Network Design with Selfish Agents
GI Seminar Report, Dagstuhl 2004

Łukasz Kaiser

Mathematische Grundlagen der Informatik
RWTH Aachen

1 Introduction

In the study of multi-player games it is usual that a network, a graph of some kind must be constructed and agents need to pay for the construction of the network together. There is a large number of practical scenarios where this situation occurs, for example when internet or telephone connections are created or when multiple transport companies have to pay for the communication network.

To illustrate the situation we will consider sea transport companies or broadband internet providers. We choose such scenarios as we do not want to analyse the problem of limited capacities of network connections, so each connection in our network is either bought by the companies and can be used as much as necessary or it is not used at all. Accordingly, each connection has a constant cost and we will say that it is bought if all companies together decide to pay for it the price that covers the cost.

Each company has a few users or ports and it has to ensure that these will be connected, it is not possible that any of these places will be disconnected from any other. We do not know how each company, or in a more general setting - each agent, will decide how much to pay for a connection, but we assume that they are *selfish*. It is not possible that companies will negotiate or communicate in some way to find the optimal solution, in network connection games they act only to maximise their own benefit, to minimise the cost necessary to connect all the places that must be connected. In such setting everyone will go for a cheaper price if possible.

The situation when no one can gain anything by changing his payments is called an *equilibrium*. When no external mechanism or regulation is present, unrestricted selfishness might lead to situations where it is impossible to find an equilibrium or the total payment of all companies in any equilibrium is much higher than the possibly optimal one. We call the ratio between cost of the best solution that everyone can agree on and the globally optimal solution the *price of stability* and the ratio between the worse solution that everyone agrees on and the optimal is the *price of anarchy*.

When the price of stability is high, so the selfishness of agents can lead to serious loss, we can introduce some mechanism to handle problematic situations. There are many kinds of mechanisms with different properties, of which the *marginal cost* and *Shapley value* are most important for network design games.

We will not discuss the design of mechanisms here, we will just introduce fair connection games that make use of a simple mechanism based on Shapley value. It turns out that even such simple mechanism can cope with many problems that unlimited selfishness poses and lead to quite good solutions.

Mechanisms and their properties were studied in detail in multiple scenarios, an overview of the results relevant for cost sharing by network construction can be found in [4]. Multi-player games where network costs are analysed were used in multiple scenarios cite like congestion, load balancing, routing and facility location, for example in [6–8, 3].

2 Notation

We will model the considered network as an undirected graph $G = (V, E)$ where vertices correspond to the places that need to be connected and edges correspond to possible connections. Each edge e has therefore a cost denoted by $c(e)$. We assume that there are N agents and each agent i has an assigned set $V_i \subseteq V$ of vertices he needs to connect.

We want to model the process where agents declare payments for each connection and buy them if the declared payment exceeds the cost of the connection. The bought connections form a network where each agent's vertices V_i have to be connected. We will model it as a play between the agents where everyone fights to minimise his total payment and we will call this a *connection game*.

Therefore we will say that a *strategy* p_i of an agent or player i is a function that assigns to each edge e a real-valued payment $p_i(e)$. In fair connection games the agents will not declare payments but only choose a set of edges they want to use and the payment will be computed according to a mechanism, which we will discuss later.

We will say that a *play* is a tuple of strategies of all players, $p = (p_1, \ldots . p_N)$ and p induces a graph $G_p = (V, E_p)$ of bought edges

$$e \in E_p \iff \sum_i p_i(e) \geq c(e).$$

We will say that a play p is correct for player i if V_i is connected in G_p and we will say that it is correct if it is correct for all players.

The goal of the agent i is to minimise his total payment, which is defined for player i with respect to a strategy p_i as

$$P_i = \sum_{e \in E} p_i(e).$$

We will also sometimes use the total payment of all players given a play p as a sum of payments of all players using their strategies,

$$P_p = P_1 + \ldots + P_N.$$

We are interested in situations when all players agree to play their strategy, so no player can decrease his payment by changing the strategy when strategies of other players remain unchanged. This corresponds to the well-known notion of (pure) Nash Equilibrium, but we have to take into account that only correct games are considered.

Therefore a play p is a (pure) Nash equilibrium if p_i minimises the total payment of player i necessary for a play that is correct for him. This must hold for each player i when all p_j remain unchanged for $j \neq i$. Additionally a play p is an c-approximate Nash equilibrium if each player can decrease the total payment only by a factor of c by choosing a different strategy.

More formally $p = (p_1, \ldots, p_N)$ is a (pure) Nash Equilibrium if for any player i and any alternative strategy p'_i for this player if the play $p' = (p_1, \ldots, p'_i, \ldots, p_N)$ where only player i changed his strategy is correct for i then $P_{p'} \geq P_p$. The play is a c-approximate Nash Equilibrium if analogous property holds but with $c * P_{p'} \geq P_p$ and of course we consider $c > 1$.

To define formally the *price of anarchy* and the *price of stability* let us consider the globally optimal solution in a game, so let c_{opt} be the smallest payment P_p among all possible plays p. Further let c_{st} be the smallest payment P_q among all possible plays q that are Nash Equilibria and let c_{an} be the biggest of these payments. Then the price of stability is defined as $\frac{c_{st}}{c_{opt}}$ and the price of anarchy as $\frac{c_{an}}{c_{opt}}$.

3 Connection Games

We have defined what connection games are and adopted Nash Equilibria to such games, let us now look at a few basic properties of such equilibria in these games.

If p is a Nash equilibrium and T^i is the tree in G_p that connects all vertices of player i (i.e. V_i), then the following holds:

(i) each player i only contributes to costs of edges in T^i,
(ii) each edge in G_p is either fully paid for or not at all,
(iii) G_p is a forest.

Indeed, for the first point please note that if player i contributed to some edge outside T^i then he could decrease his payment by refusing to pay for that edge and his vertices would remain connected by T^i. Also if there was an edge that would not be fully paid for or that creates a cycle then the players could refuse to pay for it at all without disconnecting any vertices.

In basic connection games as we defined them (pure) Nash Equilibria do not always exist. To see it please look at Figure 1 where s_1 and t_1 belong to player 1 and s_2 and t_2 belong to player 2 and each edge has cost 1. We can assume without loosing generality that the path $t_2 - s_1 - s_2 - t_1$ was bought by the players. According to the above properties the path $s_2 - t_1$ is fully paid for by player 1 as it is not in T^2 and $t_2 - s_1$ is fully paid for by player 2 as it is not in T^1. Let us assume that player 1 (or symmetrically player 2) pays some non-zero amount for $s_1 - s_2$. Then he can stop paying anything for this edge and for $s_2 - t_1$

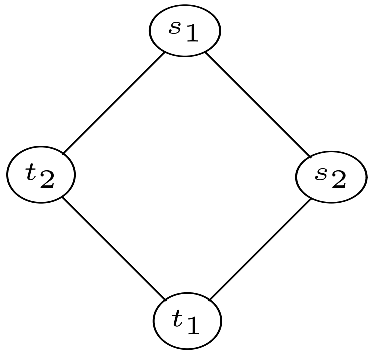

Fig. 1. Nash Equilibria do not Always Exist

for which he paid 1 and instead just buy $t_2 - t_1$ for 1. In this way the player can always decrease his payment, so no Nash Equilibrium is possible in this game.

To see that the price of anarchy can be as high as N please look at Figure 2 and notice, that if each player buys one of the edges on the leftmost path, each one paying 1, then no one can gain by changing the strategy so it is an equilibrium. Still all players could buy the edge on the right together, with everyone paying only $\frac{1}{N}$, which is also an equilibrium with a much lower total cost.

The biggest problem with connection games is that not only the price of anarchy but also the price of stability can be high. Looking at Figure 3 you can see that the optimal (lowest) total cost is $1 + 3\epsilon$ when the players but the leftmost path and three of the ϵ-edges in the square on the right. But the lowest cost achieved in an equilibrium is $N - 2$ when the players buy the two edges with cost $\frac{N}{2} - 1 - \epsilon$ and two ϵ edges in the right square. No three edges in the ϵ-rectangle can be paid for by any equilibrium for the same reasons as discussed by Figure 1.

To overcome the problem with the high price of stability we can either introduce a simple mechanism to assign payments, what we will discuss in the context of fair connection games, or analyse a restricted class of connection games, namely single source connection games, where the price of stability is 1.

In general connection games we already know that the price of stability is high, near the upper bound N where the player can buy the whole optimal tree himself. Still there are some positive results with approximate equilibria.

Theorem 1 ([2]). *In each connection game a 3-approximate Nash Equilibrium exists and can be computed in exponential time. In polynomial time we can compute a $(4, 55 + \epsilon)$-approximate equilibrium.*

The lower bound for approximate Nash Equilibria in connection games is $\frac{3}{2}$, meaning that there exist a sequence of games that any equilibrium purchasing the optimal solution must be (in the limit) ($\frac{3}{2}$)-approximate.

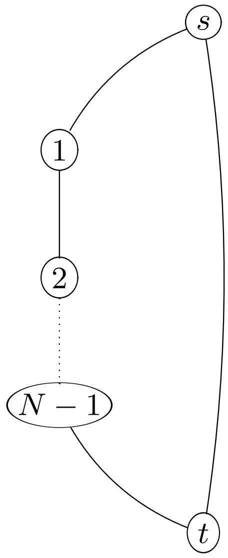

Fig. 2. Price of Anarchy is N

Even though there always are approximate Nash Equilibria it is still interesting to know if in a given game there exists any strict Nash Equilibrium and how complex it is to calculate it.

Theorem 2 ([2]). *Determining the existence of a Nash Equilibrium in a connection game in* NP*-hard if the number of players is a part of the input.*

Proof. To prove this theorem we will use a reduction to 3-SAT and the two kinds of gadgets presented on Figure 4. The gadgets of the first kind will be constructed for each variable x_i occurring in the 3-SAT formula from which we reduce and we will say that player i that needs to connect s_i and t_i in this gadget is a variable player for variable x_i. In each variable gadget we have two edges labelled with e_{iT} and e_{iF} as presented on the picture and when an assignment of variables is given we will say that e_{iT} was assigned if x_i is true and that e_{iF} was assigned in the other case. The gadgets of the second kind will be constructed for each clause in the formula and clause players will play in these gadgets.

The middle edges in the clause gadget are the same as edges in the corresponding variable gadgets that are labelled accordingly, for example on Figure 4 the clause gadget represents the clause

$$\neg x_1 \vee \neg x_2 \vee x_3$$

and the edge labelled with e_{1F} is the same edge as the one in the variable gadget for x_1 labelled in the same way.

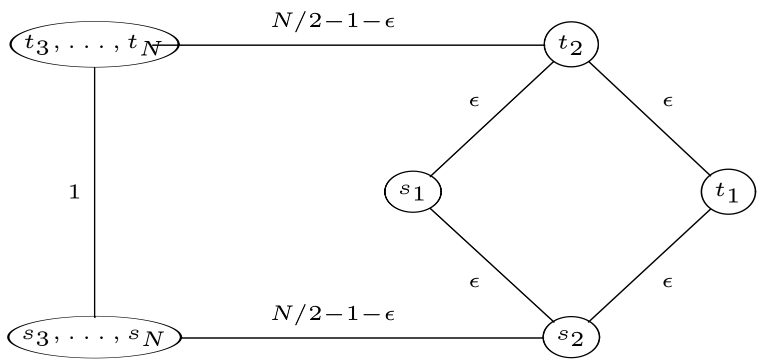

Fig. 3. Price of Stability is about $N - 2$

Let us now assume that the formula has a satisfying assignment and let each variable player pay the full cost of the path that contains the assigned edge. Since in each clause gadget one of the middle edges is now paid for by the variable player each clause player can now use it and connect the vertices he needs to connect by fully paying for two edges inside the gadget. It is easy to check that this is indeed a Nash Equilibrium in the presented connection game.

Let us now assume there is a Nash Equilibrium in the connection game. First we have to notice that the edges on the perimeter of clause gadgets are not used at all in any Nash Equilibrium. To see this one can use use arguments similar to those used to show that there is no Nash Equilibrium in the game presented on Figure 1 or just enumerate all possible cases. You should also note that no clause player will pay more than 2 in any Nash Equilibrium as he can connect his vertices using the two edges on the perimeter.

Therefore in each clause there must be at least one middle edge that will be fully paid for by the variable player. But no variable player i will pay for both e_{iF} and e_{iT} in an equilibrium as he clearly needs to pay only for one of these. So if a Nash Equilibrium exists then each variable player pays for just one edge and the corresponding assignment satisfies each clause as in each clause gadget one of the middle edges has to be paid for by the variable player. This concludes the proof.

Please note that in the special case of a connection game with two players each with two vertices the equilibrium can be found in polynomial time [2].

4 Single Source Connection Games

Let us now consider a class of connection games where each player can have only two vertices to connect and one of them is common for all players. We will denote the common vertex by s and the other vertices by $\{t_1, \ldots, t_N\}$ and T^* will be the minimum cost (Steiner) tree rooted at s and containing $\{t_1, \ldots, t_N\}$.

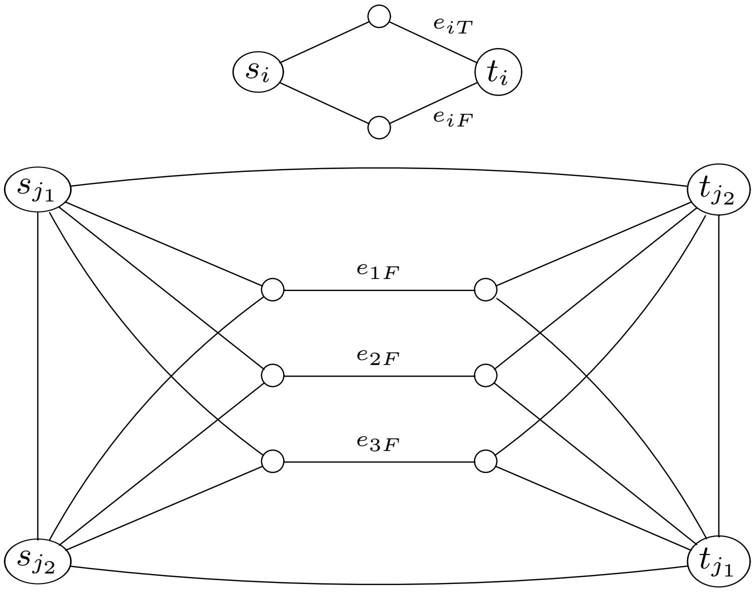

Fig. 4. Equilibrium Existence is NP-hard

By definition paying for T^* is the optimal solution in such game and we will show that indeed there is a Nash Equilibrium where players pay exactly for T^*. Let us first define the games we want to describe and state the theorem.

Definition 1. *Single source connection games are such connection games, where all players share a common vertex s and in addition each player has exactly one other vertex t_i to connect to s, so $V_i = \{s, t_i\}$ for each player i.*

Theorem 3 ([2]). *In a single source connection game there is a Nash Equilibrium purchasing the minimum cost tree T^* containing $\{s, t_1, \ldots, t_N\}$ and therefore the price of stability is 1.*

To prove this we will present an algorithm that constructs the Nash Equilibrium by assigning what each player should pay for each edge. In the algorithm we will be going in reverse breadth-first search (BFS) order through edges in T^* and in each step we will set payments of all players for the considered edge.

Let us first introduce some concepts and notation used in the algorithm. When $p_i(e)$ is the payment of player i for edge e at the given point of the run of the algorithm (we start with $p_i(e) = 0$ and increase them), then the payment of player i for the tree T^* at this point is defined as

$$p_i(T^*) = \sum_{e \in T^*} p_i(e),$$

and the total payment for the edge e at this point is defined as

$$p(e) = \sum_i p_i(e).$$

When considering an edge e we will denote the subtree of T^* disconnected from s when e is removed by T_e.

Since we assign the payments dynamically during the run of the algorithm we will also consider the *modified costs* for player i defined with respect to what the player has already paid for T^* and are defined as

$$c'(e) = \begin{cases} p_i(e) & \text{for } e \in T^*, \\ c(e) & \text{for } e \notin T^*. \end{cases}$$

Now we can present Algorithm 1 that assigns payments to players buying the whole tree T^*.

Algorithm 1 Payment Construction Algorithm

```
Initialise p_i(e) = 0 for all players and edges.
for all edges e in T* in reverse BFS order do
  for all players i with t_i ∈ T_e do
    repeat
      if e is a cut in G then
        set p_i(e) = c(e)
      else
        let χ_i be the cost of the cheapest path A from s to t_i in G\{e} under the
        modified costs c',
        set p_i(e) = min{χ_i − p_i(T*), c(e) − p(e)}
      end if
    until e is fully paid for
  end for
end for
```

Please first notice that the above algorithm returns a Nash Equilibrium if it terminates. To see it assume the contrary, namely that there is a possibility for player i to pay less and keep t_i connected with s. In such case these is an edge e in T^* for which i now refuses to pay and instead chooses to pay for some cheaper alternate path A. Let us return to the moment during the run of the algorithm when i decided to pay for e. Please notice that at that moment, what he paid for e was less than the cost of any alternate path minus what i has paid for T^* below e. This contradicts the existence of the cheaper path A, so the output of the algorithm is indeed a Nash Equilibrium.

Before we start to prove that the above algorithm succeeds to pay for each edge please take a look at Figure 5 that presents the situation during the run of the algorithm.

In this situation we assume that at some point players with vertices in T_e are unwilling to pay for one of the edges e as depicted on the figure. For each player

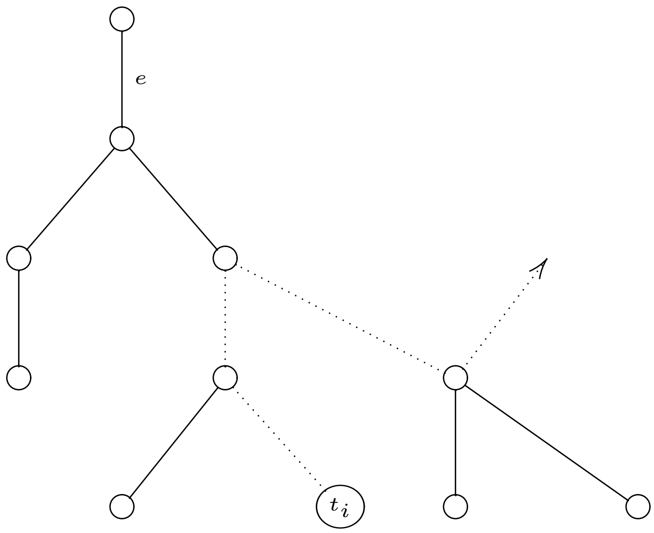

Fig. 5. Situation During the Run of the Algorithm

i with $V_i \cap T_e \neq \emptyset$ there is an alternate path A_i explaining his unwillingness to pay for e. If more than one such path exists, we choose one including as many ancestors of the player i's vertex t_i as possible.

In such situation inside the algorithm $\chi_i = c'(A_i)$ and using this fact we will build a cheaper tree than T^* from these paths and thus reach a contradiction. To complete the proof we need the following path lemma.

Lemma 1. *When A_i is an alternate path as chosen before then it has precisely three segments, one in T_e, one in $E \backslash T^*$ and the rest in $T^* \backslash T_e$.*

Proof. Please first note that once A_i reaches $T^* \backslash T_e$ then it remains there as edges there have cost $c'(f) = p_i(f) = 0$ since the algorithm runs in reverse BFS order and did not visit these edges yet.

Suppose that A_i leaves T_e and returns to T_e and we will show that this leads to a contradiction with the choice of A_i by constructing a different path that would be chosen instead of A_i in such case.

Let P_1 be the starting segment of A_i in T_e and P_2 be the segment that continues outside T_e until a vertex x back in T_e is reached. Consider y, the lowest common ancestor of x and t_i and let P_3 be the path from t_i to y and P_4 the path from y to x. You can look at Figure 6 to visualise the situation and we will argue that $P_3 \cup P_4$ should have been chosen instead of $P_1 \cup P_2$.

It should be clear that e is somewhere over y as x and t_i are both in T_e. Since e is the first edge for which the algorithm refuses to pay, so the algorithm did not fail before and therefore $P_3 \cup P_4$ is at least as cheap as $P_1 \cup P_2$. As $P_3 \cup P_4$

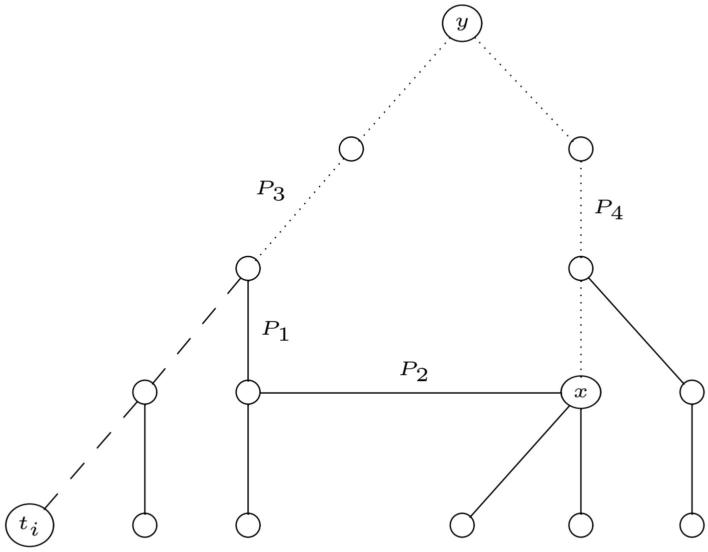

Fig. 6. Proof of Path Lemma

induces a higher ancestor y than $P_1 \cup P_2$ we have a contradiction with the choice of A_i as the one with most ancestors, which completes the proof of the lemma.

Using the above lemma we can now finish the proof of correctness of the algorithm. Let us assume that the algorithm fails to pay for some edge e so each player i with $V_i \cap T_e \neq \emptyset$ has now an alternate path A_i that leaves T_e at vertex d_i and does not return there any more according to the path lemma.

The situation is depicted on Figure 7 and we will now show that in such case T^* can not be the cheapest tree as we assumed.

Let us consider all the edges in $T^* \backslash T_e$ together with all the edges in the alternate paths A_i. Clearly these edges connect all vertices $t_i \in T_e$ with s as all paths A_i are included and also these connect all other vertices t_j with s as $T^* \backslash T_e$ is included. Therefore to reach the desired contradiction it is enough to show that the total cost of all these edges is smaller that the cost of T^*.

Since the algorithm failed to pay for the edge e we know that with respect to the payments and modified costs at that point in the run of the algorithm it holds

$$c(e) > \sum_{\{i \,:\, t_i \in T_e\}} c'(A_i) - p_i(T^*).$$

As the algorithm runs in reversed BFS order we know that the sum of payments for T^* of all players with nodes in T_e is the cost of T_e so the above can be rewritten as

$$c(e) + c(T_e) > \sum_{\{i \,:\, t_i \in T_e\}} c'(A_i). \qquad (1)$$

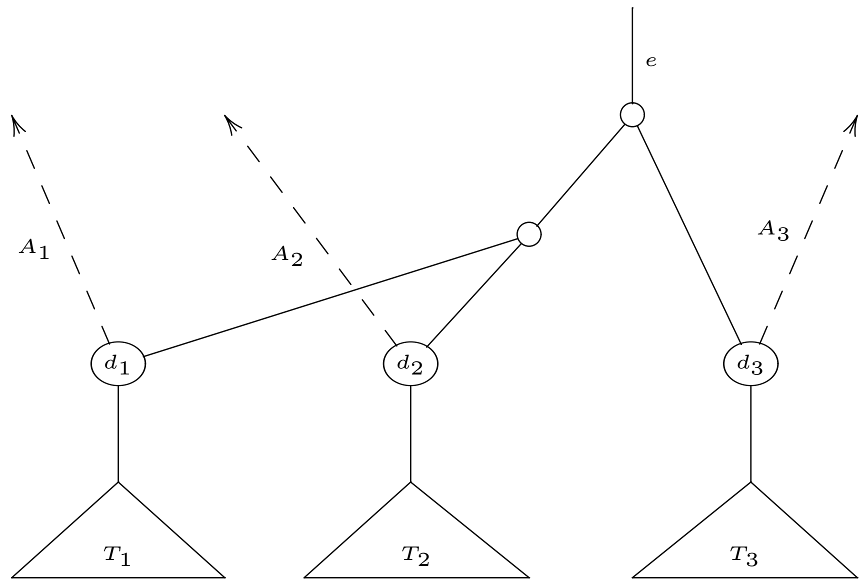

Fig. 7. Constructing Cheaper Tree

We know from the path lemma that each A_i has three parts, the part A_i^e that is inside T_e, the part A_i^{out} that is outside T^* and the final part A_i^* that is in $T^* \backslash T_e$. By definition the modified cost of A_i^{out} is the cost of this path, the modified cost of A_i^* is 0 and the modified cost of A_i^e is what i paid for it. Therefore we can add the cost of $T^* \backslash T_e$ to both sides of (1) and break each A_i into three parts obtaining

$$c(T^*) > c(T^* \backslash T_e) + \sum_{\{i \,:\, t_i \in T_e\}} c(A_i^{out}) + \sum_{\{i \,:\, t_i \in T_e\}} p_i(A_i^e).$$

Therefore to complete the proof we only need to show that the last component, the payments of all players for the paths A_i inside T_e, is bigger than the total cost of the edges belonging to paths A_i inside T_e. To see this look at the points d_i and notice that by the choice of A_i in a subtree rooted at any d_i there can be no other point d_j, as then either player i could deviate in d_j and get a cheaper path or player j could deviate in d_i and get a path with more ancestors. Therefore each tree rooted at any d_i must be fully paid for as the algorithm worked on it and did not fail, so the payments of the players are enough to cover the costs. This completes the proof of Theorem 3.

As you see we have an algorithm that assigns payments to players and constructs an equilibrium where players buy the optimal cost tree. Therefore the price of stability in the case of single source connection games is 1 and we showed how an algorithm can compute the Nash Equilibrium in such case. Although the algorithm that assigns payments is clearly polynomial it uses the optimal cost tree, which is NP-hard to compute, but for which $1,55$-approximation exists [5].

Given an α-approximate tree T_α we can construct an algorithm polynomial in $\frac{1}{\epsilon}$ for $(1+\epsilon)$-approximate equilibrium with cost at most $c(T_\alpha)$. The idea is to use the alternate paths to build a better tree if some edge is not paid for. To make only substantial improvement we use the possibility to deviate by a factor $(1+\epsilon)$.

The result about single source connection games can be extended to directed graphs [2] where a more complex argument is needed to prove the lemma analogous to the path lemma. We can also extend the definition of the game so that each player will have a maximal cost that he can not exceed. In single source connection games with such restriction we also have a Nash Equilibrium purchasing the optimal tree. To reduce this case to the case for directed graphs it is enough to add a new vertex t_i' for each player and connect it with a directed edge costing 0 to the old t_i and with an edge costing the same as the maximal allowed cost to s.

As you can see even though in general connection games it is not always possible to find a Nash Equilibrium and the price of stability might be very high the situation in single source games is much better as an equilibrium always exists and stability comes for free. Still sometimes we need to solve the problem for more general games and in such case we can use a simple mechanism.

5 Fair Connection Games

Fair connection games differ in the problem setting from the basic connection games as now players can only choose if they want to use an edge or if they do not want to use it at all. More formally each player i chooses now a set of edges $E_i \subseteq E$ such that all vertices V_i that the player needs to connect are connected by the edges in E_i.

In this setting the cost of each edge is assigned by a mechanism, a deal between the agents. We will consider only such mechanism that the cost is always divided equally between all agents that use the edge. More formally if a play is given by the edges chosen by all players

$$p = (E_1, \ldots, E_N)$$

and the set of agents using an edge e is

$$A(e) = \{i \ : \ e \in E_i\}$$

then the payment for edge e paid by player i in the play p is given by

$$p_i(e) = \begin{cases} \frac{c(e)}{|A(e)|} & \text{if } e \in E_i, \\ 0 & \text{else.} \end{cases}$$

and the total payment that agent i or all agents have to pay in p is

$$P_i(p) = \sum_{e \in E} p_i(e)\ ,\ \ P_p = \sum_{i=1}^{N} P_i(p).$$

You can see that in this way each player using edge e pays the same price for the edge and this mechanism used to determine the price is derived from Shapley cost-sharing mechanism which has a few nice properties from mechanism-design point of view [4].

In this modified formalism the graph of bought edges is also defined,

$$G_p = (V, E_1 \cup E_2 \cup \ldots \cup E_N),$$

and the definitions of Nash Equilibria and the price of stability and anarchy are analogous to the general case.

Fair connection games put only a slight constraint on the selfishness of the agents as the agents only have to agree on this very simple mechanism. Still this is enough not only to guarantee the existence of equilibria in such games but also to make the price of stability at most logarithmic with respect to the number of agents, which is a significant improvement to the basic model of connection games.

Before we start to prove these properties please notice that the class of fair connection games that we defined is a subclass of congestion games defined by Monderer and Shapley [6]. In a congestion game we have a function $f_e(k)$ that denotes the cost of the edge e when k players decide to use it, so in our case $f_e(k) = \frac{c(e)}{k}$.

One of the most important properties of congestion games is the fact that a *potential function* for these games exists. We can define for a given play p the potential of p by

$$\Phi(p) = \sum_{e \in E} \sum_{k=1}^{|A(e)|} f_e(k).$$

Let us now consider two plays p and p' where the only difference between them is that player i changed his strategy, so if $p = (E_1, \ldots, E_n)$ then $p' = (E_1, \ldots, E'_i, \ldots E_n)$. The astonishing property of the potential function [6] is that for any such two plays the change of the potential is exactly the same as the change of the total payment of the player that changed his strategy,

$$\Phi(p) - \Phi(p') = P_i(p) - P_i(p').$$

You should note that the change may affect costs of many edges and that payments of other players may also change, but the potential function measures the change of the total payment of the player that has chosen a different strategy. Games for which such potential functions exist are called *potential games* and congestion games are in this class.

Let us also introduce the notation for harmonic sum which will be used as an exact bound for the price of stability. Let

$$H(k) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$$

and you should note that $H(k)$ is approximately $\Theta(\log k)$.

Theorem 4 ([1]). *In fair connection games a (pure) Nash Equilibrium always exists and the price of stability is at most $H(N)$.*

Proof. In [6] Monderer and Shapley show that in potential games (pure) Nash Equilibria always exist. To find some equilibrium in a potential game you can just start from any play p_0 and allow any player to change his strategy so that his payment will decrease. Then you have a new play p_1 and you know that since only one player changed his strategy and decreased his payment, then $\Phi(p_0) > \Phi(p_1)$. You can repeat this getting new plays $p_2, p_3, \ldots, p_k$ with $\Phi(p_0) > \Phi(p_1) > \ldots > \Phi(p_k)$ until the potential function can not be decreased any more and in such case p_k is a Nash Equilibrium. You should note that in a game with finite number of strategies and players there is a minimum by which the potential function decreases in each step and so an infimum will always be reached.

To show the bound on the price of stability let us compute the potential function for the case of fair connection games, where for a play $p = (E_1, \ldots, E_N)$ we will denote all used edges used by $E_p = E_1 \cup \ldots \cup E_N$,

$$\Phi(p) = \sum_{e \in E} \sum_{k=1}^{|A(e)|} f_e(k) = \sum_{e \in E} \sum_{k=1}^{|A(e)|} \frac{c(e)}{k} = \sum_{e \in E_p} c(e) H(|A(e)|).$$

Therefore as $1 \le H(|A(e)|) \le H(N)$ we can use the following inequalities for any play p

$$\sum_{e \in E_p} c(e) \le \Phi(p) = \sum_{e \in E_p} c(e) H(|A(e)|) \le H(N) \sum_{e \in E_p} c(e).$$

Let now p^* be the play that represents the globally optimal solution and minimises the total cost, so $P_{p^*} = \sum_{e \in E_{p^*}} c(e)$ is minimal. We can start from this play and reach some Nash Equilibrium p in the way described before so that $\Phi(p) < \Phi(p^*)$ and therefore

$$H(N) P_{p^*} = H(N) \sum_{e \in E_{p^*}} c(e) \ge \Phi(p^*) \ge \Phi(p) \ge \sum_{e \in E_p} c(e) = P_p$$

so indeed the price of stability is at most $H(N)$, which concludes the proof.

To see that in fair connection games the price of stability can be very close to $H(N)$ please take a look at the game presented in Figure 8 assuming $\epsilon \to 0$. Indeed, the only Nash Equilibrium in this fair connection game is when each player i buys the edge from t_i to s paying $\frac{1}{i}$. If any group of players $\{i_1, \ldots, i_l\}$

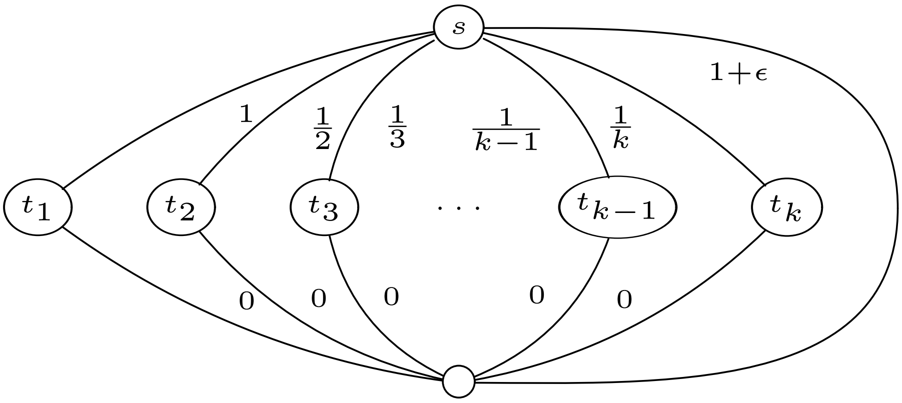

Fig. 8. Price of Stability in Fair Connection Games

decided to buy the optimal solution each of them would have to pay $\frac{1+\epsilon}{l}$. But among these players there is at least one that already pays $\frac{1}{l}$ so his payment would increase.

You should notice that the game from Figure 8 analysed in the general setting would be a single source connection game, so the algorithm from the previous chapter would find the optimal solution. The problem is that the best general equilibrium requires that payments for the edge that costs $1+\epsilon$ are not equal for all players, for example the first player could pay $\frac{3}{4}$ and the second player $\frac{1}{4}+\epsilon$ and the other players nothing. As you can see this is better for everyone but is in some sense unfair as some of the players do not pay anything. So fairness also has some cost, but luckily it is only logarithmic in the number of players and this method guarantees that an equilibrium can always be found.

6 Conclusions

Network design games are applicable in many practical situations and try to model the behaviour of selfish agents. Unluckily if there are no constraints on the selfish behaviour the price to pay for it can be very high. In connection games without any constraints both the price of anarchy and the price of stability can be maximally high. Moreover, in some situations finding an equilibrium might not be possible at all and deciding if an equilibrium exists is in general NP-hard. In the general case the only positive result that we know is the existence of 3-approximate Nash Equilibria, though only $(4,55+\epsilon)$-approximate can be computed in polynomial time.

We can try to keep the selfishness of the agents unlimited but constrain the structure of the game and analyse single source connection games. In such games an equilibrium always exists and the price of stability is 1 so the agents can always find an equilibrium with optimal cost. Still in polynomial time we can only compute $(1,55+\epsilon)$-approximate equilibria in such games.

If the structure of single source games in too constrained for the problem we can use a simple mechanism to assign how much each agent has to pay for a connection in the network. In such case we allow the agents only to choose which connections they want to use and the cost of each connection is equally divided between all agents that use it. In such fair connection games a Nash Equilibrium always exists and the price of stability by N agents is of the order $\log(N)$. Still it might happen that the restriction that the cost of each connection is divided equally between all agents using it makes it impossible to find an equilibrium with a lower cost, for example for a single source connection game, so using the fair mechanism of cost assignment also has a price.

It is possible to study other mechanisms for sharing costs of the connections among multiple agents [4]. There are also interesting possible ways to approximate the equilibria in network design games and perhaps better results can be obtained for approximate equilibria in the general setting. With the wide adoption of internet and other local and global networks the analysis of connection games and other related problems can provide important insights into the problem of creating and managing such networks.

References

1. E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proc. 44th Ann. IEEE Symp. on Foundations of Comput. Sci. (FOCS)*, 2004.
2. E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proc. 35th Ann. ACM. Symp. on Theory of Comput. (STOC)*, 2003.
3. A. Czumaj, P. Krysta, and B. Vöcking. Selfish traffic allocation for server farms. In *Proc. 34th Ann. ACM. Symp. on Theory of Comput. (STOC)*, 2002.
4. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *J. Comput. Syst. Sci*, 2001.
5. A.Zelikowsky G. Robins. Improved steiner tree approximation in graphs. In *Proc.Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA)*, 2000.
6. D. Monderer and L. Shapley. Potential games. *Games and Economic Behaviour 14*, 1996.
7. T. Roughgarden. The price of anarchy is independent of the network topology. In *Proc. 34th Ann. ACM. Symp. on Theory of Comput. (STOC)*, 2002.
8. A. Vetta. Nash equilibria in competitive societies with applications to facility location, traffic routing and auctions. In *Proc. 43th Ann. IEEE Symp. on Foundations of Comput. Sci. (FOCS)*, 2002.