On Dynamics in Basic Network Creation Games

Pascal Lenzner

Department of Computer Science,
Humboldt-Universität zu Berlin, Germany
lenzner@informatik.hu-berlin.de

Abstract. We initiate the study of game dynamics in the Sum Basic Network Creation Game, which was recently introduced by Alon et al.[SPAA'10]. In this game players are associated to vertices in a graph and are allowed to "swap" edges, that is to remove an incident edge and insert a new incident edge. By performing such moves, every player tries to minimize her connection cost, which is the sum of distances to all other vertices. When played on a tree, we prove that this game admits an ordinal potential function, which implies guaranteed convergence to a pure Nash Equilibrium. We show a cubic upper bound on the number of steps needed for any improving response dynamic to converge to a stable tree and propose and analyse a best response dynamic, where the players having the highest cost are allowed to move. For this dynamic we show an almost tight linear upper bound for the convergence speed. Furthermore, we contrast these positive results by showing that, when played on general graphs, this game allows best response cycles. This implies that there cannot exist an ordinal potential function and that fundamentally different techniques are required for analysing this case. For computing a best response we show a similar contrast: On the one hand we give a linear-time algorithm for computing a best response on trees even if players are allowed to swap multiple edges at a time. On the other hand we prove that this task is NP-hard even on simple general graphs, if more than one edge can be swapped at a time. The latter addresses a proposal by Alon et al..

1 Introduction

The importance of the Internet as well as other networks has inspired a huge body of scientific work to provide models and analyses of the networks we interact with every day. These models incorporate game theoretic notions to be able to express and analyse selfish behavior within these networks. Such behavior by players can be the creation or removal of links to influence the network structure to better suit their needs. However, most of this work focused on *static* properties of such networks, like structural properties of solution concepts. Prominent examples are bounds on the Price of Anarchy or on the Price of Stability of (pure) Nash Equilibria in games that model network creation. The problem is, that such results do not explain how selfish and myopic players can actually *find* such desired states.

G. Persiano (Ed.): SAGT 2011, LNCS 6982, pp. 254–265, 2011.

In this paper we focus on the process itself. That is, on the dynamic behavior of players which eventually leads to a state of the game having interesting properties like stability against unilateral deviations and low social cost. We initiate the study of myopic game dynamics in the SUM BASIC NETWORK CREATION GAME, which was introduced very recently by Alon et al.[2]. This elegant model incorporates important aspects of network design as well as network routing but is at the same time simple enough to provide insights into the induced dynamic process. The idea is to let players “swap” edges to resemble the natural process of weighing two decisions (possible edges) against each other. We investigate the convergence process of dynamics which allow players to myopically swap edges until a stable state of the game emerges. Furthermore, we take the mechanism design perspective and propose a specific dynamic, which yields near optimal convergence speed.

1.1 Related Work

The line of research which is closest to our work was initiated by Fabrikant et al.[5], who considered network creation with a fixed edge-cost of α. For some ranges of α they proved first bounds on the Price of Anarchy [8], which is the ratio of the social cost of the worst (pure) Nash Equilibrium and the minimum possible social cost achieved by central design. Subsequent work [9,1,4,10] has shown, that this ratio is constant for almost all values of α. Only for $\alpha \in \Theta(n)$ there remains a gap. However, there is a downside of this model: As already observed in [5], computing a best response is NP-hard, which implies, that players cannot efficiently decide if the game has reached a stable state. This computational hardness prevents myopic dynamics from being applied to finding a pure Nash Equilibrium.

Very recently, Alon et al.[2] proposed a slightly different model, which no longer depends on the parameter α but still captures important aspects of network creation. The authors consider two different cost-measures, namely the sum of distances to all other players and the maximum distance to all other players and give bounds on the price of anarchy. Here, we adopt the former measure. Alon et al. proved that in this case the star is the only equilibrium tree. Interestingly, as observed in [10], it is not true that the class of equilibria in the model without parameter is a super-class of the equilibria in the original model. Nevertheless, we believe that the model of Alon et al. is still interesting, because it models the natural process of locally weighing alternatives against each other. Furthermore, it has another striking feature: Best responses can be computed efficiently. Thus, applying myopic dynamics seems a natural choice for the task of finding stable states in the game. The authors of [2] also propose to analyse the case where players are allowed to swap more than one edge at a time.

The work of Baumann and Stiller [3] is very similar in spirit to our work. They provide deep insights into the dynamics of a related network creation game and show various structural properties, e.g. sufficient and necessary conditions for stability.

Due to space constraints we refer for further work on selfish network creation to Jackson's survey [6] and to the references in Nisan et al.[12, Chapter 19].

1.2 Model and Definitions

The Sum Basic Network Creation Game is defined as follows: Given an undirected, connected graph $G = (V, E)$, where each vertex corresponds to a player. Every player $v \in V$ selfishly aims to minimize her connection cost by performing moves in the game. A player's *connection cost* $c(v)$ is the sum of all shortest-path distances to all other players. If the graph is disconnected, then we define $c(v)$ to be infinite. At any time, a player can "swap" an incident existing edge with an incident non-edge at no cost. More formally, let u be a neighbor of v and w be a non-neighbor of v, then the edge swap (u, w) of player v removes the edge vu and creates the edge vw. Let $\Gamma_G(v)$ denote the closed neighborhood of v in G, which includes v and all neighbors of v. The set of pure strategies for player v in G is $S_G(v) = \big(\Gamma_G(v) \setminus \{v\} \times V \setminus \Gamma_G(v)\big) \cup \{\bot\}$, where $\bot$ denotes, that player v does not swap. Note, that this set depends on the current graph G and that moves of players in the game modify the graph. We allow only pure strategies and call a pure strategy $s \in S_G(v)$, which decreases player v's current connection cost most, a *best response*. Sometimes we say that a vertex x is a best response of a player v, which abbreviates, that v has a best response of the form (y, x), for some $y \neq x$.

We assume that players are lazy, in the sense that if for some player v the best possible edge-swap yields no decrease in connection cost, then player v prefers the strategy $\bot$, that is, not to swap. We say that G is *stable* or in *swap-equilibrium* if $\bot$ is a best response for every player.

Since the model does not include costs for edges, the utility of a player is simply the negative of her connection cost. Let $x \in G$ denote that G contains vertex x. The connection cost of player v in graph G is defined as $c_G(v) = \sum_{x \in G} d_G(v, x)$, where $d_G(v, x)$ is the number of edges on the shortest path from v to x in G. We omit the reference to G, if it is clear from the context. The *social cost* of a graph G is the sum of the connection costs of all players in G.

Furthermore, we use the convention, that for a graph G, we let $|G|$ denote the number of vertices in G and we define $G - x$ to be the graph G after the removal of vertex x.

1.3 Our Contribution

We provide a rigorous treatment of the induced game dynamics of the Sum Basic Network Creation Game on trees. For this case, Theorem 1 shows that the game dynamic has the desirable property that local improvements by players directly yield a global improvement in terms of the social cost. More formally, we show that the game on trees is an *ordinal potential game*[11], that is, there exists a function mapping states of the game to values with the property that pure Nash Equilibria of the game correspond to local minima of the function. A prominent feature of such games is, that a series of local improvements must eventually converge to a pure Nash Equilibrium – a stable state of the game in which no player wants to unilaterally change her strategy. Theorem 3 shows that this convergence is fast by providing a cubic upper bound on the number of steps any improving response dynamic needs to reach such a stable state. Furthermore we introduce and analyse a natural dynamic called *Max Cost Best Response Dynamic*. This dynamic is proven to be close to optimal in terms of convergence speed, since Theorem 4 shows that the number of steps needed by this dynamic almost matches the trivial lower bound. This implies, that the process of finding a pure Nash Equilibrium can be significantly sped up by introducing coordination and enforcing that best responses are played.

In contrast to these positive results on trees, Theorem 7 is a strong negative result for the SUM BASIC NETWORK CREATION GAME on general graphs. We show that in this case best response dynamics can cycle, which implies, that there cannot exist an ordinal potential function. Thus, any treatment of the game dynamics on general graphs requires fundamentally different techniques and is an interesting open problem for ongoing research.

Last, but not least, we use structural insights to obtain a linear-time algorithm for computing a best response on trees even for the case where players are allowed to swap multiple edges at a time. For the game on general graphs, we provide another sharp contrast by showing that computing a best response in the general case is NP-hard, if more than one edge can be swapped at a time. This is particularly interesting, since this addresses the proposal of Alon et al.[2] to analyse this case. Our results imply, that in this case best responses can be efficiently computed only if the game is played on trees or on very simple graphs.

Due to space constraints some proofs are omitted. They can be found in the full version of the paper.

2 Playing on a Tree

In this section we consider the special case where the given graph G is a tree. We show, that the SUM BASIC NETWORK CREATION GAME on trees belongs to the well-studied class of *ordinal potential games*[11]. This guarantees the desirable property that pure Nash Equilibria always exist and that such solutions can be found by myopic play.

Theorem 1. *The* SUM BASIC NETWORK CREATION GAME *on trees is an ordinal potential game.*

Before we prove the Theorem, we analyse the impact of an edge-swap on the connection cost of the swapping player and on the social cost.

Let $T = (V, E)$ be a tree having n vertices. Assume that player v performs the edge-swap vu to vw in the tree T. (Note, that this implies, that $vw \notin E$). Let T' be the tree obtained after this edge-swap. Let Φ and Φ' be the social cost of T and T', respectively. Let T_v and T_u be the tree T rooted at v and u, respectively. Let A be the subtree rooted at v in T_u and let B be the subtree rooted at u in T_v. See Fig. 1 for an illustration. Let $c_K(z) = \sum_{k \in K} d_K(z, k)$ denote the connection cost of player z within tree K.

Lemma 1. *The change in player v's connection cost induced by the edge-swap vu to vw is $\Delta(v) = c_B(u) - c_B(w)$.*

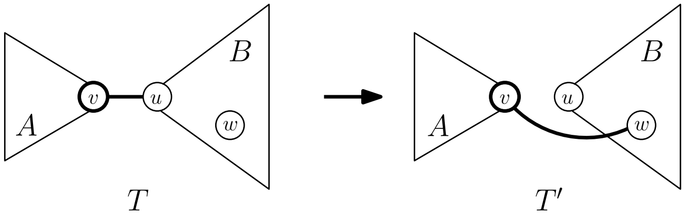

Fig. 1. Player v swaps edge vu to edge vw

The following Lemma implies the desired property, that local improvement of a player yields a global improvement in terms of social cost.

Lemma 2. *The change in social cost induced by the edge-swap vu to vw is*

$$\Delta(\Phi) = 2|A|\Delta(v) \ .$$

Proof (Theorem 1). By Lemma 2, we have that the social cost strictly decreases if and only if the connection cost of the swapping player strictly decreases. This implies, that the social cost Φ is an ordinal potential function for the Sum Basic Network Creation Game on trees. □

Theorem 1 guarantees that a pure Nash Equilibrium of this game can be reached by myopic play, even if the players do not play in an optimal way. We only need one very natural ingredient for convergence: Whenever a player moves, this move must decrease this player's connection cost. We call every dynamic where a player strictly improves by making a move (or passing if no improving move is possible) an *improving response dynamic*(IRD). Such a dynamic stops if no player can strictly improve, which implies that any IRD stops if a stable graph is obtained.

2.1 Improving Response Dynamics on Trees

For trees it was shown by Alon et al.[2] that the star is the only stable tree. Using this observation and Theorem 1, we arrive at the following Corollary.

Corollary 1. *For every tree T, every IRD converges to a star.*

Having guaranteed convergence, the natural question to ask is how many steps are needed to reach the unique pure Nash Equilibrium by myopic play. The following Theorems provide a lower and an upper bound on that number.

Theorem 2. *Let P_n be a path having n vertices. Any IRD on P_n needs at least* $\max\{0, n-3\}$ *steps to converge.*

Lemma 3. *P_n is the tree on n vertices which has maximum social cost.*

Theorem 3. *Any IRD on trees having n vertices converges in $\mathcal{O}(n^3)$ steps.*

Proof. The idea is to start with the tree having the highest potential and to bound the number of steps any IRD needs by analysing the number of steps needed if this potential is decreased by the smallest possible amount per step. By Lemma 3, we have that P_n has the maximum social cost Φ_{P_n}. Observe, that $\Phi_{P_n} = \sum_{i=1}^{n-1} 2i(n-i) = \frac{n^3-n}{3}$. Let X_n be a star having n vertices. We have $\Phi_{X_n} = 2n^2 - 4n + 2$. To transform P_n into X_n any IRD has to decrease the social cost by $\Phi_{P_n} - \Phi_{X_n} = \frac{n^3}{3} - 2n^2 + \frac{11n}{3} - 2$. Since we have an IRD, every moving player decreases her connection cost by at least 1. By Lemma 2, we have that the minimum decrease in social cost by any move is 2. Hence, at most $\frac{n^3}{6} - n^2 + \frac{11n}{6} - 1 \in \mathcal{O}(n^3)$ steps are needed to transform P_n into X_n. □

2.2 Best Response Dynamics on Trees

It is reasonable to assume, that players greedily try to decrease their connection cost most, whenever swapping an edge. In this section we analyse dynamics, where every move of a player is a best response move.

Since a best response is always an improving response, we have that every dynamic where every move is a best response must converge to a star for every tree T. We are left with the question of how fast best response dynamics converge. In the following, we analyse a specific best response dynamic, called *Max Cost Best Response Dynamic*(mcBRD), whose convergence speed almost matches the lower bound provided by Theorem 2. Hence, for best response dynamics we can significantly improve the upper bound of Theorem 3.

Definition 1. *The* Max Cost Best Response Dynamic *on a graph G is a dynamic, where in every step the player having the highest connection cost is allowed to play a best response. If two or more players have maximum connection cost, then one of them is chosen uniformly at random.*

In this section we show the following upper bound on the speed of convergence for the Max Cost Best Response Dynamic. Surprisingly, mcBRD behaves differently depending on whether the number of vertices in the tree is odd or even.

Theorem 4. *Let T be a tree having n vertices. The following holds:*

- *If n is even, then mcBRD(T) converges after at most* $\max\{0, n-3\}$ *steps and every player moves at most once.*
- *If n is odd, then at most* $\max\{0, n + \lfloor n/2 \rfloor - 5\}$ *steps are needed and every player moves at most twice.*

In order to prove Theorem 4, we first show some useful properties of the convergence process induced by the mcBRD-rule.

We begin with characterizing a player's best response on a tree. Here, the notion of a *center-vertex* is crucial.

Definition 2. *A center-vertex of a graph G is a vertex x, which satisfies*

$$x \in \arg\min_{v \in G} c(v) \ .$$

Lemma 4. *Let v be an arbitrary vertex of a tree T and let $F = T - v = \bigcup_{j=1}^{l} T_l$, where the trees T_j are connected components in the forest F. Let $u_1, \ldots, u_l$ be the neighbors of v in T, where u_j is a vertex of T_j for all $1 \leq j \leq l$. Let w_j be a center-vertex of the tree T_j. The best response of v in T is the edge-swap vu_j to vw_j, where $j \in \arg\max_j\{c_{T_j}(u_j) - c_{T_j}(w_j)\}$.*

The next Lemma provides a very useful property of neighbors in a tree.

Lemma 5. *Let u and w be neighbors in a tree T. Let T_u and T_w denote the tree T rooted at vertex u and w, respectively. Let U be the set of vertices in the subtree rooted at u in T_w. Analogously, let W be the set of vertices in the subtree rooted at w in T_u. Then we have $c(u) \leq c(w) \iff |U| \geq |W|$ and $c(u) < c(w) \iff |U| > |W|$.*

We can use Lemma 5, to show an important property of the mcBRD-process.

Lemma 6. *Let T be a tree. Every player who moves in a step of mcBRD(T) must be a leaf.*

The following Lemma provides the key to analysing mcBRD. It shows, that at some point in the dynamic a certain behavior is "triggered", which forces the dynamic to converge quickly.

Lemma 7 (First Trigger Lemma). *Let T be a tree. If the player who moves in step i of mcBRD(T) has a unique best response vertex w, then all players who move in a later step of mcBRD(T) will connect to vertex w.*

Lemma 8. *In any tree T on n vertices, there are at most two center-vertices. If this is the case, then they are neighbors and n must be even.*

Now we are ready, to prove the first part of Theorem 4.

Proof (Theorem 4, Part 1). We show, that if the number of vertices in a tree T is even, then mcBRD needs at most $\max\{0, n-3\}$ to converge and every player moves at most once.

If T has two vertices, then it is already a star and no player will move in mcBRD(T). Thus, let T be a tree having at least $n \geq 4$ vertices, where n is even. By Lemma 6, we have that in every step of mcBRD(T) a leaf l of the current tree is allowed to move. By Lemma 4, we know that player l will connect to a center-vertex of $T' - l$, where T' is the tree before player l moves. Observe, that the tree $T' - l$ has an odd number of vertices. By Lemma 8, we have that any tree having an odd number of vertices must have a unique center vertex. It follows, that the leaf who moves in the first step of mcBRD(T) has a unique best response. Let this best response be the edge-swap towards vertex w. Lemma 7 implies, that all players who move in a later step of mcBRD(T), will connect to vertex w as well. Furthermore, again by Lemma 7, after the first step of mcBRD(T) it holds, that every vertex who is already connected to vertex w will never move again. Hence, every vertex moves at most once.

By Lemma 5, we have that w must be an inner vertex of T. Thus, w has at most $n-3$ non-neighbors, which implies that the dynamic mcBRD(T) will need at most $n-3$ steps to converge to a star having w as its center-vertex. □

The next Theorem shows a lower bound on the speed of convergence for mcBRD on trees having an odd number of vertices. Surprisingly, the behavior of the dynamic on such instances is much more complex. The lower bound for odd n is roughly 50% greater than the upper bound for even n. Furthermore, the following Theorem together with Theorem 2 implies, that the analysis of mcBRD is tight.

Theorem 5. *There is a family of trees having an odd number of vertices greater than* 5*, where mcBRD can take* $n + \lfloor n/2 \rfloor - 5$ *steps to converge. Furthermore, every player moves at most twice.*

Figure 2 shows an example of a tree which belongs to the above mentioned family of trees and it sketches the convergence process induced by mcBRD.

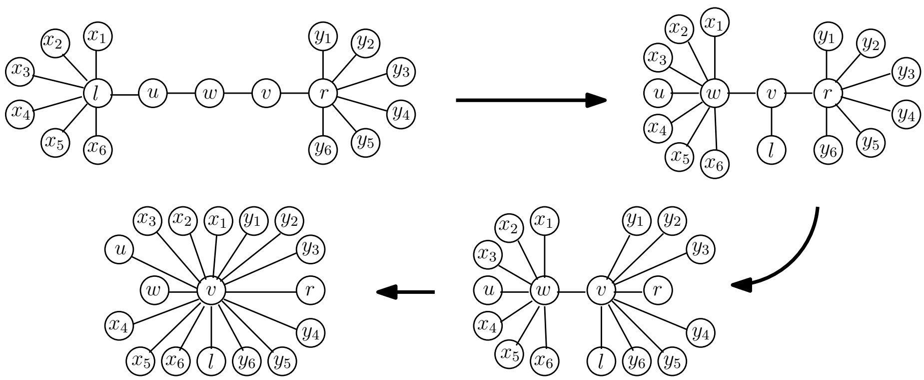

Fig. 2. Example of a tree T having 17 vertices, where mcBRD(T) takes $n+\lfloor n/2 \rfloor - 5 =$ 20 steps to converge. The vertices $x_1, \ldots, x_6, u$ move twice.

Lemma 9 (Second Trigger Lemma). *Let T be an unstable tree having n vertices. If after any step i in mcBRD(T) a vertex w of T^i has degree $\lceil n/2 \rceil$, then this vertex will be the unique best response to connect to for all players moving in a later step of mcBRD(T).*

Lemma 10. *Let T be an unstable tree having an odd number of vertices. Only vertices which are best responses of the player who moves in the first step of mcBRD(T) will be best responses in any step of mcBRD(T).*

Finally, we have set the stage to prove the second part of Theorem 4.

Proof (Theorem 4, Part 2). We show that if a tree T has an odd number of vertices, then mcBRD(T) takes at most $\max\{0, n + \lfloor n/2 \rfloor - 5\}$ steps to converge and every player moves at most twice.

If $n = 5$, then the worst case instance is a path and thus the convergence takes at most 2 steps. Hence, we assume for the following that $n \geq 7$. Observe, that there are two events that force the dynamic to converge: Let E_1 be the event, where for the first time in the convergence process a vertex w becomes the unique best response of a moving player. Let E_2 be the event, where for the first time a vertex w has degree $\lceil n/2 \rceil$.

If event E_1 occurs in step j, then, by Lemma 7, all non-neighbors of the vertex w will connect to w in the subsequent steps of mcBRD(T). Thus, mcBRD(T) will converge in at most $j + |V \setminus \Gamma(w)|$ steps, where $\Gamma(w)$ is the closed neighborhood of w. If event E_2 occurs in step j, then, by Lemma 9, all non-neighbors of w will connect to w in the subsequent steps. Thus, in this case $j + \lfloor n/2 \rfloor - 1$ steps are needed for mcBRD(T) to converge.

Let T be any tree and v be the first player to move and assume that v has two best responses p and q, since otherwise the dynamic will converge in at most $n-3$ steps. By Lemma 10, we have that in any step of mcBRD(T) a player will connect either to p or to q. Let $t_1(T)$ denote the number of steps until event E_1 is the first event to occur in mcBRD(T). Analogously, let $t_2(T)$ denote the number of steps until E_2 is the first occurring event. Let $r_1(T)$ denote the number of steps needed for convergence after event E_1. Hence, the maximum number of steps needed until mcBRD(T) converges is

$$t(T) = \max\{t_1(T) + r_1(T), t_2(T) + \lfloor n/2 \rfloor - 1\} \ .$$

We claim, that $t_1(T) + r_1(T) \leq n + \lfloor n/2 \rfloor - 5$. Observe, that $r_1(T) \leq n - 3$, since the vertex that becomes the center of the star must be an inner vertex of T and, thus, can have at most $n-3$ non-neighbors. Furthermore, if $t_1(T) \leq \lfloor n/2 \rfloor - 2$, then the claim is true. Now let $t_1(T) > \lfloor n/2 \rfloor - 2$. Note, that both p and q must be inner vertices of T. Thus, they have at least degree 2. Since event E_2 did not occur in the first $t_1(T)$ steps of mcBRD(T) we have that not all players who moved within the first $t_1(T)$ steps can be connected to p. Thus, at least $x = t_1(T) - (\lfloor n/2 \rfloor - 2)$ players have connected to q. This yields $t_1(T) + r_1(T) \leq t_1(T) + n - 3 - x \leq n + \lfloor n/2 \rfloor - 5$. On the other hand, since all players move either to p or q and both p and q have degree at least 2, it follows that $t_2(T) \leq 2(\lfloor n/2 \rfloor - 2)$. Hence, $t_2(T) + \lfloor n/2 \rfloor - 1 \leq n + \lfloor n/2 \rfloor - 5$.

Observe, that any player x who is a neighbor of either p or q will not move again until event E_1 or E_2 happens. This holds because every leaf, which is not a neighbor of p or q must have higher connection cost than x and will therefore move before x. Thus, every player moves at most twice. □

2.3 Computing a Best Response on Trees

Observe, that Lemma 4 directly yields an algorithm for computing a best response move of a player v: Compute the connection-costs of all other vertices in $T - v$ within their respective connected component to find a center-vertex for every component. Then choose the center-vertex, which yields the greatest cost decrease for player v. Clearly, the connection-cost of a player can be obtained using a BFS-computation. However the above naive approach of computing a center-vertex yields an algorithm with running time quadratic in n, since $\Omega(n)$ BFS-computations can occur. The following Lemma shows, that a center-vertex can be computed in linear time, which is clearly optimal. The algorithm crucially uses the structural property provided by Lemma 5.

Lemma 11. *Let T be a tree having n vertices. A center-vertex of T and its connection-cost can be computed in $\mathcal{O}(n)$ time.*

Proof. We give a linear time algorithm, which computes a center-vertex of T and its connection-cost. Let L be the set of leaves of T. Clearly, L can be computed in $\mathcal{O}(n)$ steps by inspecting every vertex.

Given T and L, the algorithm proceeds in two stages:

1. The algorithm computes for every vertex v of T two values n_v and c_v. This is done in reverse BFS-order: We define n_v to be the number of vertices in the already processed subtree T_v containing v and c_v to v's connection-cost to all vertices in T_v. For every leaf $l \in L$ we set $n_l := 1$ and $c_l := 0$. Let i be an inner vertex and assume that we have already processed all but one neighbor of i. Let $a_1, \ldots, a_s$ denote these neighbors. We set $n_i := 1 + n_{a_1} + \cdots + n_{a_s}$ and $c_i := n_i - 1 + c_{a_1} + \cdots + c_{a_s}$. By breaking ties arbitrarily, this computation terminates at a root-vertex r, for which all neighbors are already processed. Let $b_1, \ldots, b_q$ denote these neighbors. We set $n_r := n$ and $c_r := n - 1 + c_{b_1} + \cdots + c_{b_q}$.
2. Starting from vertex r, the algorithm performs a local search for the center-vertex with the help of Lemma 5. For all neighbors $b_i \in \{b_1, \ldots, b_q\}$ of r, the algorithm checks if $n_{b_i} > n_r - n_{b_i}$. Since T is a tree, this can hold for at most one neighbor x. In this case, x will be considered as new root-vertex. Let $c_1, \ldots, c_s, r$ be the neighbors of x. By setting $n_x := n$ and $c_x := n - 1 + c_1 + \cdots + c_s + c_r - c_x$ we arrive at the same situation as before and we now check for all neighbors $c_j \neq r$ if $n_{c_j} > n_x - n_{c_j}$ holds and proceed as above. Once no neighbor of the current root-vertex satisfies the above condition, the algorithm terminates and the current root-vertex is the desired center-vertex.

The correctness of the above algorithm follows by Lemma 5. Step 1 clearly takes time $\mathcal{O}(n)$. Step 2 takes linear time as well, since the condition is checked exactly once for every edge towards a neighbor and there are only $n - 1$ edges in T. □

Theorem 6. *If $p \geq 1$ edges can be swapped at a time, then the best response of a player v can be computed in linear time if G is a tree.*

Proofsketch. Let v be a degree d vertex in G and let $v_1, \ldots, v_d$ denote the neighbors of v in G. The $k \leq \min\{p, d\}$ edge swaps that decrease player v's connection cost most can be determined as follows.

Consider the forest $F = T_1 \cup T_2 \cup \cdots \cup T_d$ obtained by deleting v. By Lemma 4 we have that every swap in player v's best response is of the form (v_i, w_i), where w_i is a center-vertex of T_i. Thus, computing player v's best response reduces to finding a center-vertex in each tree T_i and to computing the corresponding cost decreases. By Lemma 11 we have that both tasks can be done in time linear to the number of vertices in each T_i.

3 Playing on General Graphs

3.1 Best Response Dynamics on General Graphs

Definition 3. *A cycle $x_1, \ldots, x_l$ is a* best response cycle*, if $x_1 = x_l$ and each x_i is a pure strategy profile in the* Sum Basic Network Creation Game *and for all $1 \leq k \leq l-1$ there is a player p_k whose best response move transforms the profile x_k into x_{k+1}.*

Theorem 7. *The* Sum Basic Network Creation Game *allows best response cycles.*

Proof. Consider the graph G depicted left in Figure 3 and let x_1 denote the corresponding strategy profile. Player a can decrease its connection cost and one of its best responses is to swap edge ab with edge ac. This leads to the second graph depicted in Figure 3. Call the corresponding strategy profile x_2. Now, player b has the swap bc to ba as its best response, which leads to the third graph depicted in the illustration, with x_3 as its strategy profile. Finally, player c can perform the swap ca to cb as its best response, which leads to profile $x_4 = x_1$. Thus, x_1, x_2, x_3, x_4 is a best response cycle in the Sum Basic Network Creation Game on graph G. □

Voorneveld [13] introduced the class of *best-response potential games*, which is a super-class of ordinal potential games. Furthermore he proves, that if the strategy space is countable, then a strategic game is a best-response potential game if and only if there is no best response cycle. This implies the following Corollary.

Corollary 2. *There cannot exist an ordinal potential function for the* Sum Basic Network Creation Game *on graphs containing cycles.*

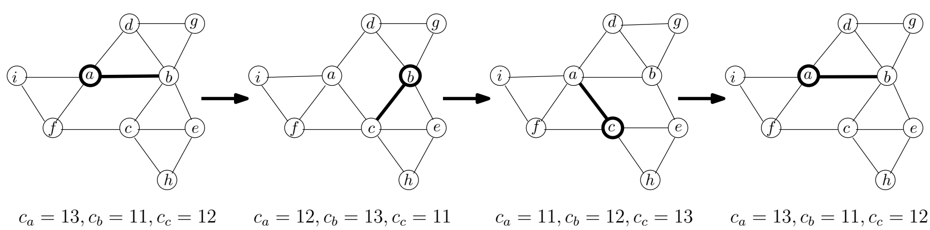

Fig. 3. Example of a graph, where the Sum Basic Network Creation Game allows a best response cycle. The steps of the cycle are shown.

3.2 Computing a Best Response in General Graphs

Given an undirected, connected graph G, then the best response for player v can be computed in $\mathcal{O}(n^2)$ time, since $|S_G(v)| < n^2$ and we can try all pure strategies to find the best one. Quite surprisingly, computing the best response is hard if we allow a player to swap $p > 1$ edges at a time.

Theorem 8. *If players are allowed to swap $p > 1$ edges at a time, then computing the best response is NP-hard even if G is planar and has maximum degree 3.*

Proofsketch. We reduce from the p-Median-Problem [7].

References

1. Albers, S., Eilts, S., Even-Dar, E., Mansour, Y., Roditty, L.: On nash equilibria for a network creation game. In: Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, SODA 2006, pp. 89–98. ACM, New York (2006)
2. Alon, N., Demaine, E.D., Hajiaghayi, M., Leighton, T.: Basic network creation games. In: SPAA 2010: Proceedings of the 22nd ACM Symposium on Parallelism in Algorithms and Architectures, pp. 106–113. ACM, New York (2010)
3. Baumann, N., Stiller, S.: The price of anarchy of a network creation game with exponential payoff. In: Monien, B., Schroeder, U.-P. (eds.) SAGT 2008. LNCS, vol. 4997, pp. 218–229. Springer, Heidelberg (2008)
4. Demaine, E.D., Hajiaghayi, M., Mahini, H., Zadimoghaddam, M.: The price of anarchy in network creation games. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, pp. 292–298. ACM, New York (2007)
5. Fabrikant, A., Luthra, A., Maneva, E., Papadimitriou, C.H., Shenker, S.: On a network creation game. In: Proceedings of the Twenty-Second Annual Symposium on Principles of Distributed Computing, PODC 2003, pp. 347–351. ACM, New York (2003)
6. Jackson, M.O.: A survey of models of network formation: Stability and efficiency. Group Formation in Economics: Networks, Clubs and Coalitions (2003)
7. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. ii: The p-medians. SIAM Journal on Applied Mathematics 37(3), 539–560 (1979)
8. Koutsoupias, E., Papadimitriou, C.: Worst-case equilibria. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 404–413. Springer, Heidelberg (1999)
9. Lin, H.: On the price of anarchy of a network creation game. Class final project (2003)
10. Mihalák, M., Schlegel, J.C.: The price of anarchy in network creation games is (mostly) constant. In: Kontogiannis, S., Koutsoupias, E., Spirakis, P.G. (eds.) Algorithmic Game Theory. LNCS, vol. 6386, pp. 276–287. Springer, Heidelberg (2010)
11. Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior 14(1), 124–143 (1996)
12. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press, New York (2007)
13. Voorneveld, M.: Best-response potential games. Economics Letters 66(3), 289–295 (2000)