Approximation Algorithms for the Capacitated Minimum Spanning Tree Problem and its Variants in Network Design

RAJA JOTHI

National Institutes of Health

AND

BALAJI RAGHAVACHARI

University of Texas at Dallas

Abstract. Given an undirected graph $G = (V, E)$ with nonnegative costs on its edges, a root node $r \in V$, a set of demands $D \subseteq V$ with demand $v \in D$ wishing to route $w(v)$ units of flow (weight) to r, and a positive number k, the *Capacitated Minimum Steiner Tree* (CMStT) problem asks for a minimum Steiner tree, rooted at r, spanning the vertices in $D \cup \{r\}$, in which the sum of the vertex weights in every subtree connected to r is at most k. When $D = V$, this problem is known as the *Capacitated Minimum Spanning Tree* (CMST) problem. Both CMsT and CMST problems are NP-hard. In this article, we present approximation algorithms for these problems and several of their variants in network design. Our main results are the following:

—We present a $(\gamma\rho_{ST} + 2)$-approximation algorithm for the CMStT problem, where γ is the *inverse Steiner ratio*, and ρ_{ST} is the best achievable approximation ratio for the Steiner tree problem. Our ratio improves the current best ratio of $2\rho_{ST} + 2$ for this problem.

—In particular, we obtain $(\gamma+2)$-approximation ratio for the CMST problem, which is an improvement over the current best ratio of 4 for this problem. For points in Euclidean and rectilinear planes, our result translates into ratios of 3.1548 and 3.5, respectively.

—For instances in the plane, under the L_p norm, with the vertices in D having uniform weights, we present a nontrivial $(\frac{7}{5}\rho_{ST} + \frac{3}{2})$-approximation algorithm for the CMStT problem. This translates into a ratio of 2.9 for the CMST problem with uniform vertex weights in the L_p metric plane. Our

---

Research supported in part by the National Science Foundation under grant CCR-9820902.

An extended abstract of this article appeared in Jothi and Raghavachari [2004a].

Authors' addresses: R. Jothi, National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, 8600 Rockville Pike, Bethesda, MD 20894, e-mail: jothi@ncbi.nlm.nih.gov; B. Raghavachari, Department of Computer Science, University of Texas at Dallas, Richardson, TX 75080, e-mail: rbk@utdallas.edu.

ratio of 2.9 solves the long-standing open problem of obtaining any ratio better than 3 for this case.

—For the CMST problem, we show how to obtain a 2-approximation for graphs in metric spaces with unit vertex weights and $k = 3, 4$.

—For the *budgeted* CMST problem, in which the weights of the subtrees connected to r could be up to αk instead of k ($\alpha \geq 1$), we obtain a ratio of $\gamma + \frac{2}{\alpha}$.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems; G.2.1 [**Discrete Mathematics**]: Combinatorics; G.2.2 [**Discrete Mathematics**]: Graph Theory

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Spanning trees, minimum spanning trees, approximation algorithms, network design

1. *Introduction*

In this article, we consider the *Capacitated Minimum Steiner Tree* (CMStT) problem, one of the extensively studied network design problems in telecommunications [Amberg et al. 1996; Voß 2001]. The CMStT problem can formally be defined as follows:

> CMStT**:** Given an undirected graph $G = (V, E)$ with nonnegative costs on its edges, a root node $r \in V$, a set of demands $D \subseteq V$ with demand $v \in D$ wishing to route $w(v)$ units of flow (weight) to r, and a positive number k, the *Capacitated Minimum Steiner Tree* (CMStT) problem asks for a minimum Steiner tree, rooted at r, spanning the vertices in $D \cup \{r\}$, in which the sum of the vertex weights in every subtree connected to r is at most k.

For the CMStT problem, the capacity constraint k must be at least as much as the largest vertex weight in order to be able to find a feasible solution. Since the case with $k = \infty$ is the minimum Steiner tree problem, which is NP-hard, the CMStT problem is NP-hard. When $D = V$, the CMStT problem is the well-known *Capacitated Minimum Spanning Tree* (CMST) problem. The CMST problem is NP-hard [Garey and Johnson 1979; Papadimitriou 1978] even for the case when vertices have unit weights and $k = 3$. The problem is polynomial-time solvable if all vertices have unit weights and $k = 2$ [Garey and Johnson 1979]. The problem can also be solved in polynomial time if vertices have 0,1 weights and $k = 1$, but remains NP-hard if vertices have 0,1 weights, $k = 2$, and all edge lengths are 0 or 1 [Garey and Johnson 1979]. Even the geometric version of the problem, in which the edge costs are defined to be the Euclidean distance between the vertices they connect, remains NP-hard.

In telecommunication network design, the CMST problem is that of designing a minimum-cost tree network by installing expensive (such as fiber-optic) cables along its edges. The cables have prespecified capacities on the amount of demand they can handle, and can be bought at a certain cost per unit length. Every source node in the network has some demand that it has to transmit to the sink node. The objective is to construct a minimum-cost tree network for simultaneous routing of demands from the source nodes to the sink node.

A generalization of the CMStT problem is the *Single Sink Buy-at-Bulk* (SSBB) problem in which we are given the option of using l different cable types, instead of just one [Salman et al. 1997]. Each cable has a different capacity constraint and the cost of the cables obeys "economy of scale." One major difference is that there is no restriction that the final network needs to be a tree. Another variation is the *Capacitated Minimum Spanning Network* problem with connectivity constraints [Jothi and Raghavachari 2004b].

The CMST problem has been well studied in computer science and operations research for the past 40 years. Numerous heuristics and exact algorithms have been proposed (see Section 1.3 for more details). Although most of the heuristics solve several well-known instances close to optimum, they do not provide any approximation guarantee on the quality of the solutions obtained. Exact procedures are limited to solving smaller instances because of their exponential running time. In this article, we present improved approximation algorithms for the CMStT and CMST problems and their variants.

1.1. PREVIOUS RESULTS. For the CMST problem with uniform vertex weights, Gavish and Altinkemer [1986] presented a modified *parallel savings algorithm* (PSA) with approximation ratio $4 - 1/(2^{\lceil \log k \rceil - 1})$. Altinkemer and Gavish [1988] gave improved approximation algorithms with ratios $3 - \frac{2}{k}$ and 4 for the uniform and nonuniform vertex weight cases, respectively. They constructed a traveling salesman tour (TSP) with length of at most twice the minimum spanning tree (MST), and partitioned the tour into segments (subtrees) of weight at most k. Partitioned subtrees were then connected to the root vertex using direct edges. Hassin et al. [2000] presented algorithms for the 1-cable SSBB problem. The algorithms by Altinkemer and Gavish [1988] and Hassin et al. [2000] can be used to obtain ratios of $2\rho_{ST}+1$ and $2\rho_{ST}+2$ for the respective uniform and nonuniform vertex weight CMStT problems.

1.2. OUR CONTRIBUTIONS. In this article, we solve the long-standing open problem of obtaining better approximation ratios for the CMST problem. Our main results are the following:

—We present a $(\gamma\rho_{ST}+2)$-approximation algorithm for the CMStT problem, where γ is the *inverse Steiner ratio*,[1] and ρ_{ST} is the best achievable approximation ratio for the Steiner tree problem. Our ratio improves the current best ratio of $2\rho_{ST}+2$ for this problem.

—In particular, we obtain $(\gamma+2)$-approximation ratio for the CMST problem, which is an improvement over the current best ratio of 4 for this problem. For points in Euclidean and rectilinear planes, our result translates into ratios of 3.1548 and 3.5, respectively.

—For instances in the plane, under the L_p norm, with the vertices in D having uniform weights, we present a nontrivial $(\frac{7}{5}\rho_{ST} + \frac{3}{2})$-approximation algorithm for the CMStT problem. This translates into a ratio of 2.9 for the CMST problem with uniform vertex weights in the L_p metric plane. Our ratio of 2.9 solves the long-standing open problem of obtaining a ratio any better than 3 for this case.

—For the CMST problem, we show how to obtain a 2-approximation for graphs in metric spaces with unit vertex weights and $k = 3, 4$.

[1]The Steiner ratio is the maximum ratio of the costs of the minimum cost Steiner tree versus the minimum-cost spanning tree for the same instance. In graphs $\gamma = 2$, and in Euclidean and rectilinear metrics it is $2/\sqrt{3}$ and $3/2$, respectively.

—For the *budgeted* CMST problem, in which the weights of the subtrees connected to r could be up to αk instead of k ($\alpha \geq 1$), we obtain a ratio of $\gamma + \frac{2}{\alpha}$.

Of the above results, the 2.9-approximation result for the CMST problem is of most significance. This is due to the fact that obtaining a ratio any better than 3 for graphs defined in the Euclidean plane (with uniform vertex weights) is not straightforward. There are several ways one can obtain a ratio of 3 for this problem (Altinkemer and Gavish [1988], a variation of the algorithm in Hassin et al. [2000], our algorithm in Section 3.1). But the question was whether one can ever obtain a ratio smaller than $3 - o(1)$ for this version of the CMST problem. We present an example (in Section 5), which shows that, with the currently available lower bounds for the CMST problem, it is not possible to obtain an approximation ratio any better than 2. We introduce a novel concept of *X-trees* to overcome the difficulties in obtaining a ratio better than 3.

Achieving ratios better than 3 and 4 for the uniform and nonuniform vertex weighted CMST problems, respectively, has been an open problem for over 15 years now. One major reason for the difficulty in finding better approximations is that there is no nontrivial lower bound for an optimal solution. There are instances for which the cost of an optimal solution can be as much as $\Omega(n/k)$ times that of an MST. The inability to find better lower bounds has greatly impeded the process of finding better approximation ratios for this problem. Even though we were not able to completely eliminate the use of MST as a lower bound, we found ways to exploit its geometric structure, thereby achieving better performance ratios. Unlike the algorithms in Altinkemer and Gavish [1988], in which the MST lower bound contributes a factor of 2 to the final ratio, our algorithms minimize the use of the MST lower bound, thereby achieving better ratios.

1.3. RELATED WORK. Algorithms for finding an MST with no constraints are well known. In fact, those algorithms can be modified to find a feasible CMST, albeit with no guaranteed approximation ratio. The celebrated clustering technique of Goemans and Williamson [1995] can be used to approximate the CMST problem by partitioning the given set of n nodes into groups of weight exactly k, form a MST for each group, and connect each tree to the root node. The approximation ratio of such an approach would be $4(1 - \frac{1}{k})^2 + 1$.

There are several exact algorithms and mathematical formulations [Chandy and Lo 1973; Chandy and Russell 1972; Elias and Ferguson 1974; Gavish 1982, 1983, 1985; Gouveia 1993, 1995; Gouveia and Ao 1991; Hall 1996; Kershenbaum and Boorstyn 1983; Malik and Yu 1993] available for solving the CMST problem. The instance sizes that can be solved by these algorithms to optimality, in reasonable amount of time, are still far from that of real-time instances.

Heuristics for the CMST problem can be classified into three categories: savings procedures, construction procedures, and improvement procedures. Esau and Williams [1966] gave an efficient and well-known savings heuristic for the CMST problem. Some of the other heuristics that use savings procedure to construct the tree include Elias and Ferguson [1974], Gavish [1991], Gavish and Altinkemer [1986], Kershenbaum [1974], Whitney [1970]. Construction procedures [Boorstyn and Frank 1977; Chandy and Russell 1972; Karnaugh 1976; Kershenbaum and Chou 1974; Martin 1967; McGregor and Shen 1977; Schneider and Zastrow 1982; Sharma and El-Bardai 1970] start with an empty tree and add the best edge or node to grow the tree. Improvement procedures [Elias and Ferguson 1974; Frank et al. 1971; Karnaugh 1976; Kershenbaum et al. 1980; Malik and Yu 1993] start with an initial feasible solution, usually obtained by running the Esau and Williams [1966] algorithm, and make improvements by adding and deleting edges as long as improvements are possible. There are other heuristics based on *tabu search* and *simulated annealing* [Aarts and Korst 1989; Osman and Christofides 1994; Sharaiha et al. 1997]. For more details on these and other heuristics, we refer the reader to the following survey papers: Amberg et al. [1996] and Voß [2001] for further information on several well-known heuristics, and Ahuja et al. [2001], Amberg et al. [1996], Domschke et al. [1992], and Glover [1989, 1990] for extensive background and comparison of several heuristics based on tabu search techniques.

2. *Preliminaries*

Let $|uv|$ denote the distance between vertices u and v. Length of an edge is also its cost. The terms *points*, *nodes*, and *vertices* will be used interchangeably in this article. For a given k, let OPT and APP denote optimal and approximate solutions, respectively, and let C_{opt} and C_{app} denote their respective costs. Let C_{mst} and C_{ST} denote the costs of an MST and an optimal Steiner tree, respectively.

In a rooted tree T, let t_v denote the subtree rooted at v. Let C_T denote the cost of tree T. Let $w(v)$ denote the weight of vertex v, and let $w(t_v)$ denote the sum of vertex weights in the subtree rooted at v. For the CMStT problem, the weight of a vertex the is not in D is assumed to be 0. By weight of a subtree, we mean the sum of the vertex weights in that subtree. We call the edges incident on r of a CMStT *spokes*. By *level* of a vertex, in a tree T rooted at r, we mean the number of tree edges on its path to r (also known as *depth*).

By *metric completion* of a given graph (whose edges obey triangle inequality), we refer to a complete graph. Throughout this article, without loss of generality, we assume that the metric completion of the input graph is available, and that the weights of vertices in $V \setminus D$ are zero. All our algorithms in this article are for the CMStT problem—a generalization of the CMST problem. The following lemma gives a lower bound on the cost of an optimal solution.

LEMMA 2.1. $C_{opt} \geq \frac{1}{k} \sum_{v \in V} w(v)|rv|$.

PROOF. Let t be the number of subtrees connected to r in OPT. Let l be one such subtree in OPT that is connected to r. Let S_l be the set of vertices in l. Let C_l be the sum of the cost of the edges incident on the vertices in l. By triangle inequality,

$$\begin{aligned} C_l &\geq \max_{v \in S_l}\{|rv|\} \\ &\geq \frac{\sum_{v \in S_l} w(v)|rv|}{\sum_{v \in S_l} w(v)} \\ &\geq \frac{\sum_{v \in S_l} w(v)|rv|}{k}. \end{aligned}$$

For t subtrees connected to r in *OPT*,

$$\begin{aligned} C_{opt} &= \sum_{l=1}^{t} C_l \\ &\geq \frac{\sum_{v \in V} w(v)|rv|}{k}. \end{aligned}$$

□

3. CMStT *Algorithms*

We first construct a ρ_{ST}-approximate Steiner tree T spanning all the vertices in $D \cup \{r\}$, and then root T at the root vertex r. Next, we prune subtrees of weight at most k in a bottom-up fashion, and add edges to connect r to the closest node in each of the pruned subtrees. In simple terms, we basically cut T into subtrees of weight at most k and connect them to the root vertex.

It is safe to assume that nodes have integer weights. The assumption is not restrictive as any CMStT problem with rational weights can be converted to an equivalent problem with integer node weights. The optimal solution for the scaled problem is identical to that of the original problem [Altinkemer and Gavish 1988].

Since our algorithm for the uniform vertex weights case is quite complex, we first present the algorithm for the general case (nonuniform vertex weights), which will help in an easier understanding of our algorithm for the uniform vertex weights case. Before we proceed to the algorithms, we present the following important lemma.

LEMMA 3.1. *For a given graph $G = (V, E)$, a set of demands $D \subseteq V$, $r \in V$, and a k, let T_f be a feasible* CMStT *and let $t_1, t_2, \ldots, t_m$ be the subtrees connected to r in T_f. Let $w(t_q)$ be the weight of a minimum weight subtree t_q connected to r. For all i, if the cost of the edge connecting subtree t_i to r is minimal, then the cost C_{sp} of all the edges incident on r (spokes) in T_f is at most $k/w(t_q)$ times the cost of an optimal solution.*

PROOF. Let Γ be the set of vertices in $t_1, \ldots, t_m$. For all i, let v_i be the vertex in t_i through which t_i is connected to r. Recall that edge rv_i is a spoke, and that it is a minimal cost edge crossing the cut between r and t_i. Then,

$$\begin{aligned} |rv_i| &\le \frac{\sum_{v \in t_i} w(v)|rv|}{\sum_{v \in t_i} w(v)} \\ &\le \frac{\sum_{v \in t_i} w(v)|rv|}{w(t_q)}. \end{aligned}$$

The cost of the all the edges incident on r is given by

$$\begin{aligned} C_{sp} &= \sum_{i=1}^{m} |rv_i| \\ &\le \frac{\sum_{v \in \Gamma} w(v)|rv|}{w(t_q)} \\ &= \frac{k}{w(t_q)} \times \frac{\sum_{v \in D} w(v)|rv|}{k} \\ &\le \frac{k}{w(t_q)} \times C_{opt} \quad \text{(by Lemma 2.1).} \end{aligned}$$

□

3.1. NONUNIFORM VERTEX WEIGHTS. The algorithm given below outputs a feasible CMStT for a given instance, whose edges obey triangle inequality. Note that during the course of the algorithm, we replace real vertices with *dummy* vertices of zero weight. These dummy vertices can be thought of as Steiner points. In the algorithm, we use c_i to denote the subtree rooted at vertex v's ith child, and p_v to denote v's parent.

Algorithm CMStT-NONUNIFORM
Input: ρ_{ST}-approximate Steiner tree T rooted at r.

(1) Choose a maximum level vertex $v \neq r$ such that $w(t_v) \geq k$. If there exists no such vertex, then for each subtree s containing a dummy vertex and connected to r, replace the Steiner edges incident on the vertices in s with the minimal cost tree τ spanning only the vertices in $s \cap D$ and r, and STOP.
(2) If $w(t_v) = k$, then replace the Steiner tree edges incident on the vertices in t_v with the edges of a minimal-cost tree τ spanning only the vertices in $t_v \cap D$. Add a new edge connecting r to the closest vertex in τ.
(3) Else if, for some i, $w(c_i) \geq k/2$, then replace the Steiner tree edges incident on the vertices in c_i with the edges of a minimal-cost tree τ spanning only the vertices in $c_i \cap D$. Add a new edge connecting r to the closest vertex in τ.
(4) Else if $\sum w(c_i) < k/2$, which means $w(v) > k/2$, then replace v with a dummy vertex. In the final solution, add v and an edge connecting v to r.
(5) Else collect a subset s of subtrees, each of which is rooted at one of v's children, such that $k/2 \leq w(s) \leq k$. Replace the Steiner tree edges incident on the vertices in s with the edges of a minimal-cost tree τ spanning only the vertices in $s \cap D$. Add a new edge connecting r to the closest vertex in τ.
(6) Go to step 1.

It can be verified that our algorithm outputs a feasible CMStT for a given k.

THEOREM 3.2. *For a given* CMStT *instance, Algorithm* CMStT-NONUNIFORM *guarantees an approximation ratio of* $(\gamma\rho_{ST} + 2)$.

PROOF. We show that the cost of the tree output by Algorithm CMStT-NONUNIFORM is at most $\gamma\rho_{ST} + 2$ times the cost of an optimal CMStT. The input to the algorithm is a ρ_{ST}-approximate Steiner tree T.

It can be easily verified from the algorithm that all the new edges added to the original tree T are either new spokes, or edges that interconnect vertices within the subtrees for which the new spokes were added. In what follows, we account for the cost of the new spokes added to T, followed by the cost of other edges in the final solution output by the algorithm.

A new spoke, incident on a subtree, is added to the original Steiner tree if and only if the weight of the subtree it connects is at least $k/2$. Notice that the algorithm outputs a tree with each subtree connected to r being disjoint and the weight of every such subtree, for which a new spoke was added, being at least $k/2$. Let C_{sp} be the cost of the spokes that the algorithm "adds" to the Steiner tree. Note that C_{sp} does not include the cost of the spokes that are already in the Steiner tree that was given as input to the algorithm. By Lemma 3.1,

$$C_{sp} \leq 2 \times C_{opt}.$$

Now, we account for the cost of other edges in the final solution. These edges are either the Steiner tree edges or the edges that replaced the Steiner tree edges. We show that the total cost of all these edges together is at most γ times the cost of the initial Steiner tree. To prove this, it suffices to prove that the cost of the edges that replace the Steiner tree edges is at most γ times the cost of the Steiner tree edges that it replaces. For every subtree formed, notice that the algorithm replaced the edges of the Steiner tree spanning the vertices in that subtree by the edges of an MST spanning only the nonzero weight vertices in that subtree. Since γ was defined to be the inverse Steiner ratio (ratio of the cost of an MST versus the cost of an optimal Steiner tree), by Steiner ratio argument, the cost of the MST spanning only the nonzero weight vertices in a subtree is at most γ times the cost of an optimal Steiner tree spanning the nonzero weight vertices in that subtree. Thus, we can conclude that the cost of the new edges is at most γ times the cost of the ρ_{ST}-approximate Steiner tree edges it replaces. The final cost of the tree output by the algorithm is given by

$$\begin{aligned} C_{app} &\le C_{sp} + \gamma\rho_{ST}C_{ST} \\ &\le 2C_{opt} + \gamma\rho_{ST}C_{opt} \\ &\le (\gamma\rho_{ST} + 2)C_{opt}. \end{aligned}$$ □

COROLLARY 3.3. *For the* CMStT *problem with uniform vertex weights, Algorithm* CMStT-NONUNIFORM *with little modification guarantees a* $(\rho_{ST} + 2)$-*approximation ratio.*

PROOF. Since we are dealing with uniform vertex weights, without loss of generality, we can assume that they are of unit weight, and thus we can eliminate Step. 4 from Algorithm CMStT-NONUNIFORM. Therefore no dummy vertices are introduced by the algorithm. Once a subtree t of size at least $k/2$ is found, instead of replacing the Steiner tree spanning the vertices in t with a MST spanning the nonzero weight vertices in t, we can just use the edges in t, minus the edge that connects t to its parent, as they are. This eliminates the γ from the final ratio. □

COROLLARY 3.4. *For the* CMST *problem, Algorithm* CMStT-NONUNIFORM *guarantees a* $(\gamma + 2)$-*approximation ratio. In particular, for points in Euclidean and rectilinear planes, it guarantees a ratio of* 3.1548 *and* 3.5*, respectively.*

PROOF. Since the MST cost is a lower bound on the CMST, the CMStT algorithm with an MST as the input will guarantee an approximation ratio of $(\gamma + 2)$ being $2/\sqrt{3}$ and $3/2$ in the Euclidean and rectilinear metrics, respectively, which completes the proof. □

3.2. UNIFORM VERTEX WEIGHTS. Although our algorithm for uniform vertex weights case is similar to Algorithm CMStT-NONUNIFORM at the top level, contrary to expectations, there are some complicated issues that have to be handled in order to obtain an approximation ratio strictly less than $\rho_{ST} + 2$. From our analysis for the non uniform vertex weights case, we can see that the weight of the minimum weight subtree connected to r plays a crucial role in the calculation of the approximation ratio. An obvious heuristic is to prune subtrees of weight as close as possible to k, so that the ratio drops considerably. We will soon see why pruning subtrees of weight strictly greater than $k/2$ is more difficult than pruning subtrees of weight greater than or equal to $k/2$. To overcome the difficulty of pruning subtrees of size strictly greater than $k/2$, we introduce the concept of *X-trees*, which we define below. We call a subtree, t_v, rooted at vertex v an *X-tree*, x, if all of the following properties

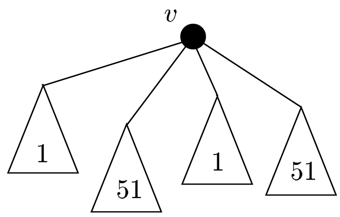

FIG. 1. An X-tree with $k = 100$.

are satisfied (see Figure 1):

—$k < w(t_v) < \frac{4}{3}k$.
—Weight of no subtree connected to v is between $\frac{2}{3}k$ and k.
—Sum of the weights of no two subtrees connected to v is between $\frac{2}{3}k$ and k.
—Sum of the weights of no three subtrees connected to v is between $\frac{2}{3}k$ and k.

Since the vertices are of uniform weight, without loss of generality, we can assume that they are of unit weight, and scale k accordingly. We also assume that a ρ_{ST}-approximate Steiner tree is given as part of the input. Note that we are trying to solves instances in L_p metric plane. Even though the maximum nodal degree in a Steiner tree on a plane is 3, we will continue as if it were 5. This is to ensure that our algorithm solves CMST instances on a plane, as the maximum degree of an MST on a L_p plane is 5 [Monma and Suri 1992; Robins and Salowe 1995]. Note that every vertex but the root in a tree, with vertex degrees at most 5, has at most four children. For points in the L_p plane, the proposition below follows from the definition of an X-tree.

PROPOSITION 3.5. *Let v_1 be a maximum level vertex in an X-tree rooted at v such that t_{v_1} is also an X-tree (v_1 could be v itself). If there is no subtree (non-X-tree) of weight greater than k rooted at one of v_1's children, then there always exist two subtrees, t_α and t_β, connected to v_1 such that $k < w(t_\alpha) + w(t_\beta) < \frac{4}{3}k$ and $\frac{1}{3}k < w(t_\alpha), w(t_\beta) < \frac{2}{3}k$.*

The algorithm given below finds a feasible CMStT for instances defined on a L_p plane. In the algorithm, we use c_i to denote the subtree rooted at child i of vertex v, and x_j to denote the X-tree rooted at child j of vertex v.

Algorithm CMStT-UNIFORM
Input: ρ_{ST}-approximate Steiner tree T rooted at r, with vertex degrees at most 5.

(1) Choose a maximum level vertex $v \neq r$ such that t_v is a non-X-tree with $w(t_v) \geq k$. If there exists no such vertex then go to step 11.
(2) If $w(t_v) = k$, then add a new edge connecting r to the closest node in t_v. Remove edge vp_v from T.
(3) Else if, for some i, $2k/3 \leq w(c_i) \leq k$, then add a new edge connecting r to the closest node in c_i. Remove the edge connecting v to c_i from T.
(4) Else if, for some i and j ($i \neq j$), $2k/3 \leq w(c_i) + w(c_j) \leq k$, then replace edges vc_i and vc_j by a minimal cost edge connecting c_i and c_j, merging the two subtrees into a single tree s. Add a new edge to connect r to the closest node in s.
(5) Else if, for some i, j, and z ($i \neq j \neq z$), $2k/3 \leq w(c_i) + w(c_j) + w(c_z) \leq k$, then replace the Steiner tree edges incident on the vertices in c_i, c_j, and c_z by a minimal-cost tree s spanning all the vertices in c_i, c_j, and c_z (Figure 2(a)). Add a new edge to, connect r to the closest node in s.

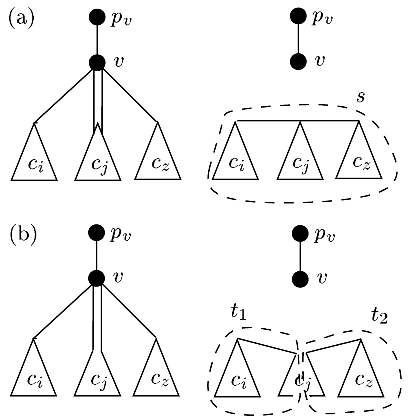

FIG. 2. Illustration of (a) Step 5 and (b) Step 6.

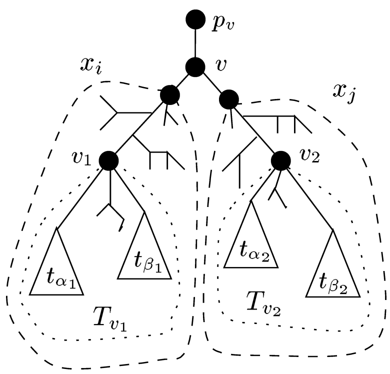

FIG. 3. X-tree within an X-tree.

(6) Else if, for some i, j, and z $(i \neq j \neq z)$, $4k/3 \leq w(c_i) + w(c_j) + w(c_z) \leq 2k$, then do the following:

Let E_i be the set of edges incident on vertices in c_i. We define E_j (E_z) with respect to c_j (respectively, c_z) analogously. Without loss of generality, let E_j be the low-cost edge set among E_i, E_j, and E_z. Use DFS on c_j to partition the vertices in c_j into two sets g_1 and g_2 such that the total weight of vertices in $(c_i \cup g_1) \cap D$ is almost the same as the total weight of vertices in $(c_z \cup g_2) \cap D$. Remove all the edges incident on the vertices in subtrees c_i, c_j and c_z. Construct a minimal cost spanning tree s_1 comprising the vertices in c_i and g_1. Similarly, construct a minimal-cost spanning tree s_2 comprising the vertices in c_z and g_2. Add new edges to connect r to the closest nodes in s_1 and s_2.

(7) Else if, for some i and j $(i \neq j)$, $2k < w(x_i) + w(x_j) < 8k/3$, do the following. Let v_1 and v_2 be two maximum level vertices in X-trees x_i and x_j, respectively, such that t_{v_1} and t_{v_2} are X-trees themselves (see Figure 3). Recall, by Proposition 3.5, that there exist two subtrees t_{α_1} and t_{β_1} (t_{α_2} and t_{β_2}), connected to v_1 (respectively, v_2) such that $k < w(t_{\alpha_1}) + w(t_{\beta_1}) < \frac{4}{3}k$ (respectively, $k < w(t_{\alpha_2}) + w(t_{\beta_2}) < \frac{4}{3}k$).

Let E_1 represent the set of edges incident on vertices in t_{α_1} (see Figure 4). Let E_2 represent the set of edges incident on vertices in t_{β_1}. We define E_4 (E_5) with respect to t_{α_2} (respectively, t_{β_2}) analogously. Let E_3 be the set of edges incident on vertices in x_i and x_j minus the edges in E_1, E_2, E_4 and E_5.

Let $G_1 = \{E_1, E_2\}$, $G_2 = \{E_3\}$, and $G_3 = \{E_4, E_5\}$ be three groups. Out of $\{E_1, E_2, E_3, E_4, E_5\}$, double two low-cost edge sets such that they belong to different groups.

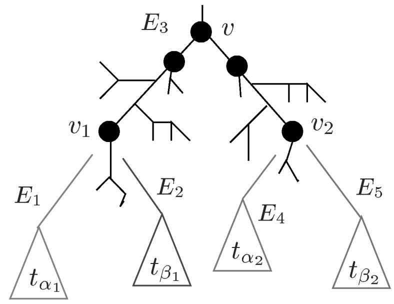

FIG. 4. Partitioning the edges within an X-tree.

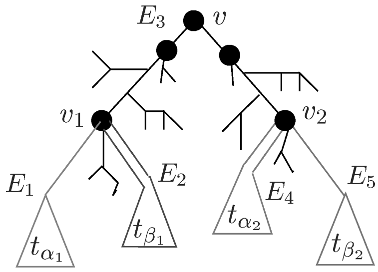

FIG. 5. Illustration of Step 7(a).

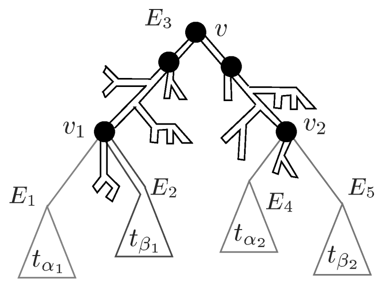

FIG. 6. Illustration of Step 7(b).

(a) If E_i and E_j were the two edges sets that were doubled, with E_i in G_1 and E_j in G_3, then form three minimal-cost subtrees s_1, s_2, and s_3 spanning the vertices in x_i and x_j as follows. Without loss of generality, let E_2 and E_4 be the two low-cost edge sets that were doubled (Figure 5). Use shortcutting to form s_1 spanning all vertices in t_{α_1} and a subset of vertices in t_{β_1}, form s_3 spanning all vertices in t_{β_2} and a subset of vertices in t_{α_2}, and form s_2 with all the left-over vertices. Remove edge vp_v. Since $k < w(t_{\alpha_1}) + w(t_{\beta_1}) < 4k/3$, $k < w(t_{\alpha_2}) + w(t_{\beta_2}) < 4k/3$, and $2k \leq w(x_i) + w(x_j) \leq 8k/3$, we can form s_1, s_2, and s_3 of almost equal weight with $2k/3 \leq w(s_1), w(s_2), w(s_3) \leq k$.

(b) If E_i and E_j were the two edges sets that were doubled, with E_i in G_1 or G_3, and E_j in G_2, then form three minimal-cost subtrees s_1, s_2, and s_3 spanning the vertices in x_i and x_j as follows. Without loss of generality, let E_2 and E_3 be the two low-cost edge sets that were doubled (see Figure 6). From t_{α_2} and t_{β_2} find a vertex w such that $|wr|$ is minimum. Without loss of generality, let t_{α_2} contain w. Use shortcutting to form s_3 spanning all the vertices in x_j minus the vertices in t_{β_2} (see Figure 7). Note that $k/3 < w(s_3) < k$, as x_j and t_{v_2} are X-trees and $k/3 < w(t_{\alpha_2}), w(t_{\beta_2}) < 2k/3$. Also, since $k/3 < w(t_{\beta_2}) < 2k/3$ and $k < w(x_i) < 4k/3$, subtrees s_1 and s_2 together will be of weight at least $4k/3$ and at most $2k$ (see Figure 7). Form subtrees s_1 and s_2, using the ideas in Step 6, such that $2k/3 \leq w(s_1), w(s_2) \leq k$, and $4k/3 \leq w(s_2) + w(s_3) \leq 2k$.

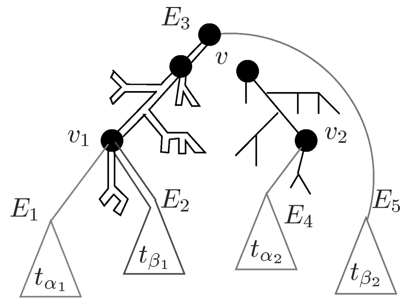

FIG. 7. Pruning a subtree of weight at least $k/3$ and at most k from an X-tree.

(c) Add new edges to connect r to the closest nodes in s_1, s_2, and s_3.

(8) Else if, for some i and j ($i \neq j$), $4k/3 \leq w(x_i) + w(c_j) < 2k$, do the following.
Let v_1 be a maximum level vertex in X-tree x_i such that t_{v_1} is an X-tree itself. Recall, by Proposition 3.5, that there exist two subtrees t_{α_1} and t_{β_1}, connected to v_1 such that $k < w(t_{\alpha_1}) + w(t_{\beta_1}) < \frac{4}{3}k$.
Let E_1 represent the set of edges incident on vertices in t_{α_1}. Let E_2 represent the set of edges incident on vertices in t_{β_1}. Let E_3 be the set of edges incident on vertices in x_i and c_j minus the edges in E_1 and E_2. Form subtrees s_1 and s_2 using the ideas in Step 6. Add new edges to connect r to the closest nodes in s_1 and s_2.

(9) Else if, $4k/3 \leq w(t_v) \leq 2k$, do the following. Let v_1 be a maximum level vertex in X-tree x_i such that t_{v_1} is an X-tree itself. Recall, by Proposition 3.5, that there exist two subtrees t_{α_1} and t_{β_1}, connected to v_1 such that $k < w(t_{\alpha_1}) + w(t_{\beta_1}) < \frac{4}{3}k$.
Let E_1 represent the set of edges incident on vertices in t_{α_1}. Let E_2 represent the set of edges incident on vertices in t_{β_1}. Let E_3 be the set of edges incident on vertices in t_v minus the edges in E_1 and E_2. Form subtrees s_1 and s_2 using the ideas in Step 6. Add new edges to connect r to the closest nodes in s_1 and s_2.

(10) Go to Step 1.

(11) While there is an X-tree, x, connected to r, pick a maximum level vertex v_1 in x such that t_{v_1} is also an X-tree. Out of the two subtrees, t_α and t_β, connected to v_1 (by Proposition 3.5), without loss of generality, let t_α be the subtree that is closer to r. Remove the edge connecting t_α to v_1, and add a new edge to connect r to the closest node in t_α.

LEMMA 3.6. *Algorithm* CMStT-UNIFORM *considers all the possible cases for a given subtree t_v.*

PROOF. Recall that we are working with a ρ_{ST}-approximate Steiner tree with maximum nodal degree at most 5. If the subtree t_v fits into one of Steps 2–6, then the algorithm will solve the case using one of those steps. Hence, for the rest of the proof, we can assume that subtree t_v does not fit into any of Steps 2–6. Since t_v is a subtree of weight greater than k and does not fit into any of Steps 2–6, there must be at least one X-tree connected to v. As can be seen from the algorithm, Steps 7–9 captures all possible cases under this setting. □

THEOREM 3.7. *For a given* CMStT *instance on a L_p plane, Algorithm* CMStT-UNIFORM *guarantees an approximation ratio of* $(\frac{7}{5}\rho_{ST} + \frac{3}{2})$.

PROOF. We show that the cost of the tree output by Algorithm CMStT-UNIFORM is at most $(\frac{7}{5}\rho_{ST} + \frac{3}{2})$ times the cost of an optimal CMStT. The input to the algorithm is a ρ_{ST}-approximate Steiner tree T with maximum nodal degree at most 5.

The algorithm "adds" a new spoke to the tree whenever it prunes a subtree of weight at least $2k/3$. There are certain situations (Steps 6 and 11) where the algorithm adds a spoke for pruned subtrees of weight less than $2k/3$. We continue our analysis as if all of the pruned subtrees were of weight at least $2k/3$. This supposition makes the analysis of spoke cost simpler. We will soon justify this supposition (in Cases 5 and 8) in a manner that does not affect the overall analysis in any way.

The cost of the spokes that were added to the initial Steiner tree is given by

$$C_{sp} \leq \frac{3}{2} \times C_{opt}$$

by an argument analogous to that proving the cost of the spokes that the algorithm adds to the initial Steiner tree in Theorem 3.2. The above inequality follows immediately from the fact that a new spoke is added to the tree if and only if the subtree it connects to r is of weight at least $2k/3$.

Now, we account for the cost of other edges—all the edges in the final solution, except for the spokes added by the algorithm—in the final solution. We show that the cost of these edges is at most $7/5$ times the cost of the Steiner tree edges that the algorithm started with. To prove this, it suffices to show that the cost of the edges that replace the Steiner tree edges is at most $7/5$ times the cost of the edges that are replaced. In what follows, we show this by presenting a case-by-case analysis depending upon which step of the algorithm was executed.

Case 1. Steps 1, 2, 3, and 10 do not add any nonspoke edges. The weight of the subtrees for which Steps 1 and 2 adds spokes to the tree is at least $2k/3$.

Case 2. The minimal-cost edge connecting c_i and c_j in Step 4 is at most the sum of the two Steiner tree edges that connect c_i and c_j to v (by triangle inequality). Hence no additional cost is involved.

Case 3. In Step 5, the cost of the tree s spanning all the vertices in c_i, c_j, and c_z is at most the cost of the tree obtained by doubling the minimum-cost edge out of the three Steiner tree edges that connect the three subtrees to v (see Figure 2(a)). Hence, we can conclude that the cost of the tree constructed in Step 5 is at most $4/3$ times the cost of the Steiner tree edges it replaces.

Case 4. In Step 6, the total cost of the trees s_1 and s_2 spanning all the vertices in c_i, c_j, and c_z is at most the total cost of the trees t_1 and t_2 obtained by doubling the minimum-cost edge set out of the three edge sets that are incident on the vertices in c_i, c_j, and c_z, respectively (see Figure 2(b)). Hence, we can conclude that the cost of the tree constructed in Step 6 is at most $4/3$ times the cost of the Steiner tree edges it replaces.

Case 5. Step 7 forms three subtrees s_1, s_2, and s_3 from X-trees x_i and x_j. Since s_1, s_2, and s_3 can be formed by doubling two low-cost edge sets (belonging to two different groups) out of the five possible edge sets and shortcutting, we can conclude that the cost of the subtrees s_1, s_2, and s_3 constructed in Step 7 is at most $7/5$ times the cost of the Steiner tree edges it replaces.

Accounting for the cost of the spokes added to the Steiner tree requires that each subtree pruned from the Steiner tree be of weight at least $2k/3$. We already proved that the cost of the spokes added to the Steiner tree is at most $3/2$ times the cost of an optimal solution. Without loss of generality, the requirement that each pruned subtree is of weight at least $2k/3$ can be interpreted as that of "charging" the spoke cost incident on a subtree to at least $2k/3$ vertices. Notice that this interpretation is valid only if the spoke connecting the subtree to the root is of minimal cost (r is connected to the closest node in the subtree).

Step 7(a) of the algorithm constructs three subtrees s_1, s_2, and s_3, each containing at least $2k/3$ vertices. This ensures that there are at least $2k/3$ vertices to which each of these subtrees can charge their spoke cost. This is not the case with Step 7(b) of the algorithm. As can be seen, subtree s_3 might be of weight less than $2k/3$. Since s_2 contains at least $2k/3$ vertices and $w(s_2) + w(s_3) \geq 4k/3$, and w is a vertex in x_j such that $|wv|$ is minimum, we can always charge the spoke costs of s_2 and s_3 to at least $4k/3$ vertices. Hence, our initial assumption that every pruned subtree is of weight at least $2k/3$ does not affect the analysis since there are at least $2k/3$ vertices for every spoke to charge.

Case 6. The analyses for Steps 8 and 9 are similar to that for Step 6 (Case 4).

Case 8. Step 11 prunes one subtree off X-tree x. The cost of the spoke $|rw|$ to connect t_α to r can be charged to all the vertices in the X-tree x as per the following argument. After disconnecting t_α from the X-tree, we are left with a subtree of $w(x) - w(t_\alpha) < k$ vertices. We do not need a new spoke for the left-over subtree as it is already connected to r using the Steiner tree edge. Hence, even for this case, our initial assumption that every pruned subtree is of weight at least $2k/3$ does not affect the analysis since there are at least $\frac{2}{3}k$ vertices to charge for the spoke added.

In all of the above cases, the cost of the edges that replace the Steiner tree edges is at most 7/5 times the cost of the Steiner tree edges that the algorithm started with. Thus, the total cost of the tree output by the algorithm is

$$\begin{aligned} C_{app} &\leq \frac{7}{5}\rho_{ST} C_{ST} + \frac{3}{2}\, C_{opt} \\ &\leq \left(\frac{7}{5}\rho_{ST} + \frac{3}{2}\right) C_{opt}. \end{aligned}$$ □

COROLLARY 3.8. *For the* CMST *problem in the* L_p *plane with uniform vertex weights, Algorithm* CMStT-UNIFORM *guarantees a* 2.9-*approximation ratio.*

3.3. UNIT VERTEX WEIGHTS WITH $k = 3, 4$. The CMST problem is polynomial time solvable if all vertices have unit weights, and $k = 2$ [Garey and Johnson 1979]. We show that an optimal solution for $k = 2$ is a 2-approximation for $k = 3, 4$. Let opt_2, opt_3, and opt_4 be the optimal costs for $k = 2$, $k = 3$, and $k = 4$, respectively.

THEOREM 3.9. *Given a undirected graph* $G = (V, E)$ *with unit vertex weights,* $r \in V$, *and* $k = 3$ *or* 4, *an optimal solution for the* CMST *problem with* $k = 2$ *is a* 2-*approximation for the* CMST *problem with* $k = 3, 4$.

PROOF. We obtain the following inequalities from the fact that one could obtain a feasible solution for $k = 2$ by doubling the edges of an optimal solution for $k = 3, 4$:

$$opt_2 \leq 2 \times opt_3 \quad \text{and} \quad opt_2 \leq 2 \times opt_4.$$

If we were to use opt_2 as a feasible solution for $k = 4$, then

$$Ratio = \frac{opt_2}{opt_4} \leq \frac{opt_2}{opt_2/2} = 2.$$

Using the same argument, we can conclude that $opt_2/opt_3 \leq 2$. □

TABLE I. COMPARISON OF OUR RATIOS WITH THOSE OF ALTINKEMER AND GAVISH [1988] FOR THE BUDGETED CMST PROBLEM

	Uniform Vertex Weights in L_p Plane		Nonuniform Vertex Weights in Euclidean Plane	
α	Our ratio $1 + \frac{2}{\alpha}$	AG's ratio $2 - \frac{2}{k} + \frac{1}{\alpha}$	Our ratio $\frac{2}{\sqrt{3}} + \frac{2}{\alpha}$	AG's ratio $2 + \frac{2}{\alpha}$
1	3	$3 - \frac{2}{k}$	3.155	4
2	2	$2.5 - \frac{2}{k}$	2.155	3
4	1.5	$2.25 - \frac{2}{k}$	1.655	2.5
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	1	2	1.155	2

4. *The Budgeted* CMST *Problem*

In this section, we show the effects of using MST cost as a lower bound for the CMST problem. For this purpose, we try to compare our algorithms for the CMST problem with those of Altinkemer and Gavish [1988]. In Altinkemer and Gavish [1988], the approximation ratios are dominated by the MST cost, while that is not the case with our algorithms.

We consider a relaxed version of the CMST problem, the *budgeted* CMST problem, in which a compromise can be made on the capacity constraint k. In this version, for a given k, the weights of the subtrees connected to r could be up to αk, where $\alpha \geq 1$. This gives flexibility when the cost of final solution has to be below a certain budget B, $C_{mst} < B < \beta C_{opt}$, where β is the approximation ratio if the capacity constraint were k.

To solve this version, we can use our algorithm for nonuniform vertex weights with αk as the capacity constraint instead of k. We obtain the following theorems for this version of the problem.

THEOREM 4.1. *For a given* CMST *instance and a k, Algorithm* CMStT-NONUNIFORM *finds a* CMST*, with the weight of every subtree connected to r being at most αk, whose cost is at most $\gamma + \frac{2}{\alpha}$ times the cost of an optimal solution to the given problem.*

PROOF. The proof is similar to the one for Theorem 3.2. □

THEOREM 4.2. *For a given* CMST *instance in the L_p plane with uniform vertex weights, and a k, Algorithm* CMStT-NONUNIFORM *finds a* CMST*, with the weight of every subtree connected to r being at most αk, whose cost is at most $1 + \frac{2}{\alpha}$ times the cost of an optimal solution to the given problem.*

PROOF. Without loss of generality, we can assume that the vertices are of unit weight and scale k accordingly. Since the vertices are of unit weight, Step 4 of the Algorithm CMST-NONUNIFORM will never be executed, and thus the concept of *dummy* or Steiner vertices never have to be introduced. This directly means that the edges within every subtree connected to r are MST edges. The rest of the proof follows along the same line as that for Theorem 3.2. □

Table I shows the comparison of our ratios with those of Altinkemer and Gavish [1988] for special cases of the budgeted CMST problem (AG denotes Altinkemer and Gavish).

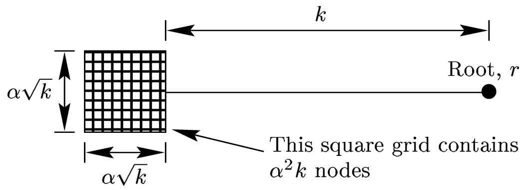

FIG. 8. A tight example for 2-approximation ratio.

5. *Conclusion*

In this article, we presented approximation algorithms for the CMStT problem. This directly translates into improved approximation guarantees for the well-studied CMST problem. Our results for the CMST problem are an improvement over the 15-year-old current best approximations for the respective problems.

Our ratios are, certainly, not tight. We believe that there is room for improvement, at least for the CMST problem with uniform vertex weights, for which we obtained a ratio of 2.9. The cost of an optimal CMST can be lower bounded by one of the following two quantities: (i) the MST cost and (ii) the spoke lower bound (Lemma 2.1). Consider Figure 8, which contains $\alpha^2 k$ points in a unit-spaced grid. MST cost of the points in the grid alone is $\alpha^2 k - 1$. Let k be the distance between r and the closest node in the grid. For capacity constraint k, the cost of an optimal solution would be $2\alpha^2 k - \alpha^2$, whereas the MST cost would be $(\alpha^2 + 1)k - 1$ and the spoke lower bound would be $\alpha^2 k$. This shows that, with the current lower bounds, one cannot get a ratio any better than 2. It would be interesting to see whether we could find a unified lower bound by combining the MST cost and the spoke cost in a some way, instead of just analyzing them separately. We do not see a reason why our of ratio of 2.9 cannot be improved to 2.

ACKNOWLEDGMENTS. We thank R. Chandrasekaran and Ovidiu Daescu for useful discussions.

REFERENCES

AARTS, E., AND KORST, J. 1989. *Simulated Annealing and Boltzman Machines*. Wiley, Chichester, U.K.

AHUJA, R., ORLIN, J., AND SHARMA, D. 2001. A composite neighborhood search algorithm for the capacitated minimum spanning tree problem. *Oper. Res. Letters 31*, 185–194.

ALTINKEMER, K., AND GAVISH, B. 1988. Heuristics with constant error guarantees for the design of tree networks. *Manage. Sci. 34*, 331–341.

AMBERG, A., DOMSCHKE, W., AND VOß, S. 1996. Capacitated minimum spanning trees: Algorithms using intelligent search. *Comb. Opt.: Theor. Pract. 1*, 9–39.

BOORSTYN, R., AND FRANK, H. 1977. Large-scale network topological optimization. *IEEE Trans. Comm. 25*, 29–47.

CHANDY, K., AND LO, T. 1973. The capacitated minimum spanning tree. *Networks 3*, 173–181.

CHANDY, K., AND RUSSELL, R. 1972. The design of multipoint linkages in a teleprocessing tree network. *IEEE Trans. Comp. 21*, 1062–1066.

DOMSCHKE, W., FORST, P., AND VOß, S. 1992. Tabu search techniques for the quadratic semi-assignment problem. In *New Directions for Operations Research in Manufacturing*. Springer, Berlin, Germany, 389–405.

ELIAS, D., AND FERGUSON, M. 1974. Topological design of multipoint teleprocessing networks. *IEEE Trans. Comm. 22*, 1753–1762.

ESAU, L., AND WILLIAMS, K. 1966. On teleprocessing system design. *IBM Sys. J. 5*, 142–147.

FRANK, H., FRISCH, I., SLYKE, R. V., AND CHOU, W. 1971. Optimal design of centralized computer design network. *Networks 1*, 43–57.

GAREY, M., AND JOHNSON, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, San Francisco, CA.

GAVISH, B. 1982. Topological design of centralized computer networks—formulations and algorithms. *Networks 12*, 355–377.

GAVISH, B. 1983. Formulations and algorithms for the capacitated minimal directed tree problem. *J. Assoc. Comput. Mach. 30*, 118–132.

GAVISH, B. 1985. Augmented lagrangean based algorithms for centralized network design. *IEEE Trans. Comm. 33*, 1247–1257.

GAVISH, B. 1991. Topological design of telecommunication networks—local access design methods. *Ann. Oper. Res. 33,* 1, 17–71.

GAVISH, B., AND ALTINKEMER, K. 1986. Parallel savings heuristics for the topological design of local access tree networks. In *Proceedings of the IEEE INFOCOM*. 130–139.

GLOVER, F. 1989. Tabu search—part i. *ORSA J. Comput. 1*, 190–206.

GLOVER, F. 1990. Tabu search—part ii. *ORSA J. Comput. 2*, 4–32.

GOEMANS, M., AND WILLIAMSON, D. 1995. A general approximation technique for constrained forest problems. *SIAM J. Comput. 24*, 296–317.

GOUVEIA, L. 1993. A comparison of directed formulations for the capacitated minimal spanning tree problem. *Telecomm. Syst. 1*, 51–76.

GOUVEIA, L. 1995. A 2n constraint formulation for the capacitated minimal spanning tree problem. *Oper. Res. 43*, 130–141.

GOUVEIA, L., AND AO, J. P. 1991. Dynamic programming based heuristics for the topological design of local access networks. *Ann. Oper. Res. 33*, 305–327.

HALL, L. 1996. Experience with a cutting plane approach for the capacitated spanning tree problem. *INFORMS J. Comput. 8*, 219–234.

HASSIN, R., RAVI, R., AND SALMAN, F. 2000. Approximation algorithms for capacitated network design problems. In *Proceedings of the Workshop on Approximation Algorithms for Combinatorial Optimization* (APPROX). 167–176.

JOTHI, R., AND RAGHAVACHARI, B. 2004a. Approximation algorithms for the capacitated minimum spanning problem and its variants in network design. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming* (ICALP). 805–818.

JOTHI, R., AND RAGHAVACHARI, B. 2004b. Survivable network design: The capacitated minimum spanning network problem. *Inform Process. Lett. 91*, 4, 183–190.

KARNAUGH, M. 1976. A new class of algorithms for multipoint network optimization. *IEEE Trans. Comm. 24*, 500–505.

KERSHENBAUM, A. 1974. Computing capacitated minimal spanning trees efficiently. *Networks 4*, 299–310.

KERSHENBAUM, A., AND BOORSTYN, R. 1983. Centralized teleprocessing network design. *Networks 13*, 279–293.

KERSHENBAUM, A., BOORSTYN, R., AND OPPENHEIM, R. 1980. Second-order greedy algorithms for centralized teleprocessing network design. *IEEE Trans. Comm. 28*, 1835–1838.

KERSHENBAUM, A., AND CHOU, W. 1974. A unified algorithm for designing multidrop teleprocessing networks. *IEEE Trans. Comm. 22*, 1762–1772.

MALIK, K., AND YU, G. 1993. A branch and bound algorithm for the capacitated minimum spanning tree problem. *Networks 23*, 525–532.

MARTIN, J. 1967. *Design of Real-Time Computer Systems*. Prentice Hall, Englewood Cliffs, NJ.

MCGREGOR, P., AND SHEN, D. 1977. Network design: An algorithm for the access facility location problem. *IEEE Trans. Comm. 25*, 61–73.

MONMA, C., AND SURI, S. 1992. Transitions in geometric minimum spanning trees. *Disc. Comput. Geom. 8*, 265–293.

OSMAN, I., AND CHRISTOFIDES, N. 1994. Capacitated clustering problems by hybrid simulated annealing and tabu search. *Intl. Trans. Oper. Res. 1*, 317–336.

PAPADIMITRIOU, C. 1978. The complexity of the capacitated tree problem. *Networks 8*, 217–230.

ROBINS, G., AND SALOWE, J. 1995. Low-degree minimum spanning trees. *Disc. Comput. Geom. 14*, 151–166.

SALMAN, F., CHERIYAN, J., RAVI, R., AND SUBRAMANIAN, S. 1997. Buy-at-bulk network design: Approximating the single-sink edge installation problem. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms* (SODA). 619–628.

SCHNEIDER, G., AND ZASTROW, M. 1982. An algorithm for the design of multilevel concentrator networks. *Comput. Netw. 6*, 563–581.

SHARAIHA, Y., GENDREAU, M., LAPORTE, G., AND OSMAN, I. 1997. A tabu search algorithm for the capacitated shortest spanning tree problem. *Networks 29*, 161–171.

SHARMA, R., AND EL-BARDAI, M. 1970. Suboptimal communications network synthesis. In *Proceedings of the IEEE International Conference on Communications*. 19.11–19.16.

VOß, S. 2001. Capacitated minimum spanning trees. In *Encyclopedia of Optimization*. Kluwer, Boston, MA, 225–235.

WHITNEY, V. 1970. A study of optimal file assignment and communication network configuration in remote access computer message processing and communications systems. Ph.D. Dissertation. University of Michigan, Ann Arbor, MI.

RECEIVED MARCH 2005; ACCEPTED APRIL 2005