Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 20, no. 1, pp. 79–100 (2016)
DOI: 10.7155/jgaa.00385

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

Thim Strothmann

Department of Computer Science, University of Paderborn, Germany

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 over 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 that 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.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
March 2015	August 2015	August 2015	August 2015	January 2016
		Published: February 2016		
	Article type: Regular paper		Communicated by: M. S. Rahman and E. Tomita	

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center "On-The-Fly Computing" (SFB 901). A preliminary version of this paper appeared in the Proceedings of the 9th International Workshop on Algorithms and Computation (WALCOM 2015), February 2015.
E-mail address: thim@mail.upb.de (Thim Strothmann)

1 Introduction

Over 30 years ago, Sleator and Tarjan [25] 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 [25]. 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) 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 & Notation

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* [25]; 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 closely 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*.

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

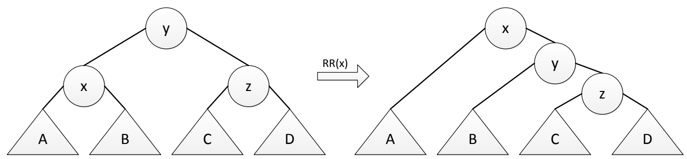


Figure 1: A single right rotation of node x. The triangles represent (possibly empty) subtrees that are not changed by the rotation.

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., [19]) over green computing [15] (because of reduced energy consumption) to adaptive virtual machine migrations [3, 23]. Self-adjusting routing schemes were examined to deal with congestion, e.g., in scale-free networks [26].

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 history that spans over more than three decades (e.g., [1, 5, 25]). However, research on self-adjusting data structures in general is a half-century old since it can be traced back to the seminal work of McCabe [21] from 1965. The focus of this paper is on splay trees [25]. 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 (e.g., [6, 8, 7, 11, 13]) and many splay tree variants have been proposed (e.g., [12, 24, 27, 16]) 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 σ. The upper bound for the amortized communication cost is based on the empirical entropies of σ. Furthermore, SplayNets have a provable online optimality under special request 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. [10]. 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 [20] 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. The BNCG model was extended in [9] by introducing communication interests of players. 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. We note that network formation with focus on social networks is a classic topic in economics and has been studied since the early 1990s [17] (see [18] for an excellent overview).

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 [14]). 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 $\Omega(\frac{n}{\log n})$. Furthermore, we show that a variant of the game in which nodes act altruistically does not yield better results in general.

2 Analysis

In general, the *SABST*-game does not converge and the dynamic process never settles on a 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.

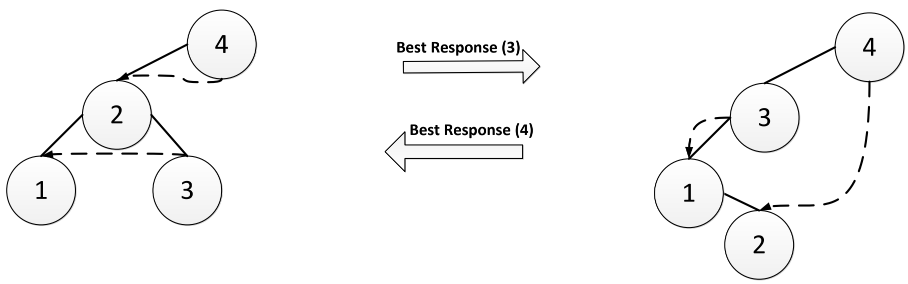
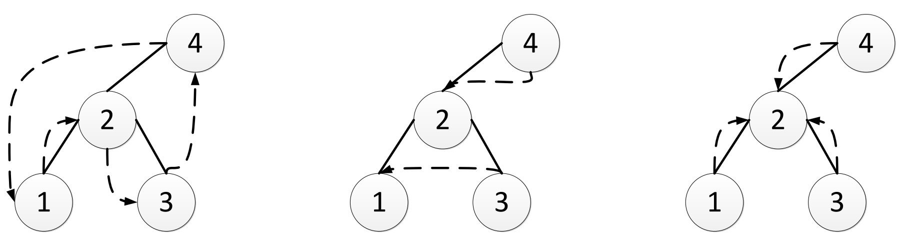


Figure 2: An example *SABST*-game instance that does not converge. Interest graph edges are dashed, connection graph edges are continuous.

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.



(a) A cyclic conflict

(b) A BST conflict

(c) A focal point conflict

Figure 3: Small examples of the three conflict classes.

2.1 Convergence & Rotation Equilibria

One can easily identify two classes of interest graphs that 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.

Proof: 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. Γ 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 Γ 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. □

Note that this result only holds for binary trees, since for k-ary trees the size of $\mathcal{H}$ is not necessarily monotonically increasing: i.e., if an 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. Γ 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.

Proof: Convergence is guaranteed since there is only one node with a non-empty interest set. Therefore, the only node actively trying to change G_C is the center node, since all other nodes have a private cost of 0. The resulting PoA of 2 depends on the fact that the center node performs rotations such that both subtrees are balanced binary trees. This is due to the fact that a balanced tree minimizes the average distance to the root. However, a socially optimal connection graph can also have a balanced binary tree of nodes above the center node to minimize the distances further: i.e., the center node is not the root of a social optimum.

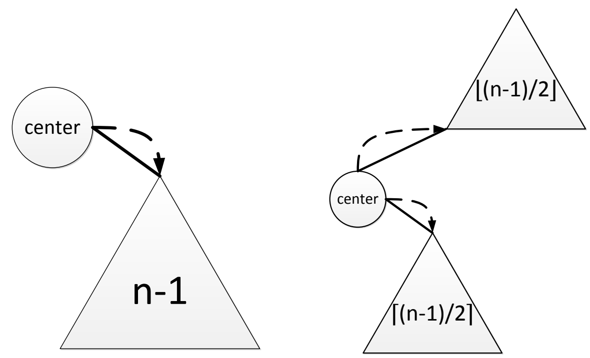
In a worst-case scenario the interest set of the center node consists only of nodes that have a smaller identifier than the center node. The resulting rotation equilibrium is visualized in Figure 4(a). A corresponding social optimum has half of these nodes above the center node in G_C (see Figure 4(b)). This results in a bisection of the social cost and the PoA is at most 2. □

Lemma 1 and 2 prove Theorem 1.

□

The rest of the 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 in 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.*



(a) Sketch of a worst-case rotation equilibrium

(b) Sketch of a social optimum

Figure 4: Comparison between a worst-case rotation equilibrium and a corresponding social optimum for the interest star.

Proof: Since e_I is neither in conflict with E'_I, nor a part of E'_I, we know that one of the following three cases must hold:

- u', v' are greater than all $w \in V'$,
- u', v' are smaller than all $w \in V'$ or
- there exists an edge $(u, v) \in E'_I$ such that u', v' are in the interval spanned by u, v.

Consequently, u', v' can be placed in a binary search tree such that they are below or above all nodes from V'. Note that in this case the lemma holds. Assume that there exists a connection graph in a rotation equilibrium G^*_C and u', v' affect the private costs of the nodes from V'. Therefore, there exist nodes $u^*, v^* \in V'$ with $v^* \in IS(u^*)$ such that u' and/or v' lie on the shortest path from u^* to v^* in G^*_C. Hence, v^* is not happy and can decrease its private cost, which contradicts the assumption that G^*_C is in a rotation equilibrium. $\square$

The proof from Lemma 3 can be expanded to easily show the following corollary.

Corollary 2 *Consider two connected components E'_I, E''_I of edges without conflicts from the interest graph $G_I = (V, E_I)$, the corresponding node sets $V' = \{v \in V | \exists u \in V \wedge (u, v) \in E'_I \vee (v, u) \in E'_I\}$ and $V'' = \{v \in V | \exists u \in V \wedge (u, v) \in E''_I \vee (v, u) \in E''_I\}$ with $E'_I \cap E''_I = \emptyset$ and $V' \cap V'' = \emptyset$ (i.e., E'_I, E''_I are maximal). If $\forall e'' \in E''_I$ it holds that e'' is not in conflict with E'_I. Then V'' does not affect the private cost of the nodes from V' in a rotation equilibrium and vice versa.*

Furthermore, it is possible to show that both statements hold if G'_I (or G''_I) contains conflicts. But since some additional notation is needed, to formally state this fact we postpone this result to the next section.

2.2 Non-Convergence & Sink Equilibria

As mentioned, 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. [14]. 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. [14] 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 wcPoS, 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* τ 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 τ starting from G_C.

Lemma 4 *Given an instance of the SABST-game Γ, a response order τ and a state s from the configuration graph $G_S = (V^*, E^*)$ of Γ. If $\forall s' \in V^*$ it holds that τ 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 τ. 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 τ that constructs the connection graph represented in s. Supplementary to the results of the last section, we can also show that a set of interest edges that contains conflicts does not affect the private cost of interest edges that are not in conflict with the set.

Lemma 5 *Consider a set of interest graph edges $E'_I \subset E_I$, 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 u, v are neither a part of V' nor is e_I in conflict with E'_I, then u' and v' do not affect the private cost of the nodes from V' in every sink equilibrium and vice versa.*

Proof: Similar to the proof of Lemma 3, one of the following three cases has to hold:

- u', v' are greater than all $w \in V'$,
- u', v' are smaller than all $w \in V'$ or
- there exists an edge $(u, v) \in E'_I$ such that u', v' are in the interval spanned by u, v.

Consequently, u', v' can be placed in G_C such that both nodes are below or above all nodes from V'. Note that as soon as u' makes its best response, it is exactly in this situation. Therefore, such a state is clearly in every sink equilibrium. W.l.o.g. assume that we are in the first case, i.e., u', v' are greater than all $w \in V'$, and that after the rotations of u', both u', v' are in the leftmost subtree of the nodes from V' (the other cases follow by a similar argument). Consequently, it remains to show that all states in sink equilibria fulfill this property: i.e. the private cost of u' is always 1.

Assume that there exists a connection graph in a sink equilibrium G^*_C and at least one node w from V' affects the private cost of u': i.e., w lies on the shortest path from u' to v' in G^*_C. Consequently, there has to exist a path in the configuration graph from the earlier mentioned state in which u' has private cost of 1 to G^*_C. However, in order to affect the private cost and to get in between u' and v', there needs to be an incentive for w to move u' or v' upwards in the tree. But since both nodes are not in $IS(w)$, this will never be the case and G^*_C cannot be in a sink equilibrium. Therefore, all nodes from V' do not affect the private cost of u' and v' and by applying the same arguments to every edge in E'_I, we also get the vice-versa statement. □

The proof from Lemma 5 can be expanded to easily show the following corollary.

Corollary 3 *Consider two connected components E'_I, E''_I of edges without conflicts from the interest graph $G_I = (V, E_I)$, the corresponding node sets $V' = \{v \in V | \exists u \in V \wedge (u, v) \in E'_I \vee (v, u) \in E'_I\}$ and $V'' = \{v \in V | \exists u \in V \wedge (u, v) \in E''_I \vee (v, u) \in E''_I\}$ with $E'_I \cap E''_I = \emptyset$ and $V' \cap V'' = \emptyset$ (i.e., E'_I, E''_I are maximal). If $\forall e'' \in E''_I$ it holds that e'' is not in conflict with E'_I. Then V'' does not affect the private cost of the nodes from V' in a rotation equilibrium and vice versa.*

Consequently, we can consider each *conflicting component* of edges independently: i.e. the set of edges that are mutually in conflict with each other.

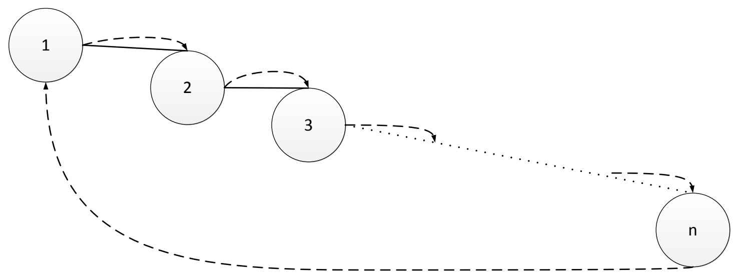


Figure 5: The connection graph for $\Gamma^{c.c.}$ after response order τ' is applied.

2.2.1 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 $V = \{1, \ldots, n\}$ and $E_I = \{(n, 1) \cup (i, i+1) : i = 1, \ldots, n-1\}$.

Theorem 4 *Let $\Gamma^{c.c.} = (G_C, G_I^{c.c.})$ be a SABST-game, the wcPoS is $\mathcal{O}(1)$.*

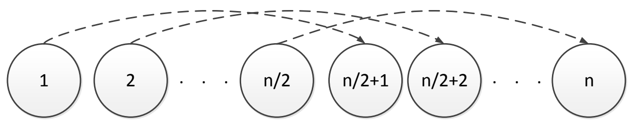
Consequently, as long as the communication interests contain only cyclic conflicts, the performance of the self-adjusting tree is asymptotically as good as the performance without conflicts. To prove Theorem 4, we need to show the following two lemmas.

Lemma 6 *For the SABST-game $\Gamma^{c.c.}$, every state in the unique sink equilibrium has social cost of $2(n-1)$.*

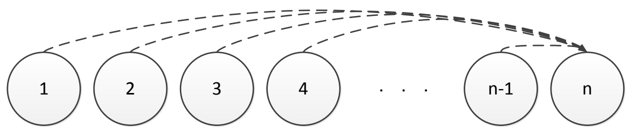
Proof: Let $\tau' = (n, \ldots, 1)$ be a response order. If τ' is applied on G_C the resulting connection graph is the one visualized in Figure 5, which is in a unique sink equilibrium. The social cost is $2(n-1)$. Now independent of a response order, there is only one unhappy node in the connection 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 $2(n-1)$, □

Lemma 7 *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) The interest graph $G_I^{d.c.}$



(b) The interest graph $G_I^{f.c.}$

Figure 6: The interest graphs for the analysis of BST conflicts and focal point conflicts

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. □

Lemma 6 and Lemma 7 together conclude the proof.

2.2.2 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 that only have 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 (see Figure 6a).

Theorem 5 *Let $\Gamma^{d.c.} = (G_C, G_I^{d.c.})$ be a SABST-game, the corresponding wcPoS is $\Omega(\frac{n}{\log(n)})$.*

To prove Theorem 5, we first prove that the configuration graph of Γ contains a unique sink equilibrium with a state that has social cost of $\Theta(n^2)$.

Lemma 8 *The configuration graph of $\Gamma^{d.c.}$ contains a state in the unique sink equilibrium with social cost of $\Theta(n^2)$.*

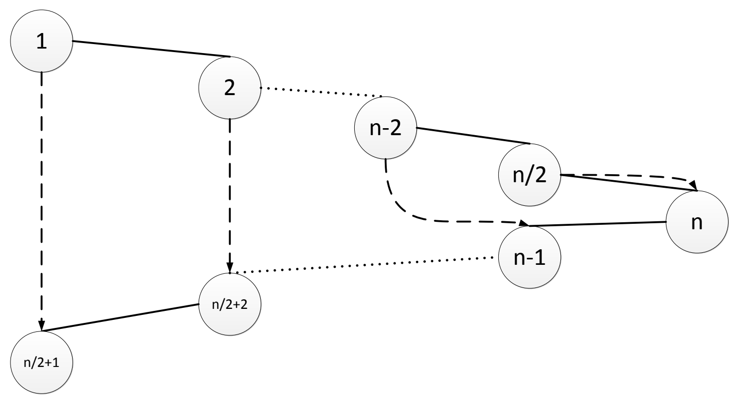


Figure 7: The connection graph for a $\Gamma^{d.c.}$ if response order $\tau' = (1, \ldots, n)$ is applied.

Proof: We pick the response order $\tau' = (1, \ldots, n)$. If we now apply τ' to any initial connection graph, we end up with the connection graph presented in Figure 7. This is due to the fact that every player performs rotations such that the nodes from its interest set are in one of its subtrees. Therefore, after the response of node 1, the node $\frac{n}{2}+1$ has to be the right child of 1. In the next step, node 2 has the node $\frac{n}{2}+2$ as its right child, whereas 1 will be the parent of 2 and $\frac{n}{2}+1$ the left child of $\frac{n}{2}+2$, i.e., the private cost of 1 doubles. We can inductively use this construction to get to the connection graph from Figure 7. 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 τ', we know that it is a state in the unique sink equilibrium of $\Gamma^{d.c.}$. □

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

Lemma 9 *A social optimum for $\Gamma^{d.c.}$ has social cost of at most $\mathcal{O}(n \log n)$.*

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

In fact, the social cost of a social optimum is overestimated by Lemma 9. By a construction similar to the social optimum of Lemma 2, we can decrease the social cost by a constant factor (see Figure 8).

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 $V = \{1, \ldots n\}$ and $E_I = \{(1, n), (2, n), \ldots, (n-1, n)\}$, which has the maximal possible focal point conflict (see Figure 6b).

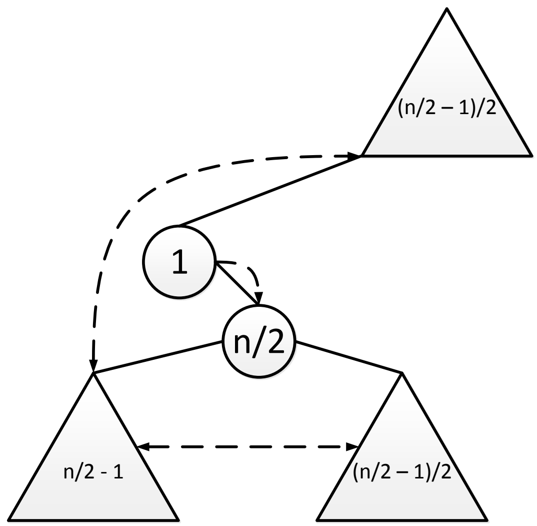


Figure 8: Sketch of a social optimum for BST conflicts

Theorem 6 *Let $\Gamma^{f.c.} = (G_C, G_I^{f.c.})$ be a SABST-game, the corresponding wc-PoS is $\Omega(\frac{n}{\log(n)})$.*

Proof: The proof technique for Theorem 6 is analogous to Theorem 5, i.e., we need to show that there is a state in the unique sink equilibrium with social cost of $\Theta(n^2)$ and that a social optimum has social cost of at most $\mathcal{O}(n \log n)$. For the social optimum we refer to Lemma 9 and Figure 8, since the results can be directly applied. To show that there is a a state in the unique sink equilibrium with high social cost we pick the response order $\tau'' = (n-1, \ldots, 1)$. Independent of the initial connection graph, τ' leads to the connection graph presented in Figure 9. By summing up the private cost of each node, we get the desired result, i.e., the social cost of $\Theta(n^2)$, and therefore the theorem holds. □

Theorems 5 and 6 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 Altruistic Behavior

In this section we want to explore the possibilities of a model in which the nodes do not act selfishly but altruistically. One way to express this formally

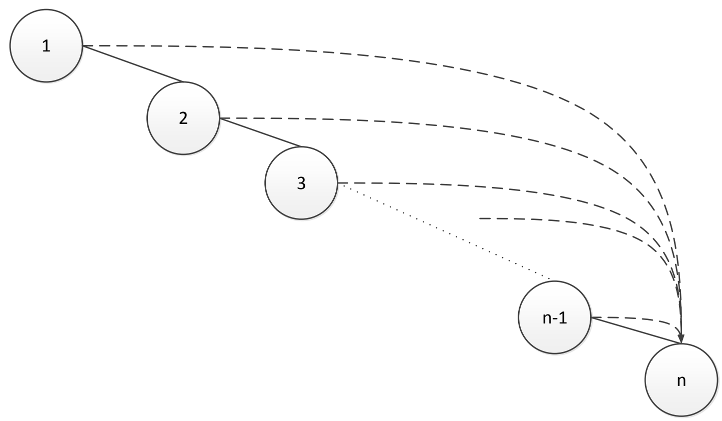


Figure 9: The connection graph for a $\Gamma^{f.c.}$ if response order τ'' is applied.

is by changing the private cost of each player to be the social cost: i.e. for the *altruistic node behavior (ANB)* version of the *SABST*-game, the private cost of each node is the sum over all connection costs for all players.

Without proof we can state that the ANB-version guarantees convergence. This results from the fact that this variant of the game is similar to an *ordinal potential game* as defined by Monderer and Shapely [22]. A game is an ordinal potential game if the incentive of all players to change their strategy can be expressed by a single global function, the so-called potential function. In the ANB-version the sum over all connection costs for all players yields this potential function.

Intuitively one might expect that the ANB-version leads to rotation equilibria with low social cost. This intuition is not completely wrong. We can construct instances of the *SABST*-game in which the ANB-model leads to a rotation equilibrium with better social cost, than the original model. However, there are also examples in which the ANB-version never reaches a social optimal connection graph, but the selfish model has a social optimum in a sink equilibrium.

Lemma 10 *There are instances of the SABST-game in which the ANB-version leads to a rotation equilibrium, which has better social cost, than the original model.*

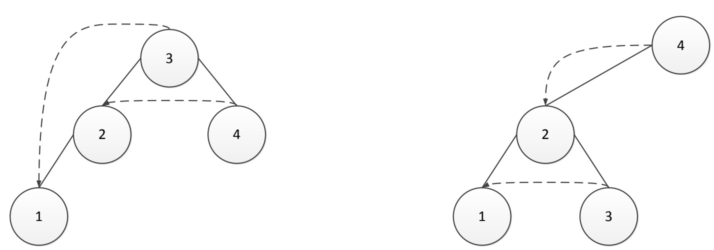
Proof: We consider one of the interest graph classes from Section 2.1: a star interest graph. As shown in Lemma 2 this interest graph leads to a Price of Anarchy of 2 if nodes behave selfishly. However, in the ANB-version every node tries to minimize the social cost by performing rotations (in contrast to only the central node). If we take the rotation equilibrium from the proof of Lemma 2 as a starting point for this proof, all nodes in the interest set of the central nodes are without loss of generality in its left subtree. Since all nodess now behave altruistically, a node in the subtree can improve the social cost by rotating over the central node and become an ancestor of it (or a child of the ancestors). Depending on the order in which the nodes perform their rotations, it is possible to reach a social optimum (see Figure 4b for a sketch of a social optimum). Independent of the order, we know that we reach a rotation equilibrium with lower social cost. □

A main reason for this result is the fact that in the ANB-version each node is able to change the structure of the connection graph, whereas in the selfish model only the nodes with non-empty interest sets will perform rotations. Thus, for the interest star we have n active nodes instead of 1. We can contrast this result with the following lemma.

Lemma 11 *There are instances of the ANB-version of the SABST-game that lead to a rotation equilibrium which has worse social cost than the state with the lowest social cost in a sink equilibrium in the original model. Furthermore, the social cost of a rotation equilibrium in the ANB-version can be as bad as the social cost of a worst-case sink equilibrium state.*

Proof: To show the lemma, we construct a small instance of the *SABST*-game with 4 nodes. The interest edge set is $E_I = \{(3,1),(4,2)\}$. In the initial connection graph node 3 is the root of the tree and $lc(3) = 2, lc(2) = 1, rc(3) = 4$. A visual representation of this instance is shown in Figure 3.

For the ANB-version the initial connection graph is a rotation equilibrium, since no node can improve the social cost by rotations. Therefore, we have social cost of 4. However, with selfish behavior we can achieve a connection graph with social cost 3(see Figure 3). In this instance of the game, 4 is the social cost of the worst state in the sink equilibrium (which in this case consists of only 2 states), which concludes the proof. □



(a) The initial connection graph of the game and the rotation equilibrium in the ANB-model.

(b) The best sink equilibrium state in the original model.

Figure 10: Visualization of the proof of Lemma 11: The rotation equilibrium in the ANB-model and the best sink equilibrium state in the original model.

A reason why the performance of the ANB-version can be bad is the fact that once it reaches a binary search tree that is in a sink equilibrium in the original model, the game stops, since all possible rotations do not decrease the social cost (by definition of a sink equilibrium). Since nodes are myopic they do not perform a rotation that increases the social cost, but could lead to lower social costs after further rotations: i.e., the process gets stuck in local minima. The selfish model does not have this drawback, therefore it reaches every state in a sink equilibrium that is reachable from the initial connection graph and thus reaches a state with better social cost.

4 Extensions

We want to conclude our analysis by comparing our results to the work of [4]. As mentioned before, their vision of a distributed self-adjusting binary search tree is quite similar to ours. However, they develop some positive results, while our work shows the disadvantages of choosing a simple double-splay strategy to adjust the tree. At first, note that their model differs from ours in many aspects. However, there are mainly two differences that affect the analysis:

1. We use simple rotations to adjust the binary search tree, while [4] uses the splay operations as introduced by [25] (see Figure 11 for their graphical description).
2. In the *SABST*-game, the interest graph is given from the start: i.e., every node knows its complete interest set from the beginning. [4] hides this knowledge from the nodes. A request pattern is analyzed, but at each timestep only one request is revealed to the tree and it has to serve that request.

However, these differences do not affect our worst-case results. Since the splay operations are just a subset of all possible rotations allowed in a binary search tree, our approach of using a simple rotations as a building block gives the nodes more freedom. This is due to the fact that we allow a node to perform as many rotations as needed on itself and its interest set. Nodes would not perform better in our worst-case scenarios if they had to rely on splay operations only.

We note that the second difference, i.e., the tree has to react to only a single request from a pattern at each timestep, is a more natural scenario than ours, in which the interest set is known a priori. However, especially for our worst-case results of Section 2.2.2 the difference does not matter. By construction of the special interest graph family each node only has one interest edge. Therefore, our result directly transfers to a scenario in which nodes only see one request per timestep. In addition we can also extend our model, so that interest edges are annotated with probabilities: i.e., $G_I = (V, E_I, w)$ where $w : E_I \to [0,1]$ with the property that $\forall v \in V : \sum_{(v,u)\in E_I} w(v,u) = 1$. Thereby, we can model that a node u has preferences among nodes in its interest set. Those nodes to which communication is more likely should be closer to u in G_C. Our results

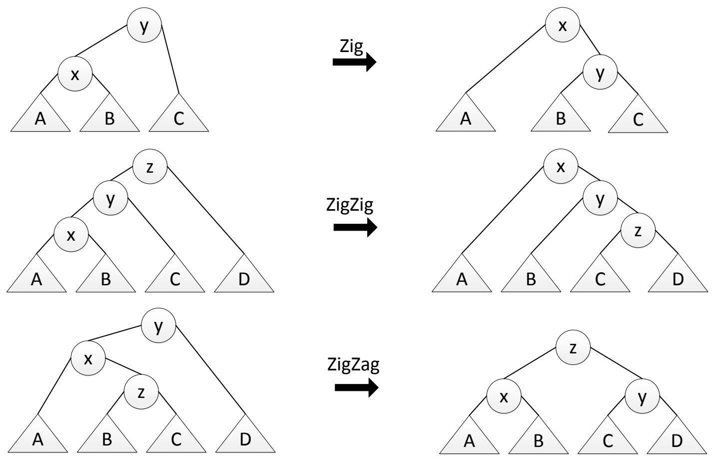


Figure 11: The three basic splay operations of splay trees.

developed in this paper hold for the case in which all interest edges adjacent to a node have the same probability

Furthermore, most of the positive results developed in [4] can be confirmed in our model, sometimes even extended. This is true for the *laminated scenario*, the *non-crossing matching scenario* (which are both covered by our results in Lemma 3) and the *multicast tree scenario* (equivalent to parts of our Theorem 1).

It remains open to show whether the performance of a distributed self-adjusting binary search tree for general interest graphs is more influenced by subgraphs that lead to bad performance or by those with a structure that is served well.

5 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 $\Omega(\frac{n}{\log n})$). Moreover, we were able to show that in general altruistic behavior does not yield better results than selfish behavior.

There are a lot of different interesting possibilities to extend our work. For example, it is challenging to analyze the *SABST*-game with 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 [14] 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] G. M. Adelson-Velsky and Y. M. Landis. An algorithm for the organization of information. *Deklady Akademii Nauk USSR 16*, 16(2):263–266, 1962.

[2] N. Alon, E. D. Demaine, M. Hajiaghayi, and T. Leighton. Basic network creation games. In *SPAA*, pages 106–113, 2010. doi:10.1145/1810479.1810502.

[3] D. Arora, M. Bienkowski, A. Feldmann, G. Schaffrath, and S. Schmid. Online strategies for intra and inter provider service migration in virtual networks. In *IPTcomm*, pages 10:1–10:11, 2011. doi:10.1145/2124436.2124449.

[4] C. Avin, B. Haeupler, Z. Lotker, C. Scheideler, and S. Schmid. Locally self-adjusting tree networks. In *IPDPS*, pages 395–406, 2013. doi:10.1109/IPDPS.2013.40.

[5] R. Bayer. Symmetric binary b-trees: Data structure and maintenance algorithms. *Acta Inf.*, 1:290–306, 1972. doi:10.1007/BF00289509.

[6] R. Cole. On the dynamic finger conjecture for splay trees (extended abstract). In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 8–17, 1990. doi:10.1145/100216.100218.

[7] R. Cole. On the dynamic finger conjecture for splay trees. part II: the proof. *SIAM J. Comput.*, 30(1):44–85, 2000. doi:10.1137/S009753979732699X.

[8] R. Cole, B. Mishra, J. P. Schmidt, and A. Siegel. On the dynamic finger conjecture for splay trees. part I: splay sorting log n-block sequences. *SIAM J. Comput.*, 30(1):1–43, 2000. doi:10.1137/S0097539797326988.

[9] A. Cord-Landwehr, M. Hüllmann, P. Kling, and A. Setzer. Basic network creation games with communication interests. In *SAGT*, pages 72–83, 2012. doi:10.1007/978-3-642-33996-7_7.

[10] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *PODC*, pages 347–351, 2003. doi:10.1145/872035.872088.

[11] M. Fürer. Randomized splay trees. In *SODA*, pages 903–904, 1999.

[12] I. Galperin and R. L. Rivest. Scapegoat trees. In *SODA*, pages 165–174, 1993.

[13] G. F. Georgakopoulos. How to splay for loglog n-competitiveness. In *WEA*, pages 570–579, 2005. doi:10.1007/11427186_49.

[14] M. X. Goemans, V. S. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *FOCS*, pages 142–154, 2005. doi:10.1109/SFCS.2005.68.

[15] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: Saving energy in data center networks. In *NSDI*, pages 249–264, 2010.

[16] J. Iacono. Alternatives to splay trees with o(log n) worst-case access times. In *SODA*, pages 516–522, 2001.

[17] M. Jackson and A. Wolinsky. A strategic model of social and economic networks. *Journal of Economic Theory*, 71:44–74, 1996. `doi:10.1006/jeth.1996.0108`.

[18] M. O. Jackson. *Social and Economic Networks*. Princeton University Press, Princeton, NJ, USA, 2008.

[19] J. C. A. Leitao, J. P. da Silva Ferreira Moura Marques, J. O. R. N. Pereira, and L. E. T. Rodrigues. X-BOT: A protocol for resilient optimization of unstructured overlays. In *SRDS*, pages 236–245, 2009. `doi:10.1109/SRDS.2009.20`.

[20] P. Lenzner. On dynamics in basic network creation games. In *SAGT*, pages 254–265, 2011. `doi:10.1007/978-3-642-24829-0_23`.

[21] J. McCabe. On serial files with relocatable records. *Operations Research*, 13(4):609–618, 1965. `doi:10.1287/opre.13.4.609`.

[22] D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996. `doi:10.1006/game.1996.0044`.

[23] Y. Shang, D. Li, and M. Xu. Energy-aware routing in data center network. In *Green Networking*, pages 1–8, 2010. `doi:10.1145/1851290.1851292`.

[24] M. Sherk. Self-adjusting k-ary search trees. In *WADS*, pages 381–392, 1989. `doi:10.1006/jagm.1995.1026`.

[25] D. D. Sleator and R. E. Tarjan. Self-adjusting binary trees. In *STOC*, pages 235–245, 1983. `doi:10.1145/800061.808752`.

[26] M. Tang, Z. Liu, X. Liang, and P. M. Hui. Self-adjusting routing schemes for time-varying traffic in scale-free networks. *Phys. Rev. E*, 80:026114, Aug 2009.

[27] C. C. Wang, J. Derryberry, and D. D. Sleator. $O(\log \log n)$-competitive dynamic binary search trees. In *SODA*, pages 374–383, 2006.