Algorithmic Aspects of Access Networks Design in B3G/4G Cellular Networks

David Amzallag*, Joseph (Seffi) Naor†, Danny Raz*

*Computer Science Department
Technion, Haifa 32000, Israel
{amzallag,danny}@cs.technion.ac.il

†Microsoft Research
One Microsoft Way, Redmond, WA 98052
naor@cs.technion.ac.il

Abstract— **The forthcoming 4G cellular systems will provide broadband wireless access to a variety of advanced data and voice services. In order to do that, these networks will have a significantly larger number of base stations and a much higher bandwidth demand from their radio access networks. This will motivate operators to replace the commonly used star based architecture, in which an RNC is connected to a set of base stations via direct links, with a more complex tree structure, in which a base station can be connected to an RNC via other base stations.**

In this paper we address algorithmic aspects of this challenging design problem, in which tree-topology is used to connect base stations and RNCs. We formulate the problem as an optimization problem and prove that it is NP-hard to approximate it in the general case. For the metric case, however, we develop an $O(\log n)$-approximation algorithm. We then study the performance of this algorithm and several other heuristics in practical scenarios. Our results indicate that a combination of a certain greedy heuristic and the proven approximation algorithm, generates a solution that produces close to optimal results in practical scenarios and can be efficiently computed for sufficiently large network sizes.

I. Introduction

The forthcoming 4G cellular networks are expected to provide a wide variety of new services, from high-quality voice and high-definition video to high-data-rate wireless channels. Although the detailed structure of 4G systems is as of yet not well defined, there is a clear consensus regarding some of the important aspects of the technologies to be implemented in these systems[1]. Assuming that video and data communications will be the main applications, 4G systems are planned to provide even higher transmission rates and larger capacity (i.e., both in term of the number of users and the traffic volume) than current 3G (IMT-2000 based) systems. Most likely, 4G systems will be designed to offer bit rates of 100 Mbit/s (peak rate in mobile environment) to 1 Gbit/s (fixed indoors) with a 5 MHz frequency bandwidth. The system capacity is expected to be at least 10 times larger than the current 3G system. In addition, these objectives should be met together with a drastic reduction in the cost (1/10 to 1/100 per bit) [1]. Such high frequencies yield a very strong signal degradation and suffer from significant diffraction resulting from small obstacles, hence forcing the reduction of cell size (in order to decrease the amount of degradation and to increase the degree of coverage), resulting in a significantly larger number of cells compared to previous generations.

Third generation network elements are functionally grouped into the Radio Access Network (RAN, or UTRAN, in UMTS systems) that handles all radio-related functionality; Core Network (CN), which is responsible for switching and routing calls and data connections to external networks, and the User Equipment (UE) that uses air interface to communicate with the base stations (see [2, Chapter 5]).

RAN architecture in current 3G systems consists of one or more Radio Network Subsystems (RNSs) as depicted in Figure 1. Each RNS is a sub-network within the RAN, comprising of one Radio Network Controller (RNC) and one or more base stations (BSs)[2]. The RNC owns and controls radio resources in its RNS; it is the service access point for all services provided by the RAN to the **CN** over the **Iu** interface. It then communicated with the BSs in its RNS (over the **Iub** interface) which are in charge of the communication

† On leave from the Computer Science Department, Technion, Haifa 32000, Israel.

[1] See International Telecommunication Union (ITU) Web Site at http://www.itu.int/home/index.html.

[2] Note that the equivalent 3GPP term for a base station is *Node B*, where RNC is the 3GPP2 terminology for the GSM Base Station Controller (BSC).

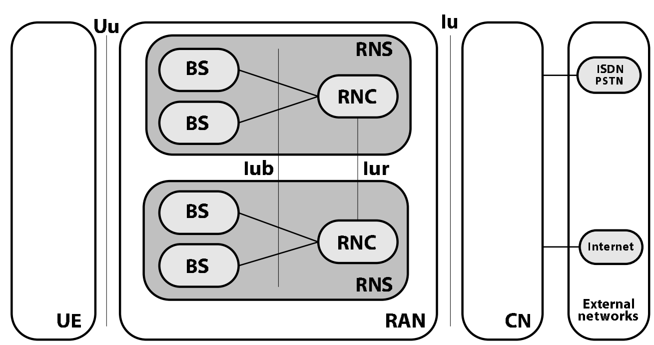

Fig. 1. RAN architecture in current 3G systems

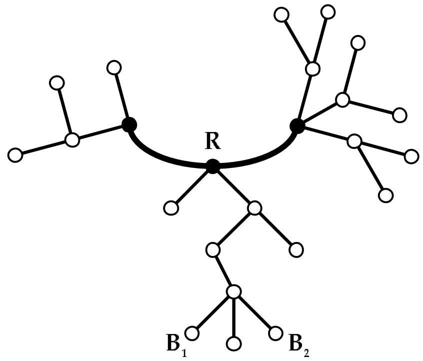

Fig. 2. Tree-topology access network (RNCs nodes are colored in black while base-stations are whites)

to the **UE** over the WCDMA radio interface. RNCs may be connected to each other via an **Iur** interface (i.e., the open interface that allows soft handover between RNCs from different manufactures).

RAN in 4G systems is expected to be considerably different from current RANs. First, as mentioned before, in 4G systems the cell size is expected to be smaller than in 3G systems [3]. Therefore, the 4G RAN will contain more base stations. A careful design must be used in order to handle this large number of base stations without a significant increase in the number of RNCs (i.e., *RNC dimensioning*). Second, the larger number of base stations (resulting in more frequent handover in the system) and the expected higher bit rate will result in a heavier load on the links between the RNCs and base stations. Finally, these changes should be made in a cost-efficient way.

In the current traditional star topology radio access networks (e.g., in UMTS systems), all base stations are directly connected to RNCs. When a tree topology is deployed (rather than a star), a base station is allowed to be connected to another base station rather than its RNC. However, base stations have no routing capabilities and they simply forward all received data towards their corresponding RNC and from the RNC to the corresponding base station. For example, in Figure 2, if a mobile user inside the coverage area of base station B_1 wishes to communicate with a mobile user inside the coverage area of base station B_2, their data traffic will be sent through RNC R. Therefore, the link connecting the parent of B_1 and B_2 to its parent, on the way to the RNC, must be capable of carrying large enough amount of traffic to handle both its own traffic as well as the traffic originating from its three children. Thus, the "heaviest" links are designed to be those that are connected directly to the RNC. These links must be able to handle all the traffic in their subtree. In the case of tree topologies, the constraints stem from technical limitations of the equipment. A base station cannot be connected to too many other base stations without creating a significant traffic reduction. Therefore, we assume that every base station and every RNC can only have a limited number of allowed connections. The planning problem is then to design the best possible tree taking these limitations as well as the cost of establishing the links into account. As we show in this paper, this is a computationally difficult task.

In this paper we rigourously study the problem of designing tree-topology based access networks for 4G cellular systems, and describe the theoretical as well as the practical aspects of the solutions to this problem.

A. Definitions and background

Consider a set $I = \{1, 2, \ldots, n\}$ of base stations and a set $J = \{1, 2, \ldots, m\}$ of RNCs. A symmetric *connection cost* $w(i_1, i_2)$ is associated with every pair of base stations i_1 and i_2; Another given cost is $w(i, j)$ representing the cost of connecting base station i to RNC j, for every $i \in I, j \in J$. A tree-topology based access network is designed as a forest, such that each of its trees is rooted at an RNC node and contains the base stations that are under the responsibility of this RNC. In addition, every node $u \in I \cup J$ is allowed to connect to no more than $b(u)$ neighbors in the forest, for some $b(u) \geq 1$.

Given a spanning tree T of a subset $I_T \subseteq I$ of base stations, rooted at RNC r, we define the *routing cost*, $d_T(r, i)$, between the RNC r and a base station $i \in I_T$, as the sum of the costs along the unique path between them in the tree T. The routing cost of the tree itself is defined by $\sum_{i \in I_T} d_T(r, i)$. Our goal is to design a tree-topology based access network with a minimum total cost. Note that the closer a connection is to the root of the tree, the higher is its contribution to the routing cost of the tree.

An important observation is that the problem of designing access networks that comprise of multiple trees (i.e., multiple RNS radio access network) is reducible to the problem of designing a network comprising of a *single* tree (i.e., a single RNS). This means that the problem of dividing base stations among the possible RNCs is directly-solvable via our model. We represent the input to the problem as a complete graph $G = (V, E)$, such that $V = I \cup J \cup \{\hat{r}\}$, where $\hat{r}$ is a "special" vertex to be defined later. There is an edge between every pair of vertices in I weighted by their corresponding connection cost. In addition, each RNC vertex in J is connected to all the vertices in I by an edge of weight equal to the connection cost between the corresponding base station and the RNC. Finally, the vertex $\hat{r}$ is connected to all the vertices of J by an edge of zero weight. All other edges of G are assumed to have an infinite weight. The degree constraints of vertices of $I \cup J$ are equal to the corresponding degree constraints given for base stations and RNCs. The degree constraint of $\hat{r}$, $b(\hat{r})$, is defined to be $|J| = m$. Hence, our access network design consists not only of the association of base stations to RNCs, but also of selecting a subset of RNCs to be deployed in the network. A consequence of the above reduction is that the problem of finding a forest with multiple RNCs is equivalent to the problem of finding a tree for a single RNC. We will therefore limit our attention to the case where $|J| = 1$.

Notice that when setting, in the above reduction, the value of $b(\hat{r})$ to be k, $k \leq m$, the model is extended to select only k out of the m RNCs to be installed on the network.

We define the *bounded-degree minimum routing cost spanning tree problem* (BDRT) as the problem of finding a minimum routing cost spanning tree, rooted at a given root r, that meets the degree constraints $b(v)$, for all $v \in V$.

Approximation algorithms and heuristics play a major role in our paper. A γ-approximation algorithm is a polynomial-time algorithm that always finds a feasible solution for which the value of the objective function is within a proved factor of γ of the optimal solution. Heuristics will be described and analyzed in comparison with the worst-case behavior of approximation algorithms, in order to design a good practical solution to the access network design problem.

B. Our contribution

In this paper we study the algorithmic aspects of access network design in 4G cellular systems. We investigate the BDRT problem as a general model that captures several aspects of radio access networks design.

We show that it is NP-hard to approximate the general problem (Section III), and then present an approximation algorithm for the case where edge weights satisfy triangle inequality. To the best of our knowledge this is the first approximation algorithm for this problem. We show, in Section III-A, that the case of $b(v) \geq 3$, for every $v \in V$, is approximable within a factor of $O(\log n)$. Four more heuristic algorithms for BDRT are presented in Section III-B and compared to the approximation algorithm. A generalization of BDRT is described in Section IV. In this generalization the traffic requirement of the base stations is also taken into account in the cost model. In this way, the model can deal with large variance in the traffic loads among the base stations, as expected in 4G networks.

In Section V we describe our simulations. We study the performance and the quality of the solutions of each of the five studied algorithms on instances of both BDRT and its generalization. Finally, we conclude that the combination of the approximation algorithm and one of the heuristics achieves a proven performance guarantee of $O(\log n)$ in the worst-case, together with a close to optimum solutions in practice, both for BDRT and its generalization.

II. Related work

Several non-star topologies for radio access networks have been proposed in the last few years [3]–[9].

Ring topologies have been proposed in [3], [4], [5]. The advantage of such a topology is, of course, its reliability; on the other hand, the delay on the path from a base station to the RNC may be significant. The authors of [3] present an $O(n^3)$-time algorithm for solving the corresponding design problem. However, such an approach is unlikely to be optimal since this problem, as modeled in [3], is the well-known traveling salesman problem (TSP) which is NP-hard to approximate in general. The algorithm presented in [3] is a "nearest neighbor" algorithm that guarantees a solution that is within a factor of $O(\log n)$ of the optimal solution of the problem [10, Chapter 3.2] only when the cost on the links satisfy triangle inequality.

Tree-topology radio access network design in UMTS cellular networks has been studied in [6]–[9]. The multiple-RNS design problem is considered in [6], [7] and [8] deal with the single-RNS version of the problem, and [9] proposes an approach to solve both problems. Note that (as we indicated earlier) both problems are algorithmically equivalent. Since all these papers use the simulated annealing technique, the quality of the solutions depends on the duration of the execution.

Fault-tolerance in access networks is considered in [7] and [8]. In both papers the cost model is very similar to the one adopted here, and the cost function can handle both wired (e.g., leased-line, fiber, coax) and wireless (microwave) interconnections. However, the constraints of the models used in these papers are different from ours. It is assumed that each base station specifies its level in the tree and a uniform out-degree bound is used for all nodes.

Tree-topology-based design for access networks has also been discussed in [9]. In this work the authors proposed a Simulated Annealing based algorithms compared with a lower bound for the single-tree version of the problem. Using this Lagrangian relaxation-based lower bound, a branch-and-bound method is proposed to compute the theoretical optimal solution to this problem for networks of small sizes. The cost model described in [9] has two important characterizations. The first one is that the cost function also depends on the level in which the base station is located in the tree. Secondly, its objective function is defined as the *sum of the connection costs of the tree* (rather than the sum of the cost of the paths between the RNC and each of the base stations, as studied in this paper). Since connections closer to the RNC aggregate more traffic, the cost model should capture the flow of the traffic throughout the tree and therefore the cost model of [9] does not reflect this behavior of traffic. Moreover, optimal solutions for these two different objectives can be far from each other by a factor of $\Theta(n)$ (as in the case of unit-weight complete graph on n vertices, all have a degree bound of two). From an algorithmically point of view, if the objective function is a minimization the total sum of connection costs, the problem can be studied under the framework of the well-known bounded-degree minimum spanning tree problem (e.g., [11], [12], [13]).

Several problems can be viewed as a generalization of the BDRT problem, where there are no limitations on the degree bounds. The *minimum routing cost spanning tree problem*, asks for a spanning tree where the sum of the routing cost is taken over all pairs of nodes and not only from the root to all other nodes.

Finding a spanning tree of minimum routing cost in general weighted undirected graphs is NP-hard [14]. (Notice that the "single-source" version of the problem *without* the degree constraints is polynomial-time solvable and can be seen as the single-source shortest path problem.) Wu et al. [15] showed that finding a minimum routing cost tree in a general weighted graph G is equivalent to solving the same problem on a complete graph in which edge weights satisfy triangle inequality. This result implies that the minimum routing cost spanning tree problem in a metric space is also NP-hard. The best result known today for this problem is a polynomial-time approximation scheme (PTAS) due to Wu et al. [15]. In this paper, the authors show that this problem has an $(1+\epsilon)$-approximate solution in time $O(n^{2\lceil \frac{2}{\epsilon} \rceil - 2})$.

Hu [16] introduced a generalization of the minimum routing cost spanning tree problem that he called *optimum communication spanning trees*. In this problem, in addition to the weight on edges, a *requirement value* $r(v_i, v_j)$ is specified for every pair of vertices v_i, v_j. The communication cost between a pair of vertices in a given spanning tree is the cost of the path between them in the tree multiplied by their requirement $r(v_i, v_j)$. The communication cost of the tree is the sum of all pairwise communication costs. Thus the routing cost is a special case of the communication cost when all the requirements are one.

Several $O(\log^2 n)$-approximation algorithms for the metric case of the minimum communication cost spanning tree problem are presented in [15] and [17]. This problem is shown to be MAX SNP-hard [17], implying that a PTAS can not be achieved unless P = NP.

In Section IV we present an extension to our model which uses similar flow requirements. This extension can be viewed as a single source, bounded-degree version of the minimum communication cost spanning tree problem.

III. The Bounded-Degree Minimum Routing Cost Spanning Tree Problem

The important goal of efficient planning of access networks is beyond our reach since this problem is NP-hard, as we mentioned before. In this paper we use two approaches for coping with hard optimization problem: approximation algorithms and heuristics. Instead of finding an optimal solution, an approximation algorithm settles for a near, yet provable, optimal solution. Heuristic algorithms, on the other hand, work well on many instances, though not necessarily on all instances.

Unfortunately, it is not even possible to design polynomial-time approximation algorithm for BDRT, unless P = NP. Such $\gamma(n)$-approximation algorithm will solve the Hamiltonian path problem, for any computable function $\gamma(n)$. Given a graph G on n vertices we can transform it to an instance G' of BDRT such that G has a Hamiltonian path connecting r and a vertex v if and only if G' has an optimal solution for BDRT of value $\xi(n) = 1+2+\ldots+(n-1) = \frac{1}{2}n(n-1)$. This transformation can be done by assigning a unit weight to the edges of G, and a weight of $\gamma(n)\cdot\xi(n)$ to non-edges so as to obtain a complete graph G'. Degree bounds for r and v are 1 and for all other vertices in G' degree bounds are taken to be two, and r is fixed to be the root. We can now state the following theorem[3].

Theorem 1: Unless P = NP, there is no polynomial-time approximation algorithm for BDRT.

When considering planning of access networks in real-life applications, the weights on edges in BDRT typically satisfy triangle inequality. Notice that in order to obtain the above inapproximability result we had to use very large edge weights that indeed violate triangle inequality. If we restrict ourselves to metric instances of BDRT in which edge weights satisfy triangle inequality, the problem remains NP-hard (even for $b(v) \geq 3$, for all $v \in V$), but as we shall see, it is no longer impossible to approximate the optimal solution.

[3]NP-hardness proofs are omitted for the lack of space.

Algorithm $\mathcal{A}'$. BDRT-APPROXIMATION

Construct a shortest-path tree, with root r, on the input complete (metric) graph; renumber the vertices so that v_i, $i = 1, 2, \ldots, n$, is the ith closest vertex to the root.
Set v_1 to be the root of the tree T; $i \leftarrow 1$.
while T is not a spanning tree **do**
    Pick the $\xi(v_i)$ vertices of least indices and assign them, from the left most child to the rightmost, as the children of vertex v_i, where $\xi(v_i) = b(v_i)$, for $i = 1$, and $\xi(v_i) = b(v_i) - 1$ otherwise.
    $i \leftarrow i + 1$
end while

Theorem 2: The metric BDRT with $b = b(v) \geq 3$, for every $v \in V$, is NP-hard.

A. An $O(\log n)$-approximation algorithm

Next we present an $O(\log n)$-approximation algorithm for the metric version of BDRT, where the degree constraints for all vertices are assumed to be greater than or equal to 3. The lower bound we use is the cost of the shortest-path tree, rooted at r, of the input graph G. Nevertheless, the total cost of a shortest-path tree is not unique, in general. Graphs can have several shortest-path trees (rooted at the same vertex) of different costs yet each preserves the shortest-distance between the root and each of the vertices. However, when edge weights satisfy triangle inequality, the shortest-path tree is *unique*, drawn as a star centered at the root vertex. Obviously, this solution is a lower bound for any instance of BDRT, since every edge of this star is counted only once in the total cost.

Our approximation algorithm, called Algorithm $\mathcal{A}'$, has two phases. First, the shortest-path tree, rooted at r, is constructed, and the vertices are renumbered by their distance from the root on the shortest-path tree (v_1 is the root itself, v_2 is the closest vertex to the root on the input graph, v_3 is the second closest vertex, and so on).

In the second phase, the output tree is constructed meeting the degree constraints of all vertices. The algorithm starts at the root vertex r, picks the $b(r)$ vertices of least index and assigns them as its children, from the left most child to the rightmost one. Moving to the next level in the constructed tree, for every vertex v, the algorithm picks the $b(v) - 1$ unpicked vertices of least index and assigns them as its children, from the left most child to the rightmost one. This process is terminated when the tree contains all the vertices of G.

Before bounding the cost of the tree constructed by Algorithm $\mathcal{A}'$, let us consider a concrete example. Let $b(v) = 3$

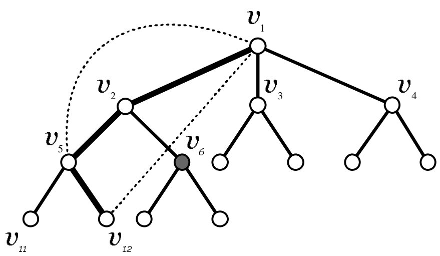

Fig. 3. Approximating BDRT

for every vertex v of G, meaning that the output tree is of the largest height, h. Since there is one vertex in the 0th level of this tree, three vertices in the first level, $3 \cdot 2$ vertices in the second level, and $3 \cdot 2^{\ell-1}$ vertices in the ℓth-level, the height h is $\lceil \log \frac{n+2}{3} \rceil \leq \log n$. Now, consider for example, vertex v_{12} in Figure 3, and let us bound its contribution to the total cost of the solution.

$$
\begin{aligned}
d(v_{12}) &= w(v_1, v_2) + w(v_2, v_5) + w(v_5, v_{12}) \\
&\leq w(v_1, v_2) + \big(w(v_1, v_2) + w(v_1, v_5)\big) \qquad (1) \\
&+ \big(w(v_1, v_5) + w(v_1, v_{12})\big) \\
&\leq w(v_1, v_{12}) + 2\big(w(v_1, v_2) + w(v_1, v_5)\big) \\
&\leq w(v_1, v_{12}) + 2\log n \cdot w(v_1, v_6) \qquad (2)
\end{aligned}
$$

Using triangle inequality we can bound this contribution by (1). Finally, since v_6 is the last internal vertex in the tree (colored in grey in Figure 3), $w(v_1, v_6) \geq w(v_1, v_j)$ for every $j \leq 6$, hence the heaviest path connecting the root to any vertex has cost at most $2\log n \cdot w(v_1, v_6)$ (by (2)).

Theorem 3: Algorithm $\mathcal{A}'$ is an $O(\log n)$-approximation algorithm for the metric BDRT with $b(v) \geq 3$, for every $v \in V$.

Proof: Let T be the tree constructed by the algorithm, and let v_k be the last internal vertex of T, that is, vertices $v_{k+1}, \ldots, v_n$ are all leaves. Recall that the routing cost, $d(v_i)$, of any vertex v_i is the sum of the weights along the unique path to the root v_1. In general, the routing cost can be computed as followed

$$
\begin{aligned}
d(v_i) &= \sum_{j=0}^{\lfloor \log i \rfloor - 1} w(v_{\lfloor \frac{i}{2^{j+1}} \rfloor}, v_{\lfloor \frac{i}{2^j} \rfloor}) \qquad (3) \\
&\leq w(v_1, v_i) + 2 \sum_{j=0}^{\lfloor \log i \rfloor - 1} w(v_1, v_{2^j}) \qquad (4) \\
&\leq w(v_1, v_i) + 2\log n \cdot w(v_1, v_k), \qquad (5)
\end{aligned}
$$

where (4) is the result of triangle-inequality, (5) follows since $w(v_1, v_k) \geq w(v_1, v_j)$ for every $j \leq k$, and $\log n$ is an upper bound on the largest height T can have.

Since the cost of the shortest-path tree of G (with v_1 as its root), as computed in the first step of the algorithm, is a lower bound on the cost of the optimal solution, OPT, we have,

$$\sum_{i=1}^{n} w(v_1, v_i) \leq \text{OPT}. \qquad (6)$$

However, since $b(v) = 3$, T has $\lceil \frac{n}{2} \rceil$ leaves and the shortest path from each leaf is no less than $w(v_1, v_k)$, we have

$$\left\lceil \frac{n}{2} \right\rceil w(v_1, v_k) \leq \text{OPT}. \qquad (7)$$

Finally, combining the above discussion gives

$$
\begin{aligned}
\sum_{i=1}^{n} d(v_i) &\leq \sum_{i=1}^{n} \big(w(v_1, v_i) + 2\log n \cdot w(v_1, v_k)\big) \qquad (8) \\
&\leq \text{OPT} + 4\log n \cdot \text{OPT} \qquad (9) \\
&= O(\log n) \cdot \text{OPT}. \qquad (10)
\end{aligned}
$$

■

When designing approximation algorithms one might be interested in improving the performance guarantee of the suggested algorithm. We next show that the $O(\log n)$-factor of Algorithm $\mathcal{A}'$ is tight, as follows by the next example.

Consider a complete graph corresponding to a set of points on the real line (Figure 4). The points are divided into groups $\{G_i\}_{i \geq 0}$ as follows: G_0 contains only the origin, G_1 contains 2 points at distance 1 and $1+\epsilon$ from the origin, on the right-side. G_2 contains 4 points at distances $1+2\epsilon, 1+3\epsilon, 1+4\epsilon$, and $1+5\epsilon$ from the origin, on the left-side. In general, G_i contains 2^i points at distances $\{1+(2^i-2)\epsilon, \ldots, 1+(2^{i+1}-3)\epsilon\}$ from the origin, taking from the left (right) side if i is odd (even).

Since distances are computed on the real line, the weights clearly satisfy triangle inequality, and the costs of a shortest-path on the constructed tree are preserved and not affected by the number of edges on the path. Taking $\epsilon = 1/n$ ensures that the solution obtained by the algorithm is of cost $\Theta(n \log n)$ (Figure 4(b)). However, the optimal solution of BDRT for this instance is of cost $\Theta(2n)$, as shown in Figure 4(a). This shows that the performance guarantee of Algorithm $\mathcal{A}'$ is tight.

B. Heuristic algorithms

In this section we concentrate on a family of greedy heuristics for solving BDRT. This is perhaps the most common approach taken in practice. Since heuristics work well on many instances, though not on all of them, they cannot be rigourously analyzed. However, the heuristics presented here are based on insights obtained from our $O(\log n)$-approximation algorithm studied in the previous section.

The following algorithms work in a similar way: They start at the root vertex r, pick the *best* $b(r)$ vertices as its children, from the left most child to the rightmost one, and then move to the next level in the constructed tree. For every vertex v ($\neq r$), the algorithm picks the best $b(v) - 1$ unpicked vertices and assign them as its children, until spanning all the vertices of G. By "best" we mean the most preferred vertices according to a given criterion. We note that ties are broken arbitrarily.

The first greedy algorithm, called GA_1, picks the vertices v in an increasing order of their ratio $d_T(r, v)/b(v)$, where $d_T(r, v)$ is the cost of the shortest-path connecting v to r in the constructed tree T. Although this algorithm might seem as the most natural one, its performance guarantee can be bad as $\Omega(n)$ as described in the following example.

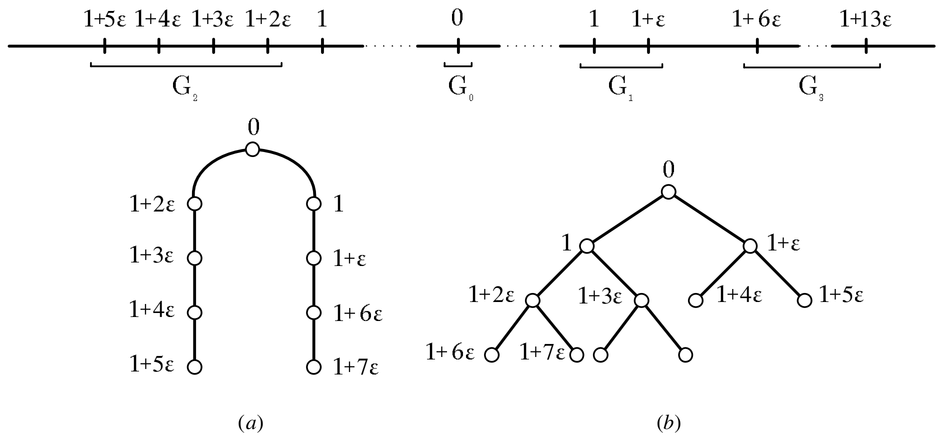

Fig. 4. A tight example for the $O(\log n)$-approximation algorithm

Observation 4: There exist instances of the metric version of BDRT for which the cost of the solution, produced by GA_1, is within a factor of $\Omega(n)$ of the optimum, where $b(v) \geq 3$ for all vertices v.

Proof: Consider a set of $n+4$ nodes given on the real line with the origin as the root (Figure 5). In this example, a set of nodes $V = \{v_1, \ldots, v_n\}$ is located at distance of 3 from the origin and a set $U = \{u_1, u_2, u_3\}$ is located at distance $\frac{n}{2}$. Each node $v \in V \cup \{0\}$ has a degree constraint of $b(v) = 3$ while nodes in U have degree constraints of n.

It is not hard to verify that the optimal solution for this instance has a cost of $O(n)$ while GA_1 builds a tree of cost $O(n^2)$. ■

The second algorithm, called GA_2, is very similar to GA_1 but here vertices are picked in an increasing order of their ratio $d_T(r,v)/(b(v))^2$. This algorithm emphasizes the importance of the degree bound in particular for those vertices with the same $d_T(r,v)/b(v)$-ratio.

The third algorithm, GA_3, picks the vertices v in an increasing order of their ratio $w(u,v)/b(v)$. Given a partially constructed tree, we denote u as the last vertex that is already picked and joined to the tree. So, we are now ready to assign u up to $b(u)-1$ children. This algorithm, as opposed to GA_1, selects the vertices v that are most closely to u in G having the largest degree constraints $b(v)$. Notice that both GA_2 and GA_3 also have the same worse performance guarantee; this can be shown in a similar way as done for GA_1 (Observation 4).

We label our approximation algorithm, $\mathcal{A}'$, as GA_4 and denote the following similar version of it as GA_5. This algorithm picks the vertices v_i in an increasing order of their ratio $v_i/b(v_i)$, where $v_i, i = 1, \ldots, n$, is the ith vertex closest to the root in the input graph G. In other words, $\{v_i\}_{i=1}^n$ are the vertices ordered as in the first phase of Algorithm $\mathcal{A}'$.

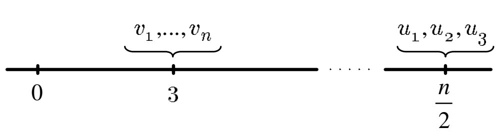

Fig. 5. A bad example for GA_1

A sample execution

A sample execution of the above five algorithms for BDRT is described in Figure 6. The input is a complete graph G on 7 vertices, given by the adjacency matrix below. The root is chosen to be vertex a and the degree bounds are defined as $b(b) = b(g) = 2$, while other vertices have a degree bound of 3.

	a	b	c	d	e	f	g
a	0	3	3	7	6	3	3
b	3	0	2	5	4	2	1
c	3	2	0	5	5	1	3
d	7	5	5	0	2	5	5
e	6	4	5	2	0	4	4
f	3	2	1	5	4	0	3
g	3	1	3	5	4	3	0

In this example, the solutions produced by algorithms GA_1, GA_2, and GA_5 have a cost of **31** (Parts (a), (b), and (e), respectively), algorithm GA_3 achieves a cost of **30** (Part (c)), and algorithm GA_4, which is the approximation algorithm, produces a solution of cost **29** (Part (d)). Notice that the theoretical lower bound in this case, namely the cost of the shortest-path tree of G, rooted at vertex a, is **25** (Part (f)).

IV. Extensions

In this section we extend our model to be more sensitive to traffic requirements of the different base stations. Notice that 4G networks will use a variety of technologies and it is very likely that some cells would support a large area and a large density of traffic, while other cells may be designed for very small traffic densities. Given an instance of BDRT, we assume now that each base station $i \in I$ has traffic requirement t_i, representing its expected traffic load. Since a base station is connected to the RNC via a path of base stations, its traffic is aggregated along that path. The routing cost of the path connecting base station i to the RNC r in the tree T is defined to be $t_i \cdot d_T(r,i)$. The *generalized bounded-degree minimum*

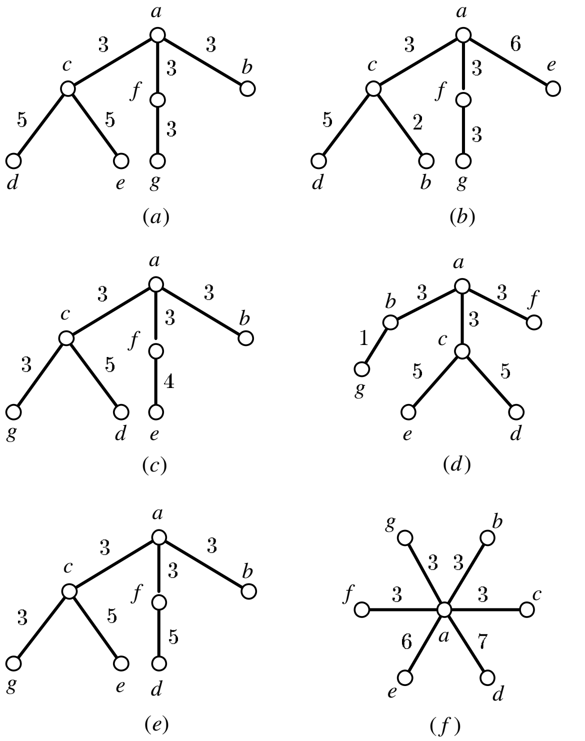

Fig. 6. Sample execution of the five algorithms. Solutions produced by GA_1, GA_2, and GA_5 have a cost of **31** (Parts (a), (b), and (e), respectively), GA_3 achieve a solution of cost of **30** (Part (c)), and GA_4, which is the approximation algorithm, produce a solution of cost **29** (Part (d)). The cost of the shortest-path tree of G, rooted at vertex a, is **25** (Part (f)).

routing cost spanning tree problem (GBDRT) is to find a spanning tree, rooted at a given root r, that meets the degree constraints $b(v)$, for all $v \in V$ and the cost

$$\sum_{i \in I} t_i \cdot d_T(r, i) \tag{11}$$

is minimized.

Obviously, BDRT is a special case of GBDRT, by taking $t_i = 1$, for all $i \in I$. Moreover, our approximation algorithm, $\mathcal{A}'$, when applied to an instance of GBDRT, gives the same performance guarantee. Notice that the lower bound in this case is the star, centered at the root vertex, computed via (11). Hence we have the following:

Theorem 5: The metric generalized bounded-degree minimum routing cost spanning tree problem (GBDRT) with $b(v) \geq 3$, for all $v \in V$, is approximable within a factor of $O(\log n)$ of the optimum.

A similar generalization can also be applied to the heuristic algorithms described in Section III-B. In this case the greedy algorithms decrease their criteria for a selection of vertex v by a factor of t_v, where v is a vertex corresponds to a base station v and t_v is its traffic requirement.

V. Simulation results

In Section III, we defined five algorithmic solutions for both BDRT and its generalization GBDRT. The first method was our approximation algorithm (Section III-A), and the other four are heuristics (Section III-B). In order to determine a good approach for solving BDRT and GBDRT in practice, we conducted two separate sets of simulations to test BDRT and GBDRT. In addition, a set of simulation testing the distance from the star lower bound using uniform weights.

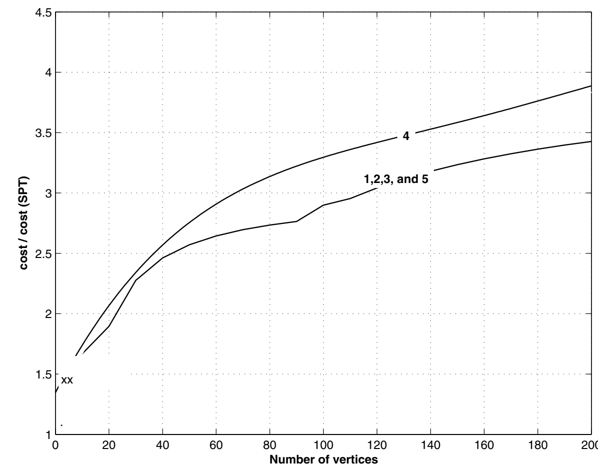

Fig. 7. Uniform case simulation of BDRT

A. Methodology

Each set of simulations was ran on a random complete graph on n vertices ($n = 10, 20, \ldots, 200$) where the (integer) degree bounds were selected uniformly at random between 3 and 8.

When simulating BDRT and GBDRT, edge weights are sampled from a $n \times n$-square given in the Euclidean plain. Each of the five algorithms was executed 1500 times, and the average cost as well as the standard deviation values were recorded. In the case of GBDRT, traffic loads are uniformly taken from the set $\{1, 2, 4, 8, \ldots, 128\}$.

Figure 7 describes the results for the uniform case of BDRT. The results indicate that the optimal solutions, produced by algorithms GA_1, GA_2, GA_3, and GA_5, are of cost within a factor of 3.4 from the lower bound. However, GA_4 was far from the optimal solution by at most 23% (in the case of $n = 80$).

B. Results

It is not hard to verify that in the uniform case (Figure 7), algorithms GA_1, GA_2, GA_3, and GA_5 are the same. Consider the greedy algorithm that picks the vertices of G in a decreasing order of their degree bounds. Clearly, changing the position of a vertex will not decrease the height of the vertex and hence the total cost of the tree will not be smaller. Assume, for example, that a vertex of level h in the constructed tree is considered to be selected by these algorithms. Since all candidates examined both by GA_1 and GA_2 are of length h from the root, the chosen vertex will be the one with the highest degree constraint. Now, since edge-weights are uniform, Algorithm GA_3 will pick the highest degree-constrained vertex as well. Finally, all the vertices in

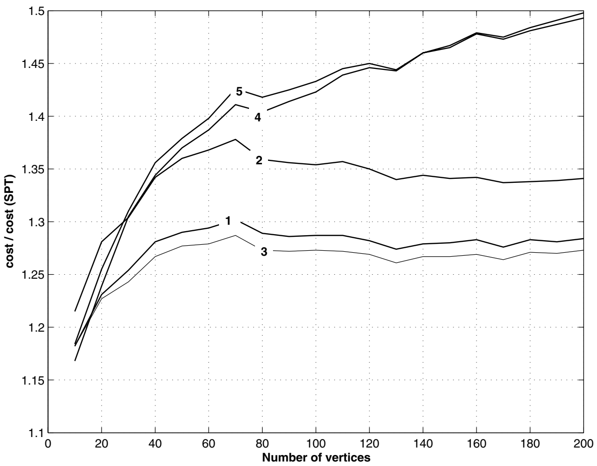

Fig. 8. BDRT simulation

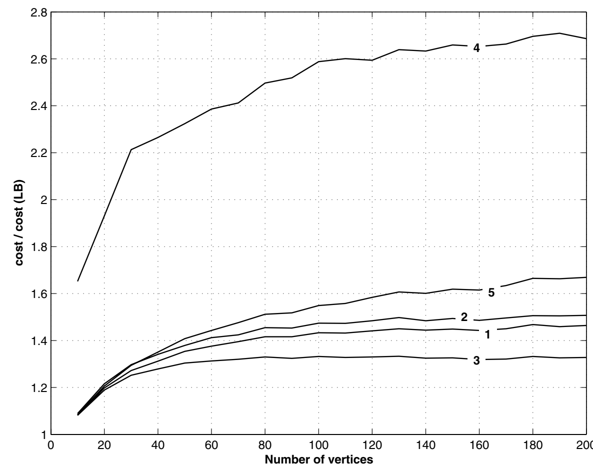

Fig. 9. GBDRT simulation

the shortest-path tree of the input graph are of the same distance from the root and therefore, the vertex of highest degree-constraint will be selected also by Algorithm GA_5. However, Algorithm GA_4 does not involve degree-constraints in its selection criteria hence it yields a different solution.

Figure 8 summarizes the results for BDRT. In this case algorithms GA_1 and GA_3 performed better than their counterparts. These algorithms achieve solutions that are within a factor of 1.28 from the lower bound. However, all five algorithms reached average costs of up to a factor of 1.49 of the lower bound, significantly far from the worst-case $O(\log n)$-factor. Standard deviation of the runs of these algorithms were between 0.14 to 0.28 for small sized graphs (Table I).

Notice that the performance of BDRT simulations (Figure 8) outperform the results of the uniform case (Figure 7) by a factor of 2.7 as in the case of GA_3. The main reason for this interesting behavior is that when a vertex is placed further down in the tree its distance to the root increases. In the non-uniform cases vertices in these positions would have lower weights and thus less effects on the overall cost.

Figure 9 summarizes the results for GBDRT. In this case, algorithms GA_1, GA_2, GA_3, and GA_5 achieve solutions that are within a factor of 1.67 from the lower bound (GA_3 has reached a factor of only 1.33). Algorithm GA_4 was far approximately 2.7 times the lower bound. Standard deviation of the runs of GA_3 were between 0.56 to 0.22 (Table II).

In addition, the worst-case running time of the algorithms, for all cases, was approximately 2 seconds for the case of $n = 200$, on a Pentium M machine, 1.4 GHz, and 256 Mb of RAM.

VI. Conclusions and Open Problems

Planning radio access networks for 4G cellular systems requires a replacement of the commonly used star based architecture (in which an RNC is connected to a set of base stations via direct links) with a new tree-topology structure.

TABLE I
ALGORITHMS FOR SOLVING BDRT

n	GA_1	GA_2	GA_3	GA_4	GA_5
10	1.182 (0.26)	1.215 (0.28)	1.182 (0.26)	1.168 (0.25)	1.184 (0.26)
20	1.231 (0.21)	1.281 (0.22)	1.227 (0.21)	1.239 (0.20)	1.255 (0.21)
30	1.254 (0.19)	1.304 (0.20)	1.243 (0.19)	1.305 (0.19)	1.310 (0.19)
40	1.281 (0.18)	1.342 (0.20)	1.267 (0.18)	1.344 (0.18)	1.356 (0.19)
50	1.290 (0.18)	1.360 (0.20)	1.277 (0.18)	1.370 (0.18)	1.379 (0.18)
60	1.294 (0.17)	1.368 (0.19)	1.279 (0.17)	1.387 (0.17)	1.398 (0.17)
70	1.302 (0.16)	1.378 (0.18)	1.287 (0.16)	1.411 (0.16)	1.426 (0.16)
80	1.289 (0.16)	1.359 (0.17)	1.273 (0.16)	1.404 (0.15)	1.418 (0.16)
90	1.286 (0.16)	1.356 (0.17)	1.272 (0.16)	1.414 (0.15)	1.425 (0.15)
100	1.287 (0.16)	1.354 (0.16)	1.273 (0.16)	1.423 (0.15)	1.433 (0.15)
110	1.287 (0.15)	1.357 (0.15)	1.272 (0.16)	1.439 (0.15)	1.445 (0.15)
120	1.282 (0.15)	1.350 (0.16)	1.269 (0.16)	1.446 (0.15)	1.450 (0.15)
130	1.274 (0.15)	1.340 (0.15)	1.261 (0.15)	1.443 (0.14)	1.444 (0.14)
140	1.279 (0.15)	1.344 (0.15)	1.267 (0.16)	1.460 (0.14)	1.460 (0.14)
150	1.280 (0.15)	1.341 (0.15)	1.267 (0.16)	1.465 (0.15)	1.467 (0.14)
160	1.283 (0.15)	1.342 (0.15)	1.269 (0.16)	1.478 (0.14)	1.479 (0.14)
170	1.276 (0.15)	1.337 (0.15)	1.264 (0.15)	1.473 (0.14)	1.475 (0.14)
180	1.283 (0.15)	1.388 (0.15)	1.271 (0.15)	1.481 (0.14)	1.484 (0.13)
190	1.281 (0.15)	1.339 (0.15)	1.270 (0.16)	1.487 (0.14)	1.491 (0.14)
200	1.284 (0.15)	1.341 (0.15)	1.273 (0.16)	1.493 (0.14)	1.489 (0.14)

TABLE II
ALGORITHMS FOR SOLVING GBDRT

n	GA_1	GA_2	GA_3	GA_4	GA_5
10	1.082 (0.56)	1.089 (0.57)	1.081 (0.56)	1.652 (0.87)	1.083 (0.56)
20	1.198 (0.45)	1.216 (0.47)	1.189 (0.45)	1.931 (0.73)	1.206 (0.46)
30	1.272 (0.41)	1.297 (0.42)	1.252 (0.40)	2.213 (0.71)	1.294 (0.41)
40	1.312 (0.36)	1.341 (0.37)	1.279 (0.35)	2.265 (0.64)	1.351 (0.37)
50	1.354 (0.33)	1.379 (0.34)	1.304 (0.32)	2.324 (0.58)	1.408 (0.35)
60	1.376 (0.31)	1.412 (0.33)	1.313 (0.30)	2.386 (0.57)	1.443 (0.33)
70	1.395 (0.30)	1.424 (0.31)	1.320 (0.29)	2.412 (0.52)	1.476 (0.31)
80	1.416 (0.29)	1.455 (0.31)	1.330 (0.27)	2.497 (0.53)	1.512 (0.31)
90	1.416 (0.27)	1.453 (0.29)	1.324 (0.26)	2.519 (0.50)	1.518 (0.29)
100	1.433 (0.27)	1.474 (0.29)	1.332 (0.25)	2.588 (0.48)	1.549 (0.28)
110	1.432 (0.25)	1.473 (0.26)	1.328 (0.23)	2.601 (0.46)	1.558 (0.26)
120	1.441 (0.26)	1.484 (0.27)	1.330 (0.23)	2.594 (0.44)	1.584 (0.27)
130	1.450 (0.25)	1.498 (0.26)	1.333 (0.23)	2.639 (0.44)	1.607 (0.27)
140	1.444 (0.24)	1.484 (0.26)	1.325 (0.22)	2.633 (0.42)	1.601 (0.26)
150	1.449 (0.24)	1.494 (0.25)	1.326 (0.22)	2.659 (0.42)	1.619 (0.26)
160	1.443 (0.23)	1.486 (0.24)	1.319 (0.21)	2.653 (0.42)	1.615 (0.25)
170	1.450 (0.23)	1.496 (0.24)	1.321 (0.21)	2.663 (0.39)	1.634 (0.24)
180	1.468 (0.23)	1.506 (0.24)	1.332 (0.21)	2.696 (0.41)	1.665 (0.25)
190	1.459 (0.23)	1.505 (0.24)	1.326 (0.21)	2.709 (0.39)	1.663 (0.24)
200	1.464 (0.22)	1.507 (0.22)	1.328 (0.20)	2.686 (0.38)	1.669 (0.23)

In the new architecture a base station can be connected to an RNC via other base stations, resulting in a complex network design problem.

We studied five algorithms for solving the metric version of BDRT. These methods involve both an approximation algorithm and a family of greedy heuristics. Our results indicate that a combination[4] of a greedy heuristic (GA_3), that picks the vertices v in an increasing order of their ratio $w(u,v)/b(v)$ and our $O(\log n)$-approximation algorithm generates a solution, which produces a close-to-optimal result in practical scenarios with a guaranteed worst case bound. Moreover, this solution can be efficiently computed for networks of large size.

Finally, two open problems are of special interest:

1) Is there a better lower bound than the shortest-path tree of the input graph? This might be a crucial challenging question towards improving the approximation factor for the metric version of BDRT. Notice that such an approximation algorithm cannot be achieved if the lower bound is indeed the cost of the shortest-path tree of the input graph. To see this consider the uniform edge-weights instance when all vertices have degree bound of 3. The optimal solution is a tree of cost $O(n \log n)$ while the shortest-path tree has cost of $n-1$.
2) As we described in this paper, when designing tree-topology radio access networks, the communication between base stations and RNCs causes a certain amount of delay in communication. In order to reduce this delay, the longest path between a base station to an RNC on every component should be limited by a given number. We can define a metric version of BDRT together with a *depth constraint* as the generalized version of BDRT. How well can this generalization be approximated?

Acknowledgements

We would like to thank Micha Berdichevsky for his help and support in making the simulations. This research was supported by REMON - Israel 4G Mobile Consortium, sponsored by Magnet Program of the Chief Scientist Office in the Ministry of Industry and Trade of Israel.

References

[1] N. Umeda, T. Otsu, and T. Masamura, "Overview of the fourth-generation mobile communication system," *NTT DoCoMo Technical Review*, vol. 2, no. 9, pp. 12–31, 2004. [Online]. Available: `http://www.ntt.co.jp/tr/0409/special.html`

[2] F. Longoni, A. Länsisalmi, and A. Toskala, "Radio Access Network Architechture," in *WCDMA for UMTS*, 3rd ed., H. Holma and A. Toskala, Eds. John Wiley & Sons, Ltd, 2004, pp. 75–98.

[3] A. Krendzel, Y. Koucheryavy, and J. Harju, "Radio Access Network Topology Planning for the 4G Networks," in *Proceedings of the 5th European Wireless Conference, Barcelona, Spain*, 2004.

[4] Y. Yamao, H. Suda, N. Umeda, and N. Nakajima, "Radio access newtork design concept for the fourth generation mobile communication system," *IEEE VTC2000*, vol. 3, pp. 2285–2289, 2000.

[5] T. Otsu, N. Umeda, and Y. Yamao, "System architechture for mobile communications systems beyond IMT-2000," in *Proceedings of the 44th Annual IEEE GLOBECOM*, vol. 1, 2001, pp. 538–542.

[4]In the sense of executing both algorithms and choosing the lower cost solution.

[6] J. Harmatos, A. Jüttner, and A. Szentesi, "Cost-based UMTS transport network topology optimization," in *Proceedings of the 14th International Conference on Computer Communication (ICCC)*, 1999.

[7] J. Harmatos, A. Szentesi, and I. Gódor, "Planning of tree-topology UMTS terrestrial access networks," in *Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2000.

[8] A. Szlovencsák, I. Gódor, J. Harmatos, and T. Cinkler, "Planning reliable UMTS terrestrial access networks," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2002.

[9] A. Jüttner, A. Orbán, and Z. Fiala, "Two new algorithms for UMTS access networks," *European Journal of Operational Research*, vol. 164, pp. 456–474, 2005.

[10] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag, 2001.

[11] M. Goemans, "Bounded degree minimum spanning tree," in *The 6th Haifa Workshop on Interdisciplinary Applications of Graph-Theory, Combinatorics and Algorithms*. The Caesarea Edmond Benjamin de Rothschild Foundation Institute for Interdisciplinary Applications of Computer Science, University of Haifa, Israel, 2006, to appear in Proceedings of the 47th Annual IEEE Symposium on Foundations on Computer Science (FOCS).

[12] S. P. Fekete, S. Khuller, M. Klemmstein, B. Raghavachari, and N. Young, "A network-flow technique for finding low-weight bounded-degree spanning trees," *J. Algorithms*, vol. 24, pp. 310–324, 1997, a preliminary version of this paper appeared in *Proceedings of the 5th International Integer Programming and Combinatorial Optimization Conference (IPCO)*, 105–117, 1996.

[13] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, "Approximation algorithms for degree-constrained minimum-cost network-design problems," in *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC)*, 1993, pp. 438–447.

[14] D. S. Johnson, J. K. Lenstra, and A. H. G. R. Kan, "The complexity of the network design problem," *Networks*, vol. 8, pp. 279–285, 1978.

[15] B. Y. Wu, G. Lancia, V. Bafna, K. M. Chao, R. Ravi, and C. Y. Tang, "A polynomial-time approximation scheme for minimum routing cost spanning trees," *SIAM J. Computing*, vol. 29, no. 3, pp. 761–778, 1999.

[16] T. C. Hu, "Optimum communication spanning trees," *SIAM J. Computing*, vol. 3, pp. 188–195, 1974.

[17] E. Reshef, "The Minimum Communication Cost Spanning Trees," Master's thesis, Weizmann Institute of Science, Israel, 1999.