International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

SPLAYNET : Towards Nearby Self-Adjusting Wireless Networks

Gaddipathi Bharathi*[1], Parchuri Kaneeja[2]

*[1]Associate Professor, Department of MCA , St. Mary's Group of Institutions, Guntur, Andhra Pradesh, India.
[2]PG Student ,Department of MCA , St. Mary's Group of Institutions, Guntur, Andhra Pradesh, India

ABSTRACT

This paper starts the investigation of locally self-changing systems: organizes whose topology adjusts progressively and in a decentralized way, to the correspondence design . Our vision can be viewed as a dispersed speculation of the self-altering datastructures presented by Sleator and Tarjan [22]: rather than their spread trees which powerfully upgrade the query costs from a solitary hub (in particular the tree root), we try to limit the steering cost between discretionary correspondence combines in the system. As an initial step, we consider disseminated twofold hunt trees (BSTs), which are appealing for their help of insatiable steering. We present a straightforward model which catches the key tradeoff between the advantages and expenses of self-changing systems. We introduce the SplayNet calculation and formally dissect its execution, and demonstrate its optimality in particular contextual investigations. We additionally present lower bound methods in light of interim cuts and edge development, to think about the restrictions of any request enhanced system. At last, we stretch out our investigation to multi-tree systems, and highlight an interesting contrast amongst great and disseminated spread trees.

Keywords : BST, Self-Adjusting Wireless Networks, SPLAYNET

I. INTRODUCTION

In the 1980s, Sleator and Tarjan [22] proposed an engaging new worldview to outline effective Binary Search Tree (BST) data structures: as opposed to advancing customary measurements, for example, the inquiry tree profundity in the most pessimistic scenario, their spread data structure self-changes with its utilization design, moving even more oftentimes got to components nearer to the root. A characteristic execution metric to assess a self-changing framework is the amortized taken a toll: the "normal cost" for a most pessimistic scenario arrangement of operations (of a specific class).

Since this fundamental work, self-altering data structures have been considered seriously, and different more productive self-changing data structures, for example, Tango BSTs [7] or multi-spread trees [23] have been proposed. Specifically, the famous Dynamic Optimality guess [7] keeps on confounding scientists: the guess asserts that spread trees execute and some other paired inquiry tree calculation up to a steady factor.

Rather than these adaptable exemplary data structures, the present disseminated data structures and systems are still enhanced

Toward static measurements, for example, the breadth or the length of the longest course: the self-altering worldview has not overflowed to appropriate arranges yet.

We, in this paper, start the investigation of a circulated general-inaction of self-streamlining data structures. This is a non-insignificant speculation of the exemplary spread tree idea: While in class-sic BSTs, a query ask for dependably begins from a similar hub, the tree root, disseminated data structures and systems, for example, skip diagrams [2], [13] need to help directing solicitations between subjective combines (or

companions) of conveying hubs; at the end of the day, both the source and additionally the goal of the solicitations wind up noticeably factor. Figure 1 shows the distinction amongst exemplary and appropriated parallel pursuit trees.

In this paper, we ask: Can we receive comparable rewards from self-modifying whole systems, by adaptively diminishing the separation between every now and again imparting hubs?

As an initial step, we investigate completely decentralized and self-modifying Binary Search Tree systems: in these systems, hubs are orchestrated in a paired tree, which regards hub identifiers. A BST topology is appealing as it bolsters avaricious directing: a hub can choose locally to which port to forward a demand given its goal address.

A. Our Contributions

This paper makes the accompanying commitments.

1) We start the investigation of self-changing appropriated reports structures and present a formal model likewise. Our model is straightforward yet catches the basic tradeoffs between the advantages of self-modifications (to be specific shorter steering ways) and their expenses (specifically reconfigurations).

2) We present a self-altering disseminated BST called Splay Net. Splay Net is a characteristic speculation of the exemplary spread tree calculation, which "spreads" communication accomplices to their normal predecessor. Splay Net is completely decentralized as in every topological change and additionally directing is neighbourhood.

3) We formally dissect the execution of Splay Net (as far as amortized costs). Specifically, we demonstrate that the general cost is upper limited by the exact entropies of the sources and goals in the correspondence design; a basic lower bound takes after from restrictive experimental entropies. We likewise demonstrate the optimality of our approach in particular contextual investigations, e.g., when the communication design takes after an item circulation. At last, we additionally display a dynamic programming calculation to ideally take care of the disconnected issue variation in polynomial time.

4) We acquaint novel lower bound procedures with think about the confinements of self-changing systems. These methods depend on interim cuts and edge extension, and may. Be of unbiased interest and find programs beyond the putting studied on this paper.

Five) in the end, we provoke the dialogue of greater complex self-adjusting networks, particularly topologies inclusive of multiple trees. We make the interesting remark that in contrast to traditional data structures in which the self-adjustment blessings of multiple timber is restricted, in a distributed placing, a single extra best can on occasion reduce the amortized fee dramatically.

In summary, our work shows that even as a few algorithmic concepts of conventional splay trees may be generalized to networks, the distributed placing calls for new analytical tools. Furthermore, our consequences spotlight that self-adjustment benefits can certainly be reaped also within the context of networks; for multi-tree networks, those benefits can even be drastically better than in traditional data structures.

In trendy, we regard our look at as a first step, and trust that our model and outcomes open a rich location for future studies.

B. Paper Organisation

The imminent segment ii introduces our formal model and presents the reader with the essential heritage. sec-tin iii describes an offline set of rules to compute foremost static disbursed bests and presents the splay net method. phase iv derives entropy-based totally upper and decrease bounds on the performance of splaynets, and segment v studies the locality and convergence properties of the splaynet algorithm in specific eventualities. phase vi then derives stepped forward lower bounds which permit us to show the optimality of splaynets in additional situations. in phase vii we initiate the dialogue of datastructures primarily based on multiple bsts. after reviewing associated paintings in section viii, we conclude our contribution in segment ix. within the appendix, a few extra technical information are supplied.

II. Version and history

We take into account a set of n nodes (or peers) v = f1; ng interacting in line with a positive communication sample. the sample is modeled by way of = (0; 1 : : : m 1): a chain of m conversation requests wherein t = (u; v) 2 v v , with supply u to destination v, henceforth now and again denoted with the aid of src(t) and dst(t), respectively. Our goal is to find a verbal exchange community g which connects the nodes v in step with the conversation sample: additionally, g have to be selected from a certain circle of relatives of favored topologies g, as an example, the set of tree topologies (the point of interest of this paper), expander graphs, or low-diameter networks, and so forth. every topology g 2 g is a graph g = (v; e). we distinguish trouble versions: (1) a static variation where g can be optimized in the direction of the verbal exchange pattern inside the feel that it may make the most, e.g., lengthy-term characteristics of , however, g is constant and can't change over time. (2) a self-adjusting variation where g may be adapted through the years.

Typically, it's miles suited that networks are adjusted easily, and we're interested by nearby transformations: converting communication pattern ends in “nearby” adjustments of the communication graph through the years.

2. as cited above, this paper focuses on a placing in which g represents the set of binary search trees (bsts), for this reason-forth sometimes clearly called tree networks. except their simplicity, bsts are appealing for their low node degree and the opportunity to path regionally: given a destination identifier (or cope with), each node can determine domestically whether to forward the packet to its left toddler, its proper baby, or its determine; see appendix a for info.

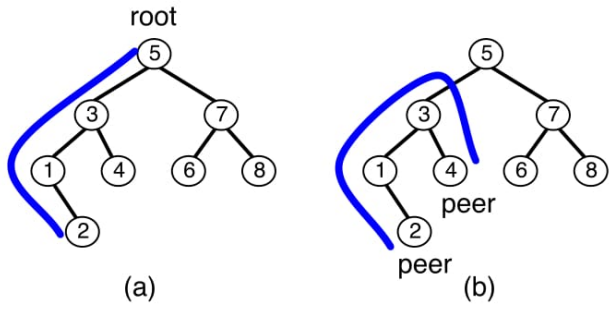

Figure 1: (a) Classic BST versus (b) conveyed BST: Classic spread tree datastructures enhance the separation of the components from the root (the query cost), while in an appropriated datastructure, correspondence happens between self-assertive hubs (the associates). Additionally take note of that in a BST organize, asks for likewise travel upwards in the tree; by the by, as we will see, steering choices are totally nearby.

The neighborhood changes of tree systems are called ro-tations. Casually, a revolution in an arranged twofold hunt tree switches up to three contiguousness connections, while keeping subtrees in place. Note that it is conceivable to change any parallel hunt tree into some other twofold pursuit tree by a succession of neighborhood changes (e.g., by enlistment over the subtree roots).

For our formal examination, we consider a disentangled syn-chronous show where initial a correspondence ask for arrives, at that point nearby system changes can be performed, lastly, the demand is fulfilled (i.e., the movement steered). In this paper, we are regularly keen on a setting where the solicitations are drawn indiscriminately from a settled however obscure correspondence lattice. Solidly, we will in some cases see the correspondence asks for as instigating a demand chart R( ) = (V ( ); E( )) over the vertices V ; the edges E( ) of R( ) are commented on with recurrence data. (At the point when clear from the unique circumstance, we will regularly discard in R( ), V ( ), E( ), and basically compose R, V , E.)

Solidly, the hub set V of R is given by the arrangement of hubs taking part in , i.e., V = fv : 9t; v 2 tg, and the arrangement of coordinated edges E is given by E = f t : t 2 [0; : ; m 1]g. The weight w(e) of each coordinated edge e = (u; v) 2 E is the recurrence f(u; v) of the demand from u to v in . In the accompanying, we will once in a while basically compose w(u; v) to indicate the weight w(e) of edge e. For instance, in a few situations the correspondence design between the hubs V may shape a tree (e.g., a multicast tree), an entire diagram, or an arrangement of separated parts (e.g., portraying a bunched correspondence design).

let a be an algorithm that given the request t and the graph gt 2 g at time t, transforms the present day graph (thru neighborhood ameliorations) to gt+1 2 g at time t + 1. we can use the notation a = ? to refer to a static (i.e., non-adjusting) “al-gorithm” which does no longer exchange the conversation community over time.

We are inquisitive about the essential tradeoff between the benefits of self-adjusting algorithms (i.e., shorter routing paths) and their charges (specifically reconfiguration costs). we introduce a maximum simple, linear price model that captures this tradeoff. concretely, we denote the fee of the community differences at time t by way of (a; gt; t), and we denote the quantity of rotations executed to exchange gt to gt+1; while a is clear from the context, we will without a doubt write t. we denote with dg( ) the distance function between nodes in g, i.e., for 2 nodes v; u 2 v we define dg(u; v) to be the wide variety of edges of a shortest course between u and v in g, and we count on messages are routed alongside the shortest paths. for a given collection of communique requests, the cost for an algorithm is given via the wide variety of changes and the distance of the conversation requests plus one (i.e., additionally a request (u; u) comes at a minimum cost of one unit).

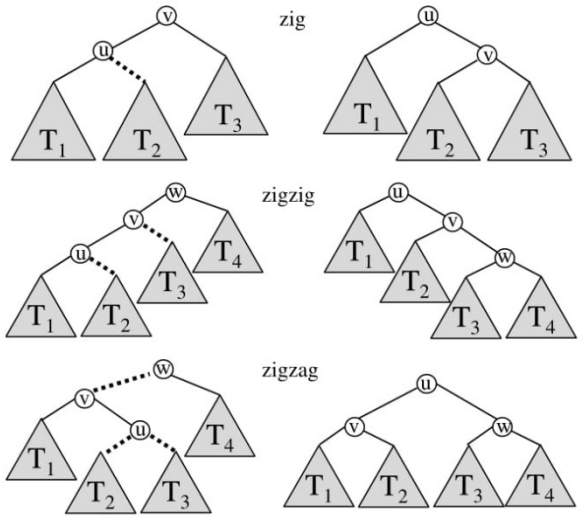

Fig. 2: Basic rotations of splay trees. The dashed bold line indicate adjacency relationships which are not maintained during the operation.

IG we need the subsequent formal definitions. definition 1 (average and amortized cost). for an algorithm a and given an initial community g0 with node distance function d( ) and a sequence = (0; 1 : : : m 1) of communication requests through the years, we outline the (average) price of a as:

1 m value(a; g0;) =xt(dgt (t) + 1 + t)(1)m =zero
the amortized value of a is defined because the worst feasible fee of a, i.e., maxg0; fee(a; g0;).

Entropy and Empirical Entropy

The entropy of the communication pattern turns out to be a useful parameter to evaluate the performance of self-adjusting SplayNets. For a discrete random variable X with possible values f x ; x , the entropy H(X) of X is n 1 ng1 that P i 2 0 defined as i=1 p(xi) log2 p(xi) where p(xi) is the probability X takes the value x . Note that, 0 log 1 is considered as 0. For a joint distribution over X; Y , the joint entropy is defined as H(X; Y) = i;j p(xi; yj) log2 1 . Also p(xi;yj) definition of the conditional entropy H(X Y):recall the n P j H(XjY) = j=1 p(yj)H(XjY = yj). sequence of communications is revealed over Since the P

Since the P time and may not be chosen from a fixed probability distribution, we are often interested in the empirical entropy of, i.e., the entropy implied by the communication fre-^ ff(x1); : : : ; f(xn)g be the empirical
quencies. Let X( ) = entropy measure of the frequency distribution of the com-munication sources (origins) occurring in the communica-tion sequence , i.e., f(xi) is the frequency with which a node xi appears as a source in the sequence, i.e., f(xi) =
and analogously, the empirical conditional entropies H(XjY) and H(Y jX).
B. Splay Trees

Our work can be regarded as a distributed generalization of splay trees, binary search trees whose topology adapts to the lookup sequence. Indeed, assuming that all requests originate from the same node, the SplayNet problem becomes equivalent to the classic splay tree problem. In the following, we hence briefly review the concept of splay trees.

For a node set V with unique identifiers (IDs) or values, we consider the family B of the set of all binary search trees over the IDs of V . Let s = (v0; v1; : : : ; vm 1), vi 2 V , be a sequence of lookup requests. In the classic offline problem (i.e., for algorithm A = ?), the goal is to find the best search tree T 2 B that minimizes the cost Cost(?; T ; s).

We will make use of the following two well-known prop-erties of optimal BSTs

III. SPLAY NETWORKS

We first acquaint ourselves with the distributed BST model by studying optimal static topologies. Subsequently, we present the SplayNet approach and introduce the simple SPLAYNET algorithm to self-adjust the network.

A. Optimal Static Distributed BST

Given a certain communication pattern or "guest graph", the optimal BST can be computed in polynomial time using a dynamic programming approach. The main insight needed is that the problem for the entire tree can be decomposed into optimal subproblems for smaller trees, and that the demand towards a given node in a subtree can be decoupled from nodes outside a given subtree: the precise topological structure of the nodes outside a subtree does not matter.
Concretely, the optimal static tree T which minimizes the sum of the weighted node distances

B. Self-Adjusting Distributed BSTs

Let us now consider self-adjusting BST networks. The SplayNet algorithm presented in this paper is a natural gener-alization of the classic splay tree algorithm. It is based on a double splay strategy: similarly to classic splay trees, SplayNet aggressively moves communicating nodes together; however, rather than splaying nodes to the root of the BST, locality is preserved in the sense that the source and the destination node are only rotated to their common ancestor (of the subtree covering the communication partners).

Concretely, consider a communication request (u; v) from node u to node v, and let T (u; v) denote the lowest common ancestor of u and v in the current network T . For an arbitrary node x, let T (x) be the subtree rooted at x. When a request (u; v) occurs, SplayNet first simply splays u to the lowest common ancestor T (u; v) of u and v, using the classic splay operations Zig, ZigZig, ZigZag from [22] (see Figure 2). We assume that the splay function returns the tree resulting from these operations. Subsequently, the idea is to splay the destination node v to the child of the lowest common ancestor T 0(u; v) of u and v in the resulting tree T 0. Observe that this common ancestor is u itself (u = T 0(u; v)), i.e., we define the double splay algorithm SplayNet to splay v such that it becomes a child of u. (Note that the child is uniquely defined: if u > v, v will be the left, if u < v, the right child of u.)

The formal algorithm listing of SplayNet is shown in Algorithm 2.

Algorithm 2 Algorithm SplayNet

1. (* upon request (u; v) in T *)
2. w := T (u; v)
3. T 0 := splay u to root of T (w)
4. splay v to the child of T 0(u)

CASE STUDIES: LOCALITY AND OPTIMALITY

The section provides insights into the properties of our algorithms in different specific settings. In particular, using different case studies, we show that our approach sometimes exhibits a desirable local convergence to the optimal network. The section also serves the purpose of introducing some analytical tools that are useful to study SplayNets.

A. Cluster Scenario

We first study a scenario which shows the locality properties of SplayNet. In this scenario, requests are clustered, and so is the resulting tree of SplayNet.

Definition 3 (Cluster Scenario). In a cluster scenario the com-munication pattern partitions the nodes into k contiguous and disjoint intervals I1 [I2 [I3 [: [Ik where nodes within an interval Ij have consecutive numbers and where commu-nication only happens between node pairs in the interval. In particular, a request (u; v) implies that u and v belong to the same interval: (u; v) 2 ! 9j : u; v 2 Ij. A maximal cluster scenario is a cluster scenario where no intervals can be divided into two intervals. (a) for an example of a maximal cluster scenario. For this case we are able to prove the following.
Theorem 8. In a maximal cluster scenario , SplayNet features the following two properties:

1) SplayNet will eventually construct a tree network in which for any communication pair (u; v) 2 Ij, for any j 2 f1; : : : ; kg, u and v are connected by a local path which only includes nodes from Ij.
2) Once this local routing property is established, it will never be violated again.

Proof: For any BST T and an interval Ij let rj be the node such that the subtree T (rj) is the smallest size sub-tree where all the nodes of Ij are in T (rj); we denote this tree by T (Ij). Let out(T (Ij)) denote the number of requests (u; v) 2 R( ) s.t. either u or v are in T (Ij), but not both.

For the (maximal) cluster scenario and a BST T , we consider the potential function = k j=1 out(T (Ij)). We will prove the theorem by first showing that when = 0, for P
any (u; v) 2 Ij, j 2 f1; : : : ; kg, u and v are connected by a local path: a path which only includes nodes from Ij; remains zero after such a request. Subsequently, we will show
Illustration for proof of Claim 1 and Theorem 8. T (rj) is the tree rooted at rj (and includes T1 and T2). that when > 0, SplayNet cannot increase , and there exists a request in which will reduce the potential.

We start with the following claim.

Claim 1. Consider Ij s.t. out(T (Ij)) = 0. Then this property is invariant for all future requests in and out(T (Ij)) will always remain 0.

Proof: A generic situation for T (Ij) is illustrated Each node not in Ij is in one of 3 possible locations: in one of the (possibly empty) subtrees T1; T2; or T3. Clearly, T1 contains only IDs smaller than Ij and must be attached to the smallest ID in Ij, and T2 contains only IDs larger than Ij and must be attached to the largest ID in Ij. T3 can contain either smaller or larger IDs depending on whether rj is a left or a right son of its parent. Since out(T (Ij)) = 0, there are no requests involving T3 so all requests remain within T (Ij) and so out(T (Ij)) will remain 0 for all future requests.
Now observe that if = 0, every out(T (Ij)) = 0, and so will remain zero also in the future.

Next we claim that no request can increase , and that if > 0, there is a request in that will decrease the potential function. Consider T (Ij) and any request (u; v). If (u; v) 2 Ij then out(T (Ij)) does not change. If u; v 2 Ti for i = 1; 2 or 3, then again out(T (Ij)) does not change. If u 2 T1 or u 2 T2 and v 2 T3, then the lowest common ancestor of u; v is in T3 and after splaying u and v, out(T (Ij)) will decrease by one. Therefore can only decrease. Now if > 0, there is some j for which out(T (Ij)) > 0. Take the request (u; v) that is a witness; after this request out(T (Ij)) will decrease by one, so will decrease. So overall given a cluster , will decrease to zero. To conclude the theorem we show that if = 0 then for any communication pair (u; v) 2 Ij, for any j 2 f1; : : : ; kg, u and v are connected by a local path which only includes nodes from Ij. Assume by contradiction that this is not the case and there is a j and (u; v) 2 Ij (assume w.l.o.g. that u < v) s.t. the path between them visits a node w 2 Ii. Also assume w.l.o.g. that w < u. Now it must be the case that either T (rj) T (ri) or T (ri) T (rj). Let's assume w.l.o.g. that T (rj) T (ri) so it must be that case that rj < w. Since it is a maximal cluster scenario there must be a path in from a node x 2 Ii, x w to a node y 2 Ii, y ri (otherwise the cluster Ii can be divided into two clusters).

B. Non-crossing Matching Scenario

SplayNet sometimes achieves an optimal performance by converging to the optimal static network implied by R( ). We have already encountered an example where SplayNets are asymptotically optimal, namely if describes a product distribution (cf Corollary 1). In the following, we will examine other optimal scenarios in more detail.

For a request (u; v) in , let I(u;v) denote the interval [min(u; v); max(u; v)].

Definition 4 (Non-Crossing Matching Scenario). In a non-crossing matching scenario, it holds for the communication pattern that for any two pairs (u1; v1) and (u2; v2) in , either:
1) I(u1;v1) (I(u2;v2) or I(u2;v2) (I(u1;v1), or

2) I(u1;v1) \ I(u2;v2)

It follows from the definition that the request graph R( ) must describe a matching, and for each request (u; v) there are no other requests that enter or leave (i.e., cross) the

interval I(u;v). provide aIHn example of a non-crossing matching scenario. If R( ) describes a non-crossing matching scenario, SplayNet will converge to an optimal solution. Formally:

Theorem 9. In a non-crossing matching scenario , SplayNet will eventually converge to a tree where any communication pair fu; vg 2 is adjacent.

Proof: To any request pair pi = (ui; vi) we assign a level L(pi) depending on the number of pairs pj = (uj; vj) it is nested in, i.e., L(pi) = jfpj : Ipi (Ipj gj. Pairs with level 0 are the outmost pairs, pairs at level 1 are nested in a single other pair and so on. We will prove by induction on the level that all pairs become (and stay) adjacent. Our induction hypothesis is that pairs at level i coverage and we will show that once that happens, pairs at level i + 1 will converge next. First we show that pairs at level 0 converge. Notice that pairs do not intersect (i.e., cross), so the pairs at level 0 represent a clustered scenario. By Theorem 8, after the convergence of clusters, requests within each interval (of pair at level 0) will remain only within the interval. After the cluster convergence, we claim that after a pair pi = (ui; vi) of level 0 communicated, it will stay adjacent forever. To see this, consider any other pair pj = (uj; vj). If pj is outside the cluster of pi we are done. Let pj be ins de the interval of pi, i.e., I(uj ;vj) (I(ui;vi). By the definition of SplayNet, after a request (ui; vi), vi is the right (resp. left) child of ui if ui < vi (resp. u1 > v1). We will show that this implies that the nodes uj; vj must be both in the same subtree, and can hence not change the adjacency relationship of ui and vi anymore. The claim follows by case distinction: (1) If ui < vi, the left

C. Multicast Scenario

To give one more example where SplayNet is optimal, we consider the multicast tree scenario. For this scenario, we may assume that an overlay network of a binary search tree structure is constructed to facilitate in-network duplication. The same tree may be used by many multicast sources and different receivers, hence the local (hop-by-hop) source and destination pairs (i.e., communication requests) are the endpoints of the tree edges

IV.CONCLUSION

we regard our work as a first step towards the layout of novel distributed datastructures and networks which adapt dy-namically to the demand. for you to cognizance on the fundamental tradeoff among advantage and cost of self-changes, we pur-posefully offered our model in a widespread and summary form, and many extra and application-particular elements want to be addressed before our technique can be examined inside the wild. the primary theoretical simplification made in this paper regards the limit to the tree topology, and the generalization to extra complex and redundant networks is an open query. furthermore, further to [22], we have focused at the amortized expenses of splaynets, and an thrilling path for future research regards the examine of the conceivable competitive ratio under arbitrary communique styles.

V. REFERENCES

[1]. B. Allen and I. Munro. Self-organizing binary search trees. J. ACM, 25:526-535, 1978.

[2]. J. Aspnes and G. Shah. Skip graphs. ACM Transactions on Algorithms (TALG), 3(4):37-es, 2007.

[3]. C. Avin, M. Borokhovich, and S. Schmid. Obst: A self-adjusting peer-to-peer overlay based on multiple bsts. In Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on, pages 1-5. IEEE, 2013.

[4]. C. Avin, M. Borokhovich, and S. Schmid. Obst: A self-adjusting peer-to-peer overlay based on multiple bsts. In Proc. 13th IEEE International Conference on Peer-to-Peer Computing (P2P), September 2013.

[5]. C. Avin, B. Haeupler, Z. Lotker, C. Scheideler, and S. Schmid. Locally self-adjusting tree networks. In Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on, pages 395-406. IEEE, 2013.

[6]. C. Avin, B. Haeupler, Z. Lotker, C. Scheideler, and S. Schmid. Locally self-adjusting tree networks. In Proc. 27th IEEE International Parallel and Distributed Processing Symposium (IPDPS), May 2013.

[7]. E. D. Demaine, D. Harmon, J. Iacono, and M. Patrascu. Dynamic optimality - almost. In Proc. 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pages 484-490, 2004.

[8]. J. D´ıaz, J. Petit, and M. Serna. A survey of graph layout problems. ACM Comput. Surv., 34(3):313-356, 2002.

[9]. J. D´ıaz, J. Petit, and M. Serna. A survey of graph layout problems. ACM Comput. Surv., 34(3):313-356, 2002.

[10]. H. Farvaresh and M. Sepehri. A branch and bound algorithm for bi-level discrete network

design problem. Networks and Spatial Economics, 13(1):67-106, 2013.
[11]. U. Feige and J. Lee. An improved approximation ratio for the minimum linear arrangement problem. Information Processing Letters, 101(1):26- 29, 2007.
[12]. L. H. Harper. Optimal assignment of numbers to vertices. J. SIAM, (12):131-135, 1964.
[13]. N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman. Skipnet: A scalable overlay network with practical locality properties. In Proc. 4th Conference on USENIX Symposium on Internet Technologies and Systems (USITS), 2003.
[14]. S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. Bulletin of the American Mathematical Society, 43(4):439- 562, 2006.
[15]. D. Huffman. A method for the construction of minimum-redundancy codes. Proceedings of the IRE, 40(9):1098-1101, 1952.
[16]. D. Knuth. Optimum binary search trees. Acta informatica, 1(1):14-25, 1971.
[17]. K. Mehlhorn. Nearly optimal binary search trees. Acta Informatica, 5(4):287-295, 1975.
[18]. G. Mitchison and R. Durbin. Optimal numberings of an n n array. SIAM J. Algebraic Discrete Methods, 7(4):571-582, 1986.
[19]. R. Ravi, A. Agrawal, and P. N. Klein. Ordering problems approximated: Single-processor scheduling and interval graph completion. In Proc. International Colloquium on Automata, Languages and Programming (ICALP), 1991.
[20]. S. Schmid, C. Avin, C. Scheideler, and B. Haeupler. Brief announce-ment: Splaynets (towards self-adjusting distributed data structures). In Proc. 26th International Symposium on Distributed Computing (DISC), 2012.
[21]. A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. Inf. Comput., 82(1):93-133, 1989.
[22]. D. Sleator and R. Tarjan. Self-adjusting binary search trees. Journal of the ACM (JACM), 32(3):652-686, 1985.
[23]. C. C. Wang, J. Derryberry, and D. D. Sleator. O(log log n)-competitive dynamic binary search trees. In Proc. 17th Annual ACM-SIAM Sympo-sium on Discrete Algorithm (SODA), pages 374-383, 2006.