

# The Impact of Communication Patterns on Distributed Self-Adjusting Binary Search Trees

Thim Strothmann

Computer Science Department, University of Paderborn (Germany),  
thim@mail.upb.de

**Abstract.** This paper introduces the problem of communication pattern adaption for a distributed self-adjusting binary search tree. We propose a simple local algorithm that is closely related to the nearly thirty-year-old idea of splay trees and evaluate its adaption performance in the distributed scenario if different communication patterns are provided. To do so, the process of self-adjustment is modeled similarly to a basic network creation game in which the nodes want to communicate with only a certain subset of all nodes. We show that, in general, the game (i.e., the process of local adjustments) does not converge, and convergence is related to certain structures of the communication interests, which we call conflicts. We classify conflicts and show that for two communication scenarios in which convergence is guaranteed, the self-adjusting tree performs well. Furthermore, we investigate the different classes of conflicts separately and show that, for a certain class of conflicts, the performance of the tree network is asymptotically as good as the performance for converging instances. However, for the other conflict classes, a distributed self-adjusting binary search tree adapts poorly.

**Key words:** Binary Search Tree, Self Optimization, Basic Network Creation Game, Sink Equilibrium, Distributed Data Structure

## 1 Introduction

Over 30 years ago, Sleator and Tarjan [15] introduced an interesting paradigm to design efficient data structures. Instead of optimizing general metrics, like tree depth, they proposed a *self-adjusting* data structure. To be more precise, the authors introduced *splay trees*, self-adjusting binary search trees in which frequently accessed elements are closer to the root. This therefore improves the average access times weighted by the popularity of the elements. Avin et al. [4] recently proposed *SplayNet*, a *distributed generalization* of splay trees, which is heavily inspired by [15]. In contrast to classical splay trees where requests (i.e., lookups) always originate from the root of the tree, communication in SplayNets happens between arbitrary node pairs in the network. As such, SplayNets can be interpreted as a distributed data structure, e.g., a structured peer-to-peer (p2p)

---

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center “On-The-Fly Computing” (SFB 901)

system or distributed hash table (DHT). Following the ideas of Avin et al., we further investigate the dynamics of a distributed locally self-adjusting tree.

An intuitive requirement to a distributed data structure is that nodes that communicate more frequently with each other become topologically closer to each other. An important factor that influences the performance of a distributed data structure is the peculiarity of the underlying communication interest pattern. Likewise to the original concept of splay trees, each node in the distributed splay tree should only have access to local information to decide whether it needs to change its position in the tree. In our specific scenario, the only kinds of information that each node has access to are its parent, its children and information about the distances to nodes it wants to communicate with. With only little knowledge about the structure of the tree and only limited possibilities to change the structure (called *rotations*), a distributed self-adjusting tree can be seen as a local algorithm whose performance is affected by the communication interests. We want to focus on this specific aspect and try to answer the question of how the performance of a distributed self-adjusting tree is influenced by different communication patterns. However, instead of using *empirical entropies* as a building block for the analysis (as done in [4]), the analytical method we use is heavily inspired by the concept of *Basic Network Creation Games (BNCG)* [2]. By doing so we can extend the analysis of [4] in convergent scenarios to a wider variety of instances. Furthermore, we contrast the previous positive results of [4] by giving concrete examples in which a distributed self-adjusting tree performs poorly, compared to an optimal static network.

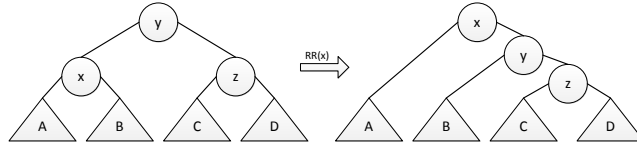
We focus on a binary search tree network structure, since trees are one of the most elemental networks. They allow a simple and local routing strategy and are a fundamental constituent of more complex networks. Additionally, many network protocols rely on spanning trees or cycle-free backbones. Taking the same line as [4], we do not see our work as an introduction for a new network structure, but as a step towards a better understanding of the inherent dynamics of self-adjusting networks and their limitations.

### 1.1 Model & Notions

We model the dynamic process of a distributed self-adjusting tree whose structure is changed as a game in which the nodes of a binary search tree are the players. An instance of the *Self-Adjusting Binary Search Tree Game (SABST-game)*  $\Gamma = (G_C, G_I)$  is given by an initial *connection graph*  $G_C = (V, E_C)$  with  $V = \{1, \dots, n\}$  being the set of players, which is required to be a binary search tree (BST), and a (*communication*) *interest graph*  $G_I = (V, E_I)$ .  $G_C$  is undirected, whereas  $G_I$  is directed. The connection graph represents the distributed self-adjusting tree network and can be altered during the game. We use  $IS(v) := \{u \in V : (v, u) \in E_I\}$  to refer to the neighborhood of player  $v$  in  $G_I$  and denote it as the *interest set* of player  $v$ . Since the connection graph is a binary search tree, we can compare two nodes by comparing their identifiers. The depth of a node  $v$  is the length of a path from the root to  $v$ . If  $v$  has a smaller depth than some node  $u$ , we say that  $v$  is above  $u$ , otherwise  $v$  is below

$u$ . We say that two edges  $(u, v), (x, y)$  from  $G_I$  *intersect* if  $x$  is in the interval  $[u, v]$  for  $u < v$  or  $[v, u]$  for  $u > v$  and  $y$  is not, or vice versa.

Given a connection graph, we formalize the *private cost* of a player  $v$  as the sum over all distances to the nodes in its interest set:  $c(v) := \sum_{u \in IS(v)} d(v, u)$ . Here  $d(v, u)$  denotes the shortest path distance between  $u$  and  $v$  in  $G_C$ . Note that by using the sum, each player tries to minimize the average distance. To improve its private cost, a player may perform *rotations* in the connection graph. These rotations are closely related to the *splay* operation of *splay trees* [15]; a single right rotation of a node (abbreviated with  $RR(x)$ ) is visualized in Figure 1 (node  $x$  rotates over the node  $y$ ). For a *response*, a player  $u$  is not only allowed to perform a single rotation on itself, but also multiple rotations on itself. Additionally,  $u$  can tell nodes from  $IS(u)$  to perform rotations. This is due to the fact that by performing rotations on only itself, a node can only move upwards in the tree. Thus,  $u$  can only move closer to a node  $v \in IS(u)$  that is in its subtrees in  $G_C$ , if it can tell  $v$  to perform rotations. Consequently, players have the opportunity to decrease their private cost as much as possible, instead of being restricted by the current connection graph. If a player  $u$  decreases its private cost by a series of rotations, we refer to this as a *better response*. If the decrease is maximal compared to all other possible better responses, we refer to this as a *best response*. To provide an easy way of computing best responses, we will stick close to the idea of the *double splay* algorithm of [4]. A node  $u$  first rotates itself upward such that it is the lowest common ancestor of all  $v \in IS(u)$  (i.e., it becomes the root of this particular subtree), then all nodes  $v$  are rotated as close as possible to  $u$ . Note that according to [4] a general optimal solution as well as best responses can be computed in polynomial time. We denote the connection graph to be in a *rotation equilibrium*, if no node can perform a better response. We say that a game *converges* if every sequence of best responses converges, irrespective of the initial connection graph. Otherwise, we say that the game is *non-convergent*.



**Fig. 1.** A single right rotation of node  $x$ . The triangles represent (possibly empty) subtrees that are not changed by the rotation.

The dynamic process of changing the connection graph (i.e., the game) proceeds in rounds. A round is finished when all players with non-empty interest sets have played a better response at least once. However, we do not enforce an order in a single round, but consider an arbitrary order. The overall quality of a connection graph  $G_C$  is measured by the *social cost*  $c(G_C) = \sum_{v \in V} c(v)$ . Our

goal is to analyze the social cost of worst-case rotation equilibria and compare them with a general optimal solution. We use the ratio of the two measures, the *Price of Anarchy (PoA)*, to do so.

## 1.2 Related Work

Self-adjusting networks have many possible application scenarios, varying from self-optimizing peer-to-peer topologies (e.g., [11]) over green computing [10] (because of reduced energy consumption) to adaptive virtual machine migrations [3, 13]. Self-adjusting routing schemes were examined to deal with congestion, e.g., in scale-free networks [16].

Our work combines ideas from two interesting and very different research areas: self-adjusting binary search trees and basic network creation games. Self-adjusting binary search trees have a long history [1, 5, 15]. The focus of this paper is on splay trees [15]. Introduced in 1985, they have an amortized time bound of  $\mathcal{O}(\log n)$  for the standard tree operations of searching, insertion and deletion. Additionally, splay trees are as efficient as static, optimal search trees for a sufficiently long sequence of node accesses. Splay trees achieve this by applying a restructuring operation for each access in the tree. This splay operation moves the recently accessed node to the root of the tree by performing rotations on the nodes. Since their establishment, splay trees have been extensively analyzed and many variants have been proposed (see [8, 14, 17] which all use the dynamics of splay trees). Closest to our work is the aforementioned paper of [4], in which a fully decentralized generalization of splay trees called *SplayNet* is presented. SplayNets adapt to a communication pattern  $\sigma$ . The upper bound for the amortized communication cost is based on the empirical entropies of  $\sigma$ . Furthermore, SplayNets have a provable online optimality under special requests scenarios.

*Basic Network Creation Games (BNCG)* were introduced by Alon et al. in 2010 [2]. They are a variant of the original *Network Creation Game (NCG)* by Fabrikant et al. [7]. In the BNCG model, an initial connection graph is given and players are allowed to change the graph by performing what are called improving *edge swaps*. For an edge swap, a node is able to exchange a single incident edge with a new edge to an arbitrary other node. In contrast to the original NCG, best responses are polynomially computable. The cost for a single node is either induced by the sum of the distances to all other nodes (**SUM**-version) or by the maximal distance (**MAX**-version). The authors showed that for the **SUM**-version of the game all trees in an equilibrium have a diameter of 2, and that the diameter of all swap equilibria is  $2^{\mathcal{O}(\sqrt{\log n})}$ . For the **MAX**-version they showed that all trees in an equilibrium have a diameter of at most 3, and that the diameter of general swap equilibria is  $\Omega(\sqrt{n})$ . Lenzner [12] proved that if the game is played on a tree, it admits an ordinal potential function, which implies guaranteed convergence to a pure nash equilibrium. However, when played on general graphs, this game allows best response cycles. For computing a best response, they show a similar contrast: a linear-time algorithm for computing a best response on trees is provided, which works even if players are allowed to swap multiple edges at a time. On the other hand, they proved that this task

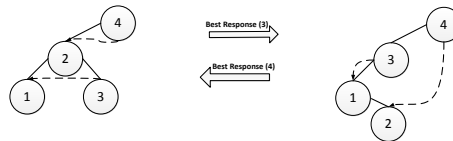
is NP-hard even on simple general graphs, in case more than one edge can be swapped. [6] extended the BNCG model by introducing what are called interests to the game. Thus, the players are now no longer interested in communicating with all other nodes, but only with a specific subset. For the MAX-version they give a tight upper bound of  $\Theta(\sqrt{n})$  for the Price of Anarchy, if the connection graph is a tree, and  $\Theta(n)$  for general connection graphs.

### 1.3 Our Contribution

To the best of our knowledge, this is the first work that evaluates dynamics of self-adjusting topologies by using (basic) network creation games. We introduce a new BNCG that is closely related to the model of [2] but incorporates the dynamics inherent to self-adjusting binary search trees. We show that the game does not converge in general, and the distributed self-adjusting binary search tree will never stop changing its structure. However, for certain interest graphs which guarantee convergence, we prove a tight upper bound on the Price of Anarchy of  $\Theta(1)$ . For non-convergent game instances, we use an altered variant of the concept sink equilibria (introduced in [9]). We define the corresponding measure *worst-case Price of Sinking* to evaluate the worst-case performance of the distributed self-adjusting tree, in contrast to an optimal solution. We prove that there exists an interest graph class such that the worst-case Price of Sinking is constant. However, we also show that, for other interest graph classes, the worst-case Price of Sinking is  $\Omega(\frac{n}{\log n})$ .

## 2 Analysis

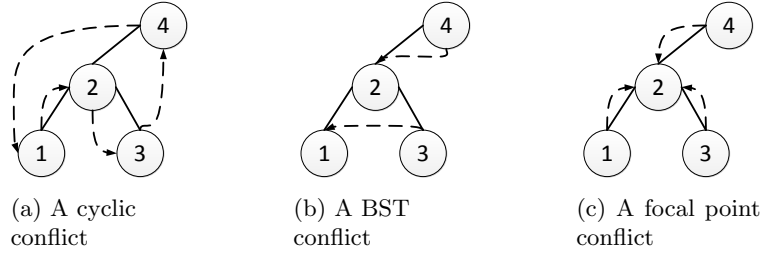
In general, the *SABST*-game does not converge and the dynamic process never settles on stable binary search tree. In fact, it is possible to construct a simple *SABST*-game with four nodes that can never converge (see Figure 2). Consequently, the Price of Anarchy cannot be computed for general instances of the game. In Section 2.1 we identify two classes of interest graphs that do converge and have a constant Price of Anarchy.



**Fig. 2.** An example *SABST*-game instance that does not converge. Interest graph edges are dashed, connection graph edges are continuous.

However, we can relate non-convergent behavior to properties of  $G_I$ , called *conflicts*. Once an interest graph contains a conflict, it is easy to show that the

game can never converge to an equilibrium. We can observe three classes of conflicts: *cyclic conflicts*, *BST conflicts* and *focal point conflicts* (see Figure 3 for examples). Cyclic conflicts are cycles in  $G_I$ . A BST conflict occurs, if nodes have more than two outgoing edges in  $G_I$  (with one small exception, see Section 2.1) or if either two edges of  $G_I$  intersect in case the nodes are ordered according to their identifier. Focal point conflicts are nodes in  $G_I$  with an indegree greater than one. In Section 2.2 we analyze the conflict classes individually.



**Fig. 3.** Small examples for the three conflict classes.

## 2.1 Convergence & Rotation Equilibria

Two classes of interest graphs imply convergence: interest graphs that are binary search trees, and interest graphs that are star graphs (a central node  $v$  has interest in all other nodes).

**Theorem 1.** *Let  $\Gamma = (G_C, G_I)$  be a SABST-game with  $G_I$  either forming a binary search tree or a star graph. Then, any sequence of best responses converges independent of the initial connection graph. The Price of Anarchy is at most 2.*

Theorem 1 implies that, for the two mentioned communication interest patterns, a distributed self-adjusting binary search tree converges to a steady BST and has almost optimal cost for communication: i.e., it has an approximation factor of at most 2 compared to the optimal BST. Theorem 1 follows from the following two lemmas.

**Lemma 1.** *Let  $\Gamma = (G_C, G_I)$  be a SABST-game with  $G_I$  forming a binary search tree.  $\Gamma$  converges to a social optimum.*

*Proof.* We call a node/player *happy* if it cannot perform a rotation to improve its private cost. Let  $\mathcal{H}$  denote the set of all nodes  $v \in V$ , with the property that the complete subtree of  $G_I$  rooted at  $v$  is happy. To prove convergence we show that the size of  $\mathcal{H}$  is monotonically increasing.

We first show that once a node has entered  $\mathcal{H}$ , it will never leave  $\mathcal{H}$ . Let  $v$  be a node from  $\mathcal{H}$  whose parent in  $G_I$  is not happy. Consequently,  $v$  and all

nodes in the subtree rooted at  $v$  in  $G_I$  are happy and they cannot decrease their private cost and form a connected component in  $G_C$ . Let  $CC_v$  be this connected component and  $v'$  be a node that is unhappy and performs a rotation. If  $v'$  and  $IS(v')$  are both above or below  $CC_v$  in  $G_C$ , then the rotations performed by  $v'$  do not affect  $v$  and its subtrees. If  $v'$  is below  $CC_v$  and  $IS(v')$  is above  $CC_v$  (or vice-versa),  $v'$  has to rotate over  $CC_v$ . To do so, it performs only right or only left rotations, because  $v'$  is either smaller or greater than all nodes in  $CC_v$ . But from the definition of a rotation (see Figure 1), we can deduce that performing only left or only right rotations does not affect the structure of the subgraph that  $v'$  rotates over. Thus, all nodes in  $CC_v$  remain happy. The last case is if  $v'$  is interested in  $v$ , above  $v$  in  $G_C$  and  $v'$  rotates  $v$  upwards. This implies that there exists at least an unhappy node  $v^-$  that is on the path from  $v'$  to  $v$  in  $G_C$ . Consequently,  $v^-$  is either above  $v'$  in  $G_I$ , a sibling of  $v'$  in  $G_I$ , or in the other subtree of  $v'$  than  $v$  in  $G_I$ . But in none of these cases can  $v^-$  be in between  $v'$  and  $v$  in  $G_C$ , since  $G_C$  is a binary search tree. Thus,  $v$  does not leave  $\mathcal{H}$  and the size of  $\mathcal{H}$  does not decrease.  $\mathcal{H}$  is monotonically increasing, because in each round the parents of the nodes already in  $\mathcal{H}$  will enter  $\mathcal{H}$  and initially all leaves from  $G_I$  are in  $\mathcal{H}$ , since their interest set is empty.

Now assume that  $\Gamma$  does not converge to a social optimum. Let  $T'$  be the connection graph in a rotation equilibrium and  $T' \neq G_I$ : i.e.,  $\exists u \in V$  with  $pos_{G_I}(u) \neq pos_{T'}(u)$ , where  $pos_{G_I}(u)$  and  $pos_{T'}(u)$  denote the position of  $u$  in  $G_I$  and  $T'$  depending on  $v$ 's depth. Let  $v'$  be the node with minimal depth in  $T'$  which has a child  $u$  with  $pos_{G_I}(u) \neq pos_{T'}(u)$ . Consequently,  $v'$  is unhappy and can perform a rotation to decrease its private cost, which contradicts the fact that  $T'$  is in a rotation equilibrium. Consequently, the connection graph in the rotation equilibrium is the same as  $G_I$  and the PoA is 1.  $\square$

Note that this result only holds for binary trees, since for general trees the size of  $\mathcal{H}$  is not monotonically increasing: i.e., if a unhappy node performs a better response, happy nodes can become unhappy again.

**Lemma 2.** *Let  $\Gamma = (G_C, G_I)$  be a SABST-game with  $G_I$  forming a star graph: i.e., all edges point from one single center node to all other nodes.  $\Gamma$  converges and has a PoA of at most 2.*

Note, that the star graph, is an exception to the conflict class of BST conflicts. However, this is the only exception, because by observation one can show that the game does not converge anymore if there is an edge  $(u, v) \in E_I$  with  $u$  being not the center node. The proof of Lemma 2 can be found in the full version. Lemma 1 and 2 prove Theorem 1. The rest of this section justifies the approach of focusing on a single connected component of edges from  $G_I$ . We say a node  $w$  affects the private cost of a node  $v$  in a rotation equilibrium if  $w$  lies on the the shortest path from  $v$  to a node  $u$  with  $u \in IS(v)$ .

**Lemma 3.** *Consider a connected component  $E'_I$  of edges without conflicts from the interest graph  $G_I = (V, E_I)$ , the corresponding node set  $V' = \{v \in V | \exists u \in V \wedge (u, v) \in E'_I \vee (v, u) \in E'_I\}$  and a single interest edge  $e_I = (u', v')$ . If  $e_I$  is*

neither a part of  $E'_I$  nor induces a conflict with  $E'_I$ ,  $u$  and  $v$  do not affect the private cost of the nodes from  $V'$  in a rotation equilibrium and vice-versa.

Again the proof can be found in the full version. We can easily extend Lemma 3 such that the single edge  $e_I$  can be replaced by a set of edges. Therefore, we can analyze multiple connected components from  $G_I$  separately. Furthermore, the proof can be extended in such a way that  $G'_I$  contains conflicts, instead of being connected. The game will not converge anymore, but has the property that a single edge (or even a set of edges) will no longer affect the private cost of  $G'_I$  eventually.

## 2.2 Non-Convergence & Sink Equilibria

As mentioned before, the three identified classes of conflicts imply non-convergent behavior. Therefore, rotation equilibria do not necessarily exist and the Price of Anarchy is no longer well defined. To overcome this obstacle, we use the solution concept *sink equilibrium*, which was introduced by Goemans et al. [9]. A sink equilibrium is not defined for a single connection graph  $G_C$  of a game instance, but for the *configuration graph* of an instance. The configuration graph  $G_S = (V^*, E^*)$  of an instance  $\Gamma = ((V, E_C), (V, E_I))$  has a vertex which is equal to the set of *valid connection graphs* (i.e., all possible BSTs) for the given node set  $V$ . The edge set  $E^*$  corresponds to better responses of the players: i.e., an edge  $(u, v)$  is in  $E^*$  if a response of a single player in the connection graph represented by  $u$  leads the connection graph in  $v$ . A *sink equilibrium* is a strongly connected component without outgoing edges in the configuration graph. Analogical to the Price of Anarchy we define a new measurement of how well selfish players perform compared to a social optimum. [9] uses the expected social cost of a sink equilibrium to compute what is called *Price of Sinking (PoS)*. However, we want to focus on the worst-case behavior of nodes. Therefore, instead of looking at the expected social cost of sink equilibria, we choose a state with worst-case social cost of all sink equilibria and compare it to the social cost of a social optimum. We call this measure the *worst-case Price of Sinking (wcPoS)*. If the wcPoS is low, then every state in a sink equilibrium has social cost close to the optimal social cost and therefore the self-adjusting binary search tree still performs well, even though it does not converge to a fixed tree.

Before analyzing the different classes of conflicts separately and giving results on their worst-case Price of Sinking, we first prove a general result about sink equilibria in the SABST-game. Due to the definition of the wcPoS, we are faced with the problem of finding a state in a sink equilibrium with maximal social cost. Lemma 4 simplifies this task. A *response order*  $\tau$  is a permutation of the players  $V$ . We say a response order is *applied* to connection graph  $G_C$  (respectively, a state from the configuration graph), when the players of the game play their responses according to  $\tau$  starting from  $G_C$ .

**Lemma 4.** *Given an instance of the SABST-game  $\Gamma$ , a response order  $\tau$  and a state  $s$  from the configuration graph  $G_S = (V^*, E^*)$  of  $\Gamma$ . If  $\forall s' \in V^*$  it holds that  $\tau$  applied on  $s'$  results in  $s$ , then  $s$  lies in a unique sink equilibrium of  $G_S$ .*



*Proof.* Assume that there is another sink equilibrium  $\mathcal{SE}'$  and let  $v'$  be a state from  $\mathcal{SE}'$ . We know that  $v^*$  can be reached from  $v'$  by  $\tau$ . But by the definition of a sink equilibrium this implies that  $v^*$  and  $v'$  are in the same sink equilibrium, which is a contradiction to the original assumption.

Therefore, we can deduce a worst-case sink equilibrium state  $s$ , if we can give a response order  $\tau$  that constructs the connection graph represented in  $s$ .

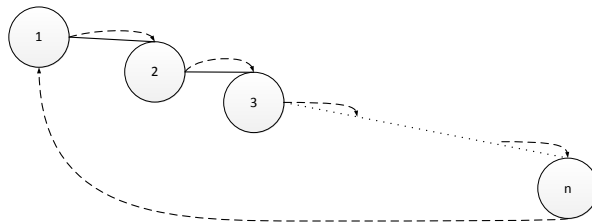
**Cyclic Conflicts** We first take a closer look on interest graphs with only cyclic conflicts. We only need to consider interest graphs that are simple cycles (i.e., cycles that do not intersect and are not contained in each other) because these cases imply a BST conflict or a focal point conflict. W.l.o.g. we focus on the cyclic conflict over all nodes  $G_I^{c.c.} = (V, E_I)$  with  $V = \{1, \dots, n\}$  and  $E_I = \{(n, 1) \cup (i, i + 1) : i = 1, \dots, n - 1\}$ .

**Theorem 2.** *Let  $\Gamma^{c.c.} = (G_C, G_I^{c.c.})$  be a SABST-game, the wcPoS is  $\mathcal{O}(1)$ .*

Consequently, as long as the communication interests contain only cyclic conflicts, the performance of the self-adjusting tree is asymptotically as good as the performance without conflicts. To prove Theorem 2, we need to show the following to lemmas.

**Lemma 5.** *For the SABST-game  $\Gamma^{c.c.}$ , every state in the unique sink equilibrium has social cost of  $2(n - 1)$ .*

*Proof.* Let  $\tau' = (n, \dots, 1)$  be a response order. If  $\tau'$  is applied on  $G_C$  the resulting connection graph is the one visualized in Figure 4, which is in a unique sink equilibrium. The social cost is  $2(n - 1)$ . Now independent of a response order, there is only one unhappy node in the connection graph that can decrease its private cost. This leads to a connection graph with social cost  $2(n - 1)$  and a single unhappy node again. Consequently, independent of a response order in each round there is a single unhappy node and social cost of  $2(n - 1)$ ,  $\square$



**Fig. 4.** The connection graph for  $\Gamma^{c.c.}$  after response order  $\tau'$  is applied.

**Lemma 6.** *Every social optimum for  $\Gamma^{c.c.}$  has social cost of  $\Omega(2(n - 1))$ .*

*Proof.* We call a connection graph edge  $e_C$  *traversed* by an interest graph edge  $e_I = (u, v)$ , if  $e_C$  is contained in the shortest path from  $u$  to  $v$  in the connection graph. We show that every connection graph edge of a social optimum is traversed by at least two interest graph edges. Let  $e'_C$  be an arbitrary connection graph edge from a socially optimal connection graph. If  $e'_C$  is removed, the connection graph is split in two connected components  $A$  and  $B$ . Since  $G_I^{c.c.}$  is a simple cycle over all nodes, there exist interest graph edges  $e'_I = (a', b')$  and  $e''_I = (a'', b'')$  with  $a', a'' \in A$ ,  $a' \neq a''$  and  $b', b'' \in B$ ,  $b' \neq b''$ . Consequently,  $e'_C$  is traversed twice.  $\square$

Lemma 5 and Lemma 6 together conclude the proof.

**BST Conflicts and Focal Point Conflicts** For BST conflicts and focal point conflicts we do not prove an upper bound for the wcPoS, but show that both conflict classes contain interest graphs such that the wcPoS is lower bounded by  $\Omega(\frac{n}{\log(n)})$ . Therefore, best responses of selfish players can lead to a state in a sink equilibrium, which has high social cost compared to a social optimum. This shows that the intuition of the *double splay* algorithm [4] performs poorly in these scenarios. We start with interest graphs with only BST conflicts. More specifically we focus on interest graphs with only *direct conflicts* in which two edges of  $G_I$  intersect if the nodes are ordered according to their identifier. Interest graphs with only direct conflicts have a node degree smaller than 2, since all other conflict types need a node degree of at least 2. We focus on interest graphs that maximize the number of direct conflicts. These are of the form  $G_I^{d.c.} = (V, E_I)$  with  $V = \{1, \dots, n\}$ ,  $n$  even and  $E_I = \{(i, i + \frac{n}{2}) : i = 1, \dots, \frac{n}{2}\}$ , because every interest edge intersects with every other interest edge.

**Theorem 3.** *Let  $\Gamma^{d.c.} = (G_C, G_I^{d.c.})$  be a SABST-game, the wcPoS is  $\Omega(\frac{n}{\log(n)})$ .*

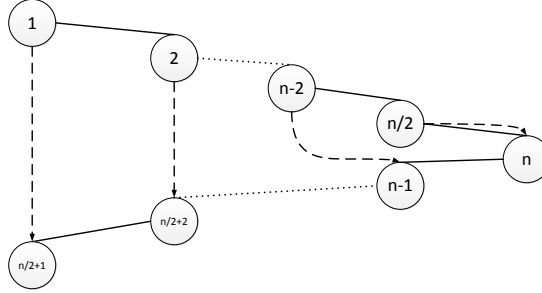
To prove Theorem 3, we first prove that the configuration graph of  $\Gamma$  contains a unique sink equilibrium with a state that has social cost of  $\Theta(n^2)$ .

**Lemma 7.** *The configuration graph of  $\Gamma^{d.c.}$  contains a state in the unique sink equilibrium with social cost of  $\Theta(n^2)$ .*

*Proof.* We pick the response order  $\tau' = (1, \dots, n)$ . If we now apply  $\tau'$  to any initial connection graph, we end up with the connection graph presented in Figure 5. The exact proof of this fact is skipped, but relies mainly on the idea that each player performs rotations such that the nodes from its interest set are in one of its subtrees. The social cost is  $\sum_{i=0}^{\frac{n}{2}-1} (2i+1) = \frac{n^2}{4} = \Theta(n^2)$ . Since this connection graph can be reached from any initial connection graph by  $\tau'$ , we know that it is a state in the unique sink equilibrium of  $\Gamma^{d.c.}$ .  $\square$

Contrasting the last lemma, we now give a general upper bound for the social cost of a social optimum for  $\Gamma$ .

**Lemma 8.** *A social optimum for  $\Gamma^{d.c.}$  has social cost of at most  $\mathcal{O}(n \log n)$ .*



**Fig. 5.** The connection graph for a  $\Gamma^{d.c.}$  if response order  $\tau' = (1, \dots, n)$  is applied.

*Proof.* We arrange the connection graph nodes such that they form a balanced binary search tree. Since every node is only interested in at most a single other node we know that the private cost for a single node can be upper bound by  $2 \log(n)$ . Therefore, the social cost are at most  $\mathcal{O}(n \log n)$ .  $\square$

For interest graphs with only focal point conflicts we can state a similar result. We use the interest graph  $G_I^{f.c.} = (V, E_I)$  with  $V = \{1, \dots, n\}$  and  $E_I = \{(1, n), (2, n), \dots, (n-1, n)\}$ , which has the maximal possible focal point conflict.

**Theorem 4.** Let  $\Gamma^{f.c.} = (G_C, G_I^{f.c.})$  be a SABST-game the wcPoS is  $\Omega(\frac{n}{\log(n)})$ .

The proof technique for Theorem 4 is analogous to Theorem 3. Moreover, Theorems 3 and 4 imply that a SABST-game  $\Gamma = (G_C, G_I)$  in which  $G_I$  contains a subgraph  $G'_I$  of size  $k$  that is either  $G_I^{f.c.}$  or  $G_I^{d.c.}$ , the wcPoS is  $\Omega(\frac{k}{\log(k)})$ . Therefore, we can conclude that the performance of a distributed self-adjusting binary search tree gets worse with increasing size of the communication patterns given by  $G_I^{f.c.}$  or  $G_I^{d.c.}$ . Notice that  $\mathcal{O}(n^2)$  is an upper bound for the social cost of a SABST-game with an interest graph with  $n$  many edges. Therefore, the upper bound for the wcPoS is  $\mathcal{O}(n)$ .

### 3 Conclusion & Open Problems

We analyzed the performance of a distributed self-adjusting binary search tree for different communication patterns. We have shown that, if the communication interests contain no conflicts or only cyclic conflicts, the performance of a self-adjusting tree is almost optimal (PoA of  $\Theta(1)$  and wcPos of  $\Theta(1)$ ). However, if the communication interests contain BST conflicts or focal point conflicts, a distributed generalization of splay trees performs poorly (wcPoS of  $\Omega(\frac{n}{\log n})$ ).

There are a lot of different possibilities to extend our work. For example, it would be interesting to analyze the SABST-game with an arbitrary combination conflicts and give upper or lower bounds for the worst-case Price of Sinking. Moreover, it is interesting to compute the Price of Sinking as defined in [9] and thereby get statements about the average performance.

## Acknowledgements:

The author thanks Alexander Skopalik for stimulating discussions that improved the quality of the paper, and Christian Scheideler for his scientific guidance.

## References

1. Georgy Maximovic Adelson-Velsky and Yevgeny Mikhaylovich Landis. An algorithm for the organization of information. *Deklady Akademii Nauk USSR* 16, 16(2):263–266, 1962.
2. Noga Alon, Erik D. Demaine, MohammadTaghi Hajiaghayi, and Tom Leighton. Basic network creation games. In *SPAA*, pages 106–113, 2010.
3. Dushyant Arora, Marcin Bienkowski, Anja Feldmann, Gregor Schaffrath, and Stefan Schmid. Online strategies for intra and inter provider service migration in virtual networks. In *IPTcomm*, 2011.
4. Chen Avin, Bernhard Haeupler, Zvi Lotker, Christian Scheideler, and Stefan Schmid. Locally self-adjusting tree networks. In *IPDPS*, pages 395–406, 2013.
5. Rudolf Bayer. Symmetric binary b-trees: Data structure and maintenance algorithms. *Acta Inf.*, 1:290–306, 1972.
6. Andreas Cord-Landwehr, Martina Hüllmann, Peter Kling, and Alexander Setzer. Basic network creation games with communication interests. In *SAGT*, pages 72–83, 2012.
7. Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *PODC*, pages 347–351, 2003.
8. Igal Galperin and Ronald L. Rivest. Scapegoat trees. In *SODA*, pages 165–174, 1993.
9. Michel X. Goemans, Vahab S. Mirrokni, and Adrian Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–154, 2005.
10. Brandon Heller, Srinivasan Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. Elastic tree: Saving energy in data center networks. In *NSDI*, pages 249–264, 2010.
11. Joao Carlos Antunes Leitao, Joao Pedro da Silva Ferreira Moura Marques, Jose Orlando Roque Nascimento Pereira, and Luis Eduardo Teixeira Rodrigues. X-BOT: A protocol for resilient optimization of unstructured overlays. In *SRDS*, pages 236–245, 2009.
12. Pascal Lenzner. On dynamics in basic network creation games. In *SAGT*, pages 254–265, 2011.
13. Yunfei Shang, Dan Li, and Mingwei Xu. Energy-aware routing in data center network. In *Green Networking*, pages 1–8, 2010.
14. Murray Sherk. Self-adjusting k-ary search trees. In *WADS*, pages 381–392, 1989.
15. Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary trees. In *STOC*, pages 235–245, 1983.
16. Ming Tang, Zonghua Liu, Xiaoming Liang, and P. M. Hui. Self-adjusting routing schemes for time-varying traffic in scale-free networks. *Phys. Rev. E*, 80:026114, Aug 2009.
17. Chengwen Chris Wang, Jonathan Derryberry, and Daniel Dominic Sleator.  $O(\log \log n)$ -competitive dynamic binary search trees. In *SODA*, pages 374–383, 2006.