For each new node w_z carrying a terminal encountered in the DFS order, we introduce a new child player and locate her terminals at the nodes w_{z-1} and w_z. At any time there is only one lonely terminal in T. Finally, when the last node w_l carrying a terminal of i is reached, we move all remaining terminals at w_T to w_l. They belong to the hierarchical players connecting downwards in T'. Child players introduced in the DFS-scan of the components T are called *personal players*, because they divide parts needed only by i (see Figure 1d).

Third, for every non-lonely terminal of i disregarded in the beginning, we introduce a *superfluous* child player connecting the terminal to v_1, which will not be assigned any payments.

Theorem 3. *For any optimum centralized solution T^* in a TCG, there exists a 2-approximate Nash equilibrium such that the purchased edges are exactly T^*.*

In the proof a special elimination order of child players is used. In any iteration a leaf player is picked, first the superfluous players and then in a bottom-up fashion to v_1 the personal and hierarchical players. One connection set for the parent player is formed by the union of all personal sets for the child players. The other connection set is the union of the community sets. In the full version we will show how to use polynomial time approximation algorithms to derive a $(3.1 + \epsilon)$-approximate Nash equilibrium on a 1.55-approximate network.

A tightness argument. For every connection game ADTW finds 3-approximate Nash equilibria purchasing T^* given only the optimum network T^* as input. Like our algorithm it uses connection sets and does not employ cost sharing of edges. The next theorem shows that no deterministic algorithm using only T^* as input can improve the guarantees even if it uses cost sharing. In this way ADTW for general connection games and our algorithm for TCGs represent optimal algorithms, but our algorithm provides a better guarantee on TCGs.

Theorem 4. *For any $\epsilon > 0$ there is a connection game [TCG] such that any deterministic algorithm using only T^* as input constructs a payment function, which is at least a $(3 - \epsilon)$ $[(2 - \epsilon)]$ approximate Nash equilibrium.*

Theorem 5. *For any $\epsilon > 0$ there is a TCG such that ADTW constructs a $(3 - \epsilon)$-approximate Nash equilibrium.*

4 Backbone Games

In this section we present the *backbone game*, an extension of the connection game to groups of terminals. Each of the k players has a set of groups of terminals. Each terminal may be located at a different node. The player strives to connect at least one terminal from each of her groups into a connected network. Different terminals may be located at the same nodes. Some important results from [4] translate directly to the backbone game by restriction to the connection game. The price of anarchy is k, and the price of stability $k - 2$. It is *NP*-complete to decide, whether a given game has a Nash equilibrium, and there is a lower bound of $\left(\frac{3}{2} - \epsilon\right)$ on approximate Nash equilibria purchasing T^*.

Finding the optimum network for a single player is the network design problem of the Group Steiner Tree (GSTP) [13]. The problem of finding a centralized optimum solution network T^* generalizes the GSTP in terms of forest connection requirements, so we term this the *Group Steiner Forest Problem (GSFP)*. There are polylogarithmic approximation algorithms for the GSTP [8], but we are not aware of any such results for the GSFP. Hence, we will concentrate on algorithms for games, in which the solution is guaranteed to be connected. The general case represents an interesting field for future work.

Single Source Backbones. In a SBG each player i has a group $\mathcal{G}_i$ of g_i terminals and must connect at least one terminal to a given source node s. Note that the price of anarchy is still k as the example establishing the bound is a single source game. However, the price of stability is 1, and cheap approximate equilibria can be found in polynomial time.

Theorem 6. *The price of stability in the SBG is 1. There is a polynomial time algorithm to find a $(1 + \epsilon)$-approximate Nash equilibrium purchasing a network T with $c(T)/c(T^*) \in O(\log n \log k \log(\max_i g_i))$.*

References

1. A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J Comp*, 24(3):445–456, 1995.
2. S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On nash equilibria for a network creation game. In *Proc 17th Ann ACM-SIAM Symp Discrete Algorithms (SODA)*, pages 89–98, 2006.
3. E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc 45th Ann IEEE Symp Foundations Comp Sci (FOCS)*, pages 295–304, 2004.
4. E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proc 35th Ann ACM Symp Theo Comp (STOC)*, pages 511–520, 2003.
5. J. Corbo and D. Parkes. The price of selfish behavior in bilateral network formation. In *Proc 24th Ann ACM Symp Principles of Distributed Comp (PODC)*, 2005.
6. A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proc 34th Ann ACM Symp Theory Comp (STOC)*, pages 287–296, 2002.
7. A. Fabrikant, A. Luthera, E. Maneva, C. Papadimitriou, and S. Shenker. On a network creation game. In *Proc 22nd Ann ACM Symp Principles of Distributed Comp (PODC)*, pages 347–351, 2003.
8. N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the Group Steiner tree problem. *J Algorithms*, 37:66–84, 2000.
9. M. Goemams and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J Comp*, 24(2):296–317, 1995.
10. M. Hoefer and P. Krysta. Geometric network design with selfish agents. In *Proc 11th Conf on Comp and Comb (COCOON), LNCS 3595*, pages 167–178, 2005.
11. M. Jackson. A survey of models of network formation: Stability and efficiency. In G. Demange and M. Wooders, editors, *Group Formation in Economics; Networks, Clubs and Coalitions*, chapter 1. Cambridge University Press, Cambridge, 2004.
12. E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc 16th Ann Symp Theoretical Aspects Comp Sci (STACS)*, pages 404–413, 1999.

13. G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Proc 15th Intl Workshop on Graph-Theoretic Concepts Comp Sci (WG), LNCS 411*, pages 196–210, 1989.
14. G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proc 10th Ann ACM-SIAM Symp Discrete Algorithms (SODA)*, pages 770–779, 2000.
15. T. Roughgarden and É.Tardos. How bad is selfish routing? *J ACM*, 49(2):236–259, 2002.
16. A. Schulz and N. Stier Moses. Selfish routing in capacitated networks. *Math Oper Res*, 29(4):961–976, 2004.
17. A. Vetta. Nash equilibria in competitive societies with application to facility location, traffic routing and auctions. In *Proc 43rd Ann IEEE Symp Foundations Comp Sci (FOCS)*, page 416, 2002.